|  |  |
| --- | --- |
| **УТВЕРЖДАЮ** |  |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л. Г. Афраймович**  **«\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 г.** |  |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к научно-исследовательской работе**

**по теме**

**«Задача производственного планирования  
 с альтернативными технологическими процессами»**

г. Н. Новгород, 2024

Оглавление

[1. Используемые обозначения 3](#_Toc154394444)

[1.1. Типы и допустимые форматы значений 3](#_Toc154394445)

[2. Формат входных данных 5](#_Toc154394446)

[2.1. Описание 5](#_Toc154394447)

[2.1.1. Информационная модель файла, содержащего информацию о производстве 5](#_Toc154394448)

[2.1.2. Информационная модель файла, содержащего информацию о заказах 8](#_Toc154394449)

[3. Формат выходных данных 11](#_Toc154394450)

[3.1. Описание 11](#_Toc154394451)

[4. Генератор входных данных 15](#_Toc154394452)

[4.1. Описание 15](#_Toc154394453)

[5. Тестирование входных данных на физическую возможность выполнения заказов 18](#_Toc154394454)

[5.1. Описание 18](#_Toc154394455)

[6. Тестирование результатов работы алгоритма на физическую возможность исполнения 19](#_Toc154394456)

[6.1. Описание 19](#_Toc154394457)

[7. Сравнительное тестирование двух результатов работы алгоритма по критерию просроченных заказов. 20](#_Toc154394458)

[7.1. Описание 20](#_Toc154394459)

[7.2. Интерпретация критерия 20](#_Toc154394460)

[8. Эффективность алгоритмов 21](#_Toc154394461)

[8.1. Описание таблицы «Эффективность алгоритмов» 21](#_Toc154394462)

[8.2. Описание таблицы «Сравнение эффективности алгоритмов» 22](#_Toc154394463)

[9. Алгоритм решения 23](#_Toc154394464)

[9.1. Фронтальный алгоритм 23](#_Toc154394465)

[9.1.1. Математическая постановка 23](#_Toc154394466)

[9.1.2. Шаги выполнения 24](#_Toc154394467)

[9.2. Алгоритмы для генерации распределения 25](#_Toc154394468)

[9.2.1. Градиентный алгоритм 25](#_Toc154394469)

[9.2.2. Ранцевый алгоритм 28](#_Toc154394470)

# Используемые обозначения

Для **обозначения допустимых форматов** входных данных будем использовать следующую запись, именуемую тэгом.

<Название\_Тэга название\_параметра1=ТИП\_ПАРАМЕТРА

название\_параметра2=ТИП\_ПАРАМЕТРА>

Все используемые типы и соответствующие им форматы значений представлены ниже.

Будем применять **сокращенную форму** записи тэга. В сокращенной форме приводится только имя тэга и не уточняется – имеет ли тег параметры и тело.

<Название\_Тэга>

Для обозначения необязательных конструкций будем использовать [] (квадратные скобки).

<Название\_Тэга [Необязательный\_параметр]>

или

<Тэг\_контейнер>

[<Необязательный\_тэг>]

</Тэг\_контейнер>

## Типы и допустимые форматы значений

|  |  |
| --- | --- |
| Тип значения | Допустимый формат значения |
| ДАТА | “ДД.ММ.ГОД ЧЧ:МН:СС”  “ДД.ММ.ГОД ЧЧ:МН”  “ДД.ММ.ГОД”  Здесь,  ДД – номер дня в месяце (1-31)  ММ – номер месяца (01-12)  ГОД – четырехзначный номер года  ЧЧ – час (00-23)  МН – минута (00-59)  СС – секунда (00-59)  ЗАМЕЧАНИЕ: Сокращенная запись  “ДД.ММ.ГОД” соответствует полной записи “ДД.ММ.ГОД 00:00:00”.  Сокращенная запись “ДД.ММ.ГОД ЧЧ:МН” соответствует полной  записи “ДД.ММ.ГОД ЧЧ:МН:00”.  Примеры:  “1.06.2005 08:30:00”  “1.06.2005 08:30”  “1.06.2005” |
| ТЕКСТ | “XXXXXXXXXXXX”  Здесь,  XXXXXXXXXXXX – произвольный текст. Минимальное число символов  в тексте – 0.  Пример:  “1825ВА1Н2НИ” |
| ЦЕЛОЕ\_ЧИСЛО | “N”  Здесь,  N – число от 0-99999999.  Пример:  “2561” |
| ДЕНЬ\_НЕДЕЛИ | “N”  Здесь,  N – номер дня недели (1-7). Так, понедельник соответствует 1,  а воскресенье – 7.  Пример:  “4” |
| ПЕРИОД\_ВРЕМЕНИ | Периоды с указанием начала и конца:  “ЧЧ:МН-ЧЧ:МН”  “ЧЧ-ЧЧ”  Периоды с указанием только начала (от начала до конца суток):  “ЧЧ:МН-”  “ЧЧ-”  Здесь,  ЧЧ – час (00-23)  МН – минута (00-59)  Примеры:  “08:30-12:40”  “13-17”  “17-” |
| ИДЕНТИФИКАТОР | “ID”  Здесь,  ID – уникальное целое число от 1 – 999999999999999.  Пример:  “122178785612331” |
| ДЛИТЕЛЬНОСТЬ | “ЧЧ.XX”  Здесь,  ЧЧ – число часов (0-9999).  XX – дробная часть часа (00-99). |

# Формат входных данных

## Описание

### Информационная модель файла, содержащего информацию о производстве

Входной файл system.xml содержит в себе информацию о предприятии, его рабочем графике, составе оборудовании, используемое для изготовления изделий.

Тэг-контейнер <SystemInformation> встречается только один раз в файле с входной информацией.

<SystemInformation>

<CalendarInformation>

<EquipmentInformation>

</SystemInformation>

В теле тэга обязательно встречаются ровно по одному разу следующие тэги-контейнеры:

* <CalendarInformation> содержит описание графика работы предприятия;
* <EquipmentInformation> содержит информацию об оборудовании.

**Информация о графике работы**

Вся информация о графике работы оборудования предприятия описывается только в теле тэга-контейнера <CalendarInformation>. Он может встретиться только один раз и только в теле тэга-контейнера <SystemInformation>.

Внутри <CalendarInformation> содержатся следующие тэги:

* <Timetable> - тэг-контейнер, объединяющий информацию о рабочих днях описанного расписания. Не содержит никаких атрибутов.
* <Day> - тэг, содержащий информацию о конкретном рабочем дне. Содержится внутри тэг-контейнера <Timetable>.

<Day day\_of\_week=ДЕНЬ\_НЕДЕЛИ time\_period=ПЕРИОД\_ВРЕМЕНИ/>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| day\_of\_week | ДЕНЬ\_НЕДЕЛИ | номер дня недели |
| time\_period | ПЕРИОД\_ВРЕМЕНИ | период времени, в который предприятие работает |

*Пример:*

<CalendarInformation>

<Timetable>

<Day day\_of\_week="1" time\_period="08:00:00-16:00:00" />

<Day day\_of\_week="2" time\_period="08:00:00-16:00:00" />

<Day day\_of\_week="3" time\_period="08:00:00-16:00:00" />

<Day day\_of\_week="4" time\_period="08:00:00-16:00:00" />

<Day day\_of\_week="5" time\_period="08:00:00-16:00:00" />

</Timetable>

</CalendarInformation>

**Информация об оборудовании**

Вся информация об оборудовании описывается только в теле тэга-контейнера <EquipmentInformation>. Этот тэг не содержит никаких параметров.

<EquipmentInformation>

<EquipmentGroup>

<Equipment>

[<Equipment>]

…

[<Equipment>]

</EquipmentGroup>

[<EquipmentGroup>]

…

[<EquipmentGroup>]

</EquipmentInformation>

Тэг-контейнер <EquipmentInformation> встречается только один раз и только в теле тэга-контейнера <SystemInformation>.

Все экземпляры оборудования описываются тэгом <Equipment>.

<EquipmentGroup>

<Equipment>

[<Equipment>]

…

[<Equipment>]

</EquipmentGroup>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| id | ИДЕНТИФИКАТОР | уникальный номер группы оборудований |
| name | ТЕКСТ | наименование группы оборудований |

Информация об конкретных экземплярах оборудования описывается в тэге <Equipment>.

<Equipment id=ИДЕНТИФИКАТОР name=ТЕКСТ>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| id | ИДЕНТИФИКАТОР | уникальный номер оборудования |
| name | ТЕКСТ | наименование конкретного оборудования |

Тэг <Equipment> может встречаться сколь угодно много раз, но только в теле тэга контейнера <EquipmentGroup>. Фактически это означает, что для того, чтобы описать некоторую единицу оборудования надо ее включить в соответствующую группу (если группы нет, то необходимо специально создать для этого оборудования собственную группу). Дублирование описания одной и той же единицы оборудования недопустимо.

*Пример:*

<EquipmentInformation>

<EquipmentGroup id="01" name="Станки">

<Equipment id="01" name="Станок 1 в цехе 1"/>

<Equipment id="02" name="Станок 2 в цехе 1"/>

<Equipment id="03" name="Станок 1 в цехе 2"/>

<Equipment id="04" name="Станок 2 в цехе 2"/>

</EquipmentGroup>

<EquipmentGroup id="03" name="Инструменты">

<Equipment id="76" name="Молоток"/>

</EquipmentGroup>

</EquipmentInformation>

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание**Пример файла**

### Информационная модель файла, содержащего информацию о заказах

Входной файл tech.xml содержит в себе информацию о заказах, их составах, технических процессах, используемых для изготовления изделий.

Тэг-контейнер <OrderInformation> встречается только один раз в файле с входной информацией.

<OrderInformation>

<Order>

…

</Order>

[<Order>]

…

[<Order>]

</OrderInformation>

Вся информация об изделиях, партиях и технологических операциях указывается в теле тэга-контейнера <Order>. Этот тэг не содержит никаких параметров и может встретиться несколько раз только в теле тэга-контейнера <OrderInformation>.

<Order id=ИДЕНТИФИКАТОР date\_begin=ДАТА date\_end=ДАТА>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| id | ИДЕНТИФИКАТОР | уникальный номер заказа |
| date\_begin | ДАТА | дата и время, начиная с которого можно запускать в производство заказанные изделия |
| date\_end | ДАТА | директивный срок, до которого заказ должен быть выполнен |

Информационная модель изделия (тэги <Detail>) включает в себя информацию о технических процессах (тэги <TechProcess>), состоящих из операций (тэги <Operation>), с помощью которых можно создать это изделие.

Информация о изделии описывается в тэге-контейнере <Detail>.

<Detail id=ИДЕНТИФИКАТОР name=ТЕКСТ count=ЦЕЛОЕ\_ЧИСЛО>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| id | ИДЕНТИФИКАТОР | уникальный номер вида изделия |
| name | ТЕКСТ | наименование изделия, которое необходимо создать |
| count | ЦЕЛОЕ\_ЧИСЛО | число изделий, которое необходимо сделать |

Информация о техническом процессе описывается в тэге-контейнере <TechProcess>. Тэг содержит только один атрибут – id, указывающий на уникальный номер технического процесса

<TechProcess id=ИДЕНТИФИКАТОР>

Информация о технологической операции описывается в тэге-контейнере <Operation>.

<Operation id=ИДЕНТИФИКАТОР name=ТЕКСТ duration=ДЛИТЕЛЬНОСТЬ equipment\_group=ИДЕНТИФИКАТОР prev\_operation\_id=ИДЕНТИФИКАТОР next\_operation\_id=ИДЕНТИФИКАТОР />

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| id | ИДЕНТИФИКАТОР | уникальный номер операции |
| name | ТЕКСТ | наименование операции |
| duration | ДЛИТЕЛЬНОСТЬ | длительность проведения операции |
| equipment\_group | ИДЕНТИФИКАТОР | уникальный номер группы оборудований, используемой для проведения операции |
| prev\_operation\_id | ИДЕНТИФИКАТОР | уникальный номер операции, предшествующей описываемой операции. Если в атрибуте указан 0, то для описываемой операции нет предыдущей операции |
| next\_operation\_id | ИДЕНТИФИКАТОР | уникальный номер операции, необходимую сделать после описываемой операции. Если в атрибуте указан 0, то для описываемой операции нет последующей операции |

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описание**Пример файла**

# Формат выходных данных

## Описание

Выходной файл result.xml содержит в себе информацию о результатах выполнения заказов, описанных в tech.xml, предприятием, описанным в system.xml.

Тэг-контейнер <Result> встречается только один раз в файле с выходной информацией.

<Result all\_start\_date\_time=ДАТА all\_end\_date\_time=ДАТА>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| all\_start\_date\_time | ДАТА | дата и время начала работ над заказами |
| all\_end\_date\_time | ДАТА | дата и время завершения работ над заказами |

В теле тэга <Result> встречаются следующие тэги:

* <Order> - тэг-контейнер, содержащий информацию о выполненном заказе. Может встретиться несколько раз.
* <Product> - тэг-контейнер, содержащий информацию об изготовленных изделиях в рамках заказа. Находится внутри тэга <Order>. Может встретиться несколько раз.
* <Operation> - тэг, содержащий информацию об операции, используемой при изготовлении изделия. Находится внутри тэга <Product>. Может встретиться несколько раз.

<Result>

<Order>

…

</Order>

[<Order>]

…

[<Order>]

</Result>

Вся информация о результатах работы с каждым отдельным заказом содержится в тэге-контейнере <Order>.

<Order order\_id=ИДЕНТИФИКАТОР order\_start\_date\_time=ДАТА order\_end\_date\_time=ДАТА>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| order\_id | ИДЕНТИФИКАТОР | уникальный номер выполненного заказа |
| order\_start\_date\_time | ДАТА | дата и время начала выполнения заказа |
| order\_end\_date\_time | ДАТА | дата и время завершения выполнения заказа |

Информация о результатах работы с каждым изделием из определённого заказа содержится в тэге-контейнере <Product>.

<Product id=ИДЕНТИФИКАТОР product\_id=ИДЕНТИФИКАТОР tech\_process\_id=ИДЕНТИФИКАТОР product\_start\_date\_time=ДАТА product\_end\_date\_time=ДАТА>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| id | ИДЕНТИФИКАТОР | уникальный номер конкретного изделия |
| product\_id | ИДЕНТИФИКАТОР | номер вида изделия |
| tech\_process\_id | ИДЕНТИФИКАТОР | номер технического процесса, с помощью которого было изготовлено изделие |
| product\_start\_date\_time | ДАТА | дата и время начала изготовления конкретного изделия |
| product\_end\_date\_time | ДАТА | дата и время завершения изготовления конкретного изделия |

Информация о выполненных операциях при изготовлении конкретного изделия содержится в тэге <Operation>.

<Operation operation\_id=ИДЕНТИФИКАТОР prev\_operation\_id=ИДЕНТИФИКАТОР next\_operation\_id=ИДЕНТИФИКАТОР equipment\_id=ИДЕНТИФИКАТОР operation\_start\_date\_time=ДАТА operation\_end\_date\_time=ДАТА>

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| operation\_id | ИДЕНТИФИКАТОР | уникальный номер операции |
| equipment\_id | ИДЕНТИФИКАТОР | уникальный номер оборудования, используемого для проведения операции |
| prev\_operation\_id | ИДЕНТИФИКАТОР | уникальный номер операции, предшествующей описываемой операции. Если в атрибуте указан 0, то для описываемой операции нет предыдущей операции |
| next\_operation\_id | ИДЕНТИФИКАТОР | уникальный номер операции, необходимую сделать после описываемой операции. Если в атрибуте указан 0, то для описываемой операции нет последующей операции |
| operation\_start\_date\_time | ДАТА | дата и время начала выполнения операции |
| operation\_end\_date\_time | ДАТА | дата и время завершения выполнения операции |

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание**Пример файла**

# Генератор входных данных

## Описание

Генератор входных данных представляет собой метод, позволяющий заполнить файлы system.xml и tech.xml, описанные в пункте 2, корректными значениями для проведения испытаний работы алгоритма. Параметры, по которым происходит генерация, описаны в файле формата json. Каждый параметр в файле имеет наименование и соответствующее ему значение. Формат записи параметра выглядит следующим образом:

“<наименование>”: <значение>

*Пример:*

"daysForSchedule": 7

В таблице ниже приведены список параметров генератора, их типы и описание.

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Описание |
| daysForSchedule | ЦЕЛОЕ\_ЧИСЛО | количество рабочих дней в расписании |
| startWorkingTime | ЦЕЛОЕ\_ЧИСЛО | число, обозначающее час начала рабочего дня, минуты и секунды равны 0 |
| endWorkingTime | ЦЕЛОЕ\_ЧИСЛО | число, обозначающее час окончания рабочего дня, минуты и секунды равны 0 |
| ordersCount | ЦЕЛОЕ\_ЧИСЛО | количество заказов |
| minEquipmentGroupCount | ЦЕЛОЕ\_ЧИСЛО | минимальное количество групп оборудований |
| maxEquipmentGroupCount | ЦЕЛОЕ\_ЧИСЛО | максимальное количество групп оборудований |
| minEquipmentCount | ЦЕЛОЕ\_ЧИСЛО | минимальное количество оборудования в группе |
| maxEquipmentCount | ЦЕЛОЕ\_ЧИСЛО | максимальное количество оборудования в группе |
| minOrderStartTime | ДАТА | минимальное значение раннего времени начала выполнения заказа |
| maxDurationStartTime | ЦЕЛОЕ\_ЧИСЛО | максимальное значение разницы между минимальным и максимальным значениями раннего времени начала выполнения заказа в днях |
| minDurationTimeInDays | ЦЕЛОЕ\_ЧИСЛО | минимальная планируемая длительность выполнения заказа в днях |
| maxDurationTimeInDays | ЦЕЛОЕ\_ЧИСЛО | максимальная планируемая длительность заказа операции в днях |
| minDetailsTypeCount | ЦЕЛОЕ\_ЧИСЛО | минимальное количество типов деталей |
| maxDetailsTypeCount | ЦЕЛОЕ\_ЧИСЛО | максимальное количество типов деталей |
| minDetailsCount | ЦЕЛОЕ\_ЧИСЛО | минимальное количество деталей одного типа |
| maxDetailsCount | ЦЕЛОЕ\_ЧИСЛО | максимальное количество деталей одного типа |
| minTechProcessCount | ЦЕЛОЕ\_ЧИСЛО | минимальное количество технологических процессов на один тип деталей |
| maxTechProcessCount | ЦЕЛОЕ\_ЧИСЛО | максимальное количество технологических процессов на один тип деталей |
| minOperationsCount | ЦЕЛОЕ\_ЧИСЛО | минимальное количество операций в одном технологическом процесса |
| maxOperationsCount | ЦЕЛОЕ\_ЧИСЛО | максимальное количество операций в одном технологическом процесса |
| minOperationDuration | ЦЕЛОЕ\_ЧИСЛО | минимальная длительность операции в секундах |
| maxOperationDuration | ЦЕЛОЕ\_ЧИСЛО | максимальная длительность операции в секундах |

# Тестирование входных данных на физическую возможность выполнения заказов

## Описание

Перед запуском алгоритма на данных производства и заказов необходимо проверить данные заказы на возможность их исполнения на данном производстве.

Для каждой детали из каждого заказа проверяются все необходимые для производства операции – выясняется, есть ли на указанном производстве необходимое для исполнения операции оборудование.

Если оборудование для выполнения отсутствует, тест сообщает, для какой операции нет необходимого оборудования.

# Тестирование результатов работы алгоритма на физическую возможность исполнения

## Описание

Для проверки результатов алгоритма на физическую возможность исполнения предлагается тестирование, включающее в себя следующие пункты:

1. Проверка, что каждая операция выполняется соответствующее ей время.
2. Проверка соответствия техпроцесса из результирующего файла реальному техпроцессу и контроль правильного порядка исполнения операций.
3. Проверка указанных временных промежутков для всех объектов результирующего файла.
4. Проверка занятости оборудования – на одном оборудовании не могут исполняться две операции одновременно.

# Сравнительное тестирование двух результатов работы алгоритма по критерию просроченных заказов.

## Описание

Для того, чтобы сравнивать результаты работы алгоритма по качеству, предлагается следующее тестирование:

Два результата сравниваются друг с другом по критерию просроченных заказов. Для каждого результата вычисляется произведение количества часов, на которые заказ был просрочен на суммарное количество часов, выполненных за границей директивного срока для каждой детали из соответствующего заказа.

