ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc452071594)

[АНАЛИЗ ЛИТЕРАТУРЫ 5](#_Toc452071595)

[МОДЕЛЬ СИСТЕМЫ ПОСТРОЕНИЯ РАСПИСАНИЯ ПАРТИЙ ДАННЫХ 12](#_Toc452071596)

[МЕТОД ПОСТРОЕНИЯ РЕШЕНИЯ ПО СОСТАВАМ ПАРТИЙ ДАННЫХ N ТИПОВ 25](#_Toc452071597)

[МЕТОД ПОСТРОЕНИЯ РЕШЕНИЯ ПО СОСТАВАМ ПАРТИЙ ДАННЫХ I – ОГО ТИПА 28](#_Toc452071598)

[МЕТОД ОПРЕДЕЛЕНИЯ ПОРЯДКА ОБРАБОТКИ ПАРТИЙ ДАННЫХ (РАСПИСАНИЙ ОБРАБОТКИ ПАРТИЙ ДАННЫХ) В ГРУППАХ 40](#_Toc452071599)

# ВВЕДЕНИЕ

На современном этапе развития информационных технологий часто появляется необходимость обработки больших массивов однотипных данных за ограниченное время. Одним из способов решения данной проблемы является применение систем конвейерного типа. В этих системах важной составляющей является поступление требований на обработку.

В данной выпускной квалификационной работе рассматривается многоуровневая система теории расписания. Теория расписаний — раздел дискретной математики, занимающийся проблемами упорядочения. В большинстве случаев в теории расписания ставится задача дискретной оптимизации: построить расписание, минимизирующее время выполнения работ, стоимость работ и т .п. Расписание — указание, на каких машинах и в какое время должны обслуживаться требования (выполняться работы).

В результате выполнения данной дипломной работы было разработано Windows-приложение, которое строит расписания обработки данных многих типов, с учетом времени перенастройки оборудования с обработки требования одного типа на другой. Учитывая то, что задача составления расписаний является NP-полной, то для её решения использовался метод градиентного поиска.

Разработанная система производит оптимизацию расписания по нескольким критериям эффективности, программный комплекс условно разделён на три уровня. Каждый из них представляет собой поиск оптимума по одному или нескольким критериям.

Производительность обработки данных при выполнении программ можно повысить путём конвейеризации [1]. Под конвейеризацией выполнения программы подразумевается разделение её на фрагменты, каждый из которых закреплён за соответствующим сегментом конвейера. Выполнение программ реализуется в многостадийной последовательной обрабатывающей системе с заданным порядком сегментов конвейера. Сегменты конвейера следуют строго друг за другом, местоположение сегмента в последовательности определяется его номером. Обозначим количество сегментов в конвейере через L, тогда порядковый номер сегмента, входящего в состав конвейера - . Если в состав конвейера входит L сегментов, то все выполняемые в системе программы должны быть разделены на L фрагментов, каждый из которых закреплён для выполнения за соответствующим сегментов конвейера. Тогда выполнение конвейеризированной программы в составе конвейера предполагает строгое закрепление её l-того фрагмента для выполнения на l-том сегменте конвейера.

В систему обработки данных могут поступать различные типы данных, обозначим их количество через n. Через i обозначим номер множества однотипных данных, которые должны быть обработаны в системе, тогда . Количество элементов в множестве однотипных данных, характеризуемых индексом i, обозначим через . Все данные i-того типа (в количестве ) обрабатываются соответствующей им программой. То есть существует  типов программ, выполняемых в системе, обрабатывающих данные i типа. Таким образом, в системе задаётся строгое соответствие между типом данных и обрабатывающей эти данные программой. Однократное выполнение программы i-го типа позволяет обработать один элемент множества данных i-того типа. Так как множество i-того типа содержит  элементов, то обрабатывающая программа i-того типа должна быть выполнена  число раз. Цель функционирования системы в этом случае состоит в обработке всех поступающих на её вход данных. 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 [2, 3]. В этом случае «партия» - это совокупность наборов данных одного (i-того, ) типа, которые обрабатываются в программе непосредственно друг за другом. Группа партий – это совокупность партий, обрабатываемых в течение одного из заданных интервалов времени . При этом состав групп партий для каждого временного интервала  определяется в соответствии с вводимыми критериями эффективности.

Выполнение программ в конвейеризированной вычислительной системе требует использования её (системы) ресурсов. В первую очередь такими ресурсами являются процессорное время каждого сегмента конвейера и оперативная память, в которой хранится выполняемая программа и обрабатываемые данные. Чтобы наиболее эффективно с точки зрения использования сегментов конвейера использовать вычислительную систему необходимо планировать запуск программ различных типов, то есть составлять расписания обработки данных разных типов. Так как поступление данных i-того типа инициирует в конвейеризованной системе выполнение программы i-того типа, то под «расписанием» понимается порядок поступления данных разных типов на вход системы или, другими словами, порядок запуска программ разных типов на выполнение.

Из-за того что обработка на всех устройствах происходит последовательно, и обработка каждой партии не может быть разбита на несколько частей, необходимо сформировать порядок поступления партий данных только на первое устройство.

Так как рассматриваемая задача по определению количества и составов партий данных i-ых типов () связана с уменьшением простоев при построении расписания обработки партий данных, то эффективность решений по составам партий, генерируемым на первом уровне иерархии подсистем, определяется в соответствии количеством партий, обработанных в группах партий данных за промежутки . Разработанный метод формирования партий так же, как и метод определения расписания обработки партий, реализует градиентную стратегию по определению решений (реализует градиентное уточнение окрестностей эффективных решений по составам партий и реализации расписаний).

# АНАЛИТИЧЕСКИЙ ОБЗОР И ОБОСНОВАНИЕ АКТУАЛЬНОСТИ ПОСТАВЛЕННОЙ ЗАДАЧИ

На современном этапе развития направления теории расписаний и задач оптимизации рассматриваются как приближенные, так и точные методы.

Теория расписаний является частью исследования операций. Теория расписаний исследует задачи, в которых необходимо упорядочить или, другими словами, определить последовательность выполнения совокупности работ, использования каких-либо средств и т.д.

Задачи упорядочения носят самый общий характер. Они возникают там, где существует возможность выбора той или иной очередности выполнения работ: при распределении работ на производстве, составлении расписания приземления самолетов, составлении расписания движения поездов, обслуживании клиентов в обслуживающих системах и т.д. Цель курса - изучить то общее, что характеризует такие задачи независимо от их конкретного содержания.

Результаты, к которым приводит то или иное упорядочение, существенно отличаются. В ряде практических случаев эти различия принимают стоимостной характер или определяются какой-либо другой величиной.

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 выполнения 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 два аспекта работы системы: формирование партий и построение расписаний их обработки. Необходимо разработать метод получения оптимального количества и состава партий данных каждого типа и метод получения оптимального расписания обработки 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 ОБРАБОТКИ ПАРТИЙ 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 СИСТЕМЫ ПОСТРОЕНИЯ РАСПИСАНИЯ ПАРТИЙ ДАННЫХ

Для обоснования методов оптимизации составов партий, составов групп партий и расписаний обработки партий в рассмотрение введены обозначения и выполнены рассуждения, приведенные ниже ([1]). Через i обозначен идентификатор типа обрабатываемых в системе данных, n – количество типов обрабатываемых в системе данных (),  – количество данных i-го типа, которые должны быть обработаны в системе. В постановке задачи являются заданным, что  (), тогда обрабатывающая программа должна быть выполнена в конвейерной системе  число раз, а для обработки однотипных данных формируются партии. Партия – это совокупность однотипных данных, обработка которых выполняется без переналадки сегментов конвейера. Партия является фиксированной, если в нее входят все данные i-го типа, либо партия может содержать не все  данных i-го типа, в этом случае в системе обрабатывается ни одна, а несколько партий данных этого типа. Через *l* обозначен индекс сегмента конвейерной системы, осуществляющего выполнение *l-*й части программы в системе (). Дисциплина обслуживания выполняемых в системе программ предполагает прохождение данными, которые они обраба­тывают, всех сегментов конвейера, при этом если *l*-й прибор приступил к обработке данных *i*-го типа, обработка не может быть прервана. Все обрабатывающие приборы конвейерной системы 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 разных типов выполняется в течение интервалов времени задаваемой длительности, для которых введены обозначения: – интервал времени, в течение которого может быть реализована обработка партий данных. Т.к. обработка партий данных i-ых типов () реализуется в течение заданных интервалов времени функционирования системы  (), тогда на основе решения по количеству и составам партий данных различных типов формируются группы партий, каждая группа партий обрабатывается в течение одного из интервалов  (формируется Z групп партий). Группа партий– это совокупность партий, обрабатываемых в течение одного интервала времени  функционирования системы. Т.к. заданными являются ограничения на интервалы времени функционирования системы  (), тогда не все сформированные партий могут быть включены в состав групп партий. Состав групп партий для интервалов  () определяется таким образом, чтобы обеспечить возможность формирования максимального количества комплектов каждого типа.

Через d обозначен идентификатор типа комплекта, который формируется из результатов обработки данных в конвейерной системе, через – общее количество типов формируемых комплектов. Комплект данных (результатов обработки данных) представляет собой совокупность данных, при этом количество данных каждого типа, входящих в комплект одного (d-го) типа, является заданным. Данные всех n типов входят в определенном количестве в состав комплекта каждого типа. Через  обозначено количество данных i-го типа, которые должны входить в один комплект d-го типа. В рассмотрение введена матрица (W), значения элементов  которой равны числу данных i-го типа, которые должны входить в комплект d-го типа. Таким образом, рассматриваются комплекты разных типов и задаются соответствующие им составы. В выполненной постановке задачи введены ограничения на длительность интервалов времени обработки партий, входящих в группы, тогда не все сформированные партии могут быть распределены по группам, не вошедшие в группы партии являются не обработанными. Комплекты разных типов могут быть сформированы только из данных, входящих в партии, включенные в соответствующие группы для обработки.

В тоже время количество обработанных в течение интервалов  данных зависит от количества и составов сформированных партий (т.е. составы групп партий (как следствие, количество сформированных комплектов) является зависящим от количества и составов партий). Решение по порядку обработки партий в каждой из групп является зависящим от составов партий данных в этой группе. Таким образом, определение количества и составов партий данных различных типов, распределение партий по группам, построение расписаний обработки партий групп с учетом ограничений на длительность интервалов  должно обеспечить формирование максимального количества комплектов типов.

Выполнена декомпозиция обобщенной цели функционирования системы на совокупность иерархически упорядоченных целей подсистем (функций, решаемых на иерархически упорядоченных уровнях системы построения расписаний обработки партий данных). Формирование решений осуществляется на уровнях системы следующим образом [1]: первый – решения по составам партий данных, второй – решения по составам групп партий; третий – решения по порядку обработки на сегментах конвейера партий, входящих в группы.

Для формирования решений по составам партий данных i-ых типов () 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 формирования решений по составам групп партий введены обозначения:  () – группа партий, обрабатываемых в течение одного из интервалов (); – количество партий данных i-го типа в группе партий ;  – вектор количества данных i-го типа в  партиях в группе . Тогда партии данных i-го типа, входящие в группу партий , определены с использованием набора параметров вида: , а группа партий – совокупность таких 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 приборах системы (должна быть определена последовательность запуска партий каждой из групп на обработку на сегментах конвейера), т.е. расписание обработки партий соответствующей группы. Расписание обработки партий z-ой 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 . 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, размер матрицы , где  – количество типов данных в партиях в группе ,  – число партий в последовательностях для группы . Порядок обработки партий группы на всех сегментах одинаков, 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 , размер матрицы ). Решение, формируемое на нижнем (третьем) уровне иерархии подсистем в системе, имеет вид: {}.

В силу наличия ограничений на время функционирования системы не все сформированные партии данных могут быть размещены в группах партий (могут быть обработаны в течение заданных интервалов ()), а цель функционирования системы состоит в обработке данных таким образом, чтобы было сформировано максимальное количество комплектов, тогда эффективное решение {| ()} обеспечивает реализацию этого требования для текущих составов партий. Т.о. для формирования решения по составам комплектов данных на основе решения по группам партий {| ()} в рассмотрение введена матрица (), значения элементов d-го столбца которой соответствуют количеству данных i-ых типов, входящих в комплекты d-го типа (элемент  соответствует количеству данных i-го типа, которые будут входить в состав комплектов d-го типа). Для идентификации количества комплектов, сформированных из данных, обработанных в группах партий (), в рассмотрение введен вектор , значение d-го элемента которого соответствует количеству комплектов d-го типа (элемент  – это количество комплектов d-го типа, сформированных из данных, обработанных в группах партий ()). Для реализации алгоритма определения составов комплектов из обработанных данных в рассмотрение введен вектор , i-ый элемент которого  – это количество данных i-го типа (), обрабатываемых в группах партий в соответствии с текущим решением (), которые могут быть распределены по комплектам  типов. Перед началом реализации алгоритма распределения данных по комплектам элемент  вектора  определяется следующим образом: ().

Для определения вида модели вычислительного процесса обработки партий данных, включенных в группы () введены следующие обозначения: – время обработки данных i-го типа на l-ом сегменте конвейера (); – время переналадки l-го сегмента с обработки данных i-го типа на обработку данных k-го типа; – время первоначальной наладки l-го сегмента на обработку данных i-го типа;  – время 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 ); – матрица моментов времени начала обработки q-ых данных в партии, занимающей в  j-ю позицию (q – порядковый номер данных в партии в j-ой позиции в  (, где – количество данных в партии, входящей в группу , 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 одну рассматриваемую группу партий , – количество партий разных типов в группе . В рассмотрение введена матрица переналадок (), элементы  которой соответствуют длительностям переналадки сегментов с обработки данных i-го типа на обработку данных k-го типа, элементы – время первоначальной наладки сегментов на обработку данных i-го типа. Определение значений  и  (; ; ) выполняется в соответствии с изложенным ниже подходом для партий данных, входящих в одну группу  (индекс z группы партий  для временных параметров модели опущен при изложении).

Для позиции (q=1) данных i-го типа в (j=1)-ой партии в  выражение для определения  формируется при учете параметра  –времени наладки сегмента на обработку данных i-го типа: . Для  (,) имеем [1]: . В этом выражении первое слагаемое определяет длительность наладки сегмента конвейера на обработку данных в первой позиции в , второе слагаемое–длительность обработки данных в первой партии, предшествующих данным в q-ой позиции. Для первой позиции данных (q=1) (j=2)-ой партии в для  имеем: если  – время переналадки первого сегмента конвейера с обработки данных i-го типа (первая позиция партии в ) на обработку данных другого типа (вторая позиция партии в ), а – количество данных i-го типа в партии в первой (j=1) позиции в , тогда , где значение  определяется следующим образом:

, где .

Первой слагаемое в выражении для  и  определяет длительность наладки сегмента на обработку данных в партии, занимающей первую позицию в , второе слагаемое – длительность обработки данных в первой партии в , третье слагаемое – длительность переналадки сегмента с обработки данных в первой партии в на обработку данных во второй партии в . Для позиций  () значения  определяются выражением вида:

, .

По аналогии обобщенное выражение для определения  при  для последовательности  имеет вид: , моменты времени начала обработки данных в *q*-ой позиции в партии с индексом : . Выражения для  и  сформированы в общем виде следующим образом [1]:

;

;

.

Использование выражений для ,  и ,  () позволяет определить временные характеристики вычислительного процесса выполнения конвейеризированных программ обработки данных на сегментах конвейера. Метод построения расписаний обработки партий предполагает добавление текущей рассматриваемой партии данных i-го типа в конец последовательностей  () и определение эффективного местоположения партии 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 .

В работе [1] выполнено формирование вида критерия эффективности принятия решений при построении расписаний обработки партий данных на нижнем уровне иерархии подсистем. Полученный критерий эффективности определения решения по порядку обработки партий в системе имеет вид [1]:

. (2.1)

Первое слагаемое в (2.1) позволяет определить суммарный простой всех L сегментов конвейера перед началом обработки данных в первой позиции (q=1) в первой партии (j=1) в  (). Второе слагаемое определяет 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 ). Третье слагаемое позволяет определить суммарный простой сегментов, вызванный ожиданием готовности данных при обработке их внутри партий.

При формировании комплектов после окончания обработки могут быть использованы только данные, партии которых включены в группы (), тогда на втором уровне иерархии решение по составам групп  должно анализироваться с точки зрения количества комплектов, которые могут быть сформировано из данных, входящих в партии, включенные в группы (). Таким образом, на втором уровне иерархии системы на основе решения по составам групп партий  необходимо определить решение по составам комплектов данных, формируемых из результатов их обработки в группах партий. Элемент  () вектора соответствует количеству комплектов d-го типа, сформированных из данных, партии которых включены в группы (), тогда общее количество комплектов, сформированных из данных, партии которых включены в группы, определяется выражением: . Т.к. цель функционирования системы в течение заданных временных интервалов предполагает обработку данных таким образом, чтобы обеспечить формирование максимального количества комплектов, тогда полученное выражение использовано в качестве критерия эффективности решений  по составам групп партий на втором уровне иерархии системы ([1]).

На первом уровне иерархии критерий эффективности решений должен учитывать решение вида , формируемое на этом уровне, и эффективное решение по составам групп партий, формируемое на втором уровне в виде . Решение  характеризуется количеством сформированных комплектов разных типов, следствием из решения по составам групп партий является решение по количеству комплектов каждого типа – вектор  (решение  обеспечивает максимальное количество комплектов). При определении значения критерия , характеризующего решение на первом уровне, используется само это решение и решение по количеству комплектов каждого типа – вектор , характеризующий эффективное решение по составам групп партий . В качестве входных параметров при определении составов партий задается количество  () данных каждого типа, которые входят в партии для обработки, из результатов обработки которых формируются комплекты. Предполагается, что все данные, подаваемые на вход системы для обработки, должны быть использованы при формировании комплектов. Тогда решение  может быть охарактеризовано количеством данных, которые должны быть обработаны и включены в комплекты, но в соответствии с решением в комплекты не вошли. В соответствии с решением количество данных i-го типа, которые должны быть обработаны в системе, определяется выражением, где – элемент матрицы (A), значение которого равно количеству данных i-го типа в h-ой партии (), – количество партий данных i-го типа – i-ый элемент вектора (M). Общее количество данных различных типов, которые должны быть обработаны в системе, в соответствии с решением  определяется выражением . Количество данных различных типов, используемых при формировании комплектов  типов, определяется выражением вида . Тогда количество данных, которые должны быть обработаны в соответствии с входными параметрами решаемой задачи, но не использованы при формировании комплектов, определяется выражением .Полученное выражение используется в качестве критерия эффективности на первом уровне принятия решений.

Значение  – это количество данных в партии, занимающей j-ю позицию в последовательностях  (). Эти рассуждения распространены на последнюю партию в последовательности . Для определения последней партии, а также данных в последней позиции в этой партии использованы обозначения: –количество партий, входящих в группу  (индекс последней партии в ); – количество данных, входящих в последнюю в  партию (индекс последних данных в этой партии). Тогда – момент времени начала обработки последних данных в партии с индексом  на L-ом сегменте конвейера, а время окончания обработки этой партии на этом сегменте определяется выражением вида: . Ограничение на время обработки партий данных, входящих в группы  () , имеет вид:при . Тогда модель многоуровневого программирования для определения эффективных составов партий, составов групп партий и расписаний обработки партий имеет вид [1]:

1) первый (верхней) уровень: , где ;

2) второй уровень: , где ;

3) третий уровень:  (), где

(2.2)



4) ограничения на длительность реализации обработки партий групп  ():

 при .

С учётом модели (2.2) процедура определений эффективных составов партий данных, составов групп партий, обрабатываемых в течение задаваемых интервалов времени функционирования системы (), расписаний обработки партий каждой из групп в конвейерной системе предполагает: 1) определение количества партий данных i-ых типов () и составов этих партий (распределение данных i-ых типов по партиям); 2) определение составов групп партий данных, обрабатываемых в течение задаваемых интервалов времени (); 3) формирование порядка обработки партий (расписания их обработки). Иерархичность функций в системе определяет порядок принятия решений: 1) решение по составам партий данных в виде [(M), (A)], где (М) – вектор количества партий данных различных типов, (А) – матрица количества данных в партиях; 2) решение по составам групп партий в виде {()}; 3) решение по порядкам обработки партий данных, входящих группы () в виде:  (). Т.к. эффективность формирования групп партий (эффективность составов партий) оценивается на основе количества комплектов разных типов, формируемых из данных, обрабатываемых в группах партий, тогда дополнительным решением на втором уровне иерархии системы является решение по составам комплектов, определяемое в виде матрицы составов комплектов , значение элемента  которой соответствует количеству данных i-го типа, включаемых в комплекты d-го типа, и вектора  количества комплектов каждого типа.

# МЕТОД ПОСТРОЕНИЯ РЕШЕНИЯ ПО СОСТАВАМ ПАРТИЙ ДАННЫХ N ТИПОВ

Для обоснования метода формирования эффективных составов партий данных в рассмотрение введены следующие обозначения: – идентификатор рассматриваемого типа данных, состав партий которого изменяется (модифицируется) на текущем шаге алгоритма; – количество партий данных рассматриваемого -го типа, состав партий которого определяется; s– индекс исходного решения, подлежащего оптимизации путем поиска нового более эффективного решения в рамках некоторой окрестности; g – индекс шага алгоритма, выполняемого по отношению к номеру шага s, который соответствует новому формируемому решению, находящемуся в окрестности  ((s+g)– номер шага алгоритма по формированию решения в окрестности  текущего локально оптимального решения); – индекс (номер) партии, состав которой изменяется (модифицируется) при реализации алгоритма на текущей его итерации (на промежуточной итерации алгоритма выполняется изменение состава партии данных с индексом (номером) , итерация алгоритма предполагает реализацию определенной последовательности его шагов); I – множество типов данных, для которых должно быть выполнено формирование составов партий (первоначальный состав этого множества имеет следующий вид: ); – копия множества I типов данных, используемая при формировании локально оптимального решения по составам этих партий; A’– матрица – аналог матрицы составов партий А, используемая при определении наилучшего решения в окрестности  текущего локально оптимального решения, которому соответствует матрица A (матрица А’ используется при формировании новых промежуточных решений);  и – матрицы, предназначенная для хранения (буферизации) составов партий данных i’-го типа, сформированных на различных итерациях алгоритма; – количество решений по составам партий данных i’-го типа, сформированных на (s+g)-ой итерации алгоритма (хранение решений реализуется в матрице  размерностью ); – количество решений по составам партий данных i’-го типа, полученных на последующей ((s+g)+1)-ой итерации алгоритма (хранение решений реализуется в матрице  размерностью ); – индекс решения по составам партий данных i’-го типа в матрице  (); – индекс решения по составам партий данных i’-го типа в матрице  (); – параметр, предназначенный для хранения идентификатора решения (номера строки матрицы ), гарантирующего максимальное по модулю значение левого дискретного градиента целевой функции ; G– максимальное по модулю значение дискретного градиента  ([2]), достигаемое в окрестностях текущего локально оптимального решения; – параметр алгоритма, реализующий буферизацию значения , используемый при определении наилучших решений по составам партий некоторого i’-го типа; – множество типов данных, для которых .

Метод формирования составов партий данных реализует поиск локально оптимального решения в рамках окрестностей, способ формирования которых формулируется ниже. Для текущего локального оптимального решения по составам партий данных i-ых типов рассматриваются два типа окрестностей. При формировании окрестности первого типа построение решений связано с изменением составов партий данных в рассматриваемом их количестве  (), при формировании окрестности второго типа построение решений связано с увеличением количества  партий данных i-го типа и заданием начального решений для этого количества партий. Таким образом, если построение решений путем изменения составов партий данных -го типа в количестве  является невозможным (выполнены условия окончания формирования составов партий данных -го типа в количестве ), тогда реализуется изменение количества  партий данных этого типа, состав которых будет определяться.

# МЕТОД ПОСТРОЕНИЯ РЕШЕНИЯ ПО СОСТАВАМ ПАРТИЙ ДАННЫХ I – ОГО ТИПА

Формулируемый метод позволяет выполнить локальную оптимизацию решения (в рамках его окрестностей разных типов). Для обоснования формулировки алгоритма определения эффективных составов партий в рассмотрение введены: 1) способ формирования решений по составам партий данных некоторого i’-го типа; 2) способ формирования для текущего локально оптимального решения окрестностей первого типа, в рамках которых будет выполняться поиск эффективных (лучших) решений.

Для обоснования способа формирования решений по составам партий данных в количестве  в рассмотрение введены следующие условия и рассуждения:

1) количество данных -го типа () в формируемых партиях не может быть меньше, чем 2, т.е. (), где h – идентификатор партии данных (номер столбца в -ой строке матрицы А); если при формировании начального решения по составам партий данных -го типа для задаваемого их количества  для h=1 (первая партия) получено, что , тогда исследование эффективности составов партий данных i’-го типа в количестве  должно быть завершено, т.к. количество данных в партиях не может быть менее 2;

2) способ формирования начального решения для заданного количества  партий данных -го типа предполагает, что (, ), а элемент  определяется следующим образом:; в дальнейшем при построении решений по составам партий данные извлекаются из партии с h=1 и распределяются по остальным партиям ();

3) задаваемые первоначально количества  партий данных i-ых типов () равны 2 (); модификация количества партий данных рассматриваемого -го типа предполагает, что параметр  увеличивается (при определении составов партий) до тех пор, пока в начальном решении по составам партий для полученного значения  выполняется условие ; при выполнении условия  (для начального решения при соответствующем значении ) исследование составов партий -го типа прекращается;

4) формирование эффективных решений по составам партий (в количестве ) связано с увеличением (на 1) количества данных в партии с индексом  (индекс столбца в соответствующей строке матрицы А) при одновременном уменьшении количества данных в партии с h=1 (последовательное увеличение количества данных в партиях с индексом ); решение по составам партий в количестве  характеризуется значением целевой функции , для которого возможно выполнение условия  (отрицательное значение левого дискретного градиента целевой функции ) – количество данных в партии с индексом  увеличивается, что приводит к уменьшению значения ;

5) формирование эффективных решений по составам партий предполагает увеличение числа данных в партии с индексом  (рассматриваемый номер партии, при неизменном составе остальных партий с индексами h )) и уменьшении количества данных в партии с h=1 в соответствии с выражением ; при выполнении условия   модификация составов партий может быть продолжена; при выполнении условия  полученное решение по составам партий в количестве  является решением, аналогичным полученному на предыдущих шагах алгоритма (дублирует полученное ранее решение), тогда полученное решение, для которого выполняется введенное условие, не интерпретируется. Сформулированному условию соответствует Теорема 1.

Теорема 1. Если при формировании решений по составу  партий данных -го типа для построенного решения выполняется условие  (где ), тогда полученное решение дублирует сформированное ранее и не рассматривается, полученное решение не используется в дальнейшем для последующего формирования новых решений.

Доказательство. При формировании составов  партий данных -го типа определяется эффективное количество данных в каждой из партий, при этом порядок следования партий значения не имеет. Алгоритм метода формирования составов партий предполагает, что для партии, индекс (номер)  которой является задаваемым в ходе выполнения алгоритма, реализуется увеличение (на единицу) количества данных в ней. При этом количество данных в остальных h-ых партиях (за исключением h=1) остается неизменным. Для партии с индексом (номером) h=1 количество данных в ней определяется выражением: . Пусть на (s-1)-ом шаге алгоритма сформировано решение по составам партий , для которого i’-ая строка матрицы А имеет следующий вид:  . При этом значения элементов  и  такие, что:  (). В соответствии со способом формирования составов партий на s-ом шаге преобразование элементов  () выполняется следующим образом:  (при ), , . Получено, что  при неизменных значениях элементов  (). На (s+1)-ом шаге выполняются преобразования элементов  ():  (), , . В результате получено, что  (т.е. ). Тогда  и  при неизменных значениях  (, ). Поэтому сформированное на (s+1)-ом шаге решение дублирует решение, полученному на (s-1)-ом шаге. Тогда при  (в общем виде при ) формируются дублирующие решения, которые не могут быть проинтерпретированы как новые сформированные решения по составам партий, что и требовалось доказать. Аналогичные рассуждения могут быть сформулированы для случаев . Тогда дублирующее решение будут получено на s-ом шаге алгоритма формирования составов партий.

Сформулированный способ модификации составов партий i’-го типа предусматривает, что на основе решений, находящихся в окрестности текущего локально оптимального решения с меньшей метрикой (окрестность ), формируются решения, находящиеся в окрестности с большей метрикой (окрестность ). Метрика k окрестности  определяется выражением вида , где – элемент i’-ой строки матрицы , соответствующей текущему локально оптимальному решению по составам партий данных i’-го типа, сформированному на s-ой итерации алгоритма, – элемент i’-ой строки матрицы , соответствующей решению по составам партий, сформированному на (s+g)-ой итерации алгоритма, которое находится в окрестности  с метрикой k решения .

В матрице  реализуется хранение промежуточных решений, находящихся в окрестности , на основе которых реализуется формирование решений, входящих в окрестность  (в матрице  хранятся решения, используемые при формировании новых решений в окрестности ). В матрице  реализуется хранение формируемых решений, находящихся в окрестности . Тогда на основе промежуточного решения по составам партий данных (-ой строки  матрицы ), формируется совокупность новых решений, хранимых в матрице , находящихся в окрестности . Если для текущего рассматриваемого решения (-ой строки  матрицы ) выполняется условие , тогда увеличение на 1 значения  (при неизменном значении ), а затем увеличение на 1 значения  (при неизменном значении ) позволяет получить одинаковые решения по составам партий данных i’-го типа. Сформулированное условие позволяет исключить повторяющиеся решения, получаемые при модификации решения по составам партий данных i’-го типа. На его основе сформулирована Теорема 2, ограничивающая количество формируемых решений.

Теорема 2. Если на основе решения по составам  партий данных -го типа (представленного в виде строки  матрицы ) реализуется формирование нового решения, при этом выполняется условие , тогда не требуется выполнять модификацию элемента  (его увеличение на 1). Должен быть выполнен переход к элементу , который в случае выполнения  увеличивается на 1. Если реализуется условие , тогда на основе решения, представленного в виде строки  матрицы  будет сформировано новое решение, дублирующее полученное решение в окрестности .

Доказательство Теоремы 2 строится по аналогии с доказательством Теоремы 1.

В соответствии с введенными в рассмотрение типами окрестностей, предполагается, что решения, входящие в окрестность первого типа, формируются путем изменения составов партий в количестве  (при неизменном значении ). Решения, входящие в окрестность второго типа, формируются путем увеличения количества партий , задания для количества партий  начального решения и последующего изменения составов партий для этого количества.

Для определения локально оптимального решения по составам партий данных i-ых типов () сформулированы: 1) способ формирования решений по составам партий данных некоторого i’-го типа, включаемых в окрестности  первого типа (предполагающий изменение составов партий данных при их количестве ), на основе решений, входящих в окрестность ; 2) обобщенный алгоритм идентификации локально оптимального решения по составам партий данных. Входными параметрами алгоритма формирования решений, включаемых в окрестность  первого типа, на основе решений, входящих в окрестность , являются: идентификатор i’-го типа данных, для которых определяется составы партий; матрица  размерностью , содержащая решения, входящие в окрестность текущего локально оптимального решения; количество партий , составы которых будут изменяться; количество  решений в окрестности ; индекс s текущего шага алгоритма идентификации локально оптимального решения.

Алгоритм формирования решений по составам партий данных i’-го типа в окрестности  на основе решений в окрестности  следующий:

1) инициализация значений параметров , , , , ;

2) инициализация номера партии , состав которой будет изменяться, значением 2 ();

3) модификация значения ; для -ой партии -го типа увеличение количества данных на 1; формирование решения по составам партий (строки  матрицы ): (),,  ; ;

4) проверка выполнения условия  (при заданном количестве партий ); в случае выполнения этого условия сформированное в виде строки  матрицы  решение не интерпретируется (в соответствии с Теоремой 1), тогда ; реализуется переход к шагу 8 (переход к формированию новых решений в окрестности  на основе решения из окрестности , хранимого в матрице ); при условии  полученное решение интерпретируется, существует возможность дальнейшего формирования решений, выполняется задание значения параметра j=1 (j–значение шага изменения номера партии h’) и реализуется переход на шаг 5;

5) осуществляется проверка условия ; при  реализуется проверка условия формирования решения по составам партий -го типа в виде  (в соответствии с Теоремой 2); если условие  не выполняется, тогда реализуется переход на шаг 6; при выполнении условия существует возможность формирования дополнительного решения по составам партий данных i’-го типа, тогда , осуществляется переход на шаг 3;

6) в случае выполнения условия  (в соответствии с Теоремой 2) при модификации ()-ой партии будет получено решение, дублирующее полученные ранее; осуществляется модификация параметра j=j+1; выполняется переход на шаг 5;

7) при выполнении  сформированы все возможные решения (хранящиеся в матрице ) с использованием одного решения с индексом  (-ой строки  матрицы ); переход на шаг 8;

8) модификация текущего значения – индекса решения, хранимого в матрице  (осуществляется переход к следующему решению в окрестности , на основании которого могут быть сформированы составы партий данных): , если  (рассмотрены не все решения из окрестности , хранимые в матрице ), тогда реализуется переход на шаг 2; в случае выполнения условия  все решения, хранимые в матрице , находящиеся в окрестности текущего локально оптимального решения, использованы для формирования решений, находящихся в окрестности , хранимых в матрице ; выполняется переход на шаг 9;

9) при  выполняется сравнение решений, хранимых в матрице , с точки зрения дублирования ими друг друга (процедура сравнения предполагает формирование копии матрицы , в копии матрицы  упорядочивание элементов в каждой -ой строке () по убыванию значений элементов, поэлементное сравнение каждой -ой строки () с другими -ыми строками (), удаление из матрицы  строк с индексами , которые соответствуют строкам в копии матрицы , дублирующим рассматриваемую -ю строку в копии ; при удалении q2’-ой строки из матрицы  реализуется изменение ); в результате повторяющиеся решения из матрицы  удалены, остальные решения, хранящиеся в этой матрице (за счет ее дублирования в процедуре сравнения), остались без изменения; инициализация значений  ; ; при  осуществляется переход на шаг 11;

10) реализация проверки условия ; при  не сформированы новые решения в окрестности , которые могут быть более эффективными, чем рассматриваемое локально оптимальное решение A(s), т.е. не сформирована окрестность  локально оптимального решения, выполняется переход на шаг 16;

11) рассмотрение решений по составам партий данных i’-го типа, находящихся в окрестности  текущего локально оптимального решения, с точки зрения их эффективности; для этого выполняется инициализация i’-ой строки матрицы А’ решением по составам партий данных, хранимым в -ой строке  матрицы : (); передача сформированного решения  на второй уровень для определения составов групп партий (при этом исходное локально оптимальное решение A(s) не изменяется);

12) получение со второго уровня иерархии эффективного решения по составам комплектов, формируемых из данных, обрабатываемых в группах, в виде матрицы , использование решения  для вычисления значения критерия  (значения );

13) для сформированного решения [M(s+g),A’(s+g)] вычисление левого (либо правого) дискретного градиентов  (); вычисление  () выполняется по формулам [2]: а) ; б) ;

14) при выполнении условия  сформированное решение [M(s+g), A’(s+g)] не является более эффективным (при минимизации критерия ), чем решение (для количества партий ); осуществляется модификация значения ; если  (т.е. рассмотрены не все решения, находящиеся в окрестности ), тогда осуществляется модификация индекса шага алгоритма g=g+1, реализуется переход на шаг 11 (переход к анализу эффективности следующего решения из окрестности ); если , тогда выполняется переход на шаг 16;

15) если для текущего рассматриваемого решения по составам партий (-ой строки  матрицы ) реализуется , тогда выполняется сравнение значения  с максимальным значением  градиента, достигаемым в рассматриваемой окрестности  (для данных i’-го типа); если , тогда текущее решение по составу партий i’-го типа данных является наилучшим среди рассмотренных решений из окрестности , индекс решения (строки  матрицы )  сохраняется: , модификация значения : ; реализуется переход к следующему решению в окрестности , для этого ; если  (рассмотрены не все сформированные решения, находящиеся в окрестности ), тогда g=g+1 и реализуется переход на шаг 11 (переход к анализу эффективности следующего решения из окрестности ); если  (рассмотрены все решения по составам партий данных i’-го типа, находящиеся в окрестности ), тогда выполняется переход на шаг 16;

16) останов алгоритма.

Выходными параметрами алгоритма определения эффективных решений в окрестности  являются: значение  количества сформированных решений по составам партий данных i’-го типа, входящих в окрестность  (при  на основе решений, входящих в окрестность , могут быть сформированы новые решения, входящие в окрестность  первого типа (при неизменном количестве партий )), при  на основе решений, входящих в окрестность , не определены решения по составам партий данных i’-го типа в , на основе которых могли быть сформированы решения, входящие в окрестность  первого типа, поэтому требуется выполнить переход к формированию окрестности второго типа путем увеличения значения  и определения для этого количества партий  начального решения по их составу; значение – идентификатора строки в матрице , соответствующей наилучшему решению по составам партий данных i’-го типа в сформированной окрестности  (при  и  не получено решение, входящее в окрестность , являющееся лучшим решением по составам партий данных i’-го типа, чем текущее локально оптимальное решение, необходим переход к окрестности  первого типа с большей метрикой для определения в ней эффективного решения по составам партий данных i’-го типа); матрица , содержащая (при ) все решения, входящие в окрестность . Реализация предложенного алгоритма формирования решений по составам партий данных i’-го типа, входящих в окрестности  с различным значением метрики k, представлена на Рис.1 для заданных значений параметров  и .



Рисунок 1 – Реализация алгоритма формирования решений по составам партий в окрестностях  (при  и )

\_\_\_\_ – решения, исключаемые из рассмотрения в соответствии с Теоремой 1;

– решения, которые не формируются в соответствии с Теоремой 2;

– решения, исключаемые из рассмотрения как дублирующие решения в окрестности 

# МЕТОД ОПРЕДЕЛЕНИЯ ПОРЯДКА ОБРАБОТКИ ПАРТИЙ ДАННЫХ (РАСПИСАНИЙ ОБРАБОТКИ ПАРТИЙ ДАННЫХ) В ГРУППАХ

В силу используемого предположения о том, что партии в сформированном расписании имеют одинаковый порядок обработки во всех последовательностях  () для отдельной рассматриваемой группы  и того, что количество типов данных ограничено, тогда является ограниченным и количество возможных решений. Для формирования расписаний обработки партий данных в системе использован метод окрестности (метод поиска лучшего решения в рамках некоторых окрестностей текущего локального оптимального решения). Для обоснования алгоритма определения эффективного порядка обработки партий данных в последовательностях  (), образующих расписание  отдельной группы , в рассмотрение введены следующие обозначения: s-индекс (номер) шага алгоритма, на котором зафиксировано текущее локально оптимальное решение по порядку обработки партий в группе ();  – количество партий данных, размещенных в последовательностях  () на предшествующих и текущем s-ом шагах алгоритма;  – вектор, i-ый элемент которого – это количество партий данных i-го типа, размещенных в последовательностях  () на предыдущих и текущем s-ом шагах алгоритма; g– количество выполненных шагов по изменению локально оптимального решения  для получения решения , соответствующего новому порядку партий в  () (т.е. индекс шага алгоритма по определению лучшего решения в окрестности текущего локально оптимального); – метрика окрестности локально оптимального решения , используемая для идентификации отличия сформированного решения  от решения ; – максимально возможная метрика окрестности исходного решения , в которой выполняется поиск более эффективного решения;  – окрестность решения , в которой выполняется поиск более эффективного решения;  – индекс элемента вектора , соответствующего составам партий данных i-го типа в группе , который содержит количество данных в партии, добавляемой в последовательности (), v – индекс текущего рассматриваемого столбца матриц  и , в i-ой строке которого значения элементов  и  соответствуют характеристикам рассматриваемой партии данных i-го типа, эффективное местоположение которой в последовательностях  определяется на текущей итерации алгоритма (, );  – индекс столбца матриц  и , в котором первоначально размещаются характеристики рассматриваемой партии данных i-го типа при добавлении этой партии в последовательности  (, ).

