

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	6
ГЛАВА 1. ПЛАНИРОВАНИЕ И ОПТИМИЗАЦИЯ СБОРОЧНЫХ ПРОЦЕССОВ В ТЯЖЕЛОМ МАШИНОСТРОЕНИИ .....	10
1.1. Подходы к организации сборочных линий .....	10
1.2. Обзор алгоритмов оптимизации процессов сборочных линий	20
1.2.1. Балансировка сборочной линии .....	20
1.2.2. Теория расписаний в современной промышленности.....	24
1.2.3. Генетические алгоритмы производственного планирования	26
1.3. ВЫБОР МЕТОДА РАЗРАБОТКИ АЛГОРИТМА ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ.....	29
1.4. Постановка цели и задач .....	32
Выводы по главе 1 .....	33
ГЛАВА 2. МАТЕМАТИЧЕСКОЕ ИССЛЕДОВАНИЕ ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧИ ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ .....	34
2.1. ФОРМАЛИЗАЦИЯ ЗАДАЧИ.....	35
2.2. ОЦЕНКА ПРИМЕНИМОСТИ ПОДХОДОВ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К РЕШЕНИЮ ПОСТАВЛЕННОЙ ЗАДАЧИ .....	37
2.3. ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ ПЛАНИРОВАНИЯ .....	40
Выводы по главе 2 .....	42
ГЛАВА 3. РАЗРАБОТКА АЛГОРИТМА ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ.....	44
3.1. РАЗРАБОТКА ГЕНЕТИЧЕСКОГО АЛГОРИТМА ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ.....	46
3.1.1. Имитационная модель производственных ресурсов и производственное планирование .....	51
3.1.2. Повышение эффективности работы имитационной модели ..	54
3.1.3. Выбор гиперпараметров генетического алгоритма .....	57

3.2. АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ .....	62
Выводы по главе 3 .....	73
ЗАКЛЮЧЕНИЕ.....	74
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	77

## ВВЕДЕНИЕ

Сегодня науки об организации управления и планирования производственной деятельности приобретают все большее значение наряду с интеллектуальными подходами к планированию. Однако появление производственных объектов, планирование, эксплуатация и реорганизация которых осуществляется в соответствии с передовыми научными разработками, до сих пор не является повсеместным. Причиной этому является высокая сложность перенесения теоретических подходов на практику, заключающаяся в большом количестве особенностей, которые накладываются реальным производством.

Нынешний темп развития машиностроительных предприятий позволяет проследить тенденцию к объединению задач производственного планирования и организации управления промышленным предприятием. Целью такой интеграции является необходимость сделать основным предметом развития не только разработку технологий производства и проектирования новейших изделий, но и процесс интеграции результатов на реальные производства и последующую максимизацию эффективности эксплуатации промышленного объекта.

Отдельной задачей выделяют планирование, управление и оптимизацию процессов сборочных производств. Это обусловлено постоянным ростом научно-технического уровня сборочного производства, которое на сегодня характеризуется самым крупным потенциалом к развитию в сравнении с другими этапами машиностроительного производственного процесса. Сборочные процессы распространены повсеместно, фигурируют как в проверенных десятилетиями, так и в новейших технологиях производства практически во всех промышленных областях. В связи с этим сборка часто становится предметом частных исследований на базе производственных комплексов, поэтому сегодня сборочные процессы нормированы, их описание формализовано, для их автоматизации разрабатывается все более новое

оборудование. Однако алгоритмы оптимизации сборочных процессов развиваются медленнее остальной научно-технической базы [1].

Различные решения задач формирования, описания новых подходов и алгоритмов для решения задач оптимизации сборочных линий предлагаются с момента начала распространения этих технических средств в массовом производстве. При этом работы, проводимые различными производственными, научными и проектными площадками централизованно не координируются, что приводит к вынужденному дублированию решений многих сходных и даже одинаковых задач. Сегодня эта проблема приводит промышленность к необоснованно большому разнообразию программного обеспечения (ПО), нацеленного на оптимизацию сборочных процессов. Возникает потребность в наличии современного и универсального решения, которое учитывало бы современные инженерные разработки узко специализированной сферы.

В тяжелом машиностроении уровень автоматизации сборочных процессов относительно невысок [1]. Частичная автоматизация отдельных сборочных операций повышает производительность труда лишь на 5-6 % в год, тогда как внедрение программных комплексов дает возможность обеспечить рост производительности труда в 2-2,5 раза.

Наиболее актуальной задачей сегодня становится автоматизация мелко- и среднесерийного машиностроения. Эта форма выпуска изделий является сейчас наиболее востребованной ввиду необходимости кастомизации и индивидуализации изделий под требования рынка практически на каждой серии выпуска изделий. Соответствие этим требованиям серьезно влияет на длительность производственных процессов – частые и длительные переналадки высокоточного оборудования приводят к увеличению длительности производственных процессов более чем в два раза [2]. Поэтому придание мелко- и среднесерийному производству преимуществ массового (ритмичность, высокий темп выпуска изделий) без потерь гибкости технологий производства является проблемой предметной области.

Основная проблема, решавшаяся в рамках проведенных исследований, заключалась в необходимости повышения производительности сборочной линии путем применения для планирования её деятельности алгоритма оптимизации. При решении этой задачи рассмотрены существующие технические решения организации конвейерного производства, применяющиеся как для мелко- и среднесерийных, так и для массовых производств, исследованы алгоритмы планирования процессов сборки изделий. В качестве объекта исследования рассмотрено мелко- и среднесерийное машиностроительное производство, где сборочные процессы осуществляются с использованием переналаживаемых сборочных линий или конвейеров. Этот выбор объясняется востребованностью программных решений в данной сфере промышленности в связи с повсеместным ростом спроса на производимую ими продукцию.

Особенности, присущие процессам тяжелого машиностроения, часто противоречат идеям прогрессивных методологий и подходов к оптимизации и организации труда. Проведенные исследования существующих решений организации сборочных линий и алгоритмов оптимизации процессов, протекающих на них, показали, что наиболее актуальным решением является использование генетических алгоритмов.

На основании выводов, полученных в процессе математической формализации и доказательства её NP-трудности, фактических производственных особенностях и ограничениях, установленных взаимосвязей разработан алгоритм, минимизирующий плановую длительность такта сборочной линии, обеспечивающий возможность планирования и оптимизации длительности необходимых вспомогательных рабочих операций (дополнительная наладка, контрольные операции), проведения наладки сборочной линии за короткое время и при минимальных затратах, а также исключение возможности проведения избыточных вспомогательных операций.

Таким образом одной из основных задач, поставленных в рамках работы, является применение полученных результатов для последующей модернизации существующих систем планирования деятельности предприятия.

Преимущественное значение для разработки алгоритма оптимизации сборочных процессов имеет поиск решения проблем планирования сборки на тяжелых машиностроительных предприятиях, определение подходов к созданию соответствующих частных моделей, учитывающих индивидуальные особенности производств. Обособленной задачей, сопряженной с разработкой алгоритма оптимизации, является проблема интеграции системы с уже существующей системой планирования.

Научная работа выполняется в рамках заключенного договора с фондом поддержки проектов национальной технологической инициативы (фонд НТИ), реализация, разработка и тестирование прототипа выполняется в рамках заключенного договора с АО "Диаконт".

# ГЛАВА 1. ПЛАНИРОВАНИЕ И ОПТИМИЗАЦИЯ СБОРОЧНЫХ ПРОЦЕССОВ В ТЯЖЕЛОМ МАШИНОСТРОЕНИИ

## 1.1. ПОДХОДЫ К ОРГАНИЗАЦИИ СБОРОЧНЫХ ЛИНИЙ

По своему характеру все виды организации сборочных процессов можно разделить на поточную и непоточную формы организации [1]. В рамках работы был рассмотрена поточная форма организации сборки, как наиболее распространенная на сегодняшний день.