## Интерпретация критерия

Нормируем общую трудоёмкость, выполненную за границей директивного срока на общую трудоёмкость заказа. Сумму результатов по каждому заказу делим на количество заказов.

*–* полная трудоёмкость заказа

*–* трудоёмкость детали ,*типа ,* из заказа , выполненная за границей директивного срока. 0 в случае, если успели выполнить деталь в срок,

# Эффективность алгоритмов

## Описание таблицы «Эффективность алгоритмов»

Эффективность алгоритмов отражается в таблице следующего вида:

* № - номер задачи;
* Количество заказов – количество заказов, описанных в tech.xml;
* Количество типов деталей - количество различных типов деталей, описанных в tech.xml;
* Среднее количество деталей каждого типа – соотношение количества деталей каждого типа на общее количестве деталей в задаче;
* Среднее количество операций на деталь – соотношение количества операций в технологическом процессе на общее количестве деталей в задаче;
* Количество атомарных ресурсов – количество оборудования, описанных в system.xml;
* Минимальное число альтернатив на деталь;
* Максимальное число альтернатив на деталь;
* Среднее число альтернатив на деталь – соотношение количества альтернативностей на общее количестве деталей в задаче;
* Количество произведённых операций;
* Среднее суммарное количество дней просрочки – соотношение количества просроченных дней на количество запусков;
* Средний критерий – соотношение значения критерия на количество запусков;
* Среднее время исполнения в секундах – соотношение времени выполнения алгоритма на количество запусков.

*Пример:*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Количество заказов** | **Количество типов деталей** | **Среднее количество деталей каждого типа** | **Среднее количество операций на деталь** | **Количество атомарных ресурсов** | **Минимальное число альтернатив на деталь** | **Максимальное число альтернатив на деталь** | **Среднее число альтернатив на деталь** | **Количество произведённых операций** | **Среднее суммарное количество дней просрочки** | **Средний критерий** | **Среднее время исполнения в секундах** |
| 1 | 1 | 1 | 1000.0 | 3.0 | 11 | 2 | 2 | 2.0 | 3000.0 | 3.4 | 12.027 | 0.0 |
| 2 | 1 | 1 | 2500.0 | 3.75 | 15 | 4 | 4 | 4.0 | 8194.9 | 18.2 | 80.336 | 3.1 |
| 3 | 1 | 2 | 1000.0 | 4.75 | 17 | 2 | 2 | 2.0 | 9071.8 | 10.4 | 31.659 | 3.0 |
| 4 | 1 | 2 | 1000.0 | 4.625 | 17 | 4 | 4 | 4.0 | 9275.6 | 40.5 | 193.288 | 2.7 |
| 5 | 2 | 4 | 1000.0 | 3.7 | 26 | 2 | 3 | 2.5 | 14262.2 | 12.5 | 22.439 | 11.5 |

## Описание таблицы «Сравнение эффективности алгоритмов»

Сравнение эффективности алгоритмов отражается в таблице следующего вида:

* № - номер задачи
* Имя файла с лучшим результатом по времени просрочки – название таблицы, у которой на текущей задаче меньше просрочка в днях
* Обгон по времени просрочки – обгон по просроченным дням
* Имя файла с лучшим результатом по критерию – название таблицы, у которой на текущей задаче меньше критерий
* Обгон по критерию

*Пример:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Имя файла с лучшим результатом по времени просрочки** | **Обгон по времени просрочки** | **Имя файла с лучшим результатом по критерию** | **Обгон по критерию** |
| 1 | resultsAlpha.csv | 0.6 | resultsAlpha.csv | 7.4035484895044 |
| 2 | resultsAlpha.csv | 10.4 | resultsAlpha.csv | 33.047764484320254 |
| 3 | resultsAlpha.csv | 3.9 | resultsAlpha.csv | 26.573714755192388 |
| 4 | resultsBackpack.csv | 4.4 | resultsAlpha.csv | 22.876350280786625 |
| 5 | resultsAlpha.csv | 7.9 | resultsAlpha.csv | 22.303498242286437 |

# Алгоритм решения

В рамках данного проекта реализовано несколько алгоритмов для планирования работы производства по предварительно описанным заказам и рабочей системы.

Основой как для базового, так и для собственного алгоритмов являются одни и те же действия – фронтальный алгоритм. Разница между алгоритмами в том, что в базовом фронтальном алгоритме альтернативности распределяются случайным образом, а в собственном распределение генерируется алгоритмом на уровень выше, а фронтальный алгоритм получает на вход готовое распределение альтернативностей.

## Фронтальный алгоритм

### Обозначения и варьируемые параметры

*Используемые обозначения:*

– количество заказов,

– количество типов изделий,

– количество типов оборудования,

– количество альтернативных технологий изделия в заказе , ,

– количество изделий типа , в заказе , ,

– количество операций, необходимых для изготовления изделия типа , в заказе , ,

– затрачиваемое время на изготовление операции , изделия типа , в заказе , , ,

– необходимое оборудование для изготовления операции , изделия типа , в заказе , , .

*Варьируемые параметры:*

– количество изделий заказа , изготавливаемых по технологии , , , .

– техпроцесс, выбранный для детали номер , типа из заказа , , , .(В базовом алгоритме техпроцесс выбирается случайно из списка техпроцессов, а в собственном алгоритме фронтальный алгоритм уже получает на вход готовое распределение)

– оборудование, на котором выполняется операция , детали номер , типа , из заказа , , , .

- время, когда мы начали и закончили операцию , детали номер , типа из заказа , , , .

### Шаги выполнения

1. Формируется первоначальный пустой список тактирования – список, который будет содержать информацию о том, в какое время проходить по списку операций и опрашивать оборудование. В данный список попадают: временя раннего начала заказов и временя, когда освобождается какое-либо оборудование. Формируется пустой фронт работ – операции,
2. Переходим к следующему в списке тактирования такту времени.
3. Если наступило время раннего начала заказа, добавляем первые операции для всех деталей из данного заказа во фронт работ.
4. Если освободилось какое-либо оборудование, помечаем оборудование как свободное и добавляем следующую за завершенной операцию во фронт, если такая есть.
5. Если операции закончились, переходим на шаг 6, иначе, на шаг 7.
6. Если заказы закончились, переходим на шаг 11, иначе, на шаг 2.
7. Проходим по списку операций и проверяем, можем ли начать какую-либо операцию.
8. Если можно начать операцию, выбираем, какую начнём и отправляем её на свободное оборудование, добавляем в список тактирования время, когда операция завершится.
9. Опрашиваем операции до тех пор, пока можем отправлять их на свободное оборудование.
10. Переходим на шаг 2.
11. Завершаем работу алгоритма.

В результате работы алгоритма получаем информацию о том, на каком оборудовании и в какое время будет выполняться каждая операция каждого заказа, а также вычисленный критерий по заказам, завершёнными с опозданием.

## Алгоритмы для генерации распределения

### Градиентный алгоритм

#### Математическая постановка

*Используемые обозначения:*

– количество заказов,

– количество типов изделий,

– количество альтернативных технологий изделия в заказе , ,

– количество изделий типа , в заказе , ,

*Варьируемые параметры:*

– количество изделий заказа , изготавливаемых по технологии , , , .

*Параметры, которые считаются в процессе работы алгоритма:*

– срок, когда были выполнены все работы над деталью , типа , из заказа , ,

– полная трудоёмкость над деталью *,* типа *,* из заказа , ,

*Ограничения:*

.

*Критерий:*

* *Матричное Представление:* Использование матрицы *Xijk*, где сумма по *k* от *1* до *ai* равняется *Bij*.
* *Целевая Функция и Критерии***:** Расчет целевой функции с помощью фронтального алгоритма, определяющего эффективность планирования. Возможность использования выпуклой комбинации двух векторов и их округления до целые неотрицательных значений, каждый из которых является допустимым решением, для генерации новых решений.
* *Генерация Решений***:** Процесс генерации новых точек X с использованием формулы , где a выбирается в зависимости от лучшего по критерию решения, с учетом целочисленности и положительности значений матрицы.  
  Если матрица получается не целочисленной или отрицательной, то полученная матрица округляется, отрицательные значения приравниваются нулю, с занесением разницы в отдельный бюджет который потом распределяется между неотрицательными значениями матрицы. Не целочисленные значения приводятся к целочисленным по такому же принципу. Сравнивая получившиеся целевые функции, выбираем место для генерирования следующей точки.
* *Оценка Эффективности:* происходит выбор лучшей точки после определенного количества итераций, и ее оценка с использованием фронтального алгоритма для сравнения времени задержки выполнения заказов записывается в файл.

Запуск Алгоритма происходит с различными значениями матрицы X, после чего происходит сравнение итогов, на основе которых составляется отчет по данному алгоритму со сравнительным анализом.

*Содержательный смысл критерия:*

Нормируем общую трудоёмкость, выполненную за границей директивного срока на общую трудоёмкость заказа. Сумму результатов по каждому заказу делим на количество заказов.

*–* полная трудоёмкость заказа

*–* трудоёмкость детали ,*типа ,* из заказа , выполненная за границей директивного срока. 0 в случае, если успели выполнить деталь в срок,

#### Шаги выполнения

1. Случайно генерируем заданное число распределений альтернативностей, для каждой запускаем фронтальный алгоритм и считаем критерий, добавляем их в общий список распределений.
2. Генерируем новое распределение альтернативностей с помощью выпуклой свёртки двух выбранных распределений альтернативностей. (Выбираем два распределения из списка одним из двух способов)
3. Добавляем новое распределение альтернативности в общий список альтернативностей, запускаем для него фронтальный алгоритм, считаем для нового распределения критерий.
4. Возвращаемся на шаг 2. Повторяем заданное число раз.
5. Из списка всех распределений находим распределение с лучшим критерием, возвращаем как результат работы всего алгоритма.

*Способы выбора двух распределений:*

1. Выбираем случайно.
2. Кластеризация – все распределения разделены на три кластера по критерию:

* кластер распределений, ближайших по критерию к лучшему;
* кластер распределений, ближайших по критерию к среднему;
* кластер распределений, ближайших по критерию к худшему.

1. Алгоритм кластеризации. После выполнения фронтального алгоритма и нахождения критерия для распределения делаем следующие шаги:
2. Находим лучший, средний и худший по критерию распределения.
3. Каждое распределение разделяем по трем кластерам по близости к худшему, среднему или лучшему.
4. Алгоритм выбора двух для генерации третьего:
5. Случайно выбираем первое распределение из кластера лучших.
6. Случайно выбираем один из кластеров – средний или худший (Средний выбирается с вероятностью в 67%, худший – 33%).
7. Случайно выбираем второе распределение из выбранного кластера.
8. Делаем приближение от второго распределения к первому.

### Ранцевый алгоритм

#### Математическая постановка

*Используемые обозначения:*

– количество заказов,

– количество типов изделий,

– количество типов оборудования,

– количество альтернативных технологий изделия в заказе , ,

– количество изделий типа в заказе , ,

– мощность (норма-часы) оборудования типа , затрачиваемая при изготовлении изделия , используя технологию , , ,

– доступная мощность (норма-часы) оборудования типа , .

*Варьируемые параметры:*

– количество изделий заказа , изготавливаемых по технологии , , , .

*Ограничения:*

.

*Критерий:*

*Содержательный смысл критерия:*

Берем перерасходованные объема по каждому из видов оборудования и нормируем его на доступную мощность вида оборудования, общий критерий равен сумме нормированных перерасходов мощностей.

#### Шаги выполнения

1. Находим начальное распределение мощностей через равномерное распределение альтернативностей (Запускаем фронтальный алгоритм с равномерным распределением альтернативностей и берём сумму норма-часов занятого оборудования до дедлайна заказа).
2. Ставим задачу ранцевого типа, в которой генерируем очень много различных распределений альтернативностей, для каждой из которых считаем критерий по формуле критерия. Находим лучшее по критерию.
3. Для лучшего распределения альтернативностей запускаем фронтальный алгоритм, запоминаем критерий и результата берём новое распределение мощностей. Возвращаемся на второй шаг и повторяем указанное количество раз.
4. После выполнения заданного количества повторений 2 и 3 шагов, среди всех запомненных результатов выполнения фронтального алгоритма берём лучший по критерию и возвращаем как результат всего алгоритма.