# ВВЕДЕНИЕ

Современные способы повышения производительности выполнения программ при обработке данных связаны с использованием вычислительных кластеров и распределенных вычислительных систем – GRID-систем. Повышение производительности выполнения программ связано с их параллельной реализацией в составе кластера, либо с распределенным выполнением в GRID-системах. Одним из возможных способов повышения производительности выполнения программ обработки данных в составе кластера является их (программ) конвейеризация. Т.е. обработка данных соответствующих им программами осуществляется в составе программного макроконвейера. Конвейеризированное выполнение программ (обработка данных) соответствует классу программных систем МПОД [1] (много программ– одни данные), что предполагает организацию обработки одного потока данных последовательностью программ (последовательностью фрагментов программ) – реализацию программного конвейера. Развитие идей конвейеризации выполнения программ предполагает обработку потоков данных различных типов (в общем случае n типов) соответствующими каждому из этих типов, данных программами при реализации последовательного обмена данными между вычислительными сегментами программного конвейера (обмен данными между программами выполняется только после окончания их обработки на сегментах конвейера).

Конвейеризация программ предполагает их разделение на фрагменты, каждый из которых закреплен для выполнения за соответствующим сегментом конвейера (вычислительным устройством, обрабатывающим прибором). Введем в рассмотрение следующие обозначения: i – номер множества однотипных данных, характеризующих одинаковые объекты, которые должны быть обработаны в системе, I – множество всех данных, которые будут обработаны в вычислительной системе, n – количество множеств данных, тогда ),  – количество элементов в множестве однотипных данных, характеризуемых индексом i (множество содержит данные об  одинаковых объектах, в общем виде ). Данные, входящие в некоторое i-е множество, обрабатываются соответствующей им программой. Индекс i соответствует программе, выполняемой в составе конвейера, обрабатывающей данные i-го типа (соответствует типу выполняемой в составе конвейера программы, обрабатывающей данные i-го типа). Однократное выполнение конвейеризированной программы i-го типа обеспечивает обработку одного элемента множества данных i-го типа. Если множество данных i-го типа содержит  элементов, то обрабатывающая эти данные программа должна быть выполнена в конвейерной системе  число раз. Цель функционирования конвейерной системы в этом случае состоит в обработке поступающих на ее вход данных выполняющимися в системе конвейеризированными программами. При этом предполагается, что выполняющие обработку данных конвейеризированные программы находятся в оперативной памяти каждого из сегментов конвейера (загружены в оперативную память вычислительных устройств кластера). Тогда управление вычислительным процессом в конвейерных системах предполагает определение порядка запуска программ обработки данных на выполнение. Т.к. объемы вычислений на каждом сегменте различны, являются различными длительности выполнения программ на соответствующих сегментах, тогда может быть сформировано расписание выполнения конвейеризированных программ обработки соответствующих данных, представляющее собой порядок запуска программ на выполнение. Определение подхода по конвейеризации обработки данных выполняющимися в многостадийных системах программами рассматривается в [2-4].

Постановка задачи управления вычислительным процессом, рассматриваемой в данной работе, предполагает, что при  () реализуется обработка единичных данных (однократный запуск на выполнение соответствующих программ), для действий с которыми должно быть сформировано расписание выполнения программ. Введем обозначение для интервала времени функционирования системы при обработке данных в виде , где z–индекс временного интервала, тогда , где Z– количество временных интервалов , в течение которых выполняется обработка данных. В этом случае управление вычислительным процессом предполагает формирование совокупности данных, обрабатываемых в течение каждого из временных интервалов и расписаний обработки данных каждой из этих совокупностей. Введем в рассмотрение понятия задания на обработку как совокупности единичных данных, обрабатываемых в течение одного временного интервала . Для задания на обработку введем в рассмотрение обозначение , тогда должно быть сформировано Z заданий (т.е.). Тогда управление вычислительным процессом состоит в формировании эффективных составов заданий  () на обработку данных и расписаний обработки данных этих заданий. Т.к. задачи теории расписаний при количестве приборов более трех и отсутствии каких либо ограничений на длительности обработки данных являются трудно разрешимыми (NP–трудными, [5]), тогда для их решения должны быть применены приближенные методы.