При поточной сборке (рисунок 1) каждый сотрудник закреплен за единственным рабочим местом и выполняет только одну операцию или группу операций. Объект сборки в свою очередь перемещается между рабочими местами согласно последовательности операций, предусмотренной технологией. Такая последовательность на предприятии закреплена специализированным документом – маршрутной картой изделия.

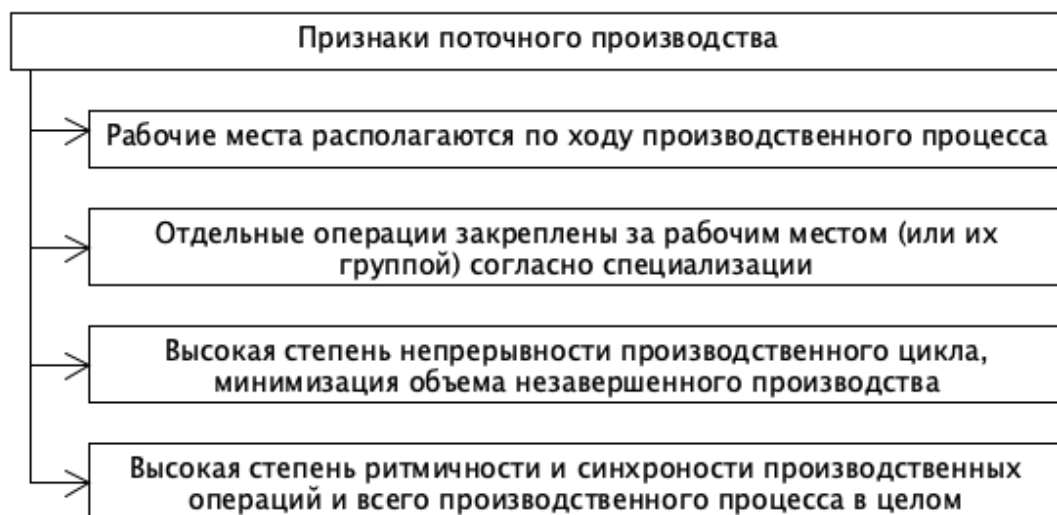


Рисунок 1– Признаки поточного производства

Поточной линией сборки или сборочной линией называют совокупность рабочих мест, которые предназначены для сборки полуфабрикатов или конечных изделий и располагаются в порядке выполнения операций технологического процесса, предусмотренного маршрутной картой изделия. Рабочее место на производственной линии принято называть рабочим постом

или рабочей станцией. Ключевым параметром, характеризующим поточную сборку, является ее такт (рисунок 2) — периодичность сдвига сборочной линии, длительность между выходом с поточной линии двух изделий. Промежуток времени с момента начала обработки заготовок на первой технологической операции МК до выхода собранного изделия с линии принято называть циклом сборочной

линии.

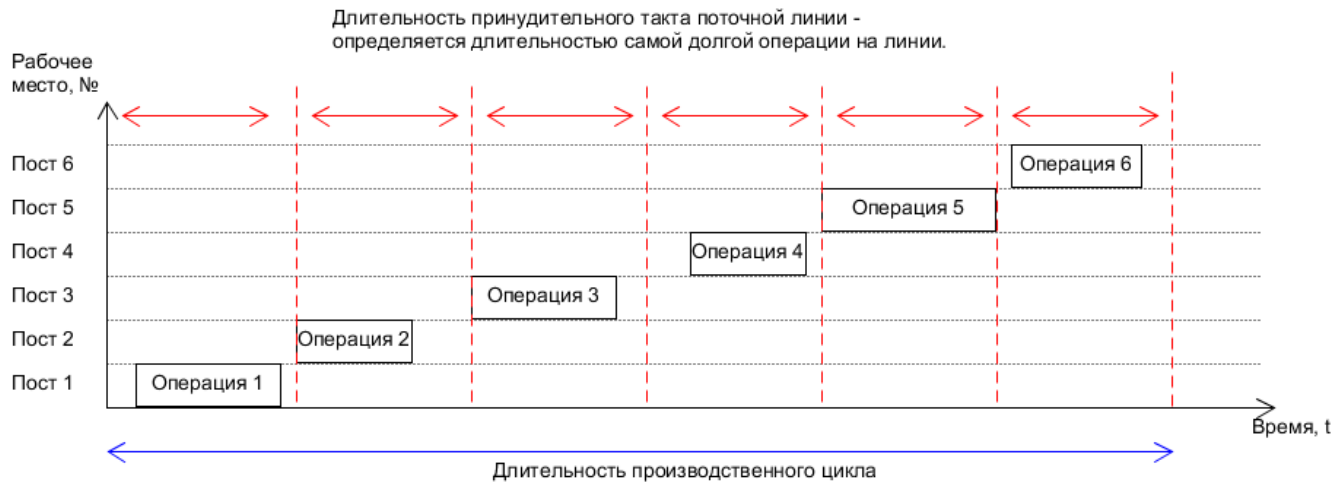


Рисунок 2 – Связь такта и цикла производства на поточной линии и принудительным тактом

Такт производственной линии, необходимый для достижения планового выпуска продукции, рассчитывается по формуле:

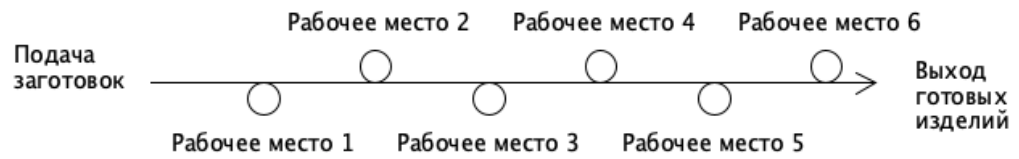
$$t_{\text{сб.л.}} = \frac{T_{\text{эф.}} - t_{\text{пер.}}}{B_{\text{пл.}}}, \quad (1)$$

где  $T_{\text{эф.}}$ , мин. – эффективный фонд рабочего времени планового периода,  $B_{\text{пл.}}$ , шт. – плановый выпуск изделий со сборочной линии, а  $t_{\text{пер.}}$ , мин. – суммарная длительность регламентированных перерывов в течение планового периода [1].

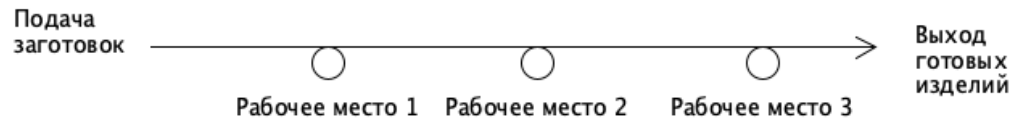
На практике сборочные линии могут отличаться своей конфигурацией. В серийных производства наиболее распространены линии с последовательной, параллельной, круговой, s-образной передачей заготовки между рабочими местами (рисунок 3).



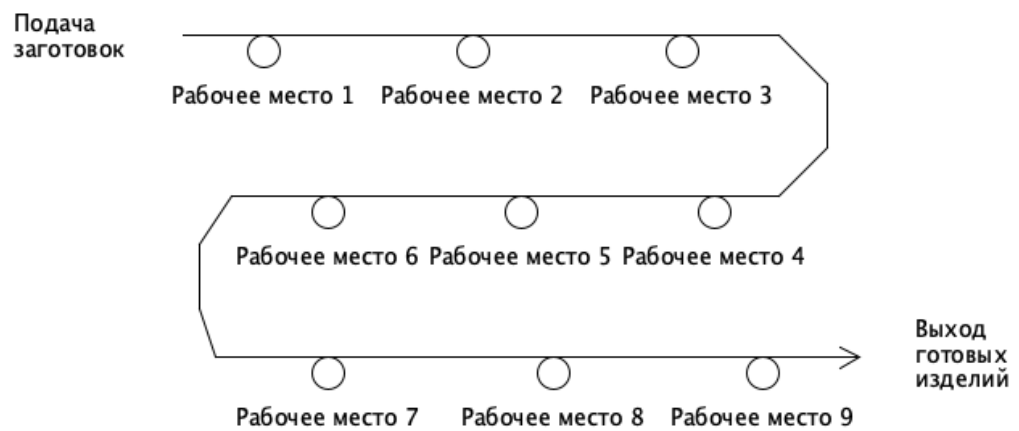
а) Параллельная передача



б) Последовательная передача



в) S-образная передача



г) Круговая передача

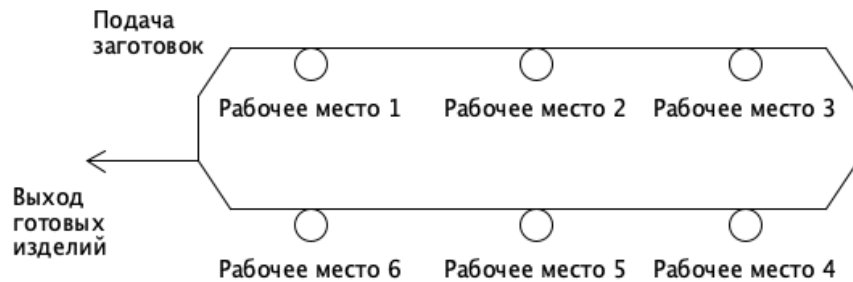


Рисунок 3 – Конфигурации сборочных линий: а) - параллельная передача, б) - последовательная передача, в) s-образная передача, г) круговая передача [1]

Конфигурация сборочной линии может накладывать дополнительные ограничения на особенности оптимизации сборочных линий, которые учитываются в существующих алгоритмах оптимизации сборочных процессов,

о чем подробно рассказано в разделе «1.2 Обзор алгоритмов оптимизации процессов сборочных линий».

Любая сборочная линия представляет собой поточную линию, специализирующуюся на сборке изделий и узлов. Поточные линии отличаются по принципу организации – сложившаяся классификация приведена на рисунке 4.

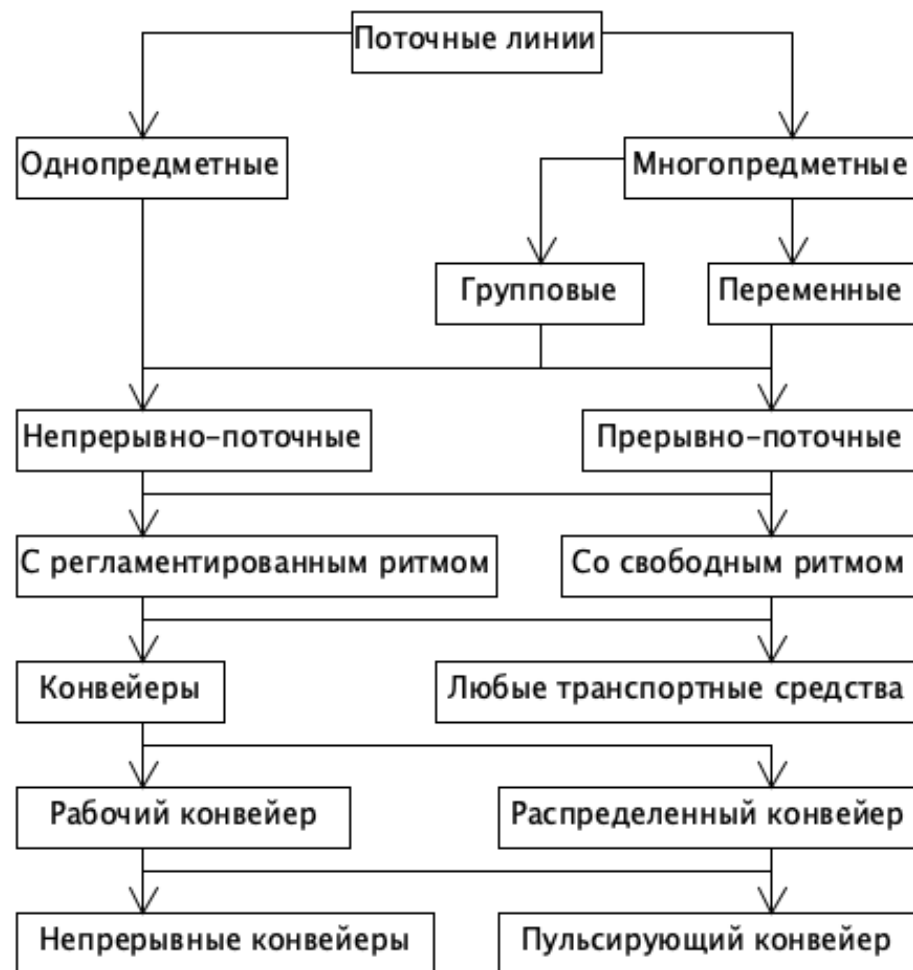


Рисунок 4 – Классификация поточных линий [2]

Поточные линии в первую очередь подразделяются по номенклатуре обрабатываемых изделий на однопредметные и многопредметные [2].

Однопредметными линиями обрабатываются (собираются) изделия одного наименования, с одинаковыми параметрами и единой спецификацией в течение длительного периода времени.

Многопредметными называют поточные линии, позволяющие одновременно или последовательно обрабатывать некоторый набор изделий различных наименований, но с похожими конфигурациями или технологическими этапами сборки. Многопредметные линии в свою очередь разделяют на:

- групповые: различные изделия обрабатываются в рамках групповой технологии и без переналадки оборудования;
- переменные: прикрепленные к ним различные типы технологически схожих изделий изготавливаются попеременно, с использованием через определенные временные интервалы универсального технологического оборудования (с переналадкой оборудования или без нее). Обычно на таких линиях после обработки одной партии изделий происходит переналадка оборудования, которая не вызывает больших потерь времени.

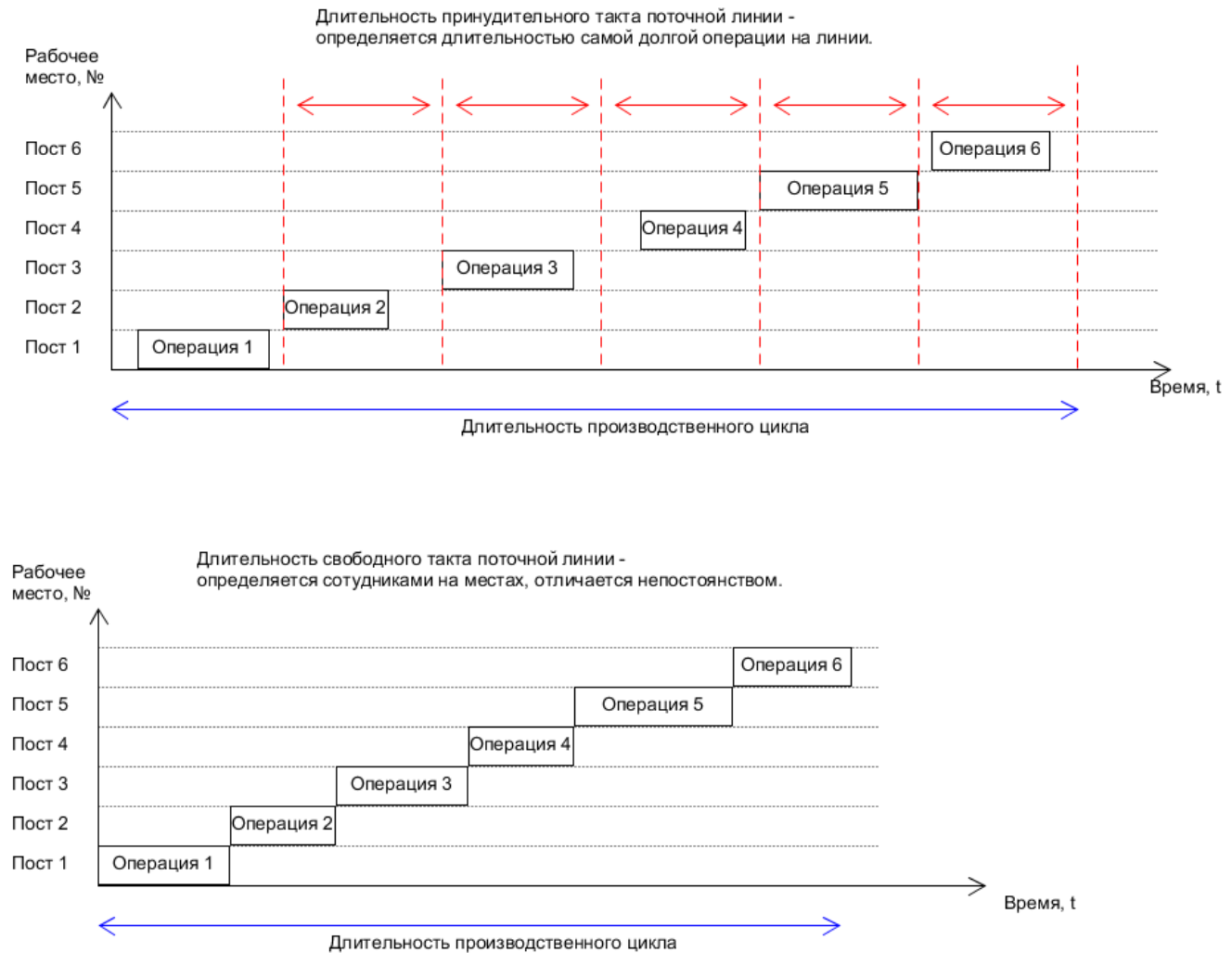
В зависимости от характера движения изделия по операциям поточные линии могут быть непрерывными и прерывными.

Непрерывной называется поточная линия, где изделие находится либо в процессе сборки, либо перемещается с предыдущей операции на следующую. Такие поточные линии нормы длительностей выполнения операций одинаковы или кратны друг другу и такту [2].

Если в процессе обработки возникают какие-либо перерывы в обработке или перемещении изделий между рабочими местами, производство называют прерывно-поточным. Перерывы в рамках работы таких линий обусловлены неполной синхронизацией операций, неравенством и не кратностью производительности оборудования на отдельных операциях. Поэтому изделие по окончании обработки на предыдущем рабочем месте может пролеживать до начала обработки на следующей – так образуются межоперационные оборотные заделы. Прерывные потоки допустимы на предприятиях, где невозможно полностью уравнивать нормы длительности выполнения операций или достичь кратности в связи с технологическими, техническими или организационными особенностями.

Такт сборочного процесса может определяться как сборочной линией, так и сотрудником, осуществляющим операцию (рисунок 5). В первом случае длительность такта называют принудительной – на нее нельзя повлиять посредством внешнего вмешательства, не спровоцировав длительный процесс подготовки и переналадки. Сборочные линии, где длительность процессов определяется длительностью работы сотрудников над операцией, характеризуются свободным тактом [1]. Такой способ формирования такта производства широко применяется на современных мелко- и среднесерийных предприятиях, поскольку обеспечивает сборочным линиям ряд преимуществ:

- гибкость специализации сборочной линии, которая обеспечивается возможностью применения её для сборки схожих изделий;
- возможность изменять характеристики изделий с высокой регулярностью, обеспечивая высокую регулярность выпуска обновленных изделий;
- возможность определения проблемных и технологически неоптимальных операций посредством сбора статистики для последующего совершенствования технологических процессов.



**Рисунок 5 – Разница между свободным и принудительным тактом производства**

Однако, применение такого подхода к дискретизации длительности производственных процессов приводит к возникновению ряда проблем в процессе планирования и оптимизации сборочных процессов в тяжелой промышленности. В первую очередь это обусловлено следующими особенностями:

- процессы сборки различных деталей в тяжелой промышленности характеризуются многократным отличием трудоемкости работ;
- современные сборочные процессы, распространенные в тяжелой промышленности, разнообразны по своему характеру и требуют длительной отладки оборудования и регулярном контроле точностных

характеристик заготовок, что приводит к увеличению длительности производственного цикла;

- получение единого производственного потока требует тщательного планирования и оптимизации плана работы сборочной линии для каждого нового производственного цикла.

Наиболее эффективным решением для вышеперечисленных проблем может стать организация сборки со сведением к непрерывной потоковой сборочной линии, которая чаще всего применяется при выпуске продукции в массовом производстве. Непрерывность процесса сборки достигается с помощью синхронизации продолжительности выполнения каждой операции сборочной линии в рамках одного производственного цикла. Синхронные операции – это так согласованные во времени операции, когда длительность каждой операции равна такту сборочной линии, при обеспечении полной загрузки оборудования и рабочих мест.

Полная синхронность операций возможна в случае организации работы автоматизированного массового производства, тогда как для мелко- и среднесерийных производств к синхронности можно приближаться при помощи оптимизаций планирования сборочных процессов, но не достигать её (при существующем технологическом уровне развития в области). Существуют следующие способы увеличения степени синхронности:

- дифференциация – деление операций на мелкие части;
- концентрация – группировка мелких операций в одну;
- комбинирование – группировка операций по рабочим местам.

Такой подход к планированию сборочных процессов эффективно применяется для производств, где номенклатура выпуска и конфигурации изделий условно постоянны на сколько-нибудь продолжительном промежутке времени. Чем менее продолжительным будет период планирования – тем сложнее осуществить планирование операций.

Метод сведения к непрерывному потоку, применяемый для мелко- и среднесерийных предприятий, формирует прерывно-поточную линию,

характеризующуюся наличием минимально допустимых перерывов между операций, необходимых ввиду технологических особенностей. Это позволяет достичь следующих преимуществ:

- максимизировать специализацию рабочих мест и обеспечить строгое разделение труда, что неизменно приводит к росту производительности и создает предпосылки для дальнейшей автоматизации ручных операций;
- стабилизировать технологический процесс и формализовать периодичность выпуска изделий;
- минимизировать и достичь условно-постоянных значений объема незавершенного производства с устранением нерабочих перерывов в работе;
- увеличить число рабочих операций, выполняемых в рамках рабочей смены и максимизировать насыщенность рабочих интервалов сотрудников;
- оптимизировать длительность цикла сборки и уменьшить количество необходимых производственных площадей.

В серийном производстве широко распространены сборочные линии с определенным ритмом сборки и комплексной автоматизацией основных и вспомогательных работ сотрудников, позволяющих приблизить производство даже мелкой серии изделий к прерывно-поточной производства. Для этого применяется принцип дифференциации операций в рамках МК, что облегчает организацию сборки поточным методом. Он предполагает дробление операций на более мелкие, что позволяет планировать процессы детально и подробно. При этом для увеличения автоматизации сборочного процесса операции концентрируются на одном рабочем месте. Метод, реализующийся в таких условиях, нельзя считать оптимальным, поскольку он имеет следующие особенности:

- высокая длительность такта выпуска, а значит большое число сборочных операций, приходящихся на одного сотрудника, которые выполняются им в течении одного такта сборочной линии;

- наличие большого числа пусконаладочных, вспомогательных операций, которые обусловлены несоответствием технологий изготовления собираемых на одной линии изделий;
- в тяжелой промышленности – резкие скачки трудоемкости работ в рамках одной рабочей смены, связанных с разницей габаритов отдельных деталей и сборочных узлов.

При планировании и оптимизации процессов сборки необходимо учитывать и ряд факторов, накладываемых реальным производством. Например, возникают сложности планирования производственного процесса с учетом фактического сменного графика сотрудников, регламентированных перерывов рабочей деятельности. Необходимо отметить, что рабочие операции при этом характеризуются переменной длительностью выполнения, она зависит от количества сотрудников, фактически привлеченных к работам. Также переход изделий от одной операции к другой на рабочих местах поточных линий образует задел - совокупность заготовок и изделий, характеризующихся разной степенью завершенности и предназначенных для обеспечения бесперебойной и непрерывной работы линии.

Оперативное планирование сборочного производства детализирует процессы сборки и деятельность сотрудников на краткий срок. Основным результатом работы алгоритма планирования и оптимизации сборочной линии является оперативный план (ОП). Помимо оптимальности, полученный план должен соответствовать ряду параметров. В первую очередь ОП учитывает особенности графика работы предприятия, производственный календарь и другие параметров, задаваемыми пользователем, а также обеспечивает сбор информации о запланированных объемах вовлечения ресурсов (как трудовых, так и технологических) в производственные процессы и длительностях выполнения отдельных операций, заделов, резервов и производственного цикла сборки в целом.

Таким образом одной из наиболее актуальных проблем в области разработки ПО для предприятий тяжелого машиностроения является разработка



алгоритма планирования и оптимизации для сборочных линий со свободным тактом, который позволит реализовать задачу сведения к непрерывному потоку производства, т.е. сформировать в рамках мелко- и среднесерийных предприятий сборочный процесс близкий к поточной форме организации.

Далее рассмотрим алгоритмы планирования и оптимизации сборочных линий, которые применяются на практике для решения актуальных производственных задач.

## **1.2. ОБЗОР АЛГОРИТМОВ ОПТИМИЗАЦИИ ПРОЦЕССОВ СБОРОЧНЫХ ЛИНИЙ**

Задача оптимизации работы сборочных линий стоит перед производствами с момента запуска первого конвейерного производства в начале XX века. С тех пор было решено множество общих и частных задач оптимизации, наиболее прогрессивные решения, предложенных в начале XXI века рассмотрены в этом разделе.

### **1.2.1. Балансировка сборочной линии**

Несмотря на то, что работы в области оптимизации конвейерных производств ведутся с начала прошлого столетия, задача была формализована математически лишь в 1955 году [3], где в качестве основной проблемы рассматривался процесс распределения сборочных операций по станциям конвейера, иными словами его конфигурации. Проблема рассматривалась с точки зрения поиска такого сбалансированного распределения работ по станциям конвейера, чтобы длительности работ были условно нормируемы. Отсюда появилось название для целого комплекса частных задач оптимизации – балансировка сборочной линии (ALBP от англ. Assembly Line Balancing Problem).

Необходимость поиска частных случаев решения задачи состояла в том, что для применения на практике полученных результатов необходимо учитывать массу ограничений и условий, накладываемых реальным производством. Более того, в связи со стремительным развитием промышленности, внешние условия и

ограничения также характеризуются непостоянством. Поэтому изучение математическими методами задачи балансировки сборочной линии породило множество различных подходов к решению. Спустя четверть века исследований они были объединены в 1986 г. в обзоре [4], где задача ALBP была рассмотрена связно с девятью условиями, которые позволяют применять наработки для внедрения на реальные предприятия. Она получила название Simple Assembly Line Balancing Problem (SALBP) (рисунок 6).

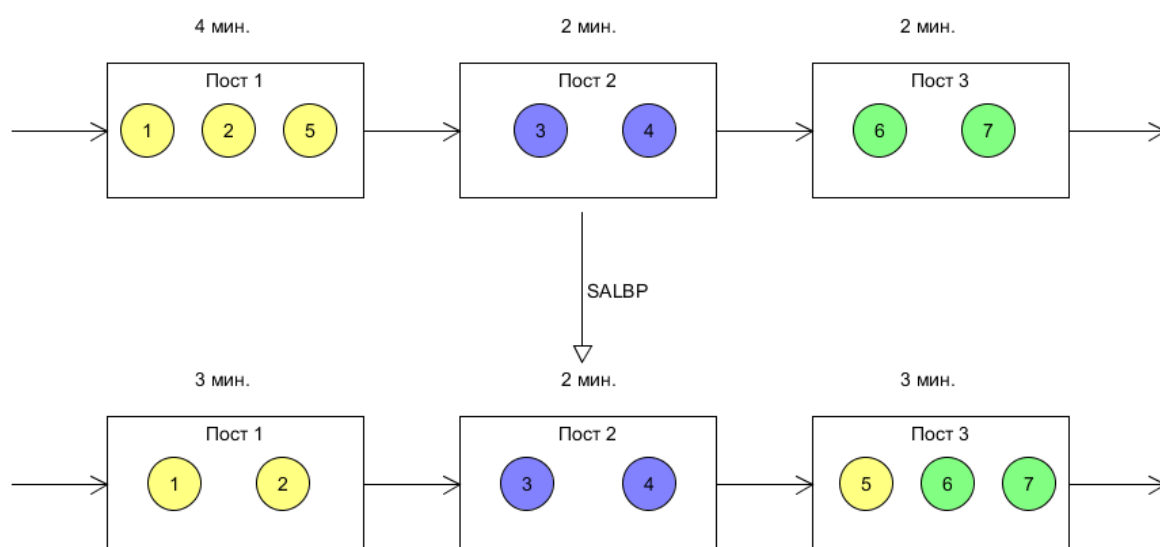


Рисунок 6 – Пример результатов нормировки длительности такта поточной линии с использованием метода SALBP [4]

Сегодня существует множество различных постановок и модификаций задачи ALBP, как и подходов к решению. Рассмотрим лишь те из них, условия и ограничения которых соответствуют поставленной в данной работе задаче – оптимизация реальной длительности выполнения производственных операций с учетом погрешности данных о фактической длительности операций, а также изменчивости длительности в зависимости от количества привлеченных к работе сотрудников.

Рассмотрим SALBP, как наиболее изученное математическое описание проблемы. Она может быть сформулирована так: необходимо назначить сборочные операции на рабочие посты упорядоченного множества рабочих станций ( $S_1, S_2, \dots, S_m$ ) так, чтобы длительность такта  $c$  была минимальной, при

сохранении числа рабочих станций  $m$ . Согласно статье [4], на модель сборочной линии накладываются следующие ограничения:

- входные данные однозначно определены;
- каждая операция атомарная и не может быть перераспределена на несколько станций;
- последовательность запуска в работу операций ограничена орграфом  $G = (V, A)$ ;
- совокупность операций необходимо выполнить за такт  $c$ , который характеризуется одинаковым значением для каждой станции;
- рабочие станции  $(S_1, S_2, \dots, S_m)$  оборудованы и укомплектованы так, что каждая может выполнить любую операцию;
- станции последовательны, приспособления для подачи заготовок из рассмотрения исключаются;
- сборочная линия не требует переналадки;
- количество станций  $m$  фиксировано и является начальным значением.

Тогда эффективность  $E$  сборочной линии определяется по формуле

$$E = \frac{t_{sum}}{(m * c)}, \quad (2)$$

где  $t_{sum} = \sum_{i=1}^n t_i$  – сумма длительностей  $t_i, i \in V$  всех сборочных операций.

Следует отметить, что в частном случае задача SALBP также является NP-трудной, когда  $m = 2$  и заданный орграф  $G = (V, A)$  не включает дуг ( $A = \emptyset$ ). Доказательство NP-трудности такого случая задач SALBP следует из того, что к нему сводится NP-трудная задача  $\max P2 || C$ , где требуется построить оптимальное по скорости выполнения расписание обслуживания  $n$  требований на двух параллельных приборах. Доказательство NP-трудности обыкновенных задач балансировки сборочных линий приведено и в [5].

Отдельно стоит отметить и стохастическую задачу ALBP. Здесь длительности производственных операций рассматриваются как случайные величины с известными функциями распределения.

Стохастическую задачу ALBP можно сформулировать следующим образом: задано конечное множество операций  $V = \{1, 2, \dots, n\}$ , длительность выполнения работ над каждой из операций описывается случайной величиной, для которой задается закон распределения вероятностей и оргграф  $G = (V, A)$ , который частично определяет порядок на множестве сборочных операций  $V^1$ .

В [6–11] приведено обобщенное описание стохастической задачи SALBP, сформулированное при снятии ограничений, накладываемых на входные данные. Напротив, критерий детерминированности заменяется на стохастический критерий. Таким образом формулировка задачи несколько меняется: в задаче минимизировалось математическое ожидание  $E_c$  длительности такта  $c$  при заданном числе рабочих станций  $m$ .

Стохастический подход к задаче SALBP существенно сложнее аналогичных по точности и интегрируемости детерминированных подходов. Они не полностью соответствуют ряду реальных производств, на это есть две основные причины:

- на практике крайне редко удастся заполучить достаточно информации позволяющей оценить достоверность распределения вероятностей случайной длительности  $x_i$  каждой операции  $i \in V$ ;
- даже если достоверность распределений подтверждена фактическими данными, они остаются актуальными только если сборочная линия функционирует в неизменных условиях.

Входные данных задачи ALBP могут быть заданы только в виде пределов возможных значений. Их неопределенность может быть выражена в виде нечетких чисел, с целью минимизации возможных погрешностей, связанных в свою очередь с неточностью входных данных. Так, например, длительности выполнения сборочных операций в задачах ALBP зачастую представляют в виде нечетких множеств.

---

<sup>1</sup> Частичное определение подразумевает что  $A \neq \emptyset$ , иными словами, между операциями существуют различные зависимости, однако при этом  $A \leq V - 1$ , что означает что в множестве могут быть операции, независимые от других.

В такой нечеткой задаче ALBP задано конечное множество сборочных операций  $V = \{1, 2, \dots, n\}$ ; длительности выполнения сборочных операций представляются нечеткими множествами; задан орграф  $G = (V, A)$ , который определяет частичный порядок на множестве  $V$  сборочных операций [12–14].

В статьях [11, 13] приведены пример решения задачи ALBP с нечеткими представлениями длительностей производственных операций. В качестве инструмента решения применяется генетический алгоритм минимизации длительности такта. В статье [13] приведено решение задачи ALBP с нечеткими длительностями сборочных операций с вычислением числового параметра.

Особенно актуальным является решение, приведенное в статье [14], где авторами рассмотрена смешанно-модельная конфигурация сборочной линии (позволяет осуществлять сборку нескольких вариантов одного изделия на одной сборочной линии). Рассматриваются три целевые функции, которые конфликтуют одна с другой. Для решения такой задачи ALBP предлагается подход, основанный на математическом программировании с нечеткой целевой функцией.

### **1.2.2. Теория расписаний в современной промышленности**

Теория расписаний представляет собой раздел дискретной математики, который образовал математическую дисциплину, рассматривающую решения ряда простейших задач календарного планирования. Совокупность этих решений сформировала научные, математические, основы календарного планирования, которыми и сегодня руководствуются предприятия при планировании своей деятельности. Наиболее значимой работой в области прикладной теории расписаний являются [15, 16], где приводятся доказательства NP-полноты для относительно простых постановок задачи теории расписаний. Авторы предлагают применять для прикладных решений, использовать эвристически найденные решения. Как наиболее простой случай, ими рассматривается схема ветвления с функциями предпочтения.

Авторы [16] предлагают прикладное, применимое на практике решение, основанное на эвристических вычислениях, пригодное для автоматизированных систем управления предприятиями (АСУП). Приведенные решения охватывают как техническую подготовку производства, так и создание баз данных технологической, плановой и учетной информации на предприятии

Пусть существует производственный участок, где одинаковые рабочие посты объединены как одна машина, всегда доступная для выполнения работ. Единичное изделие или партия будут представлены в качестве предмета труда с заданным маршрутной картой. Наиболее распространенный вариант представления формализованной задачи теории расписания [15] для планирования производственных работ в таком случае имеет следующий вид.

Предметы труда из множества  $I = \{i | 1 \leq i \leq n\}$  необходимо обработать машинами, принадлежащими множеству  $Q = \{q | 1 \leq q \leq m\}$ . Обработка состоит в выполнении машинами из множества  $Q$  операций маршрутной карты  $I_i = \{iq_j | 1 \leq j \leq j_i\}$ , которая может отличаться в зависимости от предмета труда. Выполнение параллельных операций не исключается из рассмотрения, каждая из них длится  $t_{iq_j} > 0$  и выполняется конкретной машиной  $q$ . Поэтому множество операций  $I_i = \cup_{i \in I} I_i$  может быть разбито на  $m$  классов  $I_q$  ( $1 \leq q \leq m$ ). Каждый такой класс представляет из себя перечень операций, выполняемые каждой конкретной машиной и содержит не менее двух операций, которые не могут быть прерваны или выполняться одной машиной. Задача состоит в том, чтобы сформировать порядок выполнения операций для каждой из машин  $m$  так, что:

- количество переналадок минимально;
- общая длительность выполнения работ минимизирована;
- выполняется одно или оба условия, приведенные выше.

Зачастую на практике решаются задачи для иных условий и ограничений: возможность проведения параллельных работ с одинаковой или разной скоростью, работы распределяются по машинам согласно последовательно

(конвейерным способом) или для каждого предмета труда последовательность задается индивидуально, предмет труда может обрабатываться машинами беспорядочно, но на каждой по одному разу и т.д. Сами работы также часто наделяются дополнительными характеристиками, накладывающими собственные ограничения. Это чаще всего условия длительности работ – время работы, момент появления, момент окончания, дедлайн.

Для решения непосредственно этой задачи, а также её модификаций с различными дополнительными условиями и ограничениями, применяются различные подходы. Широко известны и применяются методы формирования расписания, по нижней оценке, жадный алгоритм, бинарный поиск по ответу, поочередное заполнение машин работами (« $P \mid pmtn \mid C_{max}$ »-задача) [15].

### 1.2.3. Генетические алгоритмы производственного планирования

Еще одной традиционной задачей в контексте производственного планирования является Flow Shop Scheduling Problem, сформулированная [17, 18]. Суть проблемы состоит в формировании такой последовательности изготовления деталей, партий, которая обеспечит минимальную длительность обработки совокупности всех этих партий. Существенное ограничение для такой постановки задачи заключается в обеспечении однотипности состава каждой партии изделий.

В работе [19] рассматривается частный случай решения такой задачи для планирования деятельности многопредметных поточных линий с применением генетических алгоритмов. Для минимизации суммарной длительности циклов производства нескольких партий необходимо найти такую последовательность запуска партий деталей на конвейер, чтобы суммарное время переналадки было наименьшим.

Целевая функция задачи оптимизации плана может быть представлена в следующем виде:

$$E = \sum_{i=2}^q w_{i-1,i} \rightarrow \min, \quad (3)$$

где  $q$  – количество различных партий;  $w_{i-1,i}$  – длительность переналадки между предыдущей  $(i - 1)$ -м и следующей  $i$ -ой партией.

В алгоритме [19] предполагается использование трех различных операторов мутации, инверсии и кроссинговера. Авторы приводят также и критерий результативности и эффективности алгоритма, для двух случаев решения задачи:

- когда оптимальный результат в пределе известен заранее;
- когда оптимальные результат, к которому необходимо приблизиться заранее не установлен; тогда в качестве оценки критерия результативности применяется средневзвешенное значение целевой функции:

$$\bar{E} = \frac{1}{N} \sum_{i=1}^k n_i E_i, \quad (4)$$

где  $N$  – количество экспериментов;  $k$  – количество найденных субоптимальных значений целевой функции;  $E_i$  – оптимальное или субоптимальное значение целевой функции;  $n_i$  – частота обнаружения значения целевой функции  $E_i$ .

В качестве эффективности генетического алгоритма авторы рассматривают скорость обнаружения алгоритмом оптимального или субоптимального решения. Значение критерия эффективности принимается исходя из средневзвешенного значения длительности проведения вычислений при заданном количестве экспериментов. Отметим, что алгоритм [19] показал высокую степень эффективности, поскольку длительность вычисления субоптимального решения несущественно мала, в сравнении с количеством проведенных для его поиска решений.

В работе [20] авторы применяют генетический алгоритм с целью планирования производственной деятельности и сбыта готовой продукции.

Постановка задачи, рассмотренная авторами, заключается в поиске оптимального производственного плана, при котором время, необходимое для производства всех изделий в требуемом объеме, будет минимальным, договорные обязательства выполнены в полно объеме, и запасы на складе не



превышают определенного значения незавершенного производства. Горизонт планирования – один месяц. Список заказов на производство обновляется динамически.

Целевая функция  $M$  – длительность производства всех изделий в требуемом объеме, будет выглядеть следующим образом:

$$M = \max(M_1, M_2, \dots, M_j, \dots, M_L) \rightarrow \min, \quad (5)$$

где  $M_j$  – суммарное время работы  $j$ -й производственной линии за плановый период;  $L$  – количество производственных линий.

$$M_j = \sum_{k=1}^D M_{jk}, \quad (6)$$

где  $M_{jk}$  – время работы  $j$ -й производственной линии за  $k$ -е сутки,  $D$  – количество рабочих дней, входящих в плановый период;  $M_{jk} \leq R$ , где  $R$  – рабочее время производственных линий (часов в рабочий день), то есть время работы  $M_{jk}$  включает лишь непосредственно рабочее время, без учета профилактических, вспомогательных работ.

$$M_{jk} = \sum_{i=1}^N \frac{x_{ijk}}{U_{ij}} + hV, \quad (7)$$

где  $x_{ijk}$  – переменная, характеризующая объем производства изделий  $i$ -й заказа на  $j$ -й линии за  $k$ -е сутки,  $U_{ij}$  – производительность линии,  $N$  – количество позиций номенклатуры заказа;  $h$  – дополнительные временные издержки, возникающие на линии при переналадке и смене типа сырья;  $V$  – количество мероприятий переналадки в сутках.

При этом  $x_{ijk} \geq b$ , где  $b$  – минимальный размер партии производства, обусловленный технологическими ограничениями и требованиями к качеству продукции, в случае, когда  $U_{jk}$  равняется нулю, производство изделий  $i$ -й номенклатуры на  $j$ -й линии невозможно;

Аналогично авторами статьи предлагается математически формализованное описание процесса управления размерами складских запасов.

Для реализации генетического алгоритма авторами создан массив кодов, с использованием которого можно закодировать каждое допустимое решение последовательностью символов. Решение, закодированное таким образом, принимается в качестве организма, в каждый параметр решения – хромосомы.

Дополнительно специально разработанным алгоритмом автоматически рассчитывается матрица загрузки каждой производственной линии и соответственно временные издержки, возникающие на линии при смене производственной номенклатуры и типа сырья.

То, насколько приспособленным является полученное решение, определяется целевой функции (3). Вычисленное значение целевой функции приспособленности принимается за значение витальности организма. Также предусмотрен ряд дополнительных процедур, которые реализуют стохастический выбор определенного решения для участия в турнире, осуществляют преобразование значения целевой функции организма в вероятность его выбора для кроссовера или перехода в следующее поколение.

Средний объем перебора составил – 409, эффективность работы алгоритма – 3,98. Таким образом, и в этом случае применения генетических алгоритмов, подход показал высокую эффективность.

### **1.3. ВЫБОР МЕТОДА РАЗРАБОТКИ АЛГОРИТМА ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ**

Рассмотренные выше подходы к оптимизации сборочных процессов на производстве показывают свою эффективность для решения определенных задач узкого профиля. В процессе подбора материалов для обзора и изучения специфики предметной области было установлено, что внедрение даже незначительных на первый взгляд изменений в процесс планирования сборочных операций влечет за собой значительное повышение финансовой и технологической эффективности производственных процессов. Зачастую применение алгоритмов оптимизации позволяет осуществлять планирование на

межцеховом уровне, или на уровне отдельных производств, что позволяет параллельно планировать выполнение этапов сборочных работ и увеличивает гибкость алгоритмов.

На базе проведенного обзора существующих решений в области оптимизации сборочных процессов далее приведено обоснование выбора технологии и метода решения поставленной задачи. Представим ее описание в развернутом виде, прежде чем переходить к сравнению методик решения.

Рассмотрим производственный участок с закрепленными за ним технологическими и трудовыми ресурсами. Технологические ресурсы представлены сборочными линиями, разделенными на рабочие посты (полностью укомплектованные для выполнения любых работ), которые могут обрабатывать продукцию разных типов. Сборочные линии могут как быть идентичны, так и отличаться друг от друга количеством постов, закрепленными за ними маршрутными картами изделий. Каждый пост сборочной линии может обслуживаться как фиксированным числом сотрудников, так и переменным. Для такой конфигурации сборочного производства необходимо решить задачу производственного планирования работ над реализацией  $N$  партий изделий разного типа с минимизацией длительности производственного цикла каждой партии изделий. Производственный план должен содержать в себе следующие данные:

- выбор сборочной линии, где будет произведена каждая партия;
- оптимальная последовательность выпуска партий производственным участком, обеспечивающая минимизацию производства совокупности партий;
- подбор оптимального числа сотрудников на каждый пост сборочной линии, где предусмотрено переменное их количество;
- длительности производственного цикла и такта для предложенных вариантов планирования.

Целью алгоритма является формирование плана, который обеспечивает функциональную полноту [21] системы планирования, в рамках которой будет использован, а также отвечает следующим требованиям:

- обеспечение планирования производства, которое утверждает план производства всех видов готовых изделий и их характеристики;
- обеспечение корректности технологии производства для каждого вида изделия путем соблюдения требований маршрутной карты изделия, с учетом всех технологических ограничений;
- обеспечение возможности планирования потребности в материалах на основе технологической карты изделий каждой выпускаемой партии;
- обеспечение планирования рабочего времени производственных мощностей, которые отвечают преобразование материалов в конечные изделия.

Решения, предлагаемые для задачи ALBP, не отвечают современным требованиям производства по ряду причин. В первую очередь, простейшие примеры решений предполагают множество упрощений и допущений, из-за чего их внедрение на реальные предприятия было и остается практически невозможным. Усложненные варианты предлагаемых решений (SALBP, стохастический подход к решению) учитывают множество возникающих на реальных предприятиях условий и ограничений, однако они предполагают, что длительность такта сборочной линии постоянна и заранее известна, что не позволяет применять их для планирования мелкосерийного или единичного производства. Длительность такта сборочных линий здесь изменяется в зависимости от технологической карты изделия, которое запускается в производство.

Кроме того, задачи, относящиеся к ALBP, являются NP-трудными (доказательство приведено в [15, 16]), что не позволяет напрямую решить из алгоритмически.

Задачи теории расписаний предлагают огромное многообразие решений для разных частных случаев формирования производственных планов. Подходы,

наиболее близкие к сформулированной в этой работе задаче, были рассмотрены в обзоре выше и не могут полностью разрешить поставленной проблемы. Однако ни одна из них не рассматривает совокупность взаимодействия машин и сотрудников, выполняющих работы – расписание для них может быть составлено только сепарировано, что приводит к необходимости проведения сложного процесса согласования.

Генетические алгоритмы, которые разрабатывались для систем планирования производственной деятельности можно условно подразделить на две группы. Некоторые из них, например [19], подходят к решению задачи планирования напрямую, с использованием только лишь особенностей инструмента генетических алгоритмов. Такой вариант может рассматриваться только с точки зрения модели, поскольку сложно представить применение подобного решения на практике. В первую очередь это обусловлено отсутствием возможности учета широкого множества условий и ограничений в рамках которого функционируют реальные предприятия.

#### **1.4. ПОСТАНОВКА ЦЕЛИ И ЗАДАЧ**

Основная цель проводимой работы заключается в разработке алгоритма оптимизации планирования производственных процессов на сборочных линиях для крупного машиностроения по критерию длительности. Основными задачами, которые необходимо решить в ходе выполнения работы, являются:

- разработка алгоритма оптимизации производственных процессов;
- тестирование алгоритма на реальных данных
- интеграция разработанного программного компонента для последующего внедрения в существующую систему планирования.

В качестве критерия оптимизации рассматривается минимизация длительности выполнения производственных операций. Особенности, которые необходимо учитывать, являются причинно-следственные связи между производственными операциями, обусловленные строгими последовательностями технологий производства, а также необходимость

распределения трудовых ресурсов оптимальным образом, без возникновения избыточности сотрудников на отдельных рабочих постах сборочной линии.

### **ВЫВОДЫ ПО ГЛАВЕ 1**

1. Одной из наиболее актуальных задач в рассмотренной области является создание алгоритма оптимизации плана проведения сборочных операций с учетом организации работы сотрудников на постах сборочных линий, минимизирующего длительность цикла производства.
2. Рассматриваемые в научных работах подходы не могут полностью разрешить возникающие перед реальными производствами задач –частные задачи решаются обособленно, для конкретного, специализированного случая.
3. Генетические алгоритмы, которые применяются для поиска решения аналогичных задач, показывают высокую эффективность.
4. Сформулирована постановка задачи для разработки алгоритма оптимизации сборочной линии.

## **ГЛАВА 2. МАТЕМАТИЧЕСКОЕ ИССЛЕДОВАНИЕ ПОДХОДОВ К РЕШЕНИЮ ЗАДАЧИ ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ**

Для решения задачи, которая была сформулирована в предыдущей главе рассмотрим несколько простых решений, одно из которых часто применяется в теории расписаний.

Прежде всего возникает вопрос о переборе всех вариантов. В качестве примера рассмотрим план из тысячи операций, для которого можно менять посты производства. Для такого плана существует последовательность операций, заданная графом, отсортированным топологически. Теперь для распределения операций по постам нужно в отсортированном графе присваивать номера постов для каждой операции, причем следующий номер должен быть не меньше предыдущего, иначе нарушится условие на воспроизводимость такого плана в реальной жизни. Пусть у нас есть 10 постов, тогда нам нужно распределить 10 мест, где номер поста будет возрастать. Из комбинаторики следует количество комбинаций по формуле биномиального коэффициента. В таком числе 24 разряда, если учесть, что один план моделируется 10 секунд, полный перебор всех вариантов без учета разного персонала будет занимать 1016 лет. Тем не менее, такой вариант был реализован, что позволило найти оптимальные решения для небольших оперативных планов.

Второй подход, применимый для решения поставленной задачи, базируется на том, что в задаче имеет смысл найти лучший такт производства, из которого будет следовать лучшее время работы. Этого можно добиться, если на каждом шаге рассматривать пост с самым большим тактом, и стараться уменьшить время выполнения на этом посту путем перераспределения каких-то задач на другие, менее загруженные посты.

Однако, такой подход часто теряет сходимость в случаях, когда для улучшения результата нужно принимать не точечные, а кардинальные решения. Например, если на нескольких постах одинаковое время работы, улучшение лишь одного поста не даст улучшения максимального такта, в таких случаях

требуется улучшение на всех постах. Также после какого-то улучшения на одном посту, такт для другого может стать сильно больше. Такой алгоритм требует частного разбора многих случаев и в общем виде работать не будет. Поэтому необходимо формализовать задачу, сформулированную выше и определить методику и инструменты, гарантирующие наиболее эффективное решение задачи.

## 2.1. ФОРМАЛИЗАЦИЯ ЗАДАЧИ

Прежде чем приступить к рассмотрению других подходов, представим задачу на формальном, математическом языке.

Пусть существует некоторая производственная единица, характеризующаяся множеством сборочных линий  $A = \{Al | 1 \leq Al \leq r\}$ , где  $r$  – количество сборочных линий, и множеством трудовых ресурсов  $S = \{s | 1 \leq s \leq m\}$ , где  $m$  – количество сотрудников. Каждая сборочная линия также характеризуется множеством рабочих постов, которые она включает:  $Al_r = \{o_p | 1 \leq p\}$ , где  $p$  – порядковый номер поста.

Существует множество партий  $N = \{n | 1 \leq i \leq z\}$ , где  $z$  – общее количество партий, которые нужно выпустить в течение планового периода. Каждая партия  $n$  описывается технологической картой выпуска соответствующего изделия – то есть направленным графом технологических операций  $T_n$ , а также количеством изделий в партии  $l_n$ :

$$n = \{T_i, l_i\}. \quad (8)$$

$$T_i = \{(\{prev\}, \{next\}, manhour, [1 \leq s \leq x])_j | 1 \leq j \leq y\}, \quad (9)$$

где  $(\{prev\}, \{next\}, manhour, [1 \leq s \leq x])$  – связка из множества предшествующих операций, последующих операций, и длительности обработки для перехода между ними, а также количество сотрудников, которые могут быть задействованы в обработке,  $y$  – количество операций технологической карты.

Тогда конфигурация конкретной технологической карты – непосредственно маршрутная карта, которая будет использована в производстве – будет определяться следующим образом:



$$K_i = \{(post, s)_j | 1 \leq j \leq y\}. \quad (10)$$

Для такой конфигурации сборочного производства необходимо решить задачу производственного планирования работ над реализацией  $N$  партий изделий разного типа с минимизацией длительности производственного цикла каждой партии изделий:

$$E' = \sum_{i=1}^z l_i * \sum_{j=1}^y manhour_j \rightarrow min. \quad (11)$$

Отметим, что  $manhour_j$  – переменная, зависящая от количества сотрудников, вовлеченных в операцию, поэтому для каждой технологической карты суммарная длительность также переменная.

Производственный план описывается как

$$OS = \{(begin, end, \{resources\})_j | 1 \leq j \leq y * i\}. \quad (12)$$

Аналогично, содержание и длительность выполнения плана будут зависеть от переменны значений  $resources$ . Он должен содержать в себе следующие данные:

значение  $r$ , определяющее сборочную линию  $Al_r$ , где будет произведена каждая партия;

значения  $S_{ji}$ , для каждой производственной операