МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРCИТЕТ»**

Институт информационных технологий и управления в технических системах

(полное название института)

Кафедра информационных систем

(полное название кафедры)

## **Пояснительная записка**

к выпускной квалификационной работе бакалавра

на тему «Система построения расписаний обработки партий данных в в в в в в   
конвейерной системе. Подсистема формирования составов групп партий \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил: студент IV курса, группы ИС–42о \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

направления подготовки (специальности) 09.03.02 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_Информационные системы и технологии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки (специальности))

профиль (специализация)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Икитян Руслан Владимирович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество студента)

Руководитель \_\_\_Кротов Кирилл Викторович, к.т.н., доцент\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, степень, звание, должность)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата допуска к защите «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (инициалы, фамилия)

2016 г

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 6](#_Toc453708971)

[1 ОБОСНОВАНИЕ АКТУАЛЬНОСТИ РАЗРАБОТКИ МЕТОДОВ ПОСТРОЕНИЯ КОМПЛЕКСНЫХ РАСПИСАНИЙ ГРУППОВОЙ ОБРАБОТКИ ПАРТИЙ 9](#_Toc453708972)

[1.1 Постановка задачи 9](#_Toc453708973)

[1.2 Анализ существующих методов построения расписаний групповой обработки данных 11](#_Toc453708974)

[1.3 Обоснование подхода к решению задачи построения расписаний групповой обработки данных разных типов при наличии ограничения на интервал функционирования системы 15](#_Toc453708975)

[2 МОДЕЛЬ МНОГОУРОВНЕВОГО ПРОГРАММИРОВАНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ СОСТАВОВ ПАРТИЙ ДАННЫХ, ГРУПП ПАРТИЙ, РАСПИСАНИЙ ОБРАБОТКИ ПАРТИЙ 18](#_Toc453708976)

[2.1 Описание иерархической модели построения групп партий данных 18](#_Toc453708977)

[2.2 Метод определения эффективных составов групп партий данных 28](#_Toc453708978)

[2.3 Описание алгоритма формирования начального решения групповой обработки партий данных 29](#_Toc453708979)

[2.4 Описание алгоритма формирования оптимального решения групповой обработки партий данных 35](#_Toc453708980)

[3 ОБОСНОВАНИЕ МОДЕЛИ ПОСТРОЕНИЯ КОМПЛЕКСНЫХ РАСПИСАНИЙ ОБРАБОТКИ ДАННЫХ РАЗНЫХ ТИПОВ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЙ НА ИНТЕРВАЛ ОБРАБОТКИ 47](#_Toc453708981)

[3.1 Обоснование использования аппарата теории иерархических и неантагонистических игр при построении расписаний групповой обработки партий 47](#_Toc453708982)

[3.2 Теоретико–игровая модель построения расписаний групповой обработки 50](#_Toc453708983)

[4. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПОСТРОЕНИЯ РАСПИСАНИЙ ГРУПП ПАРТИЙ ДАННЫХ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЯ НА ИНТЕРВАЛ ОБРАБОТКИ ГРУППЫ (ВТОРОЙ УРОВЕНЬ) 64](#_Toc453708984)

[4.1 Описание программы 65](#_Toc453708985)

[4.1 Описание алгоритмов функционирования второго уровня В качестве начальных данных алгоритм принимает на вход партии данных, которые прежде формировались на верхнем уровне. 66](#_Toc453708986)

[5 АНАЛИЗ РЕЗУЛЬТАТОВ РАБОТЫ РАЗРАБОТАННЫХ МЕТОДОВ ПОСТРОЕНИЯ ГРУППОВОЙ ОБРАБОТКИ ПАРТИЙ ДАННЫХ 69](#_Toc453708987)

[5 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТИРОВАНИЯ ПРОГРАМНОГО ПРОДУКТА 71](#_Toc453708988)

[6.1 Маркетинговые исследования программного продукта 71](#_Toc453708989)

[6.1.1 Сущность и этапы маркетинговых исследований 71](#_Toc453708990)

[6.1.2 Конкурентоспособность программного продукта 71](#_Toc453708991)

[6.1.3 Оценка рыночной направленности 71](#_Toc453708992)

[6.1.4 Сегментация рынка 72](#_Toc453708993)

[6.1.5 Предпочтительный потребитель программного продукта 72](#_Toc453708994)

[6.1.6 Жизненный цикл программного продукта 72](#_Toc453708995)

[6.2 Определение затрат на проектирование 73](#_Toc453708996)

[6.2.1 Расчет трудоемкости разработки программного продукта 73](#_Toc453708997)

[6.2.2 Расчет эксплуатационных затрат разработчика 75](#_Toc453708998)

[6.2.3 Расчет сметы затрат на проектирование 76](#_Toc453708999)

[6.3 Расчет капитальных затрат 77](#_Toc453709000)

[6.4 Формирование цены предложения разработчика 77](#_Toc453709001)

[6.5 Оценка экономической эффективности проектирования программного продукта 78](#_Toc453709002)

[7 БЕЗОПАСНОСТЬ ЖИЗНИДЕЯТЕЛЬНОСТИ 81](#_Toc453709003)

[7.1 Основные сведения 81](#_Toc453709004)

[7.2 Краткая характеристика помещения и выполняемых работ 82](#_Toc453709005)

[7.3 Планировка и размещение оборудования и рабочих мест 83](#_Toc453709006)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 93](#_Toc453709007)

[ПРИЛОЖЕНИЕ А 95](#_Toc453709008)

[ПРИЛОЖЕНИЕ Б 103](#_Toc453709010)

[ПРИЛОЖЕНИЕ В 112](#_Toc453709011)

# **АННОТАЦИЯ**

Данная выпускная квалификационная работа описывает современный ме-тод оптимизации решения задачи теории расписаний. Целью данной работы является разработка и исследование иерархической системы и метода построе-ния комплексного расписания групповой обработки партий требований.

Ключевые слова: иерархическая система, конвейерная обработка, опти-мизация, требования, партии, группы, расписание.

В работе представлена иерархическая модель построения расписания групповой обработки данных различных типов при условии на временное ограничение работы всей системы, методы построения партий данный фикси-рованного и всех типов и метод построения и оптимизации расписания.

На основе полученных методов был разработан модульный программ-ный продукт, описанный в разделе 4.

Для доказательства работоспособности этой системы в целом в разделе 5 приведен сравнительный анализ всей системы с работой системы без использо-вания модуля оптимизации составов партий данных.

Экономическое обоснование программного продукта приведено в разде-ле 6.

Показатели охраны труда и методы их улучшение приведены в разделе 7.

В представленной пояснительной записке к выпускной квалификационной работе 5 основных разделов на 112 страницах, включающих:

# **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. Одним их способов формирования приближенных методов теории расписаний является «жадный» подход, применение которого к сформулированной задаче будет развито в данной работе.

# **ОБОСНОВАНИЕ АКТУАЛЬНОСТИ РАЗРАБОТКИ МЕТОДОВ ПОСТРОЕНИЯ КОМПЛЕКСНЫХ РАСПИСАНИЙ ГРУППОВОЙ ОБРАБОТКИ ПАРТИЙ**

## **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.

## **Анализ существующих методов построения расписаний групповой обработки данных**

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 [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 время. решение поставленной задачи за ограниченное

Таким образом, решение задачи определения составов групп 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

## **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).

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

## **Описание иерархической модели построения групп партий данных**

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]:

.

(2.1)

В выражении (2.1) 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 .

## **2.3 Описание алгоритма формирования начального решения групповой обработки партий данных**

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;

2) инициализация  значением 0 ();

3) инициализация множеств  () и множества  в виде: ; (инициализация значением  предполагает, что наборы вида будут добавляться в состав соответствующих множеств);

4) ; 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   
(Рисунок.1). Анализ видов последовательностей  (Рисунок.1) позволил определить следующие причины неэффективного использования ресурса времени системы при обработке партий:

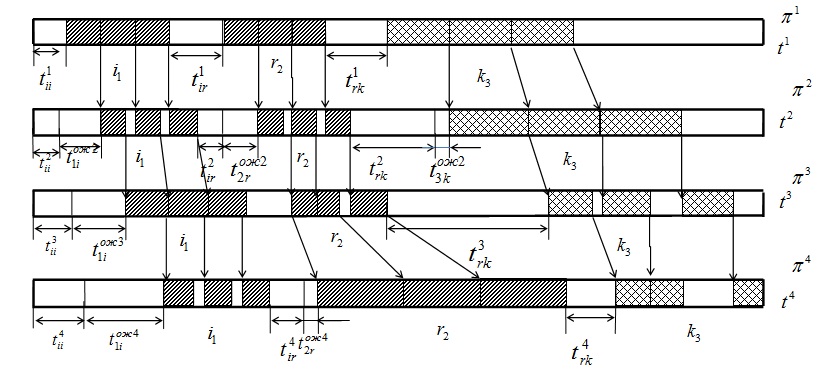
1) несогласованность длительностей обработки данных партии в *j*–ой позиции в последовательностях , которая обуславливает простои сегментов в ожидании готовности к обработке данных этой партии; в частности, 

Рисунок 1 – Заданный вид последовательностей обработки партий в группе 

для идентификации позиции удаляемой партиинесогласованность 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 ).

## **2.4 Описание алгоритма формирования оптимального решения групповой обработки партий данных**

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:

. (2.1)

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:

, . (2.2)

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

, где. (2.3)

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:

 (2.4.1)

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 . (2.4.2)

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 (2.4.1) либо (2.4.2);

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 (2.4.*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 осуществляется 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.

# **ОБОСНОВАНИЕ МОДЕЛИ ПОСТРОЕНИЯ КОМПЛЕКСНЫХ РАСПИСАНИЙ ОБРАБОТКИ ДАННЫХ РАЗНЫХ ТИПОВ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЙ НА ИНТЕРВАЛ ОБРАБОТКИ**

### **3.1 Обоснование использования аппарата теории иерархических и неантагонистических игр при построении расписаний групповой обработки партий**

В п. 2.3 структура решения обосновывается как иерархически–упорядоченная система, на каждом уровне которой существует критерий оценки эффективности. Тогда в общем виде модель системы может быть формализована в виде:

,

(3.1)

где – решения на первом, втором и третьем уровнях соответственно, – множества решений на уровнях,  – функции, с использованием которых задаются ограничения при выборе решений на этих уровнях, – эффективные решения, полученные на втором и третьем уровнях соответственно. При решении подобных задач используют понятия иерархических игр или игр по Штакельбергу. [10–12]. Задачи с иерархическими играми относятся к разделу теории игр.

В игре по Штакельбергу участники выбирают стратегии последовательно; тот, кто анонсирует стратегию первым, называется лидером. Равновесие по Штакельбергу достигается, когда лидер решает задачу оптимизации своей целевой функции, принимая во внимание оптимальный ответ последователя.

В системе уровни (участники) формируют стратегии последовательно, вышестоящий уровень (лидер) решает задачу оптимизации целевой функции основываясь на оптимальном решении нижестоящего уровня (последователя). Это является классической формулировкой равновесия по Штакельбергу.

Сформулируем задачу формирования групп (второй уровень) в теоретико–игровой постановке. Каждый игрок принимает решение, связанное с выбором состава задания . Каждая партия размещается в одном задании , тогда обмен данными между двумя группами приводит к изменению значений критериев для игроков на среднем уровне планирования. Выбор определенного состава партий в группе  одним из игроков является зависящим от выбора составов групп другими игроками. Так как каждая из партий может быть размещена только в одной из групп , то изменение решения одним (z–ым) игроком связано с обменом данными между группами (изменением состава групп  и ), что приводит к изменению значений критериев для игроков  и  на верхнем уровне планирования. Значение критерия *z*–ого игрока, обозначенное через , зависит от решения  *z*–ого игрока и (опосредовано) от решений (составов групп) других игроков. Тогда обозначение критерия –ого игрока задано в виде , где  – общее число интервалов обработки (групп). В силу того, что принятие *z*–ым игроком решения, характеризуемого значением , не связано с «потерями» игрока , «выгода» которого характеризуется критерием , то рассматриваемое взаимодействие игроков отнесено к типу неантагонистических игр.

Процедура принятия решения по идентификации состава заданий  определена как неантагонистическая игра  лиц, тогда процессу формирования эффективных решений соответствует нахождение ситуации равновесия в неантагонистической игре в виде , где  – –ая эффективная группа, т.е. эффективное решение –ого игрока.

Подходом к нахождению оптимального решения в этом случае является выполнение равновесия по Нэшу [13]. Понятие равновесия по Нэше формулируется следующим образом. Пусть  – произвольная ситуация в игре Г, а  – некоторая стратегия игрока i. Ситуация отличная от  только тем, что стратегия игрока i заменена на стратегию , т.е. , что обозначается . Ситуация  называется ситуацией равновесия по Нэшу [13], если для всех  и  имеет место неравенство (3.2).

 (3.2)

Таким образом, решение задачи составления расписаний групповой обработки данных разных типов обеспечивается определением ситуации двух равновесных типов: равновесие по Штакельбергу в иерархической игре и равновесия по Нэшу между группами на втором уровне планирования (рисунок 3.1).

**…**

**…**

**…**

**…**

**Верхний уровень**

**Средний уровень**

**Нижний уровень**

Равновесие по Штакельбергу

1–го уровня

Равновесие по Нэшу

Равновесие по Штакельбергу

2–го уровня

Рисунок 3.1 – Трёхуровневая организация системы построения расписаний обработки данных

### **3.2 Теоретико–игровая модель построения расписаний групповой обработки**

Для построения теоретико–игровой модели системы необходимо ввести некоторые обозначения:

1. *N* – множество типов данных, поступающих в систему (, *n*– количество типов данных);
2. – множество, элемент  которого – это количество данных *i*–го типа, обработка которых должна быть выполнена в системе ();
3.  – количество партий данных соответствующего *i*–того типа, формируемых на верхнем уровне принятия решений, элементы  образуют вектор *(М)*;
4. *(А) –* матрица составов партий данных различных (*i*–тых типов), элемент  которой соответствует количеству данных *i*–го типа в *u–*ой партии (); размер матрицы *(А) –* , где ; если , то  при ;
5.  – количество партий данных *i–*го типа в группе партий (если партии данных *i–*го типа входят в );
6.  – вектор количества данных *i–*го типа в  партиях в группе ;
7.  – число типов данных в партиях группы .

В соответствии с функциями уровней иерархии системы между уровнями выполняется обмен информацией следующего вида:

1) на вход первого уровня (для определения состава партий данных различных типов) подаются множества *N* и (программа работ); с выхода уровня– состав партий данных соответствующих типов,– решение в виде *[(М), (А)]*;

2) на вход второго уровня – состав партий данных; с выхода уровня – сформированные с учетом ограничения на интервал времени работы системы группы партий данных  (, состав групп партий);

3) на вход третьего уровня – совокупность групп партий данных (); с выхода уровня – сформированные порядки обработки партий данных групп для заданных интервалов времени (расписания ).

При распределении совокупности партий данных *i–*ых типов (представленной в виде решения *[(М), (А)]*) по группам партий () состав партий не меняется (значения  и , поступившие с верхнего уровня, изменены быть не могут). Партии данных некоторого *i–*го типа могут входить в различные группы партий . Через обозначим количество партий данных *i–*го типа в группе партий (если партии данных *i–*го типа входят в ), через  обозначим вектор количества данных *i–*го типа в  партиях в группе . При распределении партий данных *i–*го типа по группам для  и элементов векторов  выполняются условия: ; , где соответствующий элемент матрицы *(А)* (с первого уровня принятия решений). Для определения состава партий данных *i–*го типа, входящих в группу партий , в рассмотрение введен набор параметров вида:, тогда группа партий – это совокупность наборов, имеющая вид: , либо .

В п. 3.1 обосновали решение задачи построения расписаний групповой обработки при наличии ограничений на интервал обработки группы с использованием аппарата теории игр. Обобщенная модель задачи многоуровневого программирования отражена в формулах (3.1). Процесс определения эффективных решений в иерархической системе с использованием модели (3.1) предполагает реализацию следующей процедуры:

1) на первом уровне иерархии определяется начальное решение , передаваемое на второй уровень;

2) на втором уровне определяется начальное решение  (для решения );

3) для полученного решения  на нижнем уровне формируется эффективное решение ;

4) с использованием решения вычисляется значение ;

5) выполняется переход к новом решению  (с определением для него эффективного решения ) и так до тех пор, пока для текущего решения не будет получено эффективное решение  (и соответствующее ему решение );

6) полученное эффективное решение используется при вычислении значения , характеризующего эффективность решения ;

7) для поиска эффективного решения выполняется переход к другому решению и определение для него эффективных решений (переход к шагу 2).

В результате степень эффективности решения  (значение ) определяется на основе эффективного решения , а степень эффективности решения  (значение ) определяется на основе эффективного решения . В соответствии с этими рассуждениями модель (3.1) может быть модифицирована следующим образом:

1) первый уровень:

 при 

(3.3)

2) второй уровень:

(3.4)

 при 

3) третий уровень:

(3.5)

 при 

Введенная в рассмотрение модифицированная модель принятия решений в иерархической системе (2.3)–(2.5) в принятых выше обозначениях имеет следующий вид:

1) первые уровень:

,

(3.6)

2) второй уровень:

(3.7)

,

3) третий уровень:

(3.8)

.

Эффективность групп партий на втором уровне системы оценивается на основе эффективных расписаний (третий уровень), т.е. эффективный состав группы партий данных должен обеспечивать расписание обработки партий, при котором неэффективное использование временных ресурсов приборов системы будет минимальным. Эффективность составов партий (первый уровень) оценивается на основе эффективных составов групп партий, т.е. эффективный состав партий должен обеспечивать такой состав групп партий, при котором общее количество обработанных данных будет максимальным либо количество необработанных данных будет минимизировано (для интервалов , ).

В п. 2.3 установили, что для определения эффективности каждого из уровней должны быть введены некоторые критерии.

На нижнем уровне иерархии критерий должен учитывать эффективность использования оборудования конвейерной системы при размещении рассматриваемой партии. Тогда значение критерия оптимизации определяет простой приборов при обработке текущего количества партий, находящихся в последовательности (но не всех партий из ).

Расчет критерия необходимо производит с использованием матриц  и , введёнными в п. 2.1, элементы матрицы  определяются следующим образом:

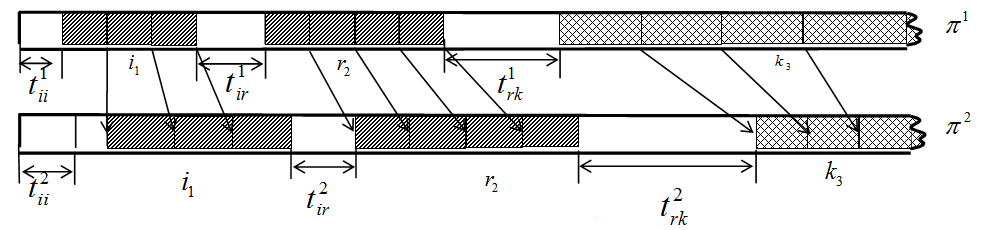
(3.9)

,

где **, – момент времени начала обработки первого экземпляра данных в партии, занимающей *j*–ю позицию в последовательности . Для расчета  используются интервалы времени переналадки приборов с обработки данных *i–*го типа на обработку данных *k–*го типа и интервалы первоначальной наладки на обработку данных *i–*го типа.

Выполним дальнейшие рассуждения для одной из групп (опуская индекс *z*). Вычисление значений  реализуется с использованием значений элементов матрицы , формализация выражений для определения выполнена в соответствии с рисунком 3.2.

а)



б)

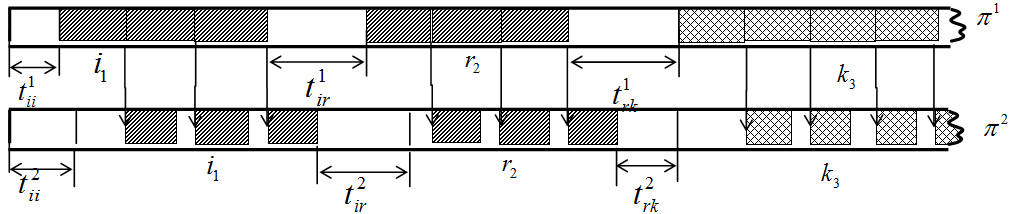


Рисунок 3.2 – Заданные виды последовательностей обработки партий, используемые при формировании выражений для 

Для первого прибора выражения для  формируются следующим образом:

1) если – время начала обработки первого экземпляра данных (*q*=1) в партии, занимающей первую позицию в последовательности , – время начала обработки партии данных *i*–го типа, занимающей в  первую позицию, тогда , где – время наладки первого прибора на обработку данных *i*–го типа;

2) выражение  определяет интервал первоначальной наладки первого прибора на обработку данных *i*–го типа, – время обработки –го экземпляра данных в этой партии, – вектор длительностей обработки данных разных типов на первом приборе (партии данных этих типов входят в группу ), тогда значение  начала обработки любого данного первой партии (в ), занимающего в ней *q*–ю позицию , определяется выражением вида:

;

(3.10)

3) если – время переналадки первого прибора с обработки данных *i*–го типа (в первой позиции партии в ) на обработку данных другого типа (партия во второй позиции в ), а  –время начала обработки первого экземпляра данных во второй партии на первом приборе (– начало обработки этой партии данных *i*–го типа в  ), тогда с учетом (3.10) значение  () определяются выражением вида:

(3.11)

,

где значение  определяется следующим образом:

, где .

(3.12)

Выражение  позволяет определить время начала обработки второй в  партии, выражение – длительность обработки предшествующих *q*–му данных в этой партии, тогда время начала обработки *q*–го данных *i*–го типа в партии, занимающей вторую позицию в , определяется следующим образом:

, .

(3.13)

По аналогии могут быть сформированы выражения для вычисления моментов времени начала обработки третьей, четвёртой партий и данных в них:

; (3.14)

; (3.15)

; (3.16)

. (3.17)

Время начала обработки партии данных *i–*го типа, занимающей *j–*ю позицию в  определяется выражением вида:

, (3.18)

время начала обработки *q*–го экземпляра данных:

. (3.19)

Выражение для  сформируем в соответствии с заданным порядком обработки партий (см. рисунки 3.2, 3.3) на основе выражений для определения начала обработки данных в них. Для *l=2* и *j=1* имеем:

, (3.20)

;…,, (3.21)

где – количество данных в первой партии.

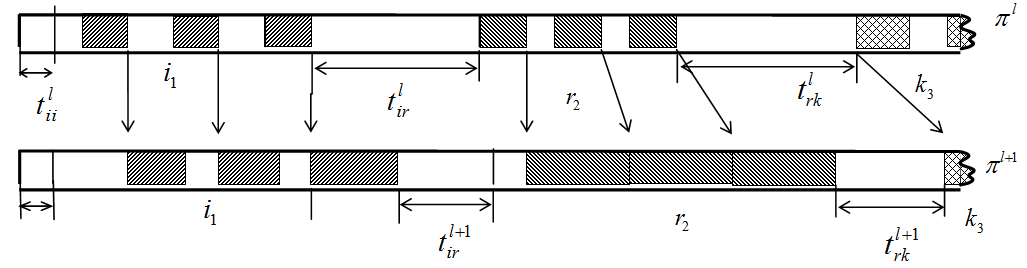


Рисунок 3.3 – Заданные виды последовательностей обработки партий, используемые при формировании выражений для .

Выражения для определения  (вторая партия в ) имеют вид:

; ; …;

. (3.22)

Выражения для  и  :

, (3.23)

, где . (3.24)

На их основе получены обобщенные выражения для определения (,) в виде:

, (3.25)

, (3.26)

. (3.27)

Формирование критерия оптимизации на третьем уровне выполнено в соответствии с заданными видами последовательностей обработки партий (см. рисунки 3.2 – 3.4), анализ которых позволил определить особенности идентификации интервалов времени простоя приборов:

1) малые длительности обработки данных *i*–го, *r*–го и *k*–го типов на приборе *l*=2 при малых длительностях его переналадки обуславливают ожидание им готовности партий к обработке и ожидание данных при их обработке внутри партии (см. рисунок 3.2 б), интервалы ожидания приборами данных при их обработке внутри партии являются одинаковыми, тогда порядок обработки партий будет определять:

a) длительность переналадок приборов с обработки данных одного типа на обработку данных другого типа; (должен быть отступ относительно первого уровня)

б) длительность простоя приборов в ожидании начала обработки партий;

2) при значительных значениях (рисунок 3.4 третий и четвёртый приборы, последовательности  и , интервалы ,  и ) наблюдаются различные интервалы времени простоя приборов (третьего и четвёртого) при обработке данных внутри партий, тогда различный порядок обработки партий определяет не только различные по величине интервалы переналадки и ожидания приборами начала обработки партий, но и различные по длительности интервалы простоя приборов при обработке данных внутри партии. Тогда критерий эффективности расписания обработки партий на нижнем уровне принятия решений учитывает:

а) время простоя приборов в ожидании начала обработки данных партий (с учетом интервалов наладки, переналадки и последующего ожидания);

б) время простоя приборов в ожидании готовности данных при их обработке внутри партии.

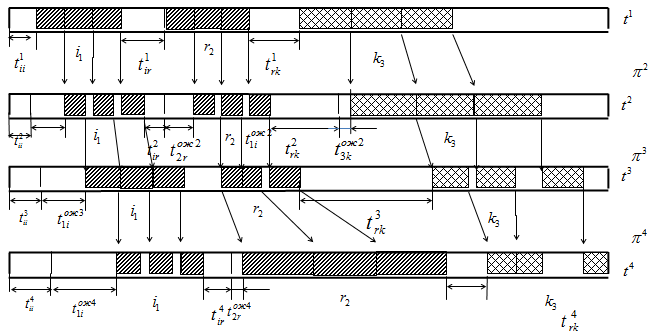


Рисунок 3.4 – Заданные виды последовательностей обработки партий, используемые при формировании критерия оптимизации на третьем уровне планирования

Простой *l*–го прибора в ожидании обработки первой в  партии равен значению , тогда суммарное время простоя всех приборов системы ** в ожидании начала обработки партий в последовательностях определяется выражением вида:. В интервал простоя *l*–го прибора в ожидании начала обработки партии после окончания обработки предыдущей партии входят: интервал переналадки прибора с обработки данных одного на обработку данных другого типа , возможный ненулевой интервал ожидания прибором начала обработки партии после окончания переналадки (Рисунок 3, где – время ожидания *l*–ым прибором начала обработки партии *i*–ых данных). Время простоя *l*–го прибора, соответствует сумме , его значение определяется выражением: , где *j* >1, – число данных в предшествующей в партии. Суммарный простой *l*–го прибора в ожидании начала обработки *j–*ых партий (**, где – общее число партий в последовательностях ) определяется следующим образом: . В этом случае суммарный простой всех *L* приборов в ожидании начала обработки партий на них определяется:

. (3.28)

Простой *l*–го прибора в ожидании готовности к обработке данных, занимающего *q*–ю позицию в *j–*ой партии в , определяется выражением вида: . Это выражение соответствует интервалу между двумя экземплярами данных (в *q*–ой и *(q–1)–*ой позициях) в *j–*ой партии в . Суммарный простой *l*–го прибора в ожидании готовности к обработке данных *j–*ой партии в вычисляется следующим образом :, где *q–* номер позиции данных в *j–*ой партии, – число данных в *j–*ой партии, , тогда общий простой *l*–го прибора при ожидании готовности к обработке данных внутри партий определяется выражением вида: , а суммарный простой всех приборов **в ожидании готовности данных внутри партий вычисляется по выражению:

. (3.29)

Критерий эффективности определения последовательности обработки партий сформирован на основе формул (2.28), (2.29) с использованием выражения  и введением индекса *z*, позволяющего идентифицировать качество расписания для конкретной группы партий , в виде:

 (3.30)

Критерий на втором уровне принятия решений (при определении состава групп партий ) характеризует общую эффективность использования ресурса времени приборов системы при реализации обработки каждой группы партий (критерий определяет суммарное время простоя приборов системы при обработке партий рассматриваемой группы). Так как на втором уровне определяется эффективный состав одной из групп партий, то формируемый критерий должен определять эффективность состава этой группы.

Суммарное время простоя приборов при обработке партий группы определяется:

1) суммой длительностей интервалов наладки приборов и возможного простоя приборов в ожидании начала обработки первого экземпляра данных в первой партии в () при **, вычисляемой для всех *L* приборов выражением вида ;

2) суммой длительностей: переналадки приборов с обработки данных одного на обработку данных другого типов, возможного простоя приборов в ожидании обработки первого в следующей в  партии, определяемой для *L* приборов выражением (3.28);

3) суммой длительностей интервалов времени простоя приборов в ожидании готовности данных при обработке партии внутри группы, определяемой для *L* приборов выражением (3.29);

4) суммой длительностей интервалов простоя *L* приборов после окончания обработки партий группы  в количестве  (т.е. сумма интервалов простоя приборов после окончания обработки партий группы на стадии «освобождения» конвейера).

Для определения последней из названных компонент суммарного времени простоя использованы обозначения:

* () – заданная длительность интервала времени, в течение которого реализуется обработка группы партий данных различных типов (все задаваемые интервалы являются одинаковыми);
* –количество партий, входящих в группу (индекс последней партии в группе);
* – количество данных, входящих в последнюю в  партию.

Тогда – момент времени начала обработки последнего экземпляра данных в партии с индексом  (последней партии в группе ), а время окончания обработки этой партии на *l–*ом приборе определяется выражением вида:

. (3.31)

В этом случае время простоя *l–*гоприбора после окончания обработки партий группы  определяется выражением вида:

, (3.32)

а суммарный простой всех приборов системы после окончания обработки группы партий вычисляется с использованием выражения вида:

 (3.33)

С учетом выражений (3.28)–(3.33) критерий эффективности принятия решений по составу групп партий на втором уровне иерархии системы примет следующий вид:

 (3.34)

Решение, формируемое для всех групп партий данных на втором уровне системы и передаваемое на первый уровень для вычисления значения критерия , имеет вид: {()}, где . Параметры и вектора , входящие в различные (*k*–е) наборы параметров, определяющих отдельную группу партий , обозначим соответственно и . Общее количество данных различных партий, входящих в группу определяется выражением вида: , где *u–* индекс соответствующего элемента в векторе , а общее количество данных, входящих во все группы {()}, обрабатываемые в системе, определяется выражением вида: . Общее количество данных, которое должно быть обработано в соответствии с программой выполнения работ, определяется выражением вида: , тогда критерий принятия эффективных решений на первом уровне системы, соответствующий количеству необработанных данных, имеет вид:

 . (3.35)

Обобщая введенные выражения для критериев, используемых при определении эффективных решений на соответствующих уровнях системы (3.30), (3.34), (3.35), представим оптимизационную модель многоуровневого программирования формирования партий, групп партий и расписаний обработки групп в следующем виде:

* – первый уровень иерархии (состав партий): ,

; (3.35)

* – второй уровень иерархии (определение состава групп партий): ,

 (3.36)

* третий уровень иерархии (определение порядка обработки партий для группы): ,

 (3.37)

* ограничение на третьем уровне иерархии для длительности реализации расписаний обработки партий группы :

***.*** (3.38)

# **4. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПОСТРОЕНИЯ РАСПИСАНИЙ ГРУПП ПАРТИЙ ДАННЫХ ПРИ НАЛИЧИИ ОГРАНИЧЕНИЯ НА ИНТЕРВАЛ ОБРАБОТКИ ГРУППЫ (ВТОРОЙ УРОВЕНЬ)**

Языком разработки программы для построения расписаний групповой обработки данных разных типов при наличии ограничения на интервал обработки группы выбран язык C#. C# объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.Язык C# является одним из самых популярных языков программирования, которые используются для разработки десктопного программного обеспечения. Он включает в себя конструкции, которые являются аналогами конструкций в псевдо-языках. Структура языка побуждает программиста использовать нисходящее проектирование, структурное программирование и пошаговую разработку модулей. Результатом такого подхода является надёжная и читаемая программа.

В данной работе используется парадигма объектно–ориентированного программирования. достоинствами, на которые опирались во время выбора стиля программирования, стали: функционального программирования являются:

Классы позволяют проводить конструирование из полезных компонент, обладающих простыми инструментами, что дает возможность абстрагироваться от деталей реализации.

Данные и операции вместе образуют определенную сущность и они не «размазываются» по всей программе, как это нередко бывает в случае процедурного программирования.

Локализация кода и данных улучшает наглядность и удобство сопровождения программного обеспечения.

Инкапсуляция информации защищает наиболее критичные данные от несанкционированного доступа.

### **4.1 Описание программы**

Программа написана с использованием парадигмы объектно–ориентированного подхода. Для каждого уровня иерархии системы разработаны классы, а в нихметоды формирования решения на каждом из уровней.

Основной является функция Button1Click(). В этой функции реализуется создание экземпляра класса первого уровня:

* GetCriterion(List<List<int>> inMatrix): в функцию передаётся структура содержащая количество типов и данных каждого из них. Функция считывает эти данные и возвращает начальное решение для первого уровня .
* GenerateSolution() в функцию передаётся структура содержащая количество типов и данных каждого из них. Функция считывает эти данные и возвращает опираясь на критерий первого уровня *f1* оптимальное решение первого уровня
* GetCriterion(List<List<int>> inMatrix) Функция получает на вход сформированный состав партий данных а возвращает оценку критерия по которой и будет строится оптимальное расписание.
* public void Algoritm\_1(): Функция формирует начальный состав группы партий опираясь на временное ограничение длительности обработки
* public void Algoritm\_2(): Функция формирует оптимальный состав группы партий опираясь на временное ограничение длительности обработки
* private List<List<int>> BuildR(List<List<int>> N) Функция формирует матрицу R на промежуточных этапах формирована оптимального решения принимая в качестве аргумента неоптимальную группу патрий Тексты разработанных модулей представлены в приложении.

### **4.1 Описание алгоритмов функционирования второго уровня** В качестве начальных данных алгоритм принимает на вход партии данных, которые прежде формировались на верхнем уровне.

Формируется начальное решение, при котором последовательно все партии распределяются по группам, внутри каждой группы составляется расписание (на нижнем уровне), запоминается самое оптимальное.

Далее проходит оптимизация сформированных составов групп, для этого сперва между самими группами происходит обмен данных, с выбором наиболее худшего по простою оборудования, а затем наиболее худшего по простою оборудования. После этого по тем же критериям обмениваются партии между группами и незавершенным производством. На каждом этапе мы текущее состояние сравниваем с оптимальным составом групп, сравнение проходит по критерию суммарного общего простоя оборудования. Общий простой оборудования расчитывается как сумма простоев при первоначальной настройке, переналадке оборудования, простоя оборудования при ожидании данных, которые ещё не выполнились на предыдущих приборах, и простое оборудования при завершении обработки всех данных.  
При каждом новом составе групп партий, каждая группа опускается на нижний уровень, где составляется оптимальное расписание для этой группы, также проверяется влезет ли группа в заданный промежуток времени.



Рисунок 4.1 Алгоритм формирования начального решения на втором уровне



Рисунок 4.2 Алгоритм формирования оптимального решения на втором уровне

# **5 АНАЛИЗ РЕЗУЛЬТАТОВ РАБОТЫ РАЗРАБОТАННЫХ МЕТОДОВ ПОСТРОЕНИЯ ГРУППОВОЙ ОБРАБОТКИ ПАРТИЙ ДАННЫХ**

Проведем анализ работы системы и докажет необходимость метода оптимизации состава партий данных на первом уровне системы. Сравнение необходимо проводить с аналогично построенной системой без модуля оптимизации составов партий данных на первом уровне.

Для сравнения возьмем фиксированные партии – используем начальное решение по составам партий данных всех типов. Фиксированные партии не будут модифицироваться, а сразу отправляться на второй уровень представленной модели (из фиксированных партий данных будут формироваться группы партий данных, проводится оптимизация расписания и проверка построенного расписания на временное ограничение )

В работе модуля оптимизации начальное решение улучшается (увеличивается окрестность решения на каждом последующем шаге алгоритма) при использовании метода оптимизации партий данных, описанного выше.

Проанализируем полученные данные.

Таблица 5.1 – зависимость количества обработанных требований от отношения времени обработки и отношения времени переналадки партий требований.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Количество обработанных партий с методом оптимизации партий данных | Количество обработанных партий без метода оптимизации партий данных |
| 1 | 1 | 64 | 64 |
| 1 | 2 | 64 | 64 |
| 1 | 4 | 64 | 64 |
| 1 | 8 | 64 | 64 |
| 1 | 10 | 64 | 64 |
| 2 | 1 | 64 | 64 |
| 2 | 2 | 64 | 64 |
| 2 | 4 | 64 | 64 |
| 2 | 8 | 64 | 50 |
| 2 | 10 | 64 | 50 |
| 4 | 1 | 56 | 8 |

Продолжение таблицы 5.1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Количество обработанных партий с методом оптимизации партий данных | Количество обработанных партий без метода оптимизации партий данных |
| 4 | 2 | 50 | 8 |
| 4 | 4 | 48 | 8 |
| 4 | 8 | 38 | 8 |
| 4 | 10 | 36 | 8 |
| 8 | 1 | 36 | 8 |
| 8 | 2 | 22 | 8 |
| 8 | 4 | 22 | 8 |
| 8 | 8 | 8 | 8 |
| 8 | 10 | 8 | 8 |
| 10 | 1 | 8 | 8 |
| 10 | 2 | 8 | 8 |
| 10 | 4 | 8 | 8 |
| 10 | 8 | 4 | 8 |
| 10 | 10 | 0 | 8 |

На основании полученных данных можно смело заявить, что при использовании методов оптимизации на первом уровне системы мы получаем лучшее решение, чем решение, основанное на фиксированных партиях. Алгоритм оптимизации получает оптимальное решение по составу групп партий данных при ограничении на время обработки .В данном разделе были получены наглядные результаты работы системы, а так же проведен сравнительный анализ и доказательство необходимости использования методов оптимизации решений на первом уровне по отношению к фиксированным партиям данных. А также на втором уровне по отношению к начальному построению групп партий данных

# **5 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТИРОВАНИЯ ПРОГРАМНОГО ПРОДУКТА**

При разработке программного продукта были проведены маркетинговые исследования.

### **6.1 Маркетинговые исследования программного продукта**

### **6.1.1 Сущность и этапы маркетинговых исследований**

Маркетинговое – систематический сбор, учет и анализ данных по маркетингу и маркетинговым проблемам в целях совершенствования качества процедур принятия решений и контроля в маркетинговой среде[2].Этапы маркетингового исследования приведены в приложении. (см. Приложение А)

### **6.1.2 Конкурентоспособность программного продукта**

Текущие системы построения расписаний имеют большее количество ограничений при работе, а также затрачивают большее время на работу.

Программный продукт основан на новом алгоритме построения рас писаний и является программным подтверждением эффективности разработанного нового алгоритма построения рас писаний, который и является отличием программного продукта, а также конкурентным преимуществом.

### **6.1.3 Оценка рыночной направленности**

Для оценки рыночной направленности продукта проведем исследование ПП по таблице A.1 (см. Приложение А). Найдем среднее арифметическое полученных оценок:



При таком значении полученного среднего балла разрабатываемый программный продукт обладает рыночной направленностью.

### **6.1.4 Сегментация рынка**

Сегментация рынка — процесс разбивки потребителей или потенциальных потребителей на рынке на различные группы (или сегменты), в рамках которых потребители имеют схожие или аналогичные запросы, удовлетворяемые определенным комплексом маркетинга. Сегментация рынка может быть произведена по нескольким признакам:

1. по географическому признаку включает разделение рынка на различные географические единицы (переменные);
2. по демографическому признаку заключается в разделении рынка в соответствии с такими переменными как: возраст, пол, размер семьи;
3. по демографическому признаку;
4. по поведенческому признаку: заключается в выделении групп покупателей на основе их знаний, квалификаций как пользователей и их реакций на товар[2].

### **6.1.5 Предпочтительный потребитель программного продукта**

Разработанная система подходит для конвейерного производства различных деталей на одной линии производства. Список всех подходящих предприятий представлен в таблице 1.2.(см. Приложение А)

### **6.1.6 Жизненный цикл программного продукта**

Программный продукт является товаром. Товар, или программный продукт, характеризуется жизненным циклом.

Жизненный цикл программного продукта – период времени, в течение которого товар обращается на рынке, начиная с момента зарождения идеи и начала его разработки и заканчивая его уходом с рынка[2].

Жизненный цикл программный продукт включает в себя этапы:

1. Разработка, капитальные вложения.

2. Выведение на рынок. Небольшой рост объёмов продаж и соответственно прибыль минимальна или её вообще нет.

3. Рост. Период быстрого роста объёма продаж. Прибыль также возрастает по мере увеличения объёма продаж.

4. Зрелость. Объёмы продаж значительны, но дальнейшего роста продаж не наблюдается. Прибыль на данном этапе стабилизировалась, так как дополнительных затрат для вывода товара на рынок не требуется.

5. Упадок. Происходит значительное снижение объёмов продаж вплоть до полного падения спроса на данный товар. Прибыль резко снижается до нуля. Жизненный цикл программного продукта приведен в приложении (см. Приложение А).

**4.1.7 Разработка и реализация уровней программного продукта**

Создание нового продукта связано с возникновением инновационной идеи, то есть освоение нового рынка, возникновением или расширением потребностей потребителя, появлением новых функций продукта.

Уровни создания и реализации программного продукта приведены в приложении (см. Приложение А).

### **6.2 Определение затрат на проектирование**

### **6.2.1 Расчет трудоемкости разработки программного продукта**

Трудоемкость проекта определяется, исходя из данных об используемых функциях программного продукта (см. Приложение А).

Общий объем разрабатываемого программного продукта (V0) определяется в тысячах условных машинных команд по формуле:

где ** – объем i – ой функции программного продукта, тыс. УМК;

*n* – общее число функций программного продукта.

*Kcп=1+0,06 = 1,06*

*T0=Tp×Kсп, где T0*–общая трудоемкость.

*T0=250,2×1,06=265,212*

*Кн –* Коэффициент новизны;

*Кт –* Коэффициент учета стандартных типов программ.

Трудоемкость считается путем суммирования затрат труда на всех стадиях разработки программного продукта.

;

Таблица 4.1 Расчет трудоемкости

|  |  |  |  |
| --- | --- | --- | --- |
| Название стадии | Уд вес стадии | Формула и расчеты | Значение |
| трудоемкость стадии ПЗ | 0.09 |  |  |
| трудоемкость стадии ЭП | 0.07 |  |  |
| трудоемкость стадии ТП | 0.07 |  |  |
| трудоемкость стадии РП | 0.61 |  |  |
| трудоемкость стадии ВН | 0.16 |  |  |

Исходя из полученной трудоемкости и численности исполнителей, можем рассчитать срок разработки программного продукта:

где Ф – среднее количество дней в месяце, равное 20,56 дней, Ч – численность разработчиков программного продукта, 1 – количество часов работы в день.

Рассчитаем эффективный фонд времени (Fном) – разница между номинальным фондом времени и потерями рабочего времени (П). .

Для рабочих составляющих:

Для ЭВМ:

### **6.2.2 Расчет эксплуатационных затрат разработчика**

Для того, чтобы определить сумму годовых эксплуатационных затрат необходимо выполнить следующие расчеты. Данные для расчета годовых эксплуатационных затрат приведены в приложении (см. Приложение А)

Таблица 4.2 – Смета годовых эксплуатационных затрат

|  |  |  |
| --- | --- | --- |
| Наименование затрат | Формула расчета | Значение, руб. |
| Расчет материальных затрат | Зм = С \* Кнз / 100  Зм =8\*27000/100 | 2160 |
| Расчет затрат на электроэнергию | Зэ = Fном \* Цэ \* W \* Кн \* Ксм  Зэ =0,35\*0,70\*2,53\*1974,00\*1 | 1223,5839 |
| Расчет оплаты труда | ФОТ = срок разработки \* Ксм \* оклад ,  ФОТ = 11,8\*1\*13000 | 1223,5839 |
| Расчет отчислений от заработной платы | Отч = ФОТ \* Кн / 100 ,  Отч =153400,00\*0,342 | 52462,80 |
| Расчет затрат на ремонт | Зр = С \* Кр / 100 ,  Зр = 27000\*0,06 | 1620,00 |
| Расчет накладных расходов | Зн = (Зм + Зэ + ФОТ + Отч + Зр) \* Кнр / 100  Зн = 2160+1223,5839+1223,5839+52462,80+16,2 | 42173,28 |
| Расчет амортизационных отчислений | Аэвм=норма амортизации \* основные средства  Аэвм=100/24 | 4,17 |
| Всего |  | 266314,66 |

Рассчитаем себестоимость часа машинного времени(Сч.м.в.) по формуле:

где – сумма годовых эксплуатационных затрат, руб.

*Сч.м.в=266314,66/1904,91\*1)= 139,80 руб*

### **6.2.3 Расчет сметы затрат на проектирование**

Стоимость всех работ, выполняемых при разработке програмного продукта, можно разделить на две части: стоимость работ по разработке и отладке программного обеспечения, выполняемых с помощью вычислительной техники (Срм = 0,75 мес.); стоимость работ, производимых без применения вычислительной техники (Срр= 0,5 мес.)

Таблица 5 – Смета затрат на материалы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Материалы | Единица | Требуемое количество | Цена за 1 единицу. | Значение, руб. |
| USB Flash–память | Руб | 1 | 700 | 700 |
| Бумага офисная | Пачка | 1 | 180 | 180 |
| Всего |  |  |  | 880 |

Таблица 6 – Смета затрат на проектирование

|  |  |  |
| --- | --- | --- |
| Наименование расходов | Формула расчетов | Значение, руб. |
| Основная заработная плата разработчика | ЗПосн =Ч\*Срр\*Оклад  ЗПосн =1\*27000\*0.76 | 9880,00 |
| Дополнительная заработная плата разработчика | ЗПдоп= ЗПосн\*0,3  ЗПдоп=9880,00\*0,3 | 2964,00 |
| Фонд оплаты труда | ФОТ = Зп.осн + Зп.осн\*Кдоп.з/п  ФОТ = 9880,00+(9880,00+0,3) | 12844,00 |
| Отчислениями | Отчисление= ФОТ \*0,342 | 4392,65 |
| Фонд оплаты труда разработчика | ФОТотч = ФОТ + Отчисление  ФОТотч =12844,00+4392,65 | 17236,65 |
| Накладные расходы | Знакл = Кнр \* ФОТотч/100  Знакл = 20/100\*17236,65 | 3447,33 |
| Затраты на разработку ПП с применением ВТ | Зпо = Срм\* Fэф \* Сч.м.в  Зпо = 1\*168\*139 | 2822,4 |
| Затраты на материалы |  | 880руб |
| Всего |  | 45051,10 |

Смета затрат является основой для формирования цены программного продукта

### **6.3 Расчет капитальных затрат**

Капитальные вложения для разработчика – расходы на покупку (Цтс), доставку (Зтр) и монтаж(Зм) технических средств, а также приобретение программного обеспечения(Цоб), необходимого для процесса создания программного продукта:

где Цтс – расходы на покупку, цена = 27000,00рублей;

Зтр – затраты на транспорт 5% от прейскурантной цены;

Зтр = 27000\*5/100= 1350,00 рублей;

Зм – затраты на монтаж 8% от прейскурантной цены;

Зтр = 27000\*8/100 = 2160,00 рублей;

Цоб – затраты на приобретение программного обеспечения (лицензия) составляют 2500 рублей;

Кр = 27000 + 1350 + 2160 + 2500 = 33010 рублей.

### **Формирование цены предложения разработчика**

Формирование цены разработчика производится методом безубыточности. Расчет цены производится на основе анализа безубыточности и обеспечения целевой прибыли. Этот метод основан на графике безубыточности.

Постоянные издержки (Ипост) – независимо от объема продукции или реализации программного продукта. Расчеты приведены в приложении.

(см. Приложение А)

где N – количество компаний, интересующихся программным продуктом TT.

При анализе рынка было выявлено, что всего 10 организаций, которых интересует программного продукта. В Приложении А перечислены эти предприятия. Посчитаем стоимость одного продукта:

6020,96

Из графика можно сказать, что при цене 6020,96руб. минимальный объем продажи составил 6.

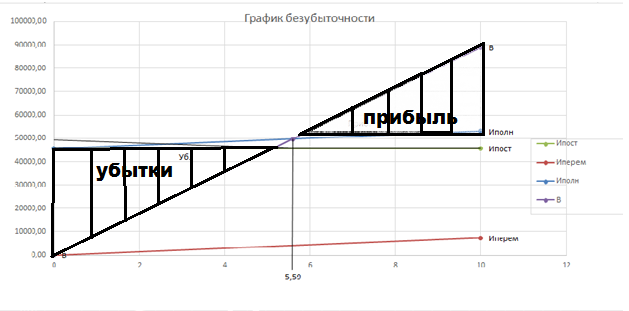


Рисунок 5 – График безубыточности

Т.Б. – точка безубыточности, которая показывает, что объем минимальных продаж составляет 6, а максимальных 10 программного продукта.

### **6.5 Оценка экономической эффективности проектирования программного продукта**

Задачей экономической оценки является определение динамики чистой текущей стоимости, т.е. суммы, ежегодно возвращающейся в виде отдачи от вложенных средств. Расчеты приведены в приложении А(см. Приложение А).

Результаты расчета сводятся в таблицу 7:

Таблица 7 – Расчет интегрального экономического эффекта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Показатели | Год | | | | |
| 2016,00 | 2017,00 | 2018,00 | 2019,00 | 2020,00 |
| Число реализаций | 0,00 | 10,00 | 9,00 | 8,00 | 7,00 |
| Объем реализации Pt, руб. | 0,00 | 60209,65 | 54188,68 | 48167,72 | 42146,75 |
| Капитальные вложения Kt, руб. | 33010,00 | 0,00 | 0,00 | 0,00 | 0,00 |
| Годовые издержки Иt, руб. | 0,00 | 52356,22 | 31413,73 | 20942,49 | 26178,11 |
| ЧДПt, руб. | –33010,00 | 7853,43 | 22774,95 | 27225,23 | 15968,65 |
| Коэф. приведения по фактору времени | 1,00 | 0,80 | 0,71 | 0,64 | 0,57 |
| ДЧДПt, руб. | –33010,00 | 6260,71 | 16210,76 | 17302,13 | 9061,04 |
| ЕИt, руб. | –33010,00 | 6260,71 | 22471,47 | 39773,60 | 48834,64 |
| Амортизация At, руб. | 0,00 | 6602,00 | 6602,00 | 6602,00 | 6602,00 |
| Прибыль Прt, руб. | 0,00 | 1251,43 | 16172,95 | 20623,23 | 9366,65 |
| Рентабельность Pt,% | 0,00 | 3,79 | 48,99 | 62,48 | 28,38 |

Рисунок 6 – График финансового профиля проекта

В таблице 8 приведены все показатели, полученные при анализе графика финансового профиля проекта.

**Таблица 8 – Показатели эффективности проекта**

|  |  |  |
| --- | --- | --- |
| Показатели | Единица измерения | Величина |
| Прибыль проекта | руб. | 27872 |
| Интегральный экономический эффект | руб. | 49663,69 |
| Рентабельность проекта | % | 84,44 |
| Срок окупаемости | Года | 0,2 |
| Период возврата капиталовложений | Года | 1,2 |

# **БЕЗОПАСНОСТЬ ЖИЗНИДЕЯТЕЛЬНОСТИ**

## **7.1 Основные сведения**

Охрана труда является важной составляющей деятельности любого предприятия. Она направлена на обеспечение эффективного, безопасного и комфортного труда работника.

В нашем расчетно-графическом задании у нас есть производственное помещение со следующими характеристиками:

– ориентация натужной стены здания: ЮВ;

– площадь остекления, м: 4,5;

– средства солнцезащиты: светлые жалюзи;

– тип и мощность ламп системы освещения, кВт: люминесцентные,0,06;

– количество ламп системы освещения: 20;

– число рабочих мест: 8.

Задачей нашего расчетно–графического задания является анализ условий труда в заданном помещении по следующим критериям:

– планировка оборудования и мест;

– тяжесть и напряженность труда;

– качество воздуха и микроклимат рабочей зоны;

– шум и вибрация;

– освещение;

– электро– и пожаробезопасность.

**7.2 Краткая характеристика помещения и выполняемых работ**

Рабочее помещение имеет размеры 7 × 7 × 3.5 м. Число работников данного помещения составляет 8 человек.

Площадь помещения 49 м2, объем помещения 171.5 м3. На одного работника приходится примерно 6 м2 по площади и 21.5 м3 по объему.

В помещении расположено 8 рабочих мест, шкаф, две кадки с растениями, мультимедийная доска и огнетушитель. Рабочие места, за исключением места начальника, отделены друг от друга перегородками высотой 1,5 м в связи тем, что работа требует умственного напряжения и концентрации внимания.

На каждом рабочем месте расположено по комплекту вычислительной техники, канцелярские принадлежности. На рабочем месте начальника отдела расположен также лазерный принтер.

В шкафу располагается документация отдела, в том числе по охране труда и пожарной безопасности, также имеется аптечка, включающая следующие средства:

– обезболивающие и противовоспалительные средства: диклофенак, пакет–контейнер портативный гипотермический (охлаждающий), капли глазные «Искусственная слеза»;

– средства для обработки кровотечения, перевязки и обработки ран: жгут, индивидуальный стерильный перевязочный пакет, бинт стерильный, бинт нестерильный, лейкопластырь, медицинские перчатки, перекись водорода, йод, зеленка, вата;

– средства при болях в сердце: нитроглицерин, валидол;

– средства для сердечно–легочной реанимации и клинической смерти: устройство для проведения искусственного дыхания;

– средства при обмороке: нашатырный спирт;

– средства при аллергиях и отравлениях: активированный уголь, супрастин, тавегил;

– средства при стрессовых реакциях: корвалол или настойка валерианы, каптопресс;

– ножницы с тупыми концами;

– блокнот и ручка.

Комплектность аптечки постоянно контролируется, лекарственные и другие средства постоянно пополняются или заменяются по мере истечения срока годности.

Ориентация натужной стены здания является юго–восточной, площадь остекления составляет 4,5 м2. Тип и мощность ламп системы освещения –люминесцентные мощностью 0,06 кВт, количество ламп системы освещения составляет 20 штук.

Виды выполняемых работ на ПЭВМ:

– программирование программных продуктов под заказ;

– осуществление модернизации и поддержки ПО.

**7.3 Планировка и размещение оборудования и рабочих мест**

На рисунке B.1 представлена схема помещения, на которой показано размещение рабочих мест и оборудования с указанием размеров, дверных и оконных проемов, проходов, офисной мебели и средств пожаротушения.

