**ЗАДАЧА ПРОИЗВОДСТВЕННОГО ПЛАНИРОВАНИЯ С АЛЬТЕРНАТИВНЫМИ ТЕХНИЧЕСКИМИ ПРОЦЕССАМИ**

**Руководство пользователя**

**Листов 8**

2024

СОДЕРЖАНИЕ

[1. Назначение программы 3](#_Toc156943080)

[2. Условия выполнения программы 4](#_Toc156943081)

[3. Выполнение программы 5](#_Toc156943082)

[3.1. Подготовка к работе 5](#_Toc156943083)

[3.2. Описание команд запуска 5](#_Toc156943084)

[3.3. Запуск функций 6](#_Toc156943085)

1. Назначение программы

Программное средство, предназначенное для решения задачи производственного планирования с альтернативными техническими процессами при помощи различных алгоритмов, а также сравнения алгоритмов по эффективности.

1. Условия выполнения программы

Программа предназначена для запуска на ЭВМ, соответствующих следующим минимальным программным и аппаратным характеристикам:

* RAM: не менее 8 ГБ;
* Процессор: минимальное требование – Intel Core i3-2130 3.4 ГГц;
* Операционная система Windows 10 с установленной Java 8.

1. Выполнение программы
   1. Подготовка к работе
2. На рабочей машине должна быть установлена java 8. Получить её можно по ссылке: https://www.java.com/ru/download
3. Необходимо скачать скомпилированную jar библиотеку по ссылке: <https://github.com/AlexeyBaldin/production-resources/blob/master/demo/ProductionResources.jar>, или скомпилировать её вручную из репозитория по ссылке: https://github.com/AlexeyBaldin/production-resources/tree/master
4. При необходимости загрузить тестовые данные по ссылке: https://github.com/AlexeyBaldin/production-resources/tree/master/Basis
   1. Описание команд запуска

Взаимодействие пользователя с программой осуществляется посредством ввода команды *java -jar ProductionResources* в консоль со следующими аргументами:

**HELP**

**GEN** <имя файла с параметрами генератора> <количество генерируемых пар производство-заказы>

**ALG BASE** <имя файла производства>.xml <имя файла заказов>.xml <имя выходного файла результатов>.xml <количество запусков алгоритма> <Тип фронтального алгоритма> <Количество потоков для фронтального алгоритма>

**ALG OWN\_ALPHA** <имя файла производства>.xml <имя файла заказов>.xml <имя выходного файла результатов>.xml <количество запусков алгоритма> <стартовое количество генераций альтернативностей> <бюджет генератора альтернативностей> <количество потоков для вариатора> <тип фронтального алгоритма> <количество потоков для фронтального алгоритма>

**ALG OWN\_BACKPACK** <имя файла производства>.xml <имя файла заказов>.xml <имя выходного файла результатов>.xml <количество запусков алгоритма> <Бюджет генератора альтернативностей> <Бюджет запусков пересчёта мощностей> <тип фронтального алгоритма> <количество потоков для фронтального алгоритма>

**TEST POSS** <имя файла производства> <имя файла заказов>

**TEST REAL** <имя файла производства> <имя файла заказов> <имя файла результатов>

**TEST COMP** <имя файла заказов> <имя первого файла результатов> < имя первого файла результатов>

**TEST BASIS BASE** <Название папки с данными производства и заказов> <Количество пар производство-заказы> <имя файла результатов>.csv <количество запусков алгоритма> <Тип фронтального алгоритма> <Количество потоков для фронтального алгоритма>

**TEST BASIS OWN\_ALPHA** <Название папки с данными производства и заказов> <Количество пар производство-заказы><Стартовое количество генераций альтернативностей> <Бюджет генератора альтернативностей> <имя файла результатов>.csv <количество запусков алгоритма> <количество потоков для вариатора> <тип фронтального алгоритма> <количество потоков для фронтального алгоритма>

**TEST BASIS OWN\_BACKPACK** <Название папки с данными производства и заказов> <Количество пар производство-заказы> <Бюджет генератора альтернативностей> <Бюджет запусков пересчёта мощностей> <имя файла результатов>.csv <количество запусков алгоритма> <тип фронтального алгоритма> <количество потоков для фронтального алгоритма>

**COMP\_RESULT\_TABLES** <имя файла с таблицей результатов первого алгоритма>.csv <имя файла с таблицей результатов второго алгоритма>.csv <имя файла с результатами сравнения>.csv

*Описание аргументов команды запуска*

**GEN | ALG | TEST | COMP\_RESULT\_TABLES | HELP** – тип выполняемых действий. GEN используется для запуска генератора производства и заказов. ALG используется для запуска алгоритма, TEST используется для запуска тестирования, COMP\_RESULT\_TABLES используется для запуска сравнения двух таблиц с результатами алгоритмов, которые генерируются в результате команды TEST BASIS. HELP используется для вывода справки

**Тип алгоритма (BASE | OWN\_ALPHA | OWN\_BACKPACK)** – следующий аргумент, если запускается алгоритм. Используется для выбора типа алгоритма. Base – базовый алгоритм, альтернативности распределяются случайно, затем отрабатывает фронтальный алгоритм. Own\_alpha – первый распределения альтернативностей генерируются случайно, последующие распределяются с помощью выпуклых сверток нескольких ранее сгенерированных распределений альтернативностей. Own\_backpack – двухуровневый алгоритм, генерируется множество случайных распределений альтернативностей, для каждой находится критерий в релаксированной до задачи о ранце задаче, данный процесс происходит несколько раз с пересчётом максимальных мощностей в релаксированной задаче.

**Тип тестирования (POSS | REAL | COMP | BASIS (BASE | OWN\_ALPHA | OWN\_BACKPACK)** — следующий аргумент, если запускается тестирование. Используется для выбора типа тестирования.

Типы тестирования:

* **POSS** — запуск тестирования на физическую возможность выполнить заказы на производстве;
* **REAL** — запуск тестирования на возможность физически уложить представленный план заказов на производстве;
* **COMP** — запуск сравнительного тестирования двух планов укладки заказов по производству;
* **BASIS (BASE | OWN\_ALPHA | OWN\_BACKPACK)** – запуск тестирования одного из видов алгоритмов на предоставленном множестве пар входных данных. Для них формируются таблицы с данными по тестированию, которые в последствие можно использовать для сравнения.

Для расчётов реализовано 2 варианта хранения и использования фронта работ, для выбора, какой из них использовать алгоритмам необходимо передавать одну из двух команд в поле «Тип фронтального алгоритма»:

* Candidates – фронт хранится в виде списка, для выбора команды на исполнение формируется список кандидатов тех, кто может исполняться в данный момент. Поддерживает только последовательный вариант выполнения (запускается за одном потоке)
* Record – фронт хранится в виде дерева с критериями качества операций, для выбора команды на исполнение из дерева берётся одна из лучших по критерию операция. Поддерживает последовательный и параллельный варианты выполнения (Один или несколько потоков)

Для каждого из аргументов затем вводится список необходимых данных.

* 1. Запуск функций

**Генератор входных данных**

Запуск генератора данных о производстве и заказах с параметрами из файла *generatorParameters.json,* генерация 5 пар производство-заказы. Тестирование вывода генератора происходит в процессе генерации.

*java -jar ProductionResources.jar GEN generatorParameters.json 5*

**Базовый алгоритм**

Запуск базового алгоритма с входными данными о производстве из файла production.xml, входными данными о заказах из файла orders.xml, вывод результата в файл res.xml, с 5 запусками для выбора лучшего, с типом фронтального алгоритма «candidates», с 1 потоком для фронтального алгоритма:

*java -jar ProductionResources ALG BASE production.xml orders.xml res.xml 5 candidates 1*

**Собственный градиентный алгоритм**

Запуск градиентного алгоритма с входными данными о производстве из файла production.xml, входными данными о заказах из файла orders.xml, вывод результата в файл res.xml, с 5 запусками для выбора лучшего, стартовой генерацией 10 распределений альтернативностей, общим бюджетом на генерацию альтернативностей 50, с 4 потоками для градиентного алгоритма, с типом фронтального алгоритма «candidates», с 1 потоком для фронтального алгоритма:

*java -jar ProductionResources ALG OWN\_ALPHA production.xml orders.xml res.xml 5 10 50 4 candidates 1*

**Собственный ранцевый алгоритм**

Запуск ранцевого алгоритма с входными данными о производстве из файла production.xml, входными данными о заказах из файла orders.xml, вывод результата в файл res.xml, с 5 запусками для выбора лучшего, общим бюджетом на генерацию альтернативностей 50, с бюджетом на пересчёт мощностей производства 3, с типом фронтального алгоритма «candidates», с 1 потоком для фронтального алгоритма:

*java -jar ProductionResources ALG OWN\_BACKPACK production.xml orders.xml res.xml 5 50 3 candidates 1*

**Тестирование на возможность выполнения заказов на производстве**

Запуск тестирования на физическую возможность выполнить заказы на производстве с входными данными о производстве из файла production.xml, входными данными о заказах из файла orders.xml:

*java -jar ProductionResources TEST POSS production.xml orders.xml*

**Тестирование на возможность физически уложить представленный план заказов на производстве**

Запуск тестирования на возможность физически уложить представленный план заказов на производстве с входными данными о производстве из файла production.xml, входными данными о заказах из файла orders.xml, входными данными о плане укладки заказов из файла res.xml:

*java -jar ProductionResources TEST REAL production.xml orders.xml res.xml*

**Сравнительное тестирование двух планов укладки заказов по производству**

Запуск сравнительного тестирования двух планов укладки заказов по производству с входными данными о заказах из файла orders.xml, входными данными о первом плане укладки заказов из файла res1.xml, втором плане укладки заказов из файла res2.xml:

*java -jar ProductionResources TEST COMP orders.xml res1.xml res2.xml*

**Тестирование эффективности базового алгоритма на предоставленном списке пар производство-заказы с выводом результатов**

Запуск тестирования эффективности базового алгоритма на предоставленном списке пар из папки folder, содержащей 5 пар производства-заказов, с сохранением результатов в таблицу table.csv, с 5 запусками для усреднения результатов, с типом фронтального алгоритма «candidates», с 1 потоком для фронтального алгоритма:

*java -jar ProductionResources TEST BASIS BASE folder 5 table.csv 5 candidates 1*

**Тестирование эффективности собственного градиентного алгоритма на предоставленном списке пар производство-заказы с выводом результатов**

Запуск тестирования эффективности собственного градиентного алгоритма на предоставленном списке пар из папки folder, содержащей 5 пар производства-заказов, стартовой генерацией 10 распределений альтернативностей, общим бюджетом на генерацию альтернативностей 50, с сохранением результатов в таблицу table.csv, с 5 запусками для усреднения результатов, c 4 потоками для градиентного алгоритма, с типом фронтального алгоритма «candidates», с 1 потоком для фронтального алгоритма:

*java -jar ProductionResources TEST BASIS OWN\_ALPHA folder 5 10 50 table.csv 5 4 candidates 1*

**Тестирование эффективности собственного ранцевого алгоритма на предоставленном списке пар производство-заказы с выводом результатов**

Запуск тестирования эффективности собственного ранцевого алгоритма на предоставленном списке пар из папки folder, содержащей 5 пар производства-заказов, общим бюджетом на генерацию альтернативностей 1000, с бюджетом на пересчёт мощностей производства 3, с сохранением результатов в таблицу table.csv, с 5 запусками для усреднения результатов, с типом фронтального алгоритма «candidates», с 1 потоком для фронтального алгоритма:

*java -jar ProductionResources TEST BASIS OWN\_BACKPACK folder 5 1000 3 table.csv 5 candidates 1*

**Сравнительное тестирование двух таблиц с результатами тестирования эффективности алгоритмов**

Запуск сравнительного тестирования двух таблиц с результатами тестирования эффективности алгоритмов. Сравниваются алгоритмы, информация о запусках которых находится в таблицах table1.csv и table2.csv. Результаты записываются в таблицу result.csv:

*java -jar ProductionResources COMP\_RESULT\_TABLES table1.csv table2.csv result.csv*

**Вывод справочной информации**

Вывод справочной информации:

java -jar ProductionResources HELP

|  |  |
| --- | --- |
|  |  |