# АНАЛИЗ ЛИТЕРАТУРЫ

Развитие современных методов построения расписаний обработки партий представлено в работах [2-11]. В [2] выполнена классификация задач управления обработкой партий (построения расписаний обработки партий). Задачи различаются по виду процесса обработки (непрерывный либо дискретный), способам представления времени моделирования (непрерывное либо дискретное), способам формирования партий и т.д. Управление обработкой партий предполагает построение расписаний для фиксированных партий, определение количества и размера партий до реализации процедуры построения расписаний (алгоритм определения составов партий никак не связан с характеристиками оборудования и процессом обработки, составы партий определяются без связи с построением расписаний), определение размеров партий совместно с решением задачи построения расписаний их обработки. В [2] рассматривается задача формирования партий и распределения их по обрабатывающим приборам при дискретном времени моделирования для непрерывного производства. При ее решении использован аппарат частично целочисленного линейного программирования (ЧЦЛП). Большая размерность модели (наличие тридцати одного ограничения), использование методов ЧЦЛП, задание ограниченного количества обрабатывающих приборов (ограниченного количества ресурсов) делают затруднительным определение решения для задач большой размерности за ограниченное время. Решение задач обработки партий рассматривается в работе [3], однако указанные там методы построения расписаний предполагают наличие фиксированных партий данных при их обработке на ограниченное количестве приборов (задача определения оптимальных составов партий в этой работе не рассматривается). В работах [4,5] рассматривается задача управления обработкой партий в непрерывном (химическом) производстве. Под партиями в этом случае подразумевается объемы материалов, участвующих в процессе производства (объемы партий определяются способом производства требуемого вида продукта, количеством конечного продукта, определяемым в соответствии с спросом). Таким образом, в [4,5] реализуется распределение обработки партий (размер которых не оптимизируется) материалов различных видов, обеспечивающих выпуск продуктов разных типов, по параллельно действующим машинам. При этом рассматриваются задачи ограниченной размерности (2 продукта, 2-3 прибора в обрабатывающей конвейерной системе). В работе [6] рассматривается решение задачи среднесрочного планирования выпуска продукции при ограниченном количестве ресурсов и последующего составления расписаний обработки сформированных при планировании партий. При этом под партией подразумевается совокупность изделий одного типа, выпуск которых закрепляется за определенным предприятием или производственным участком. Размеры партий определяются в соответствии с заказами на производство и директивными сроками их выпуска. Модель оптимизации составов партий учитывает только стоимостные параметры выпуска продукции, но не учитывает временные характеристики и особенности технологических процессов. На основе полученного решения по распределению заказов по производственным участкам (сформированным составам партий изделий) решается задача выделения для них ресурсов с целью обработки. Совместно задача планирования (определения составов партий) и управления выпуском (формирования расписаний обработки партий) в работе не решается. В [6] использованы модель большой размерности и аппарат ЧЦЛП, что ограничивает размерность решаемой задачи. В работе [7] решается задача определения количества и составов партий единичных (разнотипных) требований, обрабатываемых на одном приборе, с заданными директивными сроками обработки и стоимостью доставки партий. Составы партий разнотипных требований формируются с учетом директивных сроков. Задача предполагает наличие одного обрабатывающего прибора, для которого отсутствуют простои при обработке партий. Тогда формирование партий предполагает и одновременное автоматическое построение расписания их обработки (т.е. вопрос оптимизации использования ресурсов обрабатывающих приборов в данной работе не рассматривается). В результате решается только задача определения составов партий на основе заданных директивных сроков окончания обработки требований. Аналогичная задача формирования партий требований в соответствии с их директивными сроками завершения, обрабатываемых на параллельных машинах, рассматривается в [8]. Объединение требований в партии реализуется с использованием эвристической процедуры в соответствии со значениями параметров начала обработки и директивными сроками окончания обработки требований. Т.к. обработка двух партий на одной машине (приборе) не может пересекаться, тогда сроки начала и окончания обработки партий используются для их распределения по параллельным машинам (расписание вытекает из сформированных составов партий). В работе [9] решается задача планирования производства полупроводников, предполагающая совместное формирование составов партий обрабатываемых пластин разных типов и расписаний обработки партий в конвейерной системе с параллельно действующими машинами. При определении локально оптимальных решений по составам партий и расписаниям их обработки используется один обобщенный критерий, в котором совместно учитываются эти решения. Особенностью аппарата формирования составов партий является использование метода отжига. Реализуется поиск в окрестностях текущего локально эффективного решения, для формирования новых решений, входящих в окрестности, введены эвристические правила, оперирующие со случайно выбираемыми партиями, изменяющие как составы партий, так и расписания их обработки (изменение позиции партии, изменение обрабатывающего прибора для выполнения операции, создание новых партий). На основе сформированного совместного решения по составам партий и расписаний их обработки выполняется оценка эффективности полученного расписания с использованием дизъюнктивного графа. Предложенный в [[[1]](#footnote-1)9] подход позволяет реализовывать стохастический поиск локально оптимальных решений. Планированию производственного процесса посвящена работа [10], в которой решается задача распределения заказов на производство продукции по сменным заданиям, распределения заданий по партиям и формирование расписаний обработки партий. Для определения составов сменных заданий используется эвристическая процедура, для определения составов партий разработана имитационная процедура, позволяющая моделировать прохождение партиями конвейерной системы. Определение эффективных составов партий предполагает задание параметров имитационной модели, соответствующих размерам партий изделий каждого типа, и проведения моделирования. В соответствии с результатами моделирования выбираются значения параметров размеров партий, обеспечивающие минимальное значение критерия. После того как с использованием моделирования определены оптимальные размеры партий, выполняется формирование расписаний их обработки. Работа [11] реализует решение задачи определения составов партий компонент, из которых выполняется формирование элементов (аналог формирования комплектов из обработанных в системе изделий). Для формирования партий и расписаний их обработки введена оптимизационная модель, являющаяся многопараметрической и многоиндексной. Определение решений по количеству и составам партий осуществляется в работе путем полного перебора возможных значений этих параметров. Расписание для полученного решения формируется посредством использования эвристической процедуры. В итоге при большой размерности задачи (значительное число типов компонент и количество компонент каждого типа) прямой перебор при формировании партий не обеспечивает решение поставленной задачи за ограниченное время.

Таким образом, решение задачи определения составов партий и построения расписаний их обработки реализуется путем привлечения: 1) аппарата ЧЦЛП (однако при большой размерности задачи получение ее решения за ограниченное время является затруднительным, решения по составам партий формируются без учета расписаний их обработки); 2) методов формирования партий с учетом директивных сроков окончания обработки входящих в них требований (однако формирование групп партий, обрабатываемых в течение заданных интервалов времени, с использованием данных методов является затруднительным); 3) эвристических процедур и правил (однако, применение правил не позволяет получить решения, приближающиеся к оптимальным). В соответствии с этим разработка моделей и методов определения оптимальных составов партий, групп партий и расписаний их обработки является актуальной задачей

ПОСТАНОВКА ЗАДАЧИ

Постановка задачи предполагает наличие ограничений на время функционирования конвейеризированной системы, обрабатывающей разные типы данных, то есть существует некоторое количество интервалов фиксированной длительности, в течение которых производится конвейеризированная обработка поступающих в систему данных. Тогда введём следующие обозначения: через Z обозначим количество интервалов обработки (тогда номер интервала ), при этом длительность интервала обозначим через .

В конвейеризированной системе, которая состоит из L сегментов, реализуется обработка n типов данных n типами программ. Для повышения эффективности использования ресурсов системы рекомендуется обрабатывать данные партиями. В этом случае «партия» - это совокупность наборов данных одного (i-того, ) типа, которые обрабатываются в программе непосредственно друг за другом.

Обработка данных в системе происходит в течение Z интервалов, длительность которых , поэтому возникает необходимость формирования групп – совокупности партий, обрабатываемых в течение одного интервала функционирования конвейеризированной системы.

Так как интервалы обработки данных строго ограничены, то проблема эффективного использования ресурсов системы ставится наиболее остро. Тогда задача составления расписаний для повышения эффективности использования ресурсов системы заключается в определении порядка обработки партий данных в каждой группе. При этом необходимо учесть, что целью работы системы является обработка максимально возможного количества данных разных типов.

Входными данными для системы построения расписаний групповой обработки данных разных типов при наличии ограничений на длительность и количество интервалов обработки являются:

- количество типов данных (n);

- количество элементов в множестве данных каждого типа (, );

- количество (Z) и длительность () интервалов функционирования системы;

- количество сегментов системы (L);

- длительность обработки данных i-того типа l-ым сегментом системы (фрагментом i-той программы).

- Интервалы времени переналадки приборов с обработки данных одного типа на обработку данных другого типа

В процессе функционирования системы необходимо определить приближенно эффективное (с точки зрения вводимых в рассмотрение критериев) количество и составы партий данных, эффективный (с точки зрения обработки максимально возможного количества данных разных типов) состав групп партий данных, а так же эффективное (с точки зрения минимального времени выполения всей группы) расписание обработки партий данных разных типов в группе. Данная задача является сложной, поэтому требуется вертикальная декомпозиция целей, в результате которой задача будет разбита на подзадачи.

В данной работе рассматриваются только аспект работы системы: формирования групп партий. Необходимо разработать метод получения оптимального количества и состава групп партий данных.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать существующие методы построения расписаний обработки партий данных разных типов;

- сформулировать подход к построению расписаний групповой обработки партий при наличии ограничений на длительность интервалов обработки групп;

- выбрать математический аппарат;

- построить математическую модель системы;

- разработать метод формирования эффективного количества и состава партий каждого типа данных, обрабатываемых в системе;

- разработать метод формирования эффективных расписаний обработки групп партий;

- исследовать разработанные методы и сделать выводы об эффективности разработанных алгоритмов и областях возможного применения системы в целом.

ОБОСНОВАНИЕ ПОДХОДА К РЕШЕНИЮ ЗАДАЧИ ПОСТРОЕНИЯ РАСПИСАНИЙ ГРУППОВОЙ ОБРАБОТКИ ДАННЫХ РАЗНЫХ ТИПОВ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЯ НА ИНТЕРВАЛ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ

Задача построения расписания групповой обработки партий при наличии ограничений имеет обобщённую цель – обработать максимальное количество данных. В то же время необходимо учитывать ограниченность временных интервалов обработки групп. Данная задача трудноразрешима. В соответствии с подходом вертикальной декомпозиции описанной в [9] достижение цели может быть представлено в виде иерархии подцелей. Тогда при разбиении обобщённой цели на подцели имеют место следующие свойства иерархической обработки:

- приоритетность решений, это свойство следует из необходимости передачи данных с уровня на уровень;

- зависимость эффективного решения на вышестоящем уровне от решения на нижестоящем уровне;

- достижение обобщённой (внешней, глобальной) цели системы возможно только при достижении всех подцелей (внутренних, локальных целей).

В результате декомпозиции обобщённой цели сформирована (определена) трёхуровневая иерархически-упорядоченная структура поиска решения задачи (с локальными подцелями на каждом уровне), изображённая на рисунке 2.1.

Формирование количества и состава партий данных каждого типа

Формирование состава групп с учётом ограничения на время обработки

**Цель**: максимизировать количество обработанных данных

*Верхний уровень*

*Средний уровень*

**Цель**: минимизировать общие простои оборудования

**Цель**: минимизировать простои оборудования при обработке партий в группе

Состав и количество партий

Расписание обработки поступившей группы

Состав групп для поступившего состава и количества партий

Формирование расписания обработки партий в группах

*Нижний уровень*

Состав группы

Рисунок 2.1 – Структурная схема системы построения расписаний обработки партий данных при формировании групп и наличии ограничений на длительность функционирования приборов

Для оценки эффективности решений на каждом уровне должны быть введены критерии оценки, которые должны учитывать:

- на третьем (нижнем) уровне – эффективность использования оборудования конвейерной системы при размещении рассматриваемой партии (добавляемой в расписание для соответствующей группы) в последовательностях ;

- на втором (среднем) уровне – общую эффективность использования оборудования конвейерной системы при обработке всех партий данных группы  (анализ сформированного состава группы партий на основе построенного для неё расписания с точки зрения эффективности использования временного ресурса системы с учётом ограничений на длительность обработки);

- на первом (верхнем) уровне – общее количество данных, обработанных в системе в течение Z интервалов времени.

# МОДЕЛЬ МНОГОУРОВНЕВОГО ПРОГРАММИРОВАНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ СОСТАВОВ ПАРТИЙ ДАННЫХ, ГРУПП ПАРТИЙ, РАСПИСАНИЙ ОБРАБОТКИ ПАРТИЙ

Обоснование методов оптимизации составов групп партий (также как обоснование в [1] модели многоуровневого программирования построения комплексных расписаний обработки партий данных, обоснование в [2] методов оптимизации составов партий и расписаний обработки партий) обеспечивается введением следующих обозначений: *i* –идентификатор типа обрабатываемых в системе данных, *n –* количество типов обрабатываемых в системе данных (),  – количество данных *i-*го типа, которые должны быть обработаны в системе; l – индекс сегмента конвейерной системы, осуществляющего выполнение l-й части программы (); – интервал времени, в течение которого реализуется обработка партий данных (). В задаче заданными являются значения  (), тогда для обработки однотипных данных формируются партии. Партия – совокупность однотипных данных, обработка которых выполняется без переналадки сегментов конвейера. Партия – фиксированная, если в нее входят все данные *i-*го типа. Партия может содержать не все  данных *i*-го типа, в этом случае в системе обрабатывается ни одна, а несколько партий данных этого типа. Дисциплина обслуживания выполняемых в системе программ предполагает прохождение данными, которые они обраба­тывают, всех сегментов конвейера, при этом если l-й сегмент приступил к обработке данных i-го типа, обработка не может быть прервана. Все обрабатывающие сегменты конвейера характе­ризуются равными и неизменными во времени значениями производительно­сти их работы. Выполнение на каждом l-м сегменте назначенной ему части i‑ой программы характеризуется длительностью обработки данных. Обработка партий данных *i-*ых типов () реализуется в течение заданных интервалов времени функционирования системы  (), тогда на основе решения по количеству и составам партий формируются группы партий. Группа партий– это совокупность партий, обрабатываемых в течение одного интервала времени  функционирования системы (формируется *Z* групп партий).

Для определения решений по составам партий данных *i-*ых типов (), формируемых на первом уровне иерархии, в рассмотрение введены обозначения: – количество партий данных *i*-го типа (), формируемых на первом уровне принятия решений, элементы  образуют вектор *М*; *А–* матрица, элемент  которой – это количество данных *i*-го типа в *h-*ой партии (). Решение, формируемое на первом уровне системы имеет вид: *[М, А],* где *М*– вектор количества партий данных *i-*ых типов, *А*–матрица количества данных в партиях.

Для построения решений по составам групп партий, формируемых на втором уровне иерархии, введены обозначения:  () – группа партий, обрабатываемых в течение интервала (); – количество партий данных *i-*го типа в группе партий ;  – вектор количества данных *i-*го типа в  партиях в группе . Партии данных *i-*го типа, входящие в группу партий , определены с использованием набора параметров вида:, а группа партий – совокупность наборов вида: , где – количество типов данных, партии которых входят в . Решение, формируемое на втором уровне системы – совокупность групп партий, имеет вид: {()}.

В соответствии с полученным решением по составу групп партий требуется определить порядок обработки партий каждой из групп на сегментах конвейера, т.е. расписание обработки партий группы. Расписание обработки партий группы  обозначено как , оно представляет собой совокупность последовательностей  запуска партий на обработку на *l-*ых сегментах конвейера (). Расписание  имеет вид: . Расписание обработки партий  для группы  формируется в предположении, что порядок обработки партий данных является одинаковым на всех *L* сегментах конвейера. Для формализации вида последовательностей  расписания  в рассмотрении введена матрица порядка обработки партий в системе . Элемент , если партия данных *i*-го типа занимает в последовательности  *j*-ю позицию,  в случае, если партия данных *i-*го типа не занимает в последовательности  *j*-ю позицию (размерность матрицы , где  – количество типов данных в партиях в группе ,  – количество партий в последовательностях в группе ). Порядок обработки партий группы на всех сегментах предполагается одинаковым, поэтому достаточно определения одной матрицы порядка . В рассмотрение введена матрица – матрица количества данных *i*-го типа в партиях, занимающих в последовательности  *j*-е позиции (элемент  равен количеству данных *i*-го типа в партии, занимающей *j*-ю позицию в , размерность матрицы ). Решение, формируемое на третьем уровне, имеет вид: {}.

Выполнение обработки партий данных *i-*ых типов () реализуется в течение заданных интервалов времени функционирования системы (), поэтому на основе решения по количеству и составам партий данных формируются группы партий, каждая группа партий обрабатывается в течение одного из интервалов  (). Состав группы партий для каждого интервала  функционирования системы () определяется таким образом, чтобы обеспечить максимальную загрузку сегментов. В соответствии с полученным решением по составу групп партий требуется определить порядок обработки партий каждой из групп на сегментах конвейера, т.е. расписания обработки партий соответствующих групп. Т.к. введены ограничения на длительность обработки групп партий (на длительность реализации расписаний), тогда не все сформированные партии могут быть распределены по группам, не вошедшие в группы партии являются не обработанными. Количество обработанных в течение интервалов () данных (количество обработанных партий в каждой из групп) зависит от количества и составов сформированных партий. Расписания обработки партий каждой из групп является зависящим от составов партий данных в этой группе. Таким образом, определение количества и составов партий данных, распределение партий по группам, построение расписаний обработки партий групп с учетом ограничений на длительность интервалов должно обеспечивать обработку максимального количества данных. Входными данными для системы построения расписаний обработки партий в группах являются: типы данных, обрабатываемых в системе; количество  () данных каждого типа, которые должны быть обработаны в системе; значения длительностей интервалов времени  функционирования системы при обработке данных; количество *Z* интервалов времени, в течение которых реализуется обработка. Формируемыми выходными решениями являются: составы партий данных *i*-ых типов (); составы групп партий, обрабатываемых в течение интервалов времени (); расписания обработки партий данных каждой группы (виды последовательностей обработки партий каждой группы на сегментах конвейера). Цель функционирования системы, связанная с обработкой партий данных в течение интервалов (), представлена как совокупность иерархически упорядоченных целей подсистем (функций, решаемых на иерархически упорядоченных уровнях системы построения расписаний обработки партий данных). Формирование решений на уровнях системы осуществляется следующим образом: первый уровень – решения по составам партий данных, второй уровень – решения по составам групп партий; третий уровень – решения по порядку обработки партий, входящих- в группы, на сегментах конвейера партий.

Для формирования на нижнем уровне модели вычислительного процесса обработки партий данных, включенных в группы (), введены следующие обозначения: – длительность обработки данных *i*-го типа на *l*-ом сегменте конвейера (); – вектор длительностей обработки данных *i-*ых типов, партии которых включены в группу (); – длительность переналадки *l*-го сегмента с обработки данных *i*-го типа на обработку данных *k*-го типа; – длительность первоначальной наладки *l*-го сегмента на обработку данных *i*-го типа; – матрица длительностей переналадок *l-*го сегмента для типов данных, партии которых включены в группу ;  – момент времени начала обработки партии данных *i*-го типа, занимающей в последовательности  на *l*-ом сегменте *j*-ю позицию; – матрица моментов времени начала обработки партий данных *i*-ых типов, занимающих в  *j*-е позиции (для группы партий );  – момент времени начала обработки данных, имеющих *q*-ый порядковый номер в партии, занимающей *j*-ю позицию в последовательностях  (); – матрица моментов времени начала обработки *q-*ых данных в партии, занимающей в  *j*-ю позицию (, – количество данных в партии, занимающей *j-*ю позицию в ). Элементы матрицы  определяются следующим образом: , где **, – момент времени начала обработки первых данных в партии, занимающей *j*-ю позицию в , – количество типов данных, партии которых входят в группу , – количество партий в группе . Определение значений , (**; **; **;  ) реализуется следующим образом (индекс *z* группы партий  для простоты опущен). Для (*l=1*)-го сегмента (*j=1*)-ой партии данных *i-*го типа в  при *q=1* выражение для определения : . При *l=1* и *j=1* для  (,): . При *j>2* и  выражения имеют вид: , , где  – время переналадки первого сегмента конвейера с обработки данных *i-*го типа (первая позиция партии в ) на обработку данных другого типа (вторая позиция партии в ), определяемое следующим образом:

, где .

Выражения для  и  (*l>2*) сформированы в общем виде следующим образом [1]:

;

;

.

Использование выражений для ,  и ,  () позволяет определить характеристики вычислительного процесса выполнения конвейеризированных программ обработки данных на сегментах конвейера. Метод построения расписаний обработки партий [2] предполагает добавление текущей рассматриваемой партии данных *i-*го типа в конец последовательностей  () и определение эффективного ее местоположения в этих последовательностях. Тогда местоположение рассматриваемой партии в  может быть охарактеризовано текущими (для данного количества партий в  ()) простоями сегментов конвейера при обработке партий, нахо­дящихся в последовательностях , которые определяются как: суммарный простой всех *L* сегментов конвейера перед началом обработки данных в первой позиции (*q=1*) в первой партии (*j=1*) в  (), суммарный простой сегментов конвейера при переходе от обработки партии данных одного типа (в *(j-1)*-ой позиции в ) к обработке партии данных другого типа (в *j*-ой позиции в ), суммарный простой сегментов, вызванный ожиданием готовности данных при обработке их внутри партий. С учетом выполненных рассуждений суммарный простой всех *L* сегментов конвейера при обработке текущего количества партий, добавленных в последовательности (), имеет вид [1]:



(1)

В выражении (1) первое слагаемое представляет собой суммарный простой сегментов конвейера перед началом обработки на них партий, второе слагаемое является суммарным простоем сегментов в ожидании готовности следующей партии данных к обработке, третье слагаемое– это суммарные простои сегментов конвейера в ожидании данных при их обработке внутри партий.

Постановка задачи предполагает наличие ограничений на время функционирования системы, тогда ресурсом, управление которым реализуется при обработке партий, является время функционирования конвейерной системы. Тогда решение по составам групп партий  () на втором уровне характеризуется общей эффективностью использования ресурса времени сегментов конвейера при реализации обработки партий каждой группы  в течение интервала . Эффективность использования ресурса времени функционирования системы при обработки данных характеризуется суммарным временем простоя сегментов при выполнении операций с партиями в группах  (). Этот подход соответствует внутренней цели функционирования системы, определяющей необходимость эффективного использования ограниченного ресурса времени конвейера (т.е. управление вычислительным процессом на втором уровне иерархии осуществляется с учетом внутренней цели функционирования системы). Суммарное время простоя сегментов при обработке партий группы определяется [1]: 1) суммой длительностей интервалов наладки сегментов и возможного их простоя в ожидании начала обработки первой партии в  (**); 2) суммой длительностей переналадки сегментов с обработки данных одного типа на обработку данных другого типа, возможного простоя сегментов в ожидании обработки следующей в  партии; 3) суммой длительностей интервалов времени простоя сегментов в ожидании готовности данных при обработке партий внутри группы; 4) суммой длительностей интервалов простоя *L* сегментов после окончания обработки  партий группы  (сумма интервалов простоя сегментов после окончания обработки партий группы на стадии освобождения конвейера). В соответствии с [1] критерий определения эффективных решений по составам групп партий на втором уровне иерархии имеет вид:



(2)

Здесь –количество партий, входящих в группу (индекс последней партии в группе), – количество данных, входящих в последнюю в  партию (индекс данных, являющихся последними в партии с индексом ), – момент времени начала обработки последних данных в партии с индексом  , а – время окончания обработки этой партии на *l-*ом сегменте.