(См. Приложение B)

Планировка соответствует нормативным требованиям СанПиН 2.2.2/2.4.1340–03 «Гигиенические требования к персональным электронно–вычислительным машинам и организация работы», за исключением того, что рабочие места не удалены на расстояние 1 метра от стен. Ко всем рабочим местам обеспечивается свободный проход. Используются ЖК мониторы при работе с ПЭВМ. Площадь на каждое рабочее место в пределах нормы.

**5.4 Тяжесть и напряженность труда. Режим труда и отдыха**

Далее приведена оценка тяжести и напряженности работы. Данные сведены в таблицу B.1 (См. Приложение B)

Таким образом, класс условий труда – 2.

В работе предусмотрены технологические перерывы по 10 минут каждый час, что суммарно составляет 80–90 минут за смену. Данное значение соответствует нормам для II группы работы с ПЭВМ при 8–часовой смене.

В течении технологического перерыва выполняются упражнения для успокоения и снятия напряженности у сотрудников.

**5.5 Качество воздуха и микроклимат рабочей зоны**

Работа программиста относится к категории легких работ. Нормами микроклимата по данной приведены в таблице 2.

Таблица 2 – Нормы микроклимата для помещений с ПЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период года | Категории работ | Температура воздуха, ˚С | Относительная влажность воздуха, % | Скорость движения воздуха, м/с |
| Холодный | Легкая 1а | 22–24 | 40–60 | 0,1 |
| Легкая 1б | 21–23 | 40–60 | 0,1 |
| Теплый | Легкая 1а | 23–25 | 40–60 | 0,1 |
| Легкая 1б | 22–24 | 40–60 | 0,2 |

В нашем случае работа относиться к категории Ia. В течении года должна обеспечиваться температура 22–25 ˚С. Для холодного периода в помещении есть система отопления, обеспечивающая требуемый температурный режим. Для теплого периода наблюдаются превышения допустимого температурного режима, в связи с чем рекомендуется установить систему кондиционирования воздуха, расчет и выбор кондиционера рассмотрен в пункте 10.

Контакт со специфичными при работе с ПЭВМ веществами (тонер, озон и т.д.) не представляет опасности, т.к. имеется вентиляция. Источники выделения вредных веществ отсутствуют.

Далее приведена информация о составе воздуха на рабочих местах, сведенная в таблицу 3.

Таблица 3 – Состав воздуха на рабочих местах

|  |  |  |
| --- | --- | --- |
| Вещество | ПДК | Измеренное значение |
| Оксиды азота | 5 мг/м3 | 3,4 мг/м3 |
| Пыль | 4 мг/м3 | 3,6 мг/м3 |
| Озон | 0,1 мг/м3 | 0,05 мг/м3 |

Качество воздуха находится в пределах норм. Вещества, состоящие в воздухе, не превышают допустимых норм ПДК.

**5.6 Шум и вибрация**

Для оценки шума используют частотный спектр измеряемого уровня звукового давления, выраженного в дБ, который сравнивают с предельным спектром, приведены в таблице 4.

Таблица 4 – Допустимые уровни звукового давления и уровня звука

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Рабочие места | Уровни звукового давления, дБ, в октавных  Полосах со среднегеометрич. Частотами, Гц | | | | | | | | Уровни звука, дБ |
|  | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 |  |
| Рабочее место программиста | 71 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |

Уровень шумов от ЭВМ и другого оборудования, используемого при разработке, незначительный. В данном случае его уровень определяется только хозяйственной деятельностью человека и составляет 40 дБ, что соответствует нормам.

Меры по снижению уровня шума и вибраций:

– использование современных моделей офисной техники, а также оборудования для вентиляции воздуха, которые даже при одновременной работе не создают высоких уровней шума и вибрации;

– своевременное проведение профилактических и ремонтных работ оборудования;

– соблюдение правил эксплуатации оборудования;

– рациональный график работы оборудования;

– размещение МФУ во вспомогательных помещениях;

– облицовка стен и потолка звукоизолирующими материалами;

– экранирование источников шума и рабочих мест.

**5.7 Освещение**

Естественное освещение имеет доступ в помещение через три окна расположенных на одной из стен. Общая площадь оконных проемов 4,5 м2. Окна имеют ориентацию на юго–восток. На них установлены светлые жалюзи в качестве солнцезащиты.

Возле окон распложено 4 рабочих места. Так как имеются средства солнцезащиты, то при попадании солнечного света в окна работники не имеют негативного воздействия.

В помещении установлено 20 потолочных люминесцентных ламп. Линии источников света (ламп) располагаются над рабочими местами (См. Приложение B).

**5.8 Электро– и пожаробезопасность**

Помещение по опасности поражения электрическим током отнесем к категории без повышенной опасности, т.к. данное помещение характеризуется отсутствием условий, которые создают особую или повышенную опасность.

Помещение по категории пожарной опасности относятся к B1–B4 (пожароопасная).

В таблице 6 сведены данные о возможных классах пожаров:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс пожара | Характеристика класса |  | Подкласс пожара | Характеристика класса |
| А | Горение твердых веществ |  | А1 | Горение твердых веществ, сопровождаемое тлением (древесина, бумага) |
| E | Горение объектов, находящихся под напряжением |  | – | Горение установок и оборудования, находящихся под электрическим напряжением |

Таблица 6 – Классы пожаров

Для обеспечения пожарной безопасности помещения необходим один порошковый огнетушитель с вместимостью огнетушащего вещества 10/9 кг и покрывающим 200 м2.

Организационно–технические мероприятия по обеспечению электро– и пожаробезопасности:

– проведение инструктажей, наличие инструкций по ОТ, а также плана эвакуации при пожаре;

– наличие в помещении средств пожаротушения

Таблица 7 – Средства пожаротушения и противопожарного инвентаря( Порошковые огнетушители)

Требование к организаци рабочих мет офиса

– соблюдение правил пользования ПЭВМ.

Данное помещение полностью соответствует правилам и нормам охраны труда по электро– и пожаробезопасности.

**5.9 Статическое электричество и излучения**

Компьютер и вспомогательное электрооборудование формирует сложную электромагнитную обстановку на рабочем месте.

Установлено, что ЭМП негативно влияют на центральную нервную систему, вызывая головные боли, головокружения, тошноту, депрессию, бессонницу, отсутствие аппетита, возникновение синдрома стресса.

Поэтому проанализируем способы защиты от статического электричества и излучений в помещениях с ПЭВМ.

К основным мерам профилактики неблагоприятного влияния электромагнитного поля относятся:

– своевременное прохождение периодических медосмотров;

– отключение оборудования, на котором временно не работают, но находятся рядом;

– использование мониторов, соответствующих современным требованиям по защите от излучений, предпочтение следует отдавать жидкокристаллическим мониторам;

– выполнение режима труда и отдыха, не превышение общего времени взаимодействия с ПЭВМ за смену.

Для снижения влияние электростатического поля необходимо:

– устанавливать нейтрализаторы статического электричества;

– поддерживать в помещении относительную влажность не ниже 45–50% и ежедневно проводить влажную уборку;

–для снятия заряда несколько раз в день мыть руки и лицо водой;

–протирать экран и рабочее место специальной антистатической салфеткой;

–ограничить количество полимерных материалов в помещении.

**5.10 Эргономика и техническая эстетика**

Проанализировав рабочие места в соответствии с ГОСТ Р ИСО 9241–5–2009 “Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов”, была заполнена таблица 8 по параметрам рабочей поверхности и приведен эскиз рабочего места и рабочей поверхности (См. Приложение B)

Рациональное цветовое оформление помещения направленно на улучшение санитарно–гигиенических условий труда, повышение его производительности и безопасности. Окраска помещений ВЦ влияет на нервную систему человека, его настроение и в конечном счете на производительность труда. Основные производственные помещения целесообразно окрашивать в соответствии с цветом технических средств. Учитывая ориентацию окон помещения можно сделать вывод, что лучше всего подойдет цвет красно–оранжевый для пола и бирюзовый для стен (См. Приложение B).

**5.11 Расчет необходимой производственной мощности системы   
кондиционирования воздуха**

*Поступление тепла через световые приему Qo,кВт, определятся по формуле:*

где *q'*– количество тепла, поступающее в помещение через одинарное остекление световых проемов, облучаемых прямой радиацией, кВт/м2;

*q''* –тоже при наличии искусственного затенения,кВт/м2;

*Fo'* –площадь световых проемов,облучаемая прямой радиацией,м2;

*Fo ''*–тоже при наличии искусственного затенения,м2;

*с* –коэффициент солнцезащиты;

*tн* –расчетная температура наружного воздуха,0С;

*tв* –расчетная температура внутреннего воздуха,0С;

*Ro* –сопротивление теплопередаче заполнения светового проема,м С/кВт;

*Fo* –общая площадь остекления,м2.

Теплопоступление через ограждение Qорг.,кВт, определяется по формуле:

где *qогрi* – поступление тепла через 1 м2 i–й ограждающей поверхности,кВт/м2;

*Sогр i* –площадьi–й ограждающей поверхности,м2.

для наружной стены

для потолка и пола

для оставшихся 3х стенок

Теплопоступление от оборудования и системы искусственного освещения Qэ,кВт, оценивается по формуле:

где Ei – коэффициент, учитывающий часть энергии, выделяющуюся в виде тепла для i–го типа оборудования;

Ni – установленная мощность единицы i–го оборудования, кВт;

ni – число единиц i–го оборудования;

hi – коэффициент одновременности работы i–го оборудовании

Таблица 8 – Характеристика офисной техники и источников света

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип оборудования | N, кВт | E | h | n | Qэ |
| Комплект вычислительной техники | 0,4 | 0,3 | 0,95 | 8 | 0,912 |
| Лазерный принтер | 0,4 | 0,3 | 0,2 | 1 | 0,24 |
| Лампа люминесцентная | 0,06 | 0,5 | 1 | 20 | 0,6 |

Теплопоступление от находящихся в помещение людей Qл,кВт, определяется по формуле:

где *q* – тепло, выделяемое одним человеком, кВт;

*n* –количество работников,находящихся в помещении.

Результирующее поступление тепла в помещение *Qрез*, кВт, определяется по формуле:

По полученным данным выбираем кондиционер. В нашем случае целесообразно использование кондиционер LG LP–K3061ZA (GOLD).

Таблица 9 – Характеристика выбранного кондиционера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель кондиционера | Потребляемая,кВт | | Производительность,кВт | |
| Охлажд. | Нагрев | Охлажд. | Нагрев |
| LG LP–K3061ZA (GOLD) | 2,75 | 2,45(+2,0 | 8,15 | 8,15 (+2,0) |

**Выводы по разделу 5**

В данной работы был выполнен анализ помещения на удовлетворение нормам по различным показателям:

– планировка и размещение оборудования и рабочих мест;

– тяжесть и напряженность труда;

– качество воздуха и микроклимат рабочей зоны;

– шум и вибрация;

– освещение;

– электро– и пожаробезопасность;

– статическое электричество и излучение;

– эргономика и техническая эстетика.

В ходе анализа было выявлено несколько недочетов:

– планировка рабочих мест не удовлетворяет требованию о расстоянии от стен не менее метра;

– в теплый период года наблюдается нарушение температурного режима в помещении.

В качестве рекомендаций для устранения последнего был выполнен расчет необходимой производительности системы кондиционирования воздуха и подобран подходящий для данного помещения кондиционер.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Хьюз, Т.Хьюз.– М.: Изд–во «Вильямс», 2004.– 672 с.
2. Mendez C. A. State–of–the–art review of optimization methods for short–term scheduling of batch processes./ C. A. Mendez, J. Cerda, I. E. Grossmann, L. Harjunkoski, M. Fahl. // Computers and Chemical Engineering, 2006, #30.– P. 913–946
3. Ковалев М.М. Модели и методы календарного планирования. Курс лекций./ М.М. Ковалев. – Минск: Изд–во БГУ, 2004.– 63 с.
4. Tan Y. Comparative Study of Different Approaches to Solve Batch Process Sheduling and Optimisation Problems./ Tan Y., Huangi W., Sun Y., Yue Y.// Proceedings of the 18th International Conference on Automation & Computing, Loughborough University, Leicestershire, UK, 8 September 2012. – P. 424–444;
5. Adonyi R. Incorporating heat integration in batch process scheduling./ R. Adonyi, J. Romero, L. Puigjaner, F. Friedler.// Applied Thermal Engineering, 2003, # 23.– P. 1743–1762;
6. Kreipl S. Planning and Scheduling in supply chains: An Overview of Issues in Practice./ S. Kreipl, M. Pinedo.// Production and Operations Management, 2004, vol.17, #1. – P. 77–92;
7. Steiner G. Minimizing the weight number of late jobs with Batch setup times and delivery costs on a single machine./ G. Steiner, Zhang R.// Mulltiprocessor scheduling: Theory and Applications, Book edited by E. Levner. – Vienna, Austria: Itech Education and Published, 2007. – 436 p.;
8. Koehler F. Optimal Batch Schedules for Parrallel Machines./ F. Koehler, S. Khuller // Algorithms and Data Structures: 13th International Symposium.– Berlin: Springer – Verlag, 2013. – P. 475 – 486;
9. Yugma C. Batching and Scheduling Algoritm for the diffusion Area in Semicondactor Manufacturing./ C. Yugma, S. Davsere–Perez, C. Artiques, O. Sibille.// International Journal of Production Research, Taylor&Francis, 2012, #8.– P.2118–2132;
10. Chandra P. Managing Batch Processors to reduce lead time in a semiconductor packaging line/ P. Chandra, S. Gupta// International Jornal of Production Research, 1997, #35 (3) – P. 611– 633;
11. Surjandari I. The Batch Sheduling Model for Dynamic multiitem, Multilevel Production in an assembly Job–Shop with Parrallel Machines./ I. Surjandari, A. Rachman, A. Dhini// International Journal of Technology, 2015,#1.– P. 84–96;
12. Lawler E. L. Sequencing and scheduling: Algorithms and complexity. / E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys.// Eindhoven University of Technology, Department оf Mathematics and Computing Science.– Eindhoven, Holland, 1989 – 72 p.
13. Петросян Л.А. Теория игр./ Л.А.Петросян, Н.А.Зенкевич, Е.А.Семина. – М.: Изд–во «Высшая школа», 1999. – 300с.
14. Гермейер Ю.Б. Игры с непротивоположными интересами. / Б.Гермейер. – М.: Наука, 1976. – 327 с.
15. Г.А. Раздобреева, Е. В. Коваль, Т.В. Кулешова. Методические указания к выполнению курсовой работы «Расчёт экономической эффективности создания и использования программного продукта» – Севастополь: Изд–во СевНТУ, 2015. – 32 с.
16. Куликов, Л.М. Экономическая теория: Учебник/Л.М. Куликов. – М.: ТК Велби, Издательство Проспект, 2010. – 432с.
17. СП 1.13130.2009 «Системы противопожарной защиты. Эвакуационные пути и выходы» 72с
18. ГОСТ Р ИСО 9241–5–2009 Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов 214с

# **ПРИЛОЖЕНИЕ А**

# **Маркетинговые исследования**

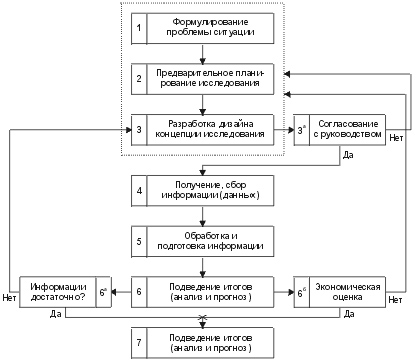


Рисунок 1 – Этапы маркетингового исследования



Рисунок 2 – Жизненный цикл програмного продукта



Рисунок 3 – Уровни создания и реализации програмного продукта

Таблица 1.1 – Оценка шансов и рисков проектируемого продукта

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Показатели | Балы | | | | | | | | |
| Опасность | | | Нейтрально | | | Шансы | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1. Объем рынка |  |  |  |  |  | + |  |  |  |
| 2. Рост рынка |  |  |  |  | + |  |  |  |  |
| 3. Финансовый потенциал |  |  |  | + |  |  |  |  |  |
| 4. Число конкурентов |  |  |  |  |  |  |  | + |  |
| 5. Поведение конкурентов |  |  |  |  |  |  | + |  |  |
| 6. Осведомленность потребителя |  |  |  |  | + |  |  |  |  |
| 7. Возможность повышения цен |  |  |  |  |  | + |  |  |  |
| 8. Изменение конъюнктуры рынка |  |  |  |  | + |  |  |  |  |
| 9. Возможность замещения продукта |  |  |  |  |  |  | + |  |  |
| 10. Потенциал сервиса |  |  |  |  |  | + |  |  |  |

**Формирование цены предложения разработчика**

Ипост = 28866,76 (руб)

Переменные (Иперем) – зависят от объема продукции.

Издержки на рекламу: 10% от Ипост

руб. (расход на рекламу)

(руб).

Рассчитаем цену программного продукта по формуле:

Ипост + Иперем= Иполн

,

Где Пр – прибыль 15% от Иполн

Из этого следует, что цена програмного продукта равна:

Таблица А.2 – Список потенциальных организаций и предприятий, которых интересует програмного продукта.

|  |  |  |
| --- | --- | --- |
| № | Название | Чем занимается |
| 1 | [ООО "Гермес"](http://alliance-catalog.ru/da_trans/) | ООО «Гермес», поставляет оборудование для предприятий конвейерного типа |
| 2 | ООО «СИР» | ООО «СИР» предлагает к поставке конвейерную продукцию для использования в различных областях промышленности Крымского региона |
| 3 | ["Евпаторийский Винный Завод"](http://tavrika.su/id15175) | Производство вина и его розлив с помощью ЭВМ конвейерного типа. |
| 4 | [Картонно–тарный комбинат](http://tavrika.su/id24838) | Изготовление тары из гофрокартона,производство конвейерного типа. |
| 5 | [Крымский кирпичный завод](http://tavrika.su/id27748) | Призводство кирпичей. |
| 6 | [ХимRussia](http://tavrika.su/id28574) | Производство автохимии и средств для профессионального клининга в Крыму. |
| 7 | Компания "Эко безопасность Крым" | Производство тентовых накрытий и каркасов для летних площадок, кафе, ресторанов, дискотек и концертных площадок – тенты и каркасы для грузовых автомобилей. |
| 8 | Производство "Абсолют–мебель" | Мебельная фабрика "Абсолют–мебель" специализируется на производстве детских парт, детских комнат, а также корпусной мебели для дома, офиса |
| 9 | ООО ПКФ "Черемош" | Изготовление фурнитуры для мебели |
| 10 | Фирма "Планета Цвета" | Мебельные фасады из МДФ с любым изображением; Стекло ламинированное пластиком; Стекла с тканью или обоями внутри. |

Таблица А.3 – Функции програмного продукта

|  |  |
| --- | --- |
| Наименование функции | Объем функций, тыс. УМК |
| Ввод, анализ входной информации, генерация кодов и процессор входного языка | |
| Организация ввода информации | 0,335 |
| Контроль | 1,05 |
| Организация ввода/вывода информации в интерактивном режиме | 0,775 |
| Организация ввода/вывода информации в сети терминалов | 1,35 |
| Генерация программ и ПС ВТ, а также ПС ВТ | |
| Формирование служебных таблиц | 2,005 |
| Управление ПС ВТ компонентами ПС ВТ, внешними устройствами | |
| Обработка ошибочных и сбойных ситуаций | 2,6 |
| Обеспечение интерфейса между компонентами | 3,43 |
| Отладка прикладных программ, вспомогательные программы и функции | |
| Справка и обучение | 0,225 |
| Итого: | 11,77 |

Таблица А.4 – Данные для расчета годовых эксплуатационных затрат (толстая)

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Стоимость ЭВМ | С | руб. | 27000,00 |
| 2. Потребляемая мощность | W | кВТ/ч | 0,35 |
| 3. Коэффициент использования по мощности | Kи |  | 0,70 |
| 4. Цена 1 кВт/ч электроэнергии | Цэ | руб. | 2,4 |
| 5. Номинальный фонд времени работы ЭВМ | Fном | час | 1974,00 |
| 6. Потери времени на ремонт и профилактику (% от Fном) | Програмного продуктаот | % | 4,00 |
| 7. Коэффициент годовых затрат на ремонт (от стоимости ЭВМ) | Кр | % | 6,00 |
| 8. Коэффициент сменности | Ксм |  | 1,00 |
| 9. Норма амортизационных отчислений на ЭВМ | Ноб | % | 4,16 |
| 13. Коэффициент начислений на фонд оплаты труда | Кн | % | 34,20 |
| 14. Коэффициент накладных расходов (% от ФОТ) | Кнр | % | 20,00 |
| 15. Коэффициент материальных затрат (% от стоимости ЭВМ) | Кмз | % | 8,00 |
| 16. Оклад разработчика | Окр | руб. | 13000,00 |

**Расчет показателей экономической эффективности**

Годовые издержки:

И2016=0

\*

Чистый денежный поток:

2) Определение показателей чистой текущей стоимости за период реализации проекта по формуле:

где E– ставка дисконтирования или норма доходности (прибыльности) от вложения средств(будем считать Е = 0,12); – расчетный год;

Рассчитаем показатели чистого дисконтированного денежного потока

3) Определение интегрального экономического эффекта

4) Рассчитаем амортизацию по формуле

где *,* – года; T = 5 (лет.)

Амортизация по годам распределяется прямолинейным методом.

5) Определим показатели прибыли

где, – объем реализации, – годовые издержки

# **ПРИЛОЖЕНИЕ Б**

**Схемы помещений и показатели охраны труда**



Рисунок B.1 – Схема помещения

Цифрами на эскизе обозначены: 1 – огнетушитель; 2 – шкаф (для хранения аптечки, документации, в том числе по охране труда и пожарной безопасности); 3 – рабочее место (рабочий стол с установленным на нем ПК и стул); 4—рабочее место начальника отдела (рабочий стол с ПК и лазерным принтером, а также стул); 5 – сетевой фильтр на 6 розеток; 6 – кадка с растением; 7 – сетевой рубильник–автомат; 8 – мультимедийная доска. (обозначение к картинке в приложение)

Таблица B.1 – Карта условий труда

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Фактор производственной среды или трудового процесса | Дата исследования | Нормативное значение (ПДК, ПДУ, допустимое значение) | Фактическое (измеренное) значение | Продолжительность действия фактора (часов или %) | Класс условий труда |
| *Тяжесть трудового процесса* | | | | | |
| Рабочая поза |  | Свободная удобная поза, возможность изменения (сидя, стоя) по желанию работника | Периодические пребывания в неудобной позе (работа с поворотом туловища, неудобным расположением конечностей) | до 25% времени смены | 2 |
| *Напряженность трудового процесса* | | | | | |
| Интеллектуальные нагрузки | | | | | |
| Восприятие сигналов (информации) и их оценка |  | Восприятие сигналов, не требуется коррекция действий | Восприятие сигналов с последующим сопоставлением фактических значений параметров с их номинальными значениями. Заключительная оценка фактических значений параметров | до 95% времени смены | 3.1 |
| Фактор производственной среды или трудового процесса | Дата исследования | Нормативное значение (ПДК, ПДУ, допустимое значение) | Фактическое (измеренное) значение | Продолжительность действия фактора (часов или %) | Класс условий труда |
| Распределение функций по степени сложности задания |  | Обработка и выполнение задания | Обработка, выполнение задания и его проверка | до 95% времени смены | 2 |
| Сенсорные нагрузки | | | | | |
| Длительность сосредоточенного  наблюдения (в % от времени смены) |  | до 25 % | 45% |  | 2 |
| Плотность сигналов (световых, звуковых) и сообщений в среднем  за 1 час работы |  | до 75 | 75–175 |  | 2 |
| Число производственных объектов  одновременного наблюдения |  | до 5 | 1–2 |  | 1 |
| Размер объекта различения (при расстоянии от глаз работающего  до объекта различения не более 0,5 м) в мм при длительности сосредоточенного наблюдения (% времени смены) |  | более 5 мм | 5–10 мм  более 50%  1– 0.3 мм  до 50%  менее  0.3 мм  до 25 % |  | 2 |
| Фактор производственной среды или трудового процесса | Дата исследования | Нормативное значение (ПДК, ПДУ, допустимое значение) | Фактическое (измеренное) значение | Продолжительность действия фактора (часов или %) | Класс условий труда |
| тельности сосредоточенного наблюдения (% времени смены) |  |  |  |  |  |
| Нагрузка на слуховой анализатор (при производственной необходимости восприятия речи или дифференцированных  сигналов) |  | Разборчивость слов и сигналов от 100% до 90% | Разборчивость слов и сигналов от 100% до 90% |  | 1 |
| Эмоциональные нагрузки | | | | | |
| Степень ответственности за результат собственной деятельности.  Значимость ошибок |  | Несет ответственность за выполнение отдельных элементов задания | Несет ответственность за выполнение отдельных элементов задания |  | 1 |
| Степень риска для собственной  жизни |  | Исключена | Исключена |  | 1 |
| Монотонность нагрузок | | | | | |
| Число элементов (приемов), необходимых |  | более 10 | 5–7 |  | 2 |
| Фактор производственной среды или трудового процесса | Дата исследования | Нормативное значение (ПДК, ПДУ, допустимое значение) | Фактическое (измеренное) значение | Продолжительность действия фактора (часов или %) | Класс условий труда |
| Продолжительность выполнения простых производственных заданий или повторяющихся операций, с |  | более 100 с | Более 100 с |  | 1 |
| Режим работы | | | | | |
| Продолжительность рабочего дня |  | 6–7 часов | 8–9 часов |  | 2 |
| Сменность работы |  | Односменная работа (без ночной смены) | Односменная работа (без ночной смены) |  | 1 |



Рисунок 2 – Схема размещения искусственного освещения

Таблица 8 – Параметры рабочего места пользователя ПЭВМ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Элемент рабочего места | Параметры | Обозначение по рис.2.2 | Фактическая величина | Диапазон регулирования | Нормативное значение  [1] |
| 1.Рабочий стол | Рабочая поверхность:  –высота | Н | 800 |  | 680–800мм |
| –ширина | – | 1200 |  | 800–1400мм |
| –глубина | В | 800 |  | 800–1400мм |
|  |  |  |  |  |
| Пространство для ног  –высота | H | 650 |  | Не менее: 600мм |
| –ширина | – | 700 |  | 500мм |
| –глубина на уровне колен | – | 500 |  | 450мм |
| –глубина на уровне вытянутых ног | a2 | 650 |  | 650мм |
| 2.Рабочий стул  (подъемно  Пово  ротный)` | Ширина сиденья | b | 500 |  | =>400мм |
| Глубина сиденья | a | 500 |  | =>400мм |
| Высота поверхности сиденья | h1 | 400 |  | 400–550мм |
| Угол наклона сиденья  –вперед  –назад | –  – | 100  50 |  | Не более: 150 50 |
| Элемент рабочего места | Параметры | Обозначение по рис.2.2 | Фактическая величина | Диапазон регулирования | Нормативное значение  [1] |
| 2.Рабочий стул  (подъемно  Поворотный) ` | Высота опорной поверхности  спинки | – | 300 |  | =>300мм |
| Радиус кривизны спинки в горизонтальной  плоскости | с | 400 |  | 260–400мм |
| Подлокотник (съемные или стационарные): |  |  |  |  |
| –длина | L | 300 |  | =>250мм |
| –ширина | b2 | 50 |  | 50–70мм |
| –высота на сиденье | h3 | 250 |  | 200–260мм |
| –расстояние между подлокотниками | T | 400 |  | 350–500мм |
| 3.Подставка для ног | Ширина | bп | 350 |  | =>300мм |
| Глубина | aп | 500 |  | =>400мм |
| Высота | hп | 100 |  | <=150мм |
| Наклон опорной поверхности | – | 15 |  | <=200 |
| 4.Пюпитр для документов  (перемещаемый | Наличие | – | – |  | Имеется |



Рисунок 3 – Схема рабочего места.1– Стол;2–Стул;3–Подставка для ног

# **ПРИЛОЖЕНИЕ В**

**Код программных модулей**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Diplom

{

class SecondLevel

{

private Groups groups, Q;

private Shedule shedule;

private List<List<int>> A;

public SecondLevel()

{

this.groups = new Groups(5);

this.Q = new Groups(5);

}

public void InitialConditions(int j)//this.groups.Z

{

for (int i = 0; i < this.groups.Z;i++)

this.groups.k.Add(0);

this.Q.k.Add(0);

this.groups.Set\_I1(j);//допустим

this.groups.Set\_I2(j);//допустим

this.groups.Set\_M(j);//допустим

this.Q.Set\_M(1);//допустим

}

private List<List<int>> BuildR(List<List<int>> N)

{

List<List<int>> Res=new List<List<int>>();

for (int i = 0; i < N.Count(); i++)

{

Res.Add(new List<int>());

}

int sum = 0, count = 0;

for (int i = 0; i < N.Count(); i++)

{

sum += N[i].Count();

}

for (int i = 0; i < N.Count(); i++)

{

for (int j = 0; j < sum; j++)

{

Res[i].Add(0);

}

}

for (int i = 0; i < N.Count(); i++)

{

for (int j = 0; j < N[i].Count(); j++)

{

Res[i][count] = N[i][j];

count++;

}

}

return Res;

}

public void Algoritm\_1()

{

while (true)

{

this.groups.Set\_Nzt(this.groups.Nz);//1

int i1 = this.groups.I2.Min();//2

this.groups.I1.Remove(i1);//2

this.groups.z = this.groups.Nzt.Min();//4

this.groups.Nzt.Remove(this.groups.Nzt.Min());//4

this.groups.hi1 = 1;//5

//6

//нужны данные 1 уроня все партии i` типа

for (int i = 1; i < groups.Nz1.Count(); i++)

// if (groups.Nz1[i] == A[i1])

{

groups.Nz1[i].Add(A[i1]);

break;

}

}

}

public List<List<List<int>>> Algoritm\_2()

{

int logi = 0;//номер группы текущий расматриваемый

//A.Clear();

/\*for (int i = 0; i < 4; i++)

{

List<int> w = new List<int>();

A.Add(w);

}\*/

//A[0].Add(12); A[0].Add(2); A[0].Add(2);

//A[1].Add(13); A[1].Add(3);

//A[2].Add(10); A[2].Add(2); A[2].Add(2); A[2].Add(2);

//A[3].Add(8); A[3].Add(2); A[3].Add(2); A[3].Add(2); A[3].Add(2);

A.RemoveAt(0);

for (int i = 0; i < A.Count(); i++)

A[i].RemoveAt(0);

for (int i = 0; i < 4; i++)

{

groups.Nz1.Add(new List<List<int>>());

}

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

groups.Nz1[i].Add(new List<int>());

}

}

Q.Nz1.Add(new List<List<int>>());

for (int i = 0; i < 4; i++)

Q.Nz1[0].Add(new List<int>());

for (int i = 0; i < A.Count(); i++)

{

for (int j = 0; j < A[i].Count(); j++)

{

if (logi >= 4)

{

Q.Nz1[0][i].Add(A[i][j]);

}

else

{

int Tz = 80;

groups.Nz1[logi][i].Add(A[i][j]);

this.shedule = new Shedule(this.BuildR(groups.Nz1[logi]), 4);

this.shedule.ConstructShedule();

if(this.shedule.GetTime() > Tz)

{

groups.Nz1[logi][i].Remove(A[i][j]);

j--;

logi++;

}

}

}

}

if(Q.Nz1[0].Count()>0)

Algoritm\_3();

return groups.Nz1;//изменить

}

public void Algoritm\_3()

{

for (int k = 0; k < groups.Nz1.Count();k++ )

for (int i = 0; i < Q.Nz1[0].Count(); i++)

for (int j = 0; j < Q.Nz1[0][i].Count();j++)

{

groups.Nz1[k][i].Add(Q.Nz1[0][i][j]);

int Tz = 80;

this.shedule = new Shedule(this.BuildR(groups.Nz1[k]), 4);

this.shedule.ConstructShedule();

if (shedule.GetTime() > Tz)

{

groups.Nz1[k][i].Remove(Q.Nz1[0][i][j]);

}

else

{

Q.Nz1[0][i].Remove(Q.Nz1[0][i][j]);

}

}

}

public List<List<int>> ReturnAMatrix()

{

List<List<int>> A1 = new List<List<int>>();

A1.Add(new List<int>());

for (int i = 0; i < 4; i++)

{

A1.Add(new List<int>());

A1[i + 1].Add(0);

}

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < groups.Nz1[i].Count(); j++)

{

for (int k = 0; k < groups.Nz1[i][j].Count(); k++)

{

A1[j + 1].Add(groups.Nz1[i][j][k]);

}

}

}

return A1;

}

public bool GenerateSolution(List<List<int>> matrixA)

{

this.A = matrixA;

this.Algoritm\_2();

return true;

}

}

}