ISSN 2074-7888, Наукові праці Донецького національного технічного університету, серія: «Проблеми моделювання та автоматизації проектування» № 1 (10)-2(11), 2012

УДК 004: 519.854

К.В. Кротов (канд. техн. наук, доцент)
Севастопольский национальный технический университет
krotov_kv@mail.ru

МОДЕЛЬ ДВУХУРОВНЕВОГО ПРОГРАММИРОВАНИЯ И МЕТОД ПОСТРОЕНИЯ РАСПИСАНИЙ ГРУППОВОЙ ОБРАБОТКИ

Обосновывается модель двухуровневого программирования формирования расписаний групповой обработки, а также метод формирования состава партий и порядка их обработки в многостадийных конвейерных системах.

Ключевые слова: модель двухуровневого программирования, групповая обработка, метод формирования состава партий, порядок обработки партий, многостадийная конвейерная система

Введение

В современных производственных и вычислительных системах возникает задача составления расписаний групповой обработки требований различных типов. При этом построение расписаний групповой обработки предполагает формирование порядка обработки партий, определение количества и состава этих партий. Формирование расписаний групповой обработки обеспечивается решением двух оптимизационных задач: определения эффективных количества и состава партий требований, определения эффективного порядка обработки партий. В этом случае задача формирования расписаний групповой обработки может быть рассмотрена как задача двухуровневого программирования (теории иерархических игр) [1,2]: на верхнем уровне планирования определяется эффективное количество и состав партий требований соответствующих типов, на нижнем уровне планирования формируется порядок обработки партий.

Анализ публикаций

Современной состояние решения задачи формирования расписаний групповой обработки требований различных типов предполагает определение порядка обработки фиксированных партий (т.е. партий, в которые включены все требования одного определенного типа), либо определения количества, состава и порядка обработки партий на ограниченном количестве приборов [3]. В обобщенном виде (эффективное количество партий, которые формируются из требований различных типов, эффективный состав партий, произвольное количество приборов) задача формирования расписаний групповой обработки требует разрешения.

Цель и постановка задач

Цель выполняемой работы состоит в совершенствовании методов формирования расписаний групповой обработки требований в многостадийных системах. Для достижения поставленной цели решаются задачи: обосновывается модель формирования расписаний групповой обработки как модель иерархической игры (двухуровневого программирования); обосновывается метод определения эффективного количества и состава партий требований, определения порядка их обработки в многостадийной системе при произвольном количестве приборов.

Основное содержание работы

Для определения вида модели двухуровневого программирования (иерархической игры) построения эффективных расписаний групповой обработки введены следующие обозначения: t_i^l– время обработки требований *i*-го типа на *l*-ом приборе ($l=\overline{1,L}$, где *L*-количество приборов в системе); t_{ik}^l – время переналадки *l*-го прибора с обработки требований *i*-го типа на обработку требований *k*-го типа; t_{ii}^l– время первоначальной наладки *l*-го прибора системы на обработку партии требований *i*-го типа; *N*– множество типов требований ($N=\{1,2,..,n\}$); n^i – количество требований *i*-го типа, которое должно быть обработано, $i=\overline{1,n}$, где n – количество типов требований; (P)– матрица порядка обработки партий требований *i*-ых типов в последовательностях π^l (порядок обработки партий на всех приборах одинаков, достаточно определения одной матрицы порядка для всей системы, элемент $p_{ij}=1$, если партия требований *i*-го типа занимает в последовательности π^l *j*-ю позицию, $p_{ij}=0$ в противном случае, размер матрицы $n\times n_p$, где n – число типов требований, n_p– число партий в последовательностях π^l); (R) – матрица количества требований в соответствующих партиях требований *i*-го типа, занимающих в последовательности π^l *j*-е позиции (элемент r_{ij} равен количеству требований *i*-го типа в партии, занимающей *j*-ю позицию в последовательности π^l, размер матрицы $n\times n_p$); t_{ji}^{nl} – время начала обработки партии требований *i*-го типа, занимающей *j*-ю позицию в последовательности π^l на *l*-ом приборе; $\overline{t_{ji}^{nl}}$ – время окончания обработки партии требований *i*-го типа в *j*-ой позиции в последовательности π^l на *l*-ом приборе; $\left(t_{ji}^{nl}\right)$– матрица моментов времени начала обработки партий требований i-ых типов, занимающих в последовательностях π^l j-е позиции; $\left(t_{jq}^{0l}\right)$– матрица моментов времени начала обработки q-ых требований партии, занимающей в последовательности π^l j-ю позицию (q – порядковый номер требования в партии в j-ой позиции в π^l, $q=\overline{1,r_{ij}}$). С использование матриц (P) и $\left(t_{jq}^{0l}\right)$ элементы матрицы $\left(t_{ji}^{nl}\right)$ определяются следующим образом: $t_{ji}^{nl}=p_{ij}\cdot t_{j1}^{0l}$, где $i=\overline{1,n}, j=\overline{1,n_p}$, t_{j1}^{0l}– момент времени начала обработки первого требования в партии, занимающей j-ю позицию в последовательности π^l. Матрицы (P), (R) (либо $\left(t_{ji}^{nl}\right)$) являются результатами составления расписания групповой обработки.

Для расчета t_{jq}^{0l} должны быть заданы интервалы времени переналадки приборов с обработки требований i-го типа на обработку требований k-го типа, а также интервалы первоначальной наладки приборов системы на обработку требований i-го типа. Эти интервалы времени сведены в матрицу переналадок (t_{ik}^{l}), недиагональные элементы t_{ik}^{l} которой соответствуют длительностям интервалов переналадки приборов с обработки требований i-го типа на обработку требований k-го типа, диагональные элементы t_{ii}^{l} соответствуют интервалам времени первоначальной наладки приборов на обработку требований i-го типа. Для определения t_{ji}^{nl} необходимо вычисление значений элементов матрицы $\left(t_{jq}^{0l}\right)$, формализация выражений для определения t_{jq}^{0l} выполнена на основе рис. 1.

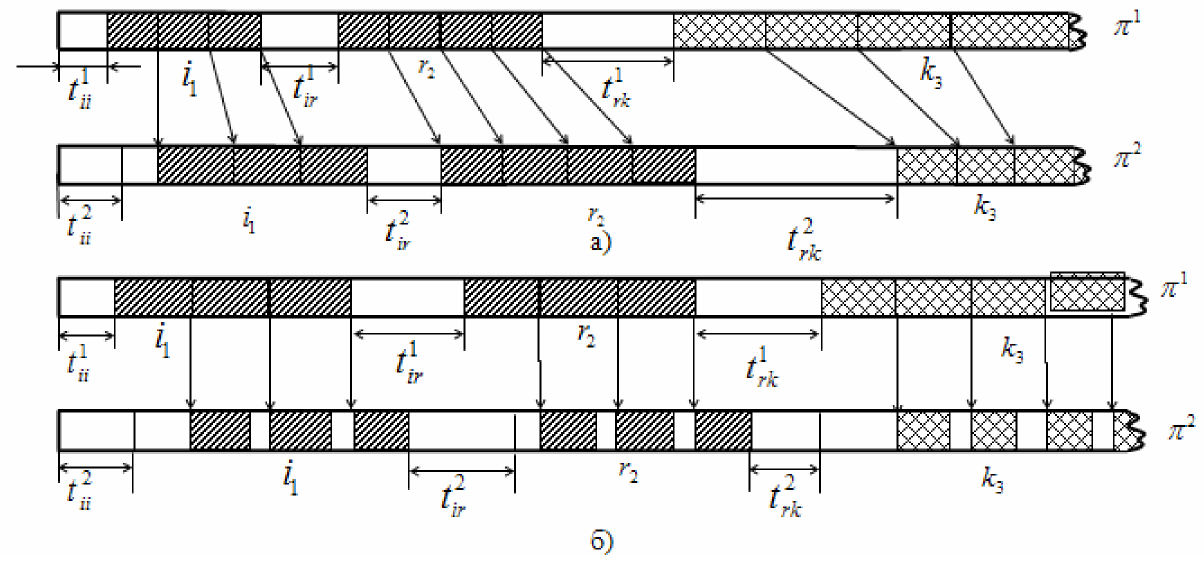

Рисунок 1 – Пример видов последовательностей обработки партий, используемые при формировании выражений для t_{jq}^{0l}

Для первого прибора выражения для t_{jq}^{01} формируются следующим образом:

1) если t_{11}^{01} - время начала обработки первого требования (q=1) в партии, занимающей первую позицию в последовательностях π^1, тогда t_{1i}^{n1} - момент времени начала обработки партии требований i-го типа, занимающая в π^1 первую позицию; ясно, что $t_{11}^{01} = t_{1i}^{n1} = t_{ii}^{1}$, где t_{ii}^{1}– время наладки первого прибора на обработку изделий i-го типа;

2) выражение $\sum\limits_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1}$ определяет интервал первоначальной наладки первого прибора на обработку требований i-го типа (партия в первой позиции в π^1); $(q-1)\sum\limits_{h=1}^{n} t_{h}^{1} \cdot p_{h1}^{1}$– время обработки предыдущих $(q-1)$-го требования в этой партии, где q- порядковый номер требования в партии в π^1, (t^1) – вектор-строка длительностей обработки требований разных типов на первом приборе, тогда значение t_{1q}^{01} начала обработки любого требования первой партии (в π^1), занимающего в ней q-ю позицию $(1 \le q \le r_{i1})$ определяется выражением вида:

$$t_{1q}^{01} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + (q-1)\sum_{h=1}^{n} t_{h}^{1} \cdot p_{h1}^{1}; \tag{1}$$

3) если $t_{i_1 i_2}^{1пер}$ - время переналадки первого прибора с обработки требований i-го типа (в первой позиции партии в π^1) на обработку требований партии другого типа (партия во второй позиции в π^1), t_{21}^{01} -время начала обработки первого требования во второй партии на первом приборе (t_{2i}^{n1}– начало обработки этой партии требований i-го типа в π^1 $\left(t_{2i}^{n1} = t_{21}^{01}\right)$), тогда с учетом (1) значения t_{21}^{01} (t_{2i}^{n1}) определяются выражениями вида:

$$t_{2i}^{n1} = t_{21}^{01} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h1}^{1} + t_{i_1 i_2}^{1пер}, \tag{2}$$

где значение $t_{i_1 i_2}^{1пер}$ определяется следующим образом:

$$t_{i_1 i_2}^{1пер} = t_{i_r i_{r'}}^{1}, \text{ где } \begin{cases} i_r = i \mid p_{i1}^{1} = 1, i = \overline{1,n} \\ i_{r'} = j \mid p_{j2}^{1} = 1, j = \overline{1,n} \end{cases}.$$

В соответствии с (2) выражение $\sum\limits_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum\limits_{h=1}^{n} t_{h}^{1} \cdot r_{h1}^{1} + t_{i_1 i_2}^{1пер}$ определяет начало обработки второй в π^1 партии, выражение $(q-1)\sum\limits_{h=1}^{n} t_{h}^{1} \cdot p_{h2}^{1}$–интервал обработки предшествующих q-му требований в этой партии, тогда время начала обработки q-го требования i-го типа в партии, занимающей вторую позицию в π^1, определяется следующим образом:

$$t_{2q}^{01} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h1}^{1} + t_{i_1 i_2}^{1пер} + (q-1)\sum_{h=1}^{n} t_{h}^{1} \cdot p_{h2}^{1}, \quad q = \overline{1, r_{i2}}. \tag{3}$$

По аналогии с (2), (3) могут быть сформированы выражения для моментов времени начала обработки третьей, четвёртой партий и требований в них. Вид выражений следующий: $t_{3i}^{n1} = \sum_{p=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h1}^{1} + t_{i_1 i_2}^{1пер} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h2}^{1} + t_{i_2 i_3}^{1пер}$;

$$t_{3q}^{01} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h1}^{1} + t_{i_1 i_2}^{1пер} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h2}^{1} + t_{i_2 i_3}^{1пер} + (q-1)\sum_{h=1}^{n} t_{h}^{1} \cdot p_{h3}^{1};$$

$$t_{4i}^{n1} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h1}^{1} + t_{i_1 i_2}^{1пер} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h2}^{1} + t_{i_2 i_3}^{1пер} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h3}^{1} + t_{i_3 i_4}^{1пер};$$

$$t_{4q}^{01} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h1}^{1} + t_{i_1 i_2}^{1пер} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h2}^{1} + t_{i_2 i_3}^{1пер} + \sum_{h=1}^{n} t_{h}^{1} \cdot r_{h3}^{1} + t_{i_3 i_4}^{1пер} + (q-1)\sum_{h=1}^{n} t_{h}^{1} \cdot p_{h4}^{1}.$$

В обобщенной форме время начала обработки партии требований i-го типа, занимающей j-ю позицию в π^1 определяется выражением вида: $t_{ji}^{n1} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{f=1}^{j-1}\sum_{h=1}^{n} t_{h}^{1} \cdot r_{hf}^{1} + \sum_{h=1}^{j-1} t_{i_h i_{h+1}}^{1пер}$. Время начала обработки q-го требования в этой партии: $t_{jq}^{01} = \sum_{h=1}^{n} t_{hh}^{1} \cdot p_{h1}^{1} + \sum_{f=1}^{j-1}\sum_{h=1}^{n} t_{h}^{1} \cdot r_{hf}^{1} + \sum_{h=1}^{j-1} t_{i_h i_{h+1}}^{1пер} + (q-1)\sum_{h=1}^{n} t_{h}^{1} \cdot p_{hj}^{1}$.

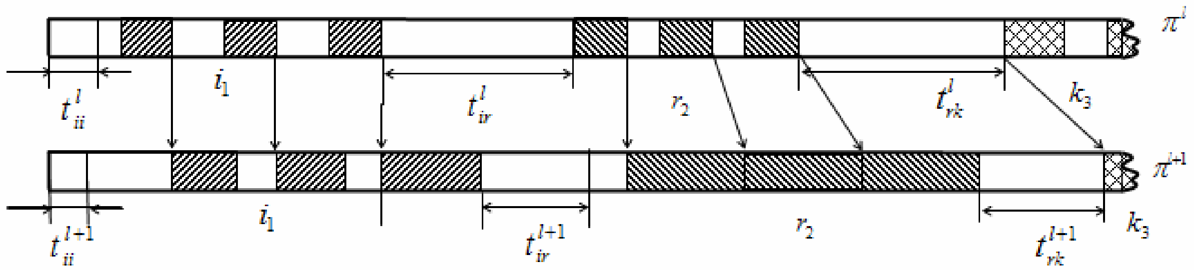

Рисунок 2– Пример видов последовательностей обработки партий, используемые при формировании выражений для t_{jq}^{0l}

Обобщённое выражение для $t_{jq}^{0l}\,(q = \overline{1, n_j}, n_j = \sum_{h=1}^{n} r_{hj})$ сформируем в соответствии с заданны порядком обработки (Рисунок 2) на основе совокупности выражений для определения начала обработки требований партий, рассматриваемых последовательно (при $l \neq 1$). В соответствии с введенными обозначениями и следуя определенным выше рассуждениям, для $l=2$ и $j=1$ имеем: $t_{11}^{02} = t_{1i}^{n2} = \max\left\{\sum_{h=1}^{n} t_{hh}^{2} \cdot p_{h1}; t_{11}^{01} + \sum_{h=1}^{n} t_{h}^{1} \cdot p_{h1}\right\}$;

$$t_{12}^{02} = \max\left\{t_{11}^{02} + \sum_{h=1}^{n} t_{h}^{2} \cdot p_{h1}; t_{12}^{01} + \sum_{h=1}^{n} t_{h}^{1} \cdot p_{h1}\right\}; \quad \ldots$$

$t_{1n_1}^{02} = \max\left\{t_{1,n_1-1}^{02} + \sum_{h=1}^{n} t_{h}^{2} \cdot p_{h1}; t_{1n_1}^{01} + \sum_{h=1}^{n} t_{h}^{1} \cdot p_{h1}\right\}$, где $n_1 = \sum_{i=1}^{n} r_{i1}$ - количество требований в первой партии. Выражения для определения $t_{2q}^{02}\,(q = \overline{1, n_2}, n_2 = \sum_{h=1}^{n} r_{h2})$ для требований i-го типа, входящих во вторую партию в π^2 на втором приборе имеют вид:

$t_{21}^{02} = t_{2i}^{n2} = max\left\{t_{1n_1}^{02} + \sum_{h=1}^{n} t_h^2 \cdot p_{h1} + t_{i_1 i_2}^{2пер}; t_{21}^{0l} + \sum_{h=1}^{n} t_h^1 p_{h,2}\right\}$; $t_{22}^{02} = max\left\{t_{21}^{02} + \sum_{h=1}^{n} t_h^2 \cdot p_{h2}; t_{22}^{0l} + \sum_{h=1}^{n} t_h^1 \cdot p_{h2}\right\}$; …; $t_{2n_2}^{02} = max\left\{t_{2,n_2-1}^{02} + \sum_{h=1}^{n} t_h^2 \cdot p_{h2}; t_{2,n_2}^{01} + \sum_{h=1}^{n} t_h^1 \cdot p_{h2}\right\}$. Выражения для t_{3i}^{n2} и t_{3q}^{02} имеют вид:

$$t_{31}^{02} = t_{3i}^{n2} = \max\left\{t_{2,n_2}^{02} + \sum_{h=1}^{n} t_h^2 \cdot p_{h2} + t_{i_2 i_3}^{2пер}; t_{31}^{01} + \sum_{h=1}^{n} t_h^1 \cdot p_{h3}\right\};$$

$t_{3q}^{02} = \max\left\{t_{3,q-1}^{02} + \sum_{h=1}^{n} t_h^2 \cdot p_{h3}; t_{3q}^{0l} + \sum_{h=1}^{n} t_h^1 \cdot p_{h3}\right\}$, где $q = \overline{2, n_3}, n_3 = \sum_{h=1}^{n} r_{h3}$. На основе сформированных выражений получены обобщенные выражения для определения t_{jq}^{0l} (с учётом особенностей для t_{11}^{0l} и t_{j1}^{0l}, при $l = \overline{1, L}$, $q = \overline{1, n_j}$, $n_j = \sum_{h=1}^{n} r_{hj}$) в виде: $t_{11}^{0l} = t_{1i}^{nl} = \max\left\{\sum_{h=1}^{n} t_h^l \cdot p_{h1}; t_{11}^{0l-1} + \sum_{h=1}^{n} t_h^{l-1} \cdot p_{h1}\right\}$; $t_{j1}^{0l} = t_{ji}^{nl} = max\left\{t_{j-1,n_{j-1}}^{0l} + \sum_{h=1}^{n} t_h^l \cdot p_{h,j-1} + t_{i_{j-1} i_j}^{lпер}; t_{j1}^{0l-1} + \sum_{h=1}^{n} t_h^{l-1} \cdot p_{hj}\right\}$; ….;

$$t_{jq}^{0l} = \max\left\{t_{jq-1}^{0l} + \sum_{h=1}^{n} t_h^l \cdot p_{hj}; t_{jq}^{0l-1} + \sum_{h=1}^{n} t_h^{l-1} \cdot p_{hj}\right\}.$$

Для реализации метода определения эффективного количества и состава партий, порядка обработки этих партий сформулирована двухуровневая задача принятия решений. На верхнем уровне планирования определяется состав партий (количество требований определённого типа в каждой из партий), эффективный с точки зрения заданного критерия. На нижнем уровне принятия решений определяется порядок обработки партий в последовательностях π^l на l-ых приборах. При этом порядок обработки партий в последовательностях π^l, так же как и состав партий является постоянным. По аналогии с градиентным методом формирования расписаний обработки требований [4] в качестве критерия (целевой функции) на нижнем уровне планирования рассматривается показатель текущей эффективности использования оборудования системы, который определяется суммарным простоем приборов в ожидании готовности требований партий к обработке. Так как решаемая задача не предполагает задание временных ограничений на обработку партий требований, то для верхнего уровня принятия решений введён критерий, характеризующий окончание обработки требований всех партий (соответствующий моменту времени окончания обработки требований партий). Формирование критерия оптимизации на нижнем уровне принятия решений выполнено в соответствии с анализом заданных видов последовательности обработки партий на приборах (рис. 1-3).

Выполненный анализ видов последовательностей позволил определить особенности идентификации интервалов времени простоя оборудования:

1) малые длительности обработки партий требований i-го, r-го и k-го типов на втором приборе при малых длительностях переналадки второго прибора

обуславливают ожидание этим прибором (l=2) готовности партий к обработке и ожидание прибором требований при их обработке внутри партии (рис. 1 б), интервалы ожидания приборами требований при их обработке внутри партии являются одинаковыми; аналогичной ситуации соответствует рис. 2, тогда порядок обработки партий будет определять: а) длительность переналадок приборов с обработки требований одного типа на обработку требований другого типа; б) длительность простоя приборов в ожидании начала обработки партий; если событие окончания переналадки является более ранним, чем окончание обработки первого требования этой партии на предшествующем приборе (рис. 2), то интервалы ожидания готовности требований внутри партии постоянны (т.е. порядок партий в рассматриваемом случае будет определять одинаковые интервалы ожидания приборами требований при обработке партии (внутри партии);

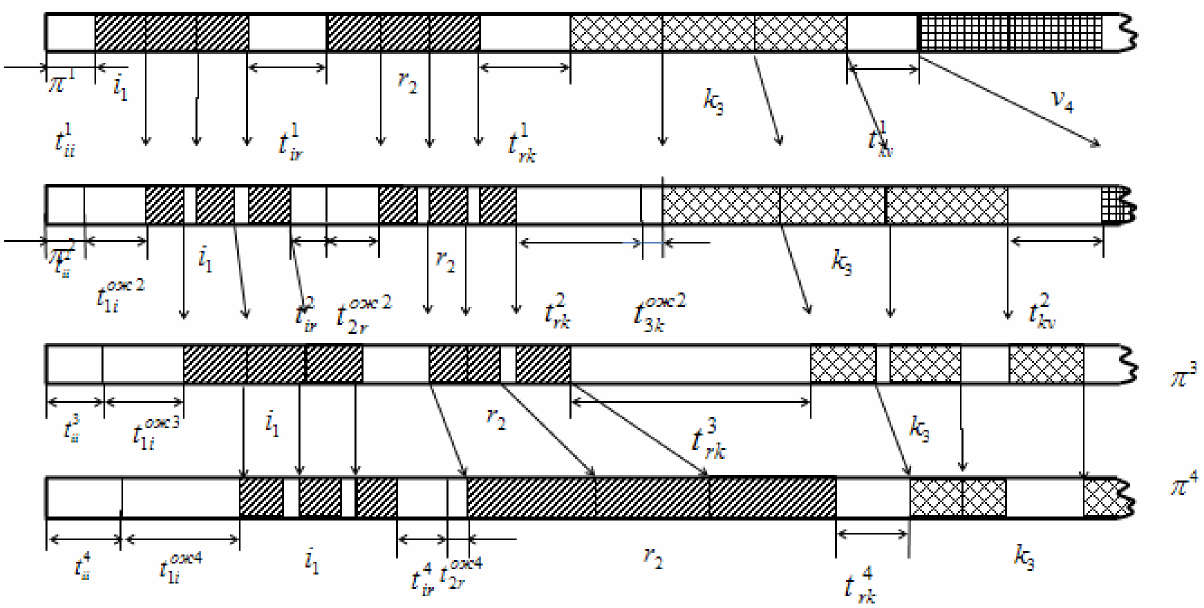

Рисунок 3 – Пример видов последовательностей обработки партий, используемые при формировании критерия оптимизации на нижнем уровне планирования

2) при значительных значениях t_{ik}^{l} (рис.3 третий и четвёртый приборы, последовательности π^3 и π^4, интервалы t_{ir}^{3}, t_{rk}^{3} и t_{rk}^{4}) наблюдаются различные по величине интервалы времени простоя приборов (третьего и четвёртого) при обработке требований внутри партий, тогда различный порядок обработки партий определяет не только различные по величине интервалы переналадки и ожидания приборами начала обработки партий, но и различные по длительности интервалы простоя приборов при обработке требований внутри партии; критерий эффективности расписания обработки партий на нижнем уровне принятия решений должен учитывать: а) время простоя приборов в ожидании начала обработки требований партий (с учетом интервалов наладки, переналадки и последующего

ожидания); б) время простоя приборов в ожидании готовности требований при их обработке внутри партии.

Простой l-го прибора в ожидании обработки первой в последовательности π^l партии равен значению t_{11}^{0l}, тогда суммарное время простоя приборов системы $(l=\overline{1,L})$ в ожидании начала обработки партий в последовательностях π^l определяется выражением вида: $\sum_{l=1}^{L} t_{11}^{0l}$. В интервал простоя l-го прибора в ожидании начала обработки последующей партии после окончания обработки предыдущей входят: интервал переналадки прибора с обработки требований i-го типа на обработку требований другого типа $(t_{i_{j-1}i_j}^{lnep})$, возможный ненулевой интервал ожидания прибором начала обработки партии после окончания переналадки $t_{j,i}^{ожl}$ (рис. 3), где $t_{j,i}^{ожl}$ - время ожидания l-ым прибором начала обработки партии i-ых требований, если эта партия в π^l занимает j-ю позицию. Тогда время простоя l-го прибора, соответствует сумме вида: $t_{i_{j-1}i_j}^{lnep} + t_{j,i}^{ожl}$, а значение этого интервала определяется выражением вида: $t_{j1}^{0l} - \left[t_{j-1,n_{j-1}}^{0l} + \sum_{h=1}^{n} t_h^l \cdot p_{h,j-1} \right]$, где $j>1$, $n_{j-1} = \sum_{h=1}^{n} r_{h,j-1}$ – число требований в предшествующей $(j\text{-}1)$-ой партии. Суммарный простой l-го прибора в ожидании начала обработки j-ых партий ($j=\overline{2,n_p}$, где n_p – общее число партий в последовательностях π^l) определён следующем образом: $\sum_{j=2}^{n_p} \left[t_{j1}^{0l} - \left[t_{j-1,n_{j-1}}^{0l} + \sum_{h=1}^{n} t_h^l \cdot p_{h,j-1} \right] \right]$. В этом случае суммарной простой всех L приборов в ожидании начала обработки партий на них определяется выражением вида:

$$\sum_{l=1}^{L} \sum_{j=2}^{n_p} \left[t_{j1}^{0l} - \left[t_{j-1,n_{j-1}}^{0l} + \sum_{h=1}^{n} t_h^l \cdot p_{h,j-1} \right] \right]. \qquad (3)$$

Простой l-го прибора в ожидании готовности к обработке требования, занимающего q-ю позицию в j-ой партии в последовательности π^l, определяется выражением вида: $t_{jq}^{0l} - \left[t_{j,q-1}^{0l} + \sum_{h=1}^{n} t_h^l \cdot p_{hj} \right]$, где $j=\overline{1,n_p}$. Это выражение соответствует интервалу между двумя требованиями (в q-ой и $(q\text{-}1)$-ой позициях) в j-ой партии в π^l. Тогда суммарный простой l-го прибора в ожидании готовности к обработке требований j-ой партии в π^l вычисляется с использованием выражения:

$$\sum_{q=2}^{n_j} \left[t_{jq}^{0l} - \left[t_{j,q-1}^{0l} + \sum_{h=1}^{n} t_h^l \cdot p_{hj} \right] \right], \qquad (4)$$

где q- номер позиции требования в j-ой партии, n_j - число требований в j-ой партии, $n_j = \sum_{h=1}^{n} r_{hj}$. На основе (4) общий простой l-го прибора при ожидании готовности к обработке требований внутри партий определяется выражением вида: $\sum_{j=1}^{n_p}\sum_{q=2}^{n_j}\left[t_{jq}^{0l}-\left[t_{j,q-1}^{0l}+\sum_{h=1}^{n}t_h^l\cdot p_{hj}\right]\right]$. Тогда суммарный простой всех приборов $(l=\overline{1,L})$ в ожидании готовности требований внутри партии вычисляется по выражению:

$$\sum_{l=2}^{L}\sum_{j=1}^{n_p}\sum_{q=2}^{n_j}\left[t_{jq}^{0l}-\left[t_{j,q-1}^{0l}+\sum_{h=1}^{n}t_h^l\cdot p_{hj}\right]\right]. \tag{5}$$

Сформированный на основе введённого выражения для $\sum_{l=1}^{L}t_{11}^{0l}$ и формул (3), (5) критерий эффективности определения последовательности обработки партий в системе имеет вид:

$$\sum_{l=2}^{L}t_{11}^{0l}+\sum_{l=1}^{L}\sum_{j=2}^{n_p}\left[t_{j1}^{0l}-\left[t_{j-1,n_{j-1}}^{0l}+\sum_{h=1}^{n}t_h^l\cdot p_{h,j-1}\right]\right]+\sum_{l=2}^{L}\sum_{j=1}^{n_p}\sum_{q=2}^{n_j}\left[t_{jq}^{0l}-\left[t_{j,q-1}^{0l}+\sum_{h=1}^{n}t_h^l\cdot p_{hj}\right]\right]. \tag{6}$$

В качестве критерия на верхнем уровне планирования (при определении количества и состава партий) определено время окончания обработки всех партий требований различных типов. Если n_p– идентификатор последней партии в π^L, n_{n_p} – количество требований в этой партии ($n_{n_p}=\sum_{h=1}^{n}r_{h,n_p}$), $t_{n_p,n_{n_p}}^{0L}$ – момент времени начала обработки последнего требования в n_p-ой партии на L-ом приборе, тогда значение критерия вычисляется с использованием выражения:

$$t_{n_p,n_{n_p}}^{0L}+\sum_{h=1}^{n}t_h^L\cdot p_{hn_p}\ . \tag{7}$$

Обобщая выполненные выше рассуждения, на основе полученных выражений (6), (7) модель двухуровневого программирования для определения количества и состава партий, а также порядка их обработки представлена в виде:

- верхний уровень планирования: $min\ f_в$, $f_{â}=t_{n_p,n_{n_p}}^{0L}+\sum_{h=1}^{n}t_h^L\cdot p_{hn_p}$;

- нижний уровень планирования: $min\ f_н$,

$$f_н=\sum_{l=2}^{L}t_{11}^{0l}+\sum_{l=1}^{L}\sum_{j=2}^{n_p}\left[t_{j1}^{0l}-\left[t_{j-1,n_{j-1}}^{0l}+\sum_{h=1}^{n}t_h^l\cdot p_{h,j-1}\right]\right]+\sum_{l=2}^{L}\sum_{j=1}^{n_p}\sum_{q=2}^{n_j}\left[t_{jq}^{0l}-\left[t_{j,q-1}^{0l}+\sum_{h=1}^{n}t_h^l\cdot p_{hj}\right]\right]. \tag{8}$$

С учётом введённых выражений (8) процедура определений эффективного состава партий и порядка их обработки в многостадийной системе предполагает: 1) на верхнем уровне планирования – определение количества партий требований i-го типа и состава этих партий (распределение требований i-го типа по партиям); 2) на нижнем уровне– формирование порядка обработки партий (расписания их обработки), определение эффективного решения, передача его на верхний уровень принятия решения; 3) вычисление значений целевой функции на верхнем уровне на основе полученного с нижнего решения, оценка эффективности сформированного на верхнем уровне состава партий, переход к более эффективному решению. Для принятия решений на верхнем уровне по количественному составу партий требований i-го типа должен быть реализован алгоритм формирования числа требований в партиях. При этом предполагается, что число партий m изменяется следующим образом: $m=2,3...,$ а в партиях не может быть менее, чем 2 требования i-го типа. В рассмотрение введен вектор состава партий *(A)*, h-ый элемент которого равен количеству требований в партии. Правила формирования вектора состава партий *(A)* заданы выражениями вида: 1) $a_1 = n^i - \sum_{h=2}^{m} a_h$, 2) $a_h^0 = 2$ ($h=\overline{2,m}$), 3) $a_1^0 = n^i - \sum_{h=2}^{m} a_h^0$, где a_h^0–значения элементов вектора *(A)*, соответствующего начальному составу m партий. В соответствии с введёнными выражениями вид вектора *(A)* при распределении n^i требований формируется следующим образом: 1) элементы $a_h^0 = 2$ ($h=\overline{2,m}$) инициализируются значением 2 (минимальное количество требований в партиях); 2) элемент $a_1^0 = n^i - \sum_{h=2}^{m} a_h^0$, если $a_1^0 \geq 2$, то количество требований n^i может быть распределено по партиям в количестве m, в случае $a_1^0 < 2$ количество требований n^i не может быть распределено по числу партий m, процесс формирования составов партий для требований i-го типа должен быть завершен (исследуются составы партий от $m=2$ и т.д. до тех пор, пока не будет выполнено условие $a_1^0 < 2$ при $a_1^0 = n^i - \sum_{h=2}^{m} a_h^0$ ($a_h^0 = 2$, $h=\overline{2,m}$), т.е. условие окончания формирования партий предполагает, что при начальном определении их состава (для заданного m) $a_1^0 < 2$). Алгоритм формирования составов m партий требований i-го типа состоит из последовательных шагов (s–индекс шага алгоритма формирования состава партий, первоначально $m=2$): 1) число партий m задано, вычислены значения элементов вектора *(A)*, соответствующего начальному составу партий ($a_h^0 = 2$; $h=\overline{2,m}$; $a_1^0 = n^i - \sum_{h=2}^{m} a_h^0$), если $a_1^0 < 2$, то процесс формирования составов партий для различных m прекращается; 2) элемент a_2 увеличивается на 1 ($a_2(s+1) = a_2(s)+1$), $h=3$, для a_h ($h=\overline{3,m}$) задаются значения 2 ($a_h = 2$, $h=\overline{3,m}$) выполняется вычисление элемента $a_1(s+1) = n^i - \sum_{h=2}^{m} a_h(s+1)$, результат–вектор *(A(s+1)); 3)* если $a_2(s+1) > a_1(s+1)$, то новый состав m партий исследован быть не может (вычисленные значения элементов *(A)* соответствуют составу партий, который был сформирован на предыдущих шагах алгоритма), переход к шагу 9; 4) элемент a_h увеличивается на 1 $(a_h(s+1) = a_h(s)+1)$, выполняется вычисление элемента $a_1(s+1)$, если $a_2(s+1) < a_1(s+1)$, то сформирован новый состав партии *(A(s+1));* 5) если $a_2(s+1) > a_1(s+1)$, то состав партий исследован быть не может, переход к шагу 9; 6) выполняется изменение значения номера h элемента вектора *(A)* на 1 ($h = h + 1$), если $h \le m$, то переход к шагу 4; 7) если $h > m$ и $a_m < a_2$, то $h = 3$, выполняется переход к шагу 4; 8) если $h > m$ и $a_m = a_2$, то переход к шагу 2; 9) увеличение значения количества партий *m,* переход к шагу 1. Пример реализации алгоритма формирования *m* партий для n^i требований приведены на рис. 4.

m=2
3,2
2,3

m=2	m=3
4,2	2,2,2
3,3	1,3,2
2,4	

m=2	m=3
5,2	3,2,2
4,3	2,3,2
3,4	

m=2	m=3	m=4	m=5
8,2	6,2,2	4,2,2,2	2,2,2,2,2
7,3	5,3,2	3,3,2,2	
6,4	4,3,3	2,3,3,2	
5,5	4,4,2		
4,6	3,4,3		

m=2	m=3	m=4	m=5	m=6
10,2	8,2,2	6,2,2,2	4,2,2,2,2	2,2,2,2,2,2
9,3	7,3,2	5,3,2,2	3,3,2,2,2	
8,4	6,3,3	4,3,3,2	2,3,3,2,2	
7,5	6,4,2	3,3,3,3		
6,6	5,4,3	4,4,2,2		
5,7	4,4,4	3,4,3,2		
	5,5,2			
	4,5,3			

m=2	m=3	m=4	m=5	m=6
11,2	9,2,2	7,2,2,2	5,2,2,2,2	3,2,2,2,2,2
10,3	8,3,2	6,3,2,2	4,3,2,2,2	2,3,2,2,2,2
9,4	7,3,3	5,3,3,2	3,3,3,2,2	
8,5	7,4,2	4,3,3,3	2,3,3,3,2	
7,6	6,4,3	5,4,2,2		
6,7	5,4,4	4,4,3,2		
	6,5,2	3,4,3,3		
	5,5,3			
	4,5,4			

Рисунок 4 – Пример реализация алгоритма формирования состава партий

Анализ примеров показывает, что сформированные в соответствии с алгоритмом составы партий, для которых $a_2(s+1) > a_1(s+1)$, являются повторяющими полученные ранее и не исследуются (составы партий ниже пунктирной линии). Сформулированное условие окончания процесса формирования состава партий может быть обобщено следующим образом: для случая, когда увеличение количества требований в партиях с $h > 1$ реализуется в некотором произвольном (отличном от сформулированного выше) порядке необходимо выполнение условия $a_h(s+1) > a_1(s+1)$ при $h = \overline{2,m}$ для прекращения процесса формирования нового состава партий. Т.о. необходимое условие остановки процесса распределения требований по партиям: количество требований в любой из партий с $h \ge 2$ превышает количество требований в первой партии (исходной партии, из которой извлекаются требования и распределяются по всем остальным партиям), т.е. $a_h > a_1$, где $h = \overline{2,m_i}$. Исходными данными при формировании решения на верхнем уровне планирования (решения по количеству и составу партий требований *i*-го типа) являются: 1) множество *N* идентификаторов типов требований, партии которых должны быть обработаны *(N ={1,2,…})*; 2) множество значений количества требований каждого типа, которые должны быть обработаны в системе ($\{n^1, n^2, .., n^n\}$, где n – общее число типов требований). Вид решения, формируемого на верхнем уровне планирования и передаваемого на нижний: $[i, m_i(s), A_i(s)]$, где $m_i(s)$ – число партий требований i-го типа, упорядочивание которых в последовательностях π^l реализуются на s-ом шаре алгоритма, $A_i(s)$ – вектор, элементы которого представляют собой количества требований в соответствующих $m_i(s)$ партиях. Эффективное решение, формируемое на верхнем уровне планирования – это количество и состав партий для требований определенного i-го типа, а также соответствующий ему эффективный порядок обработки этих партий в многостадийной системе. Вид эффективного решения, формируемого системой планирования – совокупность матриц $(P)^*$ и $(R)^*$,– представлен в следующей форме: $[(P)^*, (R)^*]$. Для расчёта критерия $f_{\hat{a}}(s)$ на верхнем уровне планирования решение, передаваемое с нижнего уровня, имеет вид $[(P), (R)]$ (в этом случае выполняется расчёт всех матриц $(t_{i,j}^{0l})$, при $l = \overline{1, L}$), в соответствии с видом критерия на верхнем уровне для сокращения объёма вычислений решение, передаваемые с нижнего уровня должно быть задано в виде $[(P), (t_{i,j}^{0L})]$.

В силу того, что порядок партий в последовательностях π^l одинаков, количество типов требований в множестве N ограничено, тогда является ограниченным и число возможных решений. В тоже время при реализации градиентного метода формирования решений на нижнем уровне (по аналогии с [4]) первые шаги алгоритма, связанные с: 1) размещением добавленной в конец последовательности π^l партии; 2) изменением её положения в π^l на одну позицию ближе к началу последовательностей,– могут не гарантировать $^{-}\nabla f_н < 0$ (где $^{-}\nabla f_н$ – левый дискретный градиент целевой функции f_i [4]). В результате формирование новых решений, связанных с изменением позиции добавленной в π^l партии, выполняться не будет (эффективное решение может быть не получено). В силу сказанного на нижнем уровне принятия решений для формирования порядка обработки партий использован метод окрестности, связанный с поиском некоторого более эффективного решения в окрестностях текущего. При этом метрика окрестности исходного решения вычисляется по формуле: $w_1(s+g) = \sum_{i=1}^{n} \sum_{j=1}^{m(s)} | p_{i,j}(s+g) - p_{i,j}(s) | / 2$, где n – количество типов требований, $m(s) = \sum_{i=1} m_i(s)$ – общее количество партий требований разных типов, добавленных до s-го (и включая его) шага алгоритма. Метрика $w_1(s+g)$ определяет число элементов в матрице *(P(s+g))* (число позиций в последовательности π^l), в которой изменены значения по отношения к исходному решению *(P(s))*. Начальное решение *(P(s))* формируется путём добавления партии требований в количестве $a_h(s)$ $(h = \overline{1, m_i(s)})$ в конец последовательностей π^l $(l = \overline{1, L})$. Через g

обозначен номер промежуточного решения *(P(s+g))*, сформированного на основе исходного *(P(s))*, находящегося в его окрестности $[\delta P(s)]$, характеризуемой метрикой $w_1(s+g)$. Формирование решения *(P(s+g))* предполагает изменение положения рассматриваемой партии в последовательности π^l относительно положения этой партии, которое соответствует решению *(P(s))*. Алгоритм определения порядка партий на нижнем уровне планирования (решение верхнего уровня имеет вид $[i, m_i(s), A_i(s)]$) содержит последовательность шагов:

1) некоторое сформированное на предыдущем *(s-1)*-ом шаге алгоритма решение определяется соответствующим видом матриц *(P(s-1))* и *(R(s-1))*, индекс *h* определяющий номер элемента вектора $A_i(s)$ (номер партии $h=\overline{1,m_i(s)}$, требования которой в количестве $a_h(s)$ размещается в последовательностях π^l) инициализируется значением 1;

2) если $m(s-1)=\sum_{h=1} m_i(s-1)$ – количество партий требований, добавленных до *s*-го шага алгоритма, тогда для партии требований *i*-го типа (задан в текущем решении, поступающем с верхнего уровня) определяется индекс столбца $v^{\max}=[m(s-1)+1]$ для матриц *(P(s)),(R(s))*, в котором будут размещаться параметры, соответствующие добавляемой партии;

3) добавляемая в последовательности π^l партия требований *i*-го типа в количестве a_h элементов ($h=\overline{1,m_i}$) размещается в конце последовательностей π^l; для этого в *i*-ой строке $v^{\max}$-го столбца матриц *(P)* и *(R)* задаются значения, соответствующие этой партии (элементы $p_{i,v^{\max}}$ и $r_{i,v^{\max}}$ инициализируются значениями 1 и a_h); модификация матриц *(P(s-1)),(R(s-1))*, связанная с инициализацией элементов $v^{\max}$-ых столбцов, выполняется следующим образом: $p_{k,v^{\max}}=1$, при *k=i*, $k=\overline{1,n}$; $p_{k,v^{\max}}=0$, при *k≠i*, $k=\overline{1,n}$; $r_{k,v^{\max}}=a_h$, при *k=i*, $k=\overline{1,n}$; $r_{k,v^{\max}}=0$, при *k≠i*, $k=\overline{1,n}$;

4) значение *g* индекса шага текущего промежуточного решения инициализируется 1 (*(s+g)* – номер промежуточного шага алгоритма, связанного с определением локального эффективного решения по размещению партии требований *i*-го типа в количестве a_h в последовательностях π^l в окрестности текущего *s*-го решения), индексу текущего рассматриваемого столбца *v* матриц *(P(s))* и *(R(s))* присваивается значение $v^{\max} (v=v^{\max})$;

5) на основе матриц *(P(s))* и *(R(s))* реализуется вычисление элементов матриц $(t^{0l}(s))$ $(l=\overline{1,L})$, матрицы $(t^{0l}(s))$ вычисляются в порядке, начиная с *l*=1 до *l*=L;

6) для полученного вида матриц *(P(s))*, *(R(s))* и матрицы $(t^{0l}(s))$ определяется значение критерия $f_н(s)$ и значения элементов матрицы $(t^{nl}(s))$;

7) в $\pi^l\ (l=\overline{1,L})$ изменяется порядок партий таким образом, что рассматриваемая партия требований *i*-го типа перемещается на одну позицию в начало $\pi^l(l=\overline{1,L})$; выполняемые действия с матрицами *(P)* и *(R)* имеют вид: $p_{i,v-1}(s+g)=1$, $p_{i,v}(s+g)=0$, $p_{k,v-1}(s+g)=0$, $p_{k,v}(s+g)=1$, $r_{i,v-1}(s+g)=r_{i,v}(s+(g-1))$, $r_{i,v}(s+g)=0$, $r_{i,v-1}(s+g)=0$, $r_{i,v}(s+g)=r_{i,v-1}(s+(g-1))$, где k – индекс строки в матрице *(P(s+(g-1))*, в которой элемент $p_{i,v-1}$ в *(v-1)*-ом столбце равен 1 (предыдущая позиция для рассматриваемой партии, которую она занимает на *(s+g)*-ом шаге алгоритма); в результате формируются модифицированные матрицы *(P(s+g))* и *(R(s+g))*; индекс *v* текущего рассматриваемого столбца, идентифицирующий местоположение в π^l рассматриваемой партии, модифицируется: $v=v-1$;

8) с использованием матриц *(P(s+g))*, *(R(s+g))* вычисляются матрицы $(t^{0l}(s+g))$, $(t^{nl}(s+g))$, значение критерия $f_н(s+g)$;

9) при $f_н(s)>f_н(s+g)$ получено решение более эффективное, чем текущее, тогда полученное решение [*P(s+g)),(R(s+g)*] интерпретируется как локально оптимальное $[(P)^*,(R)^*]$, значение *s=s+1*, значение индекса *g* шага поиска следующего локального оптимального решения в окрестности текущего $[(P)^*,(R)^*]$ задаётся равным 1, положение элементов $p_{i,v}=1$ и $r_{i,v}=a_h$ как характеризующих некоторое локальное эффективное решение для партии требований *i*-го типа в количестве a_h фиксируется; выполняется переход к шагу 5;

10) если $f_н(s)<f_н(s+g)$, проверяется выполнение условия $w_1(s+g)\le w_1^{\max}$ ($w_1^{\max}$ - максимально возможная окрестность исходного решения *[P(s)),(R(s)]*, в которой выполняется поиск более эффективного решения *[(P(s+g)),(R(s+g))]*;

11) при выполнения условия $w_1(s+g)\le w_1^{\max}$ реализуется дальнейший поиск более эффективного решения в окрестности $w_1^{\max}$ исходного расписания *[P(s)),(R(s)]*, модифицируется индекс *g* промежуточного шага поиска решения *(g=g+1)* , если *v≠ 1* (рассматриваемая партия является первой в π^l ($p_{i,1}=1$, $r_{i,1}=a_h$)) выполняется переход к шагу 7,

12) если условие $w_1(s+g)\le w_1^{\max}$ не выполняется, полученное локально эффективное решение $[(P)^*,(R)^*]$ фиксируется (фиксируется полученный в результате порядок партий);

13) модификация значений $m_i=m_i-1,\ h=h+1$; если $m_i>0$, то *s=s+1*; $v^{\max}(s+1)=v^{\max}(s)+1$; выполняется переход к шагу 3.

Реализация алгоритма формирования порядка обработки партий позволяет выполнить упорядочивание партий требований в количестве m_i (в каждой из которых a_h требований $(h=\overline{1,m_i})$) в последовательностях π^l, определить расписание обработки партий на приборах.

Алгоритм оптимизации состава партий, реализуемый на верхнем уровне планирования, предполагает (с учётом идеологии «жадных» алгоритмов): 1) формирование эффективного решения для количества и состава партий требований типов *i=1,2* (требований с минимальными идентификаторами в *N*); 2) формирование эффективного количества партий и их составов для требований типа *i>2*. Естественно, что формирование количества и состава партий требований соответствующих типов непосредственно связано с определением порядка обработки партий в последовательностях π^l. Для определения эффективного решения по количеству и составу партий требований с *i=1,2* введены обозначения: $m_1(s)$ – количество партий требований 1-го типа, рассматриваемое на s-ом шаге алгоритма; $G_в$ – глобальное значение целевой функции $f_в$, получаемое при оптимизации числа партий и их состава для требований типов *i=1,2*. Для определения эффективных количества и состава партий при *i=1,2* предложен следующий метод дискретной оптимизации: при определении количества и состава партий требований типа *i=1* реализуется метод окрестности с учётом глобального значения целевой функции $G_в$ при переходе от меньшего количества партий к большему; при определении количества и состава партий требований с *i=2* реализован градиентный метод, предполагающий уменьшение значения целевой функции $f_â$ по одному из направлений (определение $^{-}\nabla_h f_в(s)$ при увеличении числа требований в соответствующей *h*-ой партии $(h=\overline{2,m_2})$, т.е анализ изменения значения целевой функции $f_в(s)$ в каждом из *h*-ых направлений в связи с изменением числа требований в соответствующей *h*-ой партии ($h=\overline{2,m_2}$)). При этом поиск эффективных решений для требований типов *i=1,2* реализуется одновременно.

Для требований типа *i=1* выполняется поиск более эффективного решения в окрестности исходного (некоторого локально эффективного). Окрестность ограничивает решения с одинаковым (с исходным) количеством партий (равным m_1) и различным (по количеству требований) их составом. При невозможности определения более эффективного решения в задаваемый окрестности (среди решений с одинаковым значением m_1), выполняется переход к поиску эффективного решения в группе с большим количеством партий (m_1+1). Таким образом, внутри каждой группы выполняется поиск локально эффективных решений (локального экстремума) после чего определяется глобально эффективное решение (среди групп решений) путём сравнения со значением глобального экстремума $G_в$. При этом локально эффективное решение для отдельных групп с одинаковыми m_1 определяются путем: 1) фиксации определённого количества партий требований типа *i=1*; 2) задания количества требований в каждой партий типа *i=1*; 3) формирования решений с различным количеством и составом партий требований типа *i=2*; 4) выбор такого количества и состава партий требований типа *i=2*, которые при данном зафиксированном количестве и составе партий требований типа *i=1* будут обеспечивать локальный экстремум целевой функции $f_в(s)$ (определение эффективного решения $[2, m_2^*, A_2^*]$ с использованием градиентного алгоритма, рассматриваемого ниже), 5) переход к другому составу партий требований типа *i=1*, в результате– определение эффективного решения по количеству и составу партий требований типа *i=1* внутри группы решений с одинаковыми значениями параметра m_1. Определение локально эффективного решения среди решений одной группы (с одинаковым числом партий m_1) для *i=1* выполняется с использованием метрики окрестности решений, вычисляемой по выражению вида: $w_2(s+g) = [\sum_{h=1}^{m_1} |a_h(s+g) - a_h(s)|]/2$, где g – индекс промежуточного шага поиска более эффективного (чем исходное решение $[1, m_1(s), A_1(s)]$) состава партий внутри группы с одинаковым значением m_1. При построении эффективного решения для требований типа *i=1,2* предполагается, что порядок партий требований типа *i=1* остаётся неизменным (т.е партии требований типа *i=1* следуют друг за другом в фиксированном порядке).

Формулировка алгоритма метода окрестности для определения эффективного решения по количеству и составу партий требований типа *i=1* содержит последовательность шагов: 1) задание значения параметра $m_1 = 2$, начального глобального эффективного значения $G_â$ целевой функции $f_в$ равным M ($G_в = M$,где M – достаточно большое число);

2) задание значения $f_в(0) = M$, значения номера шага *s=1*, значения индекса шага поиска более эффективного решения *g=0* (в окрестности текущего решения $[1, m_1(s), A_1(s)]$ определяются решения $[1, m_1(s+g), A_1(s+g)]$ с целью исследования их большей эффективности);

3) формируется состав партии требований типа *i=1* для соответствующего значения m_1 следующим образом: $a_h = 2$ при $h = \overline{2, m_1}$, $a_1 = n^i - \sum_{h=1}^{m_1} a_h$; если выполняется условие окончания процесса изменения количества партий и формирования их состава в виде $a_1 < 2$, т.е. корректный состав партий в количестве m_1 задан быть не может, то выполняется переход к шагу 12;

4) если условие $a_h < a_1 (h = \overline{2, m_1})$ выполняется, формируется решение $[1, m_1(s), A_1(s)]$, для которого определяется начальный вид матриц *(P(s))* и *(R(s))* следующим образом: $p_{1,h} = 1, r_{1,h} = a_h$ при $h = \overline{1, m_1}$; $p_{k,h} = 0$, $r_{k,h} = 0$ при *k≠1*, $h = \overline{1, m_1}$ (т.е. последовательность партий требований типа *i=1* (матрицы *(P(s))* и *(R(s))* формируется автоматически при получении решения

$[1, m_1(s), A_1(s)]$, последовательность партий требований типа *i=1* не изменяется в дальнейшем); если условие $a_h < a_1(j = \overline{2, m_1})$ не выполняется, то переход к шагу 11;

5) для *i=2* выполняется обращение к градиентному алгоритму определения эффективного количества и состава партий, в результате реализации которого партии требований типа *i=2* (в различном количестве m_2 и составе A_2) размещаются эффективным образом в последовательностях π^1 (характеризуемых матрицами *[P(s))*, *(R(s)]*, образованными партиями требований типа *i=1*); результатом обращения к градиентному методу является решение $([1, m_1, A_1(s+g)][2, m_2^*, A_2^*])$ и соответственно $[(P(s))^*, (R(s))^*]$ для рассматриваемого количества партий m_1, их состава $A_1(s)$; для полученного решения $([1, m_1, A_1(s+g)][2, m_2^*, A_2^*])$ вычисляется значение $f_в(s+g)$;

6) если $f_в(s+g) < f_в(s)$, то полученное решение более эффективно, в этом случае полученное решение фиксируется как локально эффективное в группе решений (т.е. $([1, m_1^*, A_1^*][2, m_2^*, A_2^*]) = ([1, m_1, A_1(s+g)][2, m_2^*, A_2^*])$ и, соответственно, $[(P)^*, (R)^*] = [(P(s+g)), (R(s+g)]]$), характеризуемых одинаковым значением параметра m_1; в окрестностях $([1, m_1^*, A_1^*][2, m_2^*, A_2^*])$ для требований типа *i=1* будет в дальнейшем выполняться поиск более эффективного решения; задание значения номера промежуточного шага поиска эффективного решения в окрестности текущего *g=0*, номер шага *s=s+1*; переход к шагу 10;

7) если $f_в(s+g) > f_в(s)$, то выполняется вычисление метрики $w_2(s+g)$ полученного решения (меры отличия решения $[1, m_1, A_1(s+g)]$ от решения $[1, m_1, A_1(s)]$), если $w_2(s+g) > w_2^{\max}$, то текущее решение $[1, m_1^*, A_1^*]$ фиксируется как локально оптимальное, которое совместно с соответствующим решением $[2, m_2^*, A_2^*(s)]$ характеризуется значением $f_{â}^*(s)$; переход к шагу 10;

8) если метрики $w_2(s+g) \le w_2^{\max}$, то *g=g+1*, выполняется переход к шагу 9;

9) в соответствии с алгоритмом формирования состава партий при неизменном значении m_1 выполняется формирование состава партий (вектор $A_1(s+g)$); выполняется переход к шагу 4;

10) сравнение значений $f_в^*(s)$ для локально эффективного решения $([1, m_1^*, A_1^*][2, m_2^*, A_2^*])$ со значением глобального экстремума (значение глобального экстремума $G_в$); если $f_в^*(s) < G_в$, то текущее локально эффективное решение $([1, m_1^*, A_1^*][2, m_2^*, A_2^*])$ фиксируется как глобально эффективное для требований типа *i=1,2*;

11) изменение значения $m_1 = m_1 + 1$; переход к шагу 2;

12) получено глобально эффективное решение $([1, m_1^*, A_1^*][2, m_2^*, A_2^*])$ по количеству и составу требований типа *i=1,2*, характеризуемое

значением $f_в^*(s) = G_в$, которому соответствует вид последовательностей π^l в виде матриц $[(P)^*,(R)^*]$.

Дальнейшие шаги алгоритма направлены на определение эффективного количества и состава партий для требований типа *i>3* (последовательности π^l (матрицы $[(P)^*,(R)^*]$) дополняются эффективными по составу партиями требований типа *i>3*). Алгоритм градиентного метода определения эффективного количества и состава партий требований *i*-го типа предусматривает выполнение следующих шагов:

1) задание для m_i значения 2 ($m_i = 2$); определение начального состава партий: $a_h = 2$; $h = \overline{2, m_i}$; $a_1 = n^i - \sum_{h=2}^{m_i} a_h$, на основе количества и состава партий формируется решение $[i, m_i, A_i]$;

2) полученное на верхнем уровне планирования решение $[i, m_i, A_i]$ предаётся на нижний уровень принятия решений; на нижнем уровне выполняются определение эффективного расписания обработки партий в виде: $[(P)^*,(R)^*,\{(t^{0l})^* \mid l = \overline{1, L}\}]$ (в виде $[(P)^*,(R)^*,(t^{0L})^*]$), которое передаётся на верхний уровень для соответствующего решения $[i, m_i = 2, A_i]$; на основе решения $[(P)^*,(R)^*,(t^{0L})^*]$ выполняется вычисление начального (для данной группы решений, характеризуемых одинаковыми значениями m_i) значения целевой функции $f_в$ (определение значения $f_в$ для сформированного начального состава партий);

3) задание значения идентификатора *h* партии, в которой изменяется состав требований, равным 2 *(h=2)*;

4) для партии с рассматриваемым номером *h* – увеличение количества требований на 1 ($a_h(s+1) = a_h(s) + 1$), формирование нового состава партий: $a_k(s+1) = a_k(s)$ при $k = \overline{2, m_i}$, $k \neq h$, $a_1(s+1) = n^i - \sum_{h=1}^{m_i} a_h(s+1)$; если выполняется условие окончания процесса формирования состава партий $a_h(s+1) < a_1(s+1)$, то переход к шагу 8; если $a_h(s+1) > a_1(s+1)$ ($h = \overline{2, m_i}$) не выполняется, то формирование нового решения $[i, m_i, A_i(s+1)]$, формирование эффективного расписания в виде $[(P(s+1)^*),(R(s+1)^*),(t^{0L}(s+1))^*]$ на нижнем уровне, передача этого решения на верхний уровень планирования; расчёт для решения $[i, m_i, A_i(s+1)]$ на основе решения с нижнего уровня $[(P(s+1)^*),(R(s+1)^*),(t^{0L}(s+1))^*]$ значения критерия $f_в(s+1)$;

5) определение значения левого дискретного градиента ${}^{-}\nabla_h f_в(s) < 0$ в виде: ${}^{-}\nabla_h f_в(s) = (f_в(s+1) - f_в(s)) < 0$ при условии $f_в(s+1) < f_в(s)$, где *h* – номер партии (аналог направления *h* изменения значения целевой функции);

6) если получено ${}^{-}\nabla_h f_в(s)<0$, то сформированное решение $[i,m_i,A_i(s+1)]$ является более эффективным, чем решение $[i,m_i,A_i(s)]$, $s=s+1$, выполняется переход к шагу 4;
7) если условие ${}^{-}\nabla_h f_в(s+1)<0$ не выполняется (для решения $[i,m_i,A_i(s)]$ получено значение правого дискретного градиента ${}^{+}\nabla_h f_в(s)=(f_в(s+1)-f_в(s))>0$), то изменение состава *h*-ой партии требований прекращается (изменение состава *h*-ой партии требований не позволяет оптимизировать значение функции $f_в$ в *h*-ом направлении), выполняется модификация значения номера партии *h=h+1*, решение $[i,m_i,A_i(s)]$ рассматривается как локально эффективное решение ($[i,m_i^*,A_i^*]=[i,m_i,A_i(s)]$), оптимизируемое при изменении состава (*h=h+1)*-ой партии, $s=s+1$; если $h\le m_i$, то выполняется переход к шагу 4; если $h>m_i$, то выполняется переход к шагу 8;
8) в рамках группы решений с одинаковым значением m_i получено локально эффективное решение $[i,m_i^*,A_i^*]$, значение целевой функции $f_в^*$ для которого сравнивается с глобальным значением целевой функции $G_в$, если $f_в^*<G_в$, то текущее решение $[i,m_i^*,A_i^*]$ фиксируется как глобально эффективное (с сохранением соответствующего решения $[(P)^*,(R)^*]$);
9) выполняется модификация числа партий $m_i=m_i+1$; формируется новый состав партий в виде: $a_h=2;(h=\overline{2,m_i});a_1=n^i-\sum_{h=2}^{m_i}a_h$; если для определенного состава партий не выполняется условие прекращения процесса формирования партий и их состава в виде $a_1<2$, то для этого состава партий формируется решение в виде $[i,m_i(s),A_i(s)]$, которое является исходным (начальным) при исследовании эффективности состава партий с заданным таким образом их числом m_i, выполняется переход к шагу 2;
10) если при определении начального состава партий $a_1<2$, то выполняется условие прекращения процесса формирования начального состава партий, тогда новый состав партий в количестве m_i сформирован быть не может, тогда $[1,m_i^*,A_i^*]$- полученное глобальное эффективное решение по определению числа m_i^* и состава партий $(A_i)^*$требований *i*-го типа, а соответствующее решение на нижнем уровне $[(P^*),(R^*)]$ является глобально эффективным решением, определяющим порядок обработки партий всех типов требований (до *i*-го включительно), которое будет использоваться в дальнейшем для размещения в последовательностях π^l новых требований.

Графическая интерпретация градиентного алгоритма определения количества m_i и состава партий (A_i) представлена на рис. 5 (для случая 12 требований типа *i*).

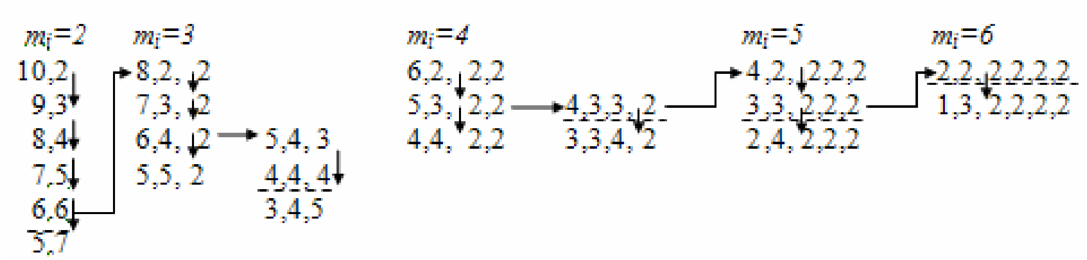

Рисунок 5 – Пример реализации градиентного алгоритма метода определения эффективного количества и состава партий

При $m_i = 2$ переход от состава партий $a_i = (6{,}6)$ к составу партий $a_i = (5{,}7)$ невозможен, т.к. элемент $a_2 > a_1$ и процесс изменения состава партий для $m_i = 2$ должен быть завершен, для $m_i = 3$ переход от состава партий (6,4,2) к составу партий (5,5,2) не обеспечивает $^{-}\nabla_h f_в(s) < 0$, изменяется состав партии, номер которой *h=3*, увеличение числа требований в партии с *h=3* обеспечивает выполнение $^{-}\nabla_h f_в(s) < 0$ до тех пор, пока не выполняется условие прекращения формирования состава партий при $m_i = 3$ ($a_3 > a_1$), тогда выполняется поиск эффективного состава партий при дальнейшем изменении (увеличении) m_i. Понятно, что эффективные решения внутри каждой группы сравниваются со значением $G_в$, в результате выбирается такое количество и состав партий требований типа *i*, которые гарантируют глобальный $min\, f_в$.

После того, как сформулирован градиентный алгоритм определения количества и состава партий требований *i*-го типа, должен быть сформулирован обобщённый алгоритм определения числа и составов для всех *i*-ых требований (*i*≥*3*). Действия, связанные с выполнением данного алгоритма определены следующим образом: 1)извлечение из множества *N* типов требований *i*-го типа, определяемого условием: $i = \min_k \{k \mid k \in N\}$, *N(s)=N(s-1)\{i};* 2) на верхнем уровне планирования осуществляется обращение к градиентному методу определения эффективного количества и состава партий требований *i*-го типа; передаваемые данные в градиентный метод определения количества и состава партий – тип рассматриваемых требований и их общее количество n^i, результатом обращения к градиентному методу являются локально оптимальные решения для верхнего и нижнего уровней в виде: $[i, m_i^*, A_i^*]$ и $[(P^*), (R^*)]$; 3) локально оптимальные решения $[(P^*), (R^*)]$ и $[i, m_i^*, A_i^*]$ фиксируются, если $N \neq \varnothing$, то выполняется переход к шагу 1.

Выполненное обоснование модели двухуровневого программирования определения количества и состава партий требований при групповой обработке, а также метода формирования составов партий позволяет выполнить идентификацию эффективных составов партий, отличных от фиксированных составов партий в существующих методах.

Определение как состава партий, так и порядка их обработки выполняется для многостадийной обрабатывающей системы с произвольным числом приборов. Развитие рассмотренного подхода связано с определением групп обрабатываемых партий при построении сменно-суточных заданий в групповой обработке.

Библиографический список

1.Петросян Л.А. Теория игр./ Л.А.Петросян, Н.А.Зенкевич, Е.А.Семина. – М.: Изд-во «Высшая школа», 1999. – 300с.

2. Гермейер Ю.Б. Игры с непротивоположными интересами. / Ю.Б.Гермейер. – М.: Наука, 1976. – 327 с.

3. Ковалев М.М. Модели и методы календарного планирования. Курс лекций./ М.М. Ковалев. – Минск: Изд-во БГУ, 2004.– 63 с.

4. Кротов К.В. Градиентный метод составления расписаний в многостадийной системе с одинаковым порядком обслуживания требований и одинаковым временем поступления./ К.В.Кротов // Сборник научных трудов «Оптимизация производственных процессов».Вып.12- Севастополь: Изд-во СевНТУ, 2010.-с.154-160 .

Надійшла до редакції 09.09.2011.
Рецензент: канд. техн. наук, доц. Красник М.Ю.

К.В.Кротов
Севастопольський національний технічний університет

Модель дворівневого програмування і метод побудови розкладів групової обробки. Обґрунтовується модель дворівневого програмування для формування розкладів групової обробки, а також метод формування складу партій та порядку їх обробки в багатостадійних конвеєрних системах.
Ключові слова: модель дворівневого програмування, групова обробка, метод формування складу партій, порядок обробки партій, багатостадійна конвеєрна система

K.V.Krotov
Sevastopol National Technical University

The model of two-level programming and the method of construction of schedules of group processing. The model of two-level programming of the construction of schedules of group processing and the method of construction of the composition of the parties and the order of their processing in multiphase conveyor systems are designed.
Keywords: model of two-level programming, schedules of group processing, method of construction the composition of parties, order of processing of parties, multiphase conveyor systems