Решение, формируемое для всех групп партий данных на втором уровне системы и передаваемое на первый уровень для вычисления значения критерия, имеет вид: {()}, где ,– вектор количества данных *i-*го типа в  партиях в группе , – количество типов данных, партии которых входят в . Решение на первом уровне представляется в виде набора , где элементы  () соответствуют количеству партий данных *i-*ых типов, элементы  соответствуют количеству данных *i-*го типа в *h-*ой партии (). При  не все партии данных *i*-го типа были размещены в группах партий  (). Партии данных *i-*го типа, не вошедшие в группы, будут размещены в множестве *Q*, которое представляет собой набор параметров вида: , где – количество партий данных *i*-го типа, не включенных ни в одну из групп партий,  –вектор составов этих партий, – количество типов данных, партии которых не включены в состав групп. Количество данных *i-*го типа, которые распределены по партиям определяется выражением , общее количество данных *i-*го типа, обрабатываемых в группах партий  (), определяется выражением . В том случае, если , тогда ()– это количество данных *i-*го типа, распределенных по партиям, но не вошедших в группы партий  (). На основе полученного выражения общее количество данных разных типов (), включаемых в составы соответствующих партий на первом (верхнем) уровне, но не вошедших в группы партий  () для обработки, будет определено следующим образом: . Полученное выражение использовано в качестве критерия эффективности составов партий на первом (верхнем) уровне принятия решений, оно характеризует эффективность решения  на основе анализа как самого этого решения, так и решения {| } со второго уровня иерархии. Критерий на первом уровне соответствует внешней цели функционирования системы, определяющей необходимость обработки максимального количества данных разных типов в ограниченные интервалы времени.

При задании значений интервалов () ограничение на время обработки партий данных, входящих в группы , имеет вид [1]:  (). В силу выполненных рассуждений модель многоуровневого программирования определения составов партий, групп партий и расписаний обработки партий в группах имеет вид [1]:

- первый уровень (определение составов партий данных *i-*ых типов ()):

, где ;

- второй уровень (составы групп партий ):

, где

(3)



- третий уровень (порядки обработки партий в группах): (), где



- ограничения на третьем уровне для длительности реализации расписания обработки партий группы  ():

.

(4)

# МЕТОД ОПРЕДЕЛЕНИЯ ЭФФЕКТИВНЫХ СОСТАВОВ ГРУПП ПАРТИЙ ДАННЫХ

Для формирования решений по составам групп партий на втором уровне иерархии к введенным обозначениям наборов вида: (), введено множество *Q* – множество партий, не размещенных ни в одной из групп , заданное в виде:  (в *Q* определяются: количество партий  данных *i*-го типа, вектор состава этих партий (размерность вектора )). Тогда решение, формируемое на втором уровне иерархии системы на *s*-ом шаге алгоритма определения составов групп партий, имеет вид:  (где ) и , где  – количество наборов заданного вида в множестве *Q*. Реализация метода построения эффективных с точки зрения введенного критерия составов групп партий на втором уровне предполагает выполнение двух стадий: 1) формирование начального решения (начального состава групп партий); 2) переход к локально оптимальному решению (формирование эффективных составов групп партий). Формирование алгоритма построения начального решения  предваряется введением в рассмотрение обозначений: 1) – индекс текущей рассматриваемой (формируемой) группы партий; 2) *i’*– тип данных, партия которых размещается в рассматриваемой руппе  (номер элемента в векторе *M*, номер строки в матрице *A*); 2)  – *i’-*ый элемент вектора *M*, соответствующий количеству партий данных *i’-*го типа, размещаемых в группах  () и множестве *Q*; 3)  – количество партий данных *i’*-го типа, размещенных в группах  и в множестве *Q*; 4)  – номер партии данных *i’*-го типа, размещаемой в группе , количество данных в которой соответствует значению элемента  матрицы *А*; 5) – множество типов данных, партии которых размещаются в группах  (); 6)– множество номеров групп , в которых партии данных могут быть размещены; 7) – текущее динамически изменяемое множество номеров групп партий, в которых партии данных могут быть размещены (с которым оперирует алгоритм). В этом случае партий данных размещаются только в тех группах , для идентификаторов *z* которых выполняется условие .

Формирование начального решения (начальных составов групп партий) предполагает распределение партий, полученных в решении *[M,A]* с первого уровня иерархии, по множествам наборов параметров вида  и множеству *Q* с учетом ограничений на время обработки партий в группах. Наряду с ограничением на время реализации обработки партий группы  (модель (3)), необходимо определить условие полного заполнения интервала обработкой партий, входящих в группу . Это условие вытекает непосредственно из указанного ограничения и имеет вид:

 .

(5)

Если при размещении в группе  некоторой текущей -ой партии данных *i’*-го типа введенное условие выполняется, тогда в эту группу партии добавлены быть не могут и ее идентификатор должен быть ударен из множества .

Начальная инициализация исходных данных для реализации алгоритма формирования начального состава групп партий выполнена следующим образом: 1) (), т.е. первоначально множества  совокупностей параметров вида не содержат ни одного элемента; 3) инициализация  значением 0 (); 4) инициализация множеств  () и множества  в виде: ; (инициализация значением  предполагает, что наборы вида будут добавляться в состав соответствующих множеств); 5) ; 6) ; 7)  (), т.е. для каждого *i-*го типа данных задан идентификатор партии, размещаемой в группах  ().

Алгоритм формирования начального состава групп партий  содержит следующие шаги:

1) инициализация множества  номеров групп партий, в которых на данной итерации алгоритма будут размещаться партии: ;

2) определение типа данных *i’*, партии которого будет размещаться в группах (): , ;

3) инициализация параметра  (количества размещенных в группах () и в множестве *Q* партий данных *i’*-го типа) значением 0 ();

4) определение номера текущей рассматриваемой группы партий , в которой будут размещаться партия данных -го типа: , (номер заполняемой группы партий исключается из множества );

5) задание номера  текущей рассматриваемой партии данных *i‘*-го типа, добавляемой в группу партий , равным 1 (, рассматривается первая партия из , добавляемая в группу партий ).

6) для -го типа и группы  выполняется проверка условия , в случае выполнения условия реализуется переход на шаг 7; в том случае, если условие не выполняется, реализуется формирование набора параметров  для рассматриваемого *-*го типа данных, набор параметров  включается в состав группы партий  : ; ;

7) определение значений параметров в наборе (для *i*-го типа данных):  ;

8) сформированная группа партий передается на третий уровень иерархии системы для формирования расписаний обработки партий , ей соответствующего;

9) проверка для расписания  выполнения ограничения на длительность его реализации (выражение (4)); если ограничение выполняется, тогда , ;

10) проверка для сформированного состава группы  (в которую добавлена рассматриваемая партия с количеством данных в ней ) выполнения условия (5);

11) если ограничение на длительность реализации расписания обработки партий (4) выполнено, условие полного заполнения интервала  обработкой партий (5) не выполняется, тогда при  реализуется переход к шагу 6, при  переход к шаг 22;

12) если условие (5) полного заполнения интервала  обработкой партий выполнено, тогда  (т.е. партии в группе  в последующем размещены быть не могут);

13) при выполнении условия (5) (автоматически выполняется условие (4)) реализуется проверка условий  и ; при их выполнении осуществляется определение номера *z’* следующей рассматриваемой группы партий  на основе выражения , , переход в шагу 6; если при реализации условия (5) выполняется условие  (все партии *i’*-го типа размещены в группах партий () и множестве *Q*), тогда выполняется переход на шаг 22;

14) если для расписания , соответствующего группе партий , ограничение вида (4) на время реализации не выполняется, тогда ;  (рассматриваемая партия данных -го типа с количеством элементов в ней  исключается из группы , так как она не может быть в ней размещена в силу не выполнения ограничения (4));

15) если в результате , тогда ,  (из группы  исключается набор , партии данных -го типа в группу  не включены);

16) если , тогда выполняется определение номера следующей рассматриваемой группы , в которую размещается рассматриваемая -я партия данных *i’-*го типа: , ; реализуется переход к шагу 6;

17) если , тогда рассматриваемая партия данных в количестве  элементов ни в одной из групп  () размещена быть не может, тогда выполняется проверка условия  для соответствующего *i’*-го типа данных;

18) при выполнении условия  для соответствующего *i’*-го типа данных (набор параметров вида  сформирован на предыдущих шагах алгоритма), тогда выполняется модификация значений параметров: , ;

19) если , тогда формируется набор параметров вида , который включается в множество *Q*:  ; выполняется инициализация значений параметров, входящих в набор: , ;

20) модификация значений параметров  и : , ;

21) если , тогда осуществляется присваивание , реализуется определение номера формируемой группы следующим образом:, ; выполняется переход на шаг 6; если , тогда реализуется переход на шаг 22;

22) проверка условия , в случае его выполнения множество  содержит типы данных, партии которых должны быть размещены в группах () и множестве *Q*; тогда реализуется определение *i’*-го типа данных, партии которого будут рассматриваться на следующей итерации алгоритма: , , выполняется присваивание , реализуется переход на шаг 3; при условии  реализуется переход на шаг 23;

23) останов алгоритма.

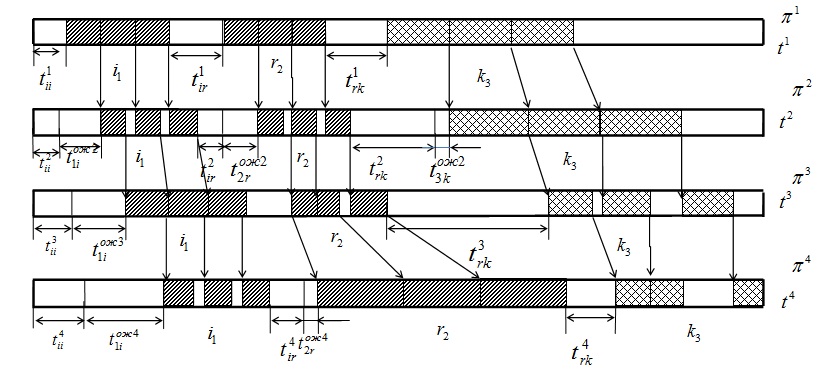
Сформулированный алгоритм позволяет получить начальное решение по составам групп партий (), на основании которого требуется сформировать локально оптимальное решение  по составам групп партий. С точки зрения введенного в рассмотрение на втором уровне иерархии критерия эффективные составы групп  предполагают минимальное суммарное время простоя сегментов конвейера при обработке партий в группах. Для оптимизации составов групп партий данных использован следующий подход. Переход к новым составам групп партий связан с извлечением из каждой группы () такой партии, обработка которой является причиной неэффективного использования ресурса времени системы. Формирование условия определения партии, извлекаемой из группы  (), выполняется на основе анализа видов последовательностей , входящих в расписание . Определение условий идентификации партий данных, исключаемых из группы  (), выполняется на основе заданного вида последовательностей обработки партий данных различных типов (Рис.1). Анализ видов последовательностей  (Рис.1) позволил определить следующие причины неэффективного использования ресурса времени системы при обработке партий: 1) несогласованность длительностей обработки данных партии в *j*-ой позиции в последовательностях , которая обуславливает простои сегментов в ожидании готовности к обработке данных этой партии; в частности, 

Рисунок 1 – Заданный вид последовательностей обработки партий в группе  для идентификации позиции удаляемой партии

несогласованность длительностей обработки данных *k*-го типа (партия в позиции *j=3* в последовательностях ) обуславливает простой третьего и четвертого сегментов в ходе обработки этой партии; в то же время простои сегментов в ожидании готовности к обработке данных в партии (в *j*-ой позиции в ) являются причиной «сдвига» последовательности партий в конец интервала *,* что обуславливает неэффективное использование ресурса времени системы на стадии освобождения конвейера (на заключительной стадии обработки партий группы ); тогда исключение из группы  партии, находящейся в соответствующей *j-*ой позиции в , и замена её партией данных другого типа позволяет уменьшить несогласованность в длительностях обработки данных этого типа на различных сегментах и уменьшить время простоя сегментов при обработке партий группы;

2) длительная переналадка сегментов конвейера на обработку данных другого типа (для партии, расположенной в *j*-ой позиции в ) обуславливает их простои в ожидании начала обработки партии, является причиной «сдвига» последовательности партий в конец интервала , при этом несогласованность моментов времени окончания переналадки *l*-го сегмента на обработку партии *i*-го типа (в *j*-ой позиции) и окончания обработки данных с *q=1* в этой же партии на (*l*-1)-ом сегменте (при незначительной длительности интервала переналадки) обуславливает простой *l*-го сегмента в ожидании начала обработки партии; наличие интервала ожидания сегментом начала обработки партии вместе с соответствующим интервалом переналадки обуславливает неэффективное использование ресурса времени системы (суммарный интервал простоя сегмента перед началом обработки партии, который в то же время обуславливает «сдвиг» партий в конец интервала ).

Идентификация степени неэффективного использования ресурса времени системы при обработке партий данных в группе  выполняется на основе анализа видов последовательностей  расписания , соответствующего этой группе. Определение значений интервалов, характеризующих неэффективное использования ресурса времени системы при обработке партий, реализуется при зафиксированном (в соответствии с ) порядке партий. Поэтому идентифицируется партия данных, занимающая определённую *j*-ю позицию в последовательностях . Для идентификации исключаемой из партии необходимо определить суммарное время неэффективного использования ресурса системы при обработке каждой партии в группе . Если **– номер позиции партии данных в последовательностях  при условии, что , тогда интервал времени простоя *l*-го сегмента в ожидании начала обработки партии данных, занимающей в *-*ю позицию будет определён выражением вида:

.

В этот интервал входят: время переналадки сегмента на обработку данных в партии в *-*ой позиции в , возможное время простоя *l*-го сегмента в ожидании начала обработки этой партии. Так как **-я позиция для рассматриваемой партии является одинаковой во всех последовательностях  , тогда суммарное время простоя сегментов в ожидании начала обработки партии, занимающей  **-ю позицию в последовательностях , определяется выражением вида:

, .

Обозначим номер позиции партии в , исключаемой из группы , через . Тогда – это позиция партии , определяемая в соответствии с следующим условием:

, где. (5)

Время простоя *l*-го сегмента в ожидании начала обработки первой партии группы определяется значением . В этот интервал входят: интервал времени первоначальной наладки сегментов на обработку первой партии в , интервал возможного простоя сегментов в ожидании начала обработки. Суммарное время простоя всех *L* сегментов в ожидании начала обработки первой партии в  вычисляется в соответствии с выражением . На основе этого выражения и выражения (5) индекс  – номер её *j*-й позиции в  (при ), который определяется следующим образом:

 (6.1)

Таким образом, – это номер позиции партии в  (исключаемой из ), выполнение операций с которой является причиной максимального суммарного простоя сегментов конвейера при их наладке либо переналадке на обработку данных *i*-го типа в этой партии.

Время простоя -го сегмента в ходе обработки партии, занимающей *j*-ю позицию в последовательности (простой сегмента в ожидании готовности к обработке данных, входящих в партию), определяется выражением: , где –количество данных в партии, занимающей *j-*ю позицию в последовательностях  (),. Тогда суммарные простои сегментов конвейера в ожидании готовности к обработке данных в партии, занимающей *j*-ю позицию в последовательностях  () расписания , соответствующего текущему составу группа , определяются выражением. Обозначим через  индекс исключаемый из  партии, обработка которой в системе обуславливает максимальные (среди всех *j*-х партий, ) простои сегментов в ожидании готовности данных к обработке; значение индекса  соответствует номеру позиции партии в последовательности , которая должна быть исключена из группы для обеспечения более «плотного» расписания обработки. Значение индекса  является таким, что для него выполняется условие следующего вида:

, при . (6.2)

Таким образом, выражения (6.1) и (6.2) позволяют идентифицировать в группе  партии, которые могут быть из неё удалены (для замены партиями из множества *Q*) для обеспечения более эффективного использования ресурса времени системы. Для формирования метода построения эффективного состава групп партий  в рассмотрение введены следующие обозначения: 1) – индекс текущей группы партий, эффективный состав которой определяется на текущем шаге алгоритма; 2)()– индекс (номер) партии (номер позиции партии в  расписания ), вызывающей максимальный суммарный простой сегментов конвейера при её обработке в соответствии с условиями (6.1) либо (6.2); 3)  () – идентификатор типа данных, партия которых исключается из группы  (партия данных -го типа, занимающая в  ()-ю позицию (), определяемую в соответствии с условиями (6.*p*) ()); 4)  () – количество данных в партии, которая извлекается из группы  (партия данных -го типа, занимающая в  ()-ю позицию (), определяемую в соответствии с условиями (6.*p*)); 5) – переменная-буфер, предназначенная для хранения идентификатора группы партий , исключение из которой партии данных и добавление в которую партии из множества *Q* обеспечивает наилучшее решение в рамках окрестности текущего эффективного решения ; 6) – переменная-буфер, предназначенная для хранения *-*го типа данных, исключение партии которых из группы  обеспечивает наилучшее решение в рамках окрестности текущего эффективного решения  (для типов данных ); 7) – переменная-буфер, предназначенная для хранения количества данных*-*го типа в партии, исключение которой из группы  обеспечивает наилучшее решение в рамках окрестности текущего эффективного решения ; 8)  – индекс (номер) типа данных, партия которых исключается из множества *Q* и добавляется в группу  (взамен исключаемой из этой группы партии -го типа () в количестве  элементов); 9)  – количество данных в партии -го типа, которая исключается из множества *Q* и добавляется в группу  (взамен исключаемой из этой группы партии -го типа () в количестве  элементов); 10) – переменная-буфер, предназначенная для хранения типа данных , партия которых в группе  замещает партии данных -го типа, что обеспечивает наилучшее решение в рамках окрестности текущего эффективного решения ; 11) – переменная-буфер, предназначенная для хранения количества данных -го типа в партии, которая замещает партию данных -го типа, что обеспечивает наилучшее решение в рамках окрестности текущего эффективного решения ; 12) – набор вида , с параметрами которого выполняются преобразования, в группе (), соответствующей текущему решению ; 13) – набор вида , с параметрами которого выполняются преобразования для данных -го типа, в множестве *Q*; 14) – промежуточное решение по составам групп (), получаемое на основе решения ; 15) – промежуточное решение по составу рассматриваемой группы партий , входящее в решение  получаемое путем исключения из группы  в исходном решении  партии данных -го типа (); 16) – промежуточное решение по составу рассматриваемой группы партий , входящее в решение  получаемое путем добавления в группу  в решении  партии данных -го типа в количестве  элементов; 17) – значение критерия , соответствующее наилучшему текущему сформированному решению  в рамках окрестности локального оптимального решения ; 18) – значение критерия , соответствующее текущему локально оптимальному решению .

Перед началом реализации алгоритма определения локального оптимального решения  выполняется инициализация его параметров следующим образом: ; ; ; ;. Алгоритм определения эффективного решения в рамках окрестности решения  реализуется в случае, если , и предполагает выполнение рассматриваемой ниже последовательности шагов:

1) вычисление для текущего локально оптимального решения  значения целевой функции  (для формирования расписаний (), соотвествующих решению , выполняется передача этого решения на третий уровень иерархии системы, реализуется формирование расписаний (), передача расписаний на второй уровень иерархии, вычисление значения  критерия , инициализация , );

2) задание номера *z’* текущей рассматриваемой группы партий следующим образом:  (группа партий , с которой при реализации алгоритма будет выполняться обмен партиями из множества *Q*);

3) задание значения индекса *p* условия, в соответствии с которым в последовательностях  () выделяются партии с номерами позиций , равным 1 (*p=1)*;

4) в соответствии с условием (6*.p*) реализуется определение позиции  партии в последовательностяхрасписания , обработка которой вызывает максимальный суммарный простой сегментов (для идентификации  используется решение в виде матриц и третьего уровня); в соответствии с номером  позиции партии в последовательностях  , определенной с использованием условия (6*.p*), по матрице  идентифицируется номер строки *i*, для которой  при  (номер строки *i* соответствует номеру элемента вектора  типов данных, партии которых включены в состав группы ); по значению *i* определение типа данных , партия которых исключается из группы : ; в соответствии с определенным номером строки *i* осуществляется идентификация по матрице  количества данных в рассматриваемой партии и инициализация параметра : ;

5) в соответствии с типом данных  в группе , как множестве наборов  выполняется определение набора  (наборсоответствует -му типу данных), с которым будут выполняться преобразования; инициализация промежуточного решения , на основании которого будет формироваться новое решение по составу группы :; в решении (по составу группы , входящей в решение ) реализуется исключение партии данных -го типа :

a) для -го типа данных и их количества  (в партии исключаемой из группы ) определение для набора  в векторе  индекса  партии такого, что ;

б) для набора  выполняется преобразование вектора  следующим образом:



в) переупорядочивание элементов вектора ;  при ;

г) преобразование количества партий данных -го типа: ; если , тогда сформировано промежуточное решение  по составу группы , которое будет дополняться партией данных -го типа к количестве  элементов из множества *Q*; если , тогда  ; ;

6) для множества *Q* наборовпараметров вида  выполняется инициализация номера *k* рассматриваемого набора*: k=1*;

7) в соответствии со значением *k* в множестве *Q* выполняется определение набора , на основании которого реализуется идентификация типа данных , партии которых будут размещаться в группе  (– тип данных такой, что  при заданном значении индекса *k* , в соответствии с значением индекса *k* набора параметров в множестве *Q* определяется тип данных, партия которых добавляется в группу ); для рассматриваемого типа данных  выполняется инициализация номера партии , которая будет размещаться в группе : ;

8) для текущего значения параметра *k* в множестве *Q* осуществляется определение набора параметров , с которым выполняются преобразования;

9) выполняется модификация индекса  партии данных -го типа, включаемой в группу : ; осуществляется проверка выполнения условия ; при его реализации для набора параметров  осуществляется определение в векторе  в соответствии с индексом  количества данных  в партии: ; если условие  не выполняется, тогда осуществоляется переход на шаг 14;

10) проверка выполнения условия , в случае, если условие не выполняется, тогда реализуется переход на шаг 11, в случае выполнения условия реализуется проверка условия , если это условие выполняется, тогда осуществляется переход на шаг 9, если условие не выполняется, то переход на шаг 11;

11) промежуточное решение  по составу группы партий  модифицируется путем добавления партии данных -го типа из множества *Q* в группу , в результате на основе решения  формируется решение (предварительно выполняется инициализация : ):

а) для рассматриваемого -го типа данных, партия которых добавляется в группу партий (модификация решения , формирование решения ), выполняются следующие действия:

-если ( при , ), тогда ; ;

-если ( при , ), тогда ; ; ; ; ;

12) сформированное промежуточное решение  передается на третий уровень иерархии для построения расписания ; выполняется проверка реализации ограничения на длительность обработки партий, входящих в группу (проверка ограничения для сформированного решения  по составу группы ); в случае, если ограничение на длительность реализации обработки партий данных, входящих в группу, не выполняется, осуществляется переход на шаг 9; при выполнении условия (4) , реализуется переход на шаг 13;

13) выполняется вычисление значения целевой функции  для сформированного решения , в состав которого входит сформированное решение  по составу группы ; для полученного значения целевой функции  проверяется условие; при его выполнении на основе исходного решения  может быть сформировано лучшее решение по составам групп партий; в этом случае изменяемые параметры решения , посредством оперирования которыми сформировано решение , буферизируются: ; ; ; ; ; ; осуществляется на шаг 9; в случае, если условие выполняется условие , тогда реализуется переход на шаг 9;

14) модификация индекса *k* набора следующим образом: *k=k+1*, (реализуется переход к рассмотрению партий данных другого типа для добавления их в рассматриваемую группу ); если , тогда выполняется переход на шаг 7; если , тогда для рассматриваемой группы партий  были сформированы все возможные решения, связанные с исключением из нее партии -го типа в количестве  элементов и добавлении в нее партий -ых типов (для которых ); выполняется переход на шаг 15;

15) осуществляется модификация индекса *p* условия определения позиции  партии, которая может быть исключена из рассматриваемой группы  для формирования новых решений в окрестности текущего решения :  ; реализуется проверка условия , в случае его выполнения осуществляется переход на шаг 4; если условие  не выполняется, тогда реализуется переход на шаг 16;

16) реализуется модификация индекса *z’*  группы , состав которой будет изменяться: ; реализуется проверка условия ; в случае его выполнения осуществляется переход на шаг 3; если условие  не выполняется, тогда осуществляется переход на шаг 17;

17) выполняется сравнение сохраненного значения  критерия  для текущего локально оптимального решения  со значением  этого критерия, соответствующего лучшему решению  в окрестности, в том случае, если , тогда решение  лучше, чем  и текущее локально оптимальное решение должно быть переопределено, при этом выполняется переход на шаг 18; при выполнении условия  в окрестности решения  лучшего решения не найдено, реализуется переход на шаг 19;

18) на основе текущего локально оптимального решения  с использованием буферизированных значений параметров , , , ,  осуществляется формирование нового локально оптимального решения:

а) в соответствии с значением  в решении  определяется группа партий , из которой требуется исключить партию данных -го типа в количестве  элементов; в соответствии с типов данных  в группе (множестве наборов параметров)  определяется набор , с которым будут выполняться преобразования;

б) в решении по составу группы  реализуется исключение партии данных -го типа следующим образом: для -го типа данных и количества  определение для набора  в векторе  индекса  партии, для которой ; для набора  выполняется преобразование вектора  следующим образом:



в) переупорядочивание элементов вектора ;  при ;

г) модификация ; если , тогда сформировано промежуточное решение по составу группы , которое будет дополняться партиями данных -ого типа в количестве  элементов из множества *Q*; если , тогда  ; ;

д) дополнение состава множества *Q* партией данных -го типа в количестве  элементов (партией исключаемой из группы ); действия по размещению партии данных -го типа в множестве *Q* реализуются следующим образом:

- если ( при , ), то ; ;

-если ( при , ), тогда ; ; ; ; ;

е) исключение партии данных -го типа в количестве  элементов из множества *Q*, которое реализуется следующим образом: для -го типа данных осуществляется определение набора параметров , в векторе  которого идентифицируется индекс  партии такой, что ; для  выполняется преобразование вектора  следующим образом:



ж) переупорядочивание элементов вектора ;  при ; ; если , то удаление партии данных -го типа завершено; если , тогда  ; ;

з) добавление партии данных -го типа в количестве  элементов в группу партий  осуществляется следующим образом:

- если ( при , ), тогда ; ;

-если ( при , ), тогда ; ; ; ; ;

в итоге сформировано новое локально эффективное решение по составу групп партий данных ; осуществляется переход на шаг 1;

19) останов алгоритма.

1. [↑](#footnote-ref-1)