Вектор  используется при определении количества  партий данных различных типов, размещенных в последовательностях (). Первоначальная инициализация элементов () выполнена следующим образом:  (). Значение количества партий , размещенных в  (), определяется выражением: . Расчет метрики  окрестности решения : , где – количество типов данных, партии которых включены в рассматриваемую группу , – общее количество партий данных разных типов, размещенных в последовательностях  (). Метрика  определяет число элементов в матрице  (число позиций в последовательностях ), в которых изменены значения по отношения к виду матрицы , соответствующей локально оптимальному решению . Начальное решение  по определению позиции партии данных i-го типа в последовательностях () формируется путём добавления -ой партии данных i-го типа в количестве ) в конец этих последовательностей. Через g обозначен номер промежуточного решения , которое формируется на основе исходного , находящегося в окрестности , характеризуемой метрикой . Формирование решения предполагает изменение положения рассматриваемой партии в последовательностях  относительно ее положения, которое соответствует решению .

Алгоритм построения эффективного расписания обработки партий, входящих в группу , предполагает последовательное размещение в последовательностях () всех партии данных i-ых типов () из этой группы и идентификацию эффективного местоположения каждой из этих партий в . Таким образом, алгоритм реализует жадный подход к оптимизации, при котором каждое последующее эффективное решение по размещению одной из партий группы  в последовательностях  формируется на основе эффективных решений по порядку обработки партий, рассмотренных на предыдущих шагах алгоритма. Порядок шагов алгоритма построения эффективного расписания обработки партий рассмотрен с точки зрения размещения в последовательностях () партий данных некоторого i-го типа в количестве  и определения эффективного положения (эффективных позиций) этих партий в последовательностях  (). Алгоритм определения эффективного положения (позиций) партий данных i-го типа в последовательностях  () имеет следующий порядок шагов:

1) некоторое сформированное на предыдущем (s-1)-ом шаге алгоритма решение определяется соответствующим видом матриц  и , номер партии  (номер элемента вектора  ()), данные которой в количестве  размещается в последовательностях ()), инициализируется значением 1 ();

2) в силу того, что – количество партий данных, добавленных в () до s-го шага алгоритма, тогда для партии данных i-го типа определяется индекс  столбца матриц ,, в котором будут размещаться параметры, ей соответствующие: ;

3) добавляемая в последовательности  партия данных i-го типа в количестве  элементов размещается в конце последовательностей (); для этого в i-ой строке -го столбца матриц  и  задаются значения, соответствующие этой партии (элементы , и  инициализируются значениями 1 и  соответственно); модификация матриц  и , связанная с инициализацией элементов -ых столбцов, выполняется следующим образом: ; ; ,  при  и ; сформированное начальное решение по размещению в последовательностях () рассматриваемой партии данных i-го типа фиксируется как локально оптимальное решение: ;

4) значение g индекса шага текущего промежуточного решения инициализируется значением 1 ((s+g) – номер промежуточного шага алгоритма, связанного с определением локального оптимального решения по размещению партии данных i-го типа в количестве  элементов в окрестности текущего решения ), индексу текущего рассматриваемого столбца v матриц  и  присваивается значение ;



5) на основе матриц ,  реализуется вычисление элементов матриц  ;

6) для полученного вида матриц ,  и матриц   определяется значение критерия  и значения элементов матрицы ;

7) в последовательностях  изменяется порядок партий таким образом, что рассматриваемая партия данных i-го типа перемещается на одну позицию в начало ; выполняемые действия с матрицами  и  имеют вид: , , , , , , , , где k – индекс строки в матрицах  и , в которой элементы  и  (предыдущая позиция для рассматриваемой партии, которую она займет на (s+g)-ом шаге алгоритма); в результате формируются модифицированные матрицы , ; индекс v столбца, идентифицирующий местоположение парии в  , модифицируется: ;

8) с использованием матриц ,  вычисляются матрицы и , а также значение критерия;

9) приполучено решение не худшее, чем текущее, тогда решение  фиксируется как локально эффективное: , фиксируется критерия ; значение s=s+g, значение индекса g шага поиска следующего локального оптимального решения в окрестности текущего  задаётся равным 1, выполняется переход к шагу 7;

10) если , тогда проверяется выполнение условия  (– метрика максимально возможной окрестности исходного решения , являющегося локально эффективным, в которой выполняется поиск более эффективного решения ;

11) при выполнения условия  реализуется дальнейший поиск более эффективного решения в окрестности  исходного расписания , модифицируются: индекс g промежуточного шага поиска решения (g=g+1), значение метрики окрестности ; проверяется выполнение условия  (рассматриваемая партия является первой в  () (, )); при истинности условия  выполняется переход к шагу 7, при выполнении условия  выполняется переход к шагу 13;

12) при выполнении условия  реализуется переход к шагу 13;

13) полученное локально эффективное решение фиксируется (фиксируется полученный в результате порядок партий), изменение положения рассматриваемой партии данных i-го типа в количестве  в дальнейшем не реализуется;

14) модификация значений количества партий данных i-го типа, которые должны быть размещены в последовательностях  : ; изменение значения счетчика количества размещенных в   партий данных i-го типа: ; если , тогда реализуется изменение индекса следующей рассматриваемой партии данных i-го типа: ; изменение значения шага алгоритма: s=s+1; изменение номера столбца матриц , , в котором на следующем шаге алгоритма будут размещаться значения параметров, характеризующих добавляемую в   партию данных:; выполнить переход к шагу 3;

15) при выполнении условия  в последовательностях   размещены все партии данных i-го типа, включенные в состав группы партий  в соответствии с решением .

Алгоритм формирования порядка обработки партий позволяет выполнить упорядочивание партий данных каждого i-го типа () в количестве  (в каждой из которых  данных ) в последовательностях  , т.е. определить расписание обработки партий на сегментах конвейера. Реализация алгоритма обработки партий данных должна быть выполнена совместно с реализацией алгоритма формирования составов партий данных и алгоритма определения групп партий, соответствующих их сформированным составам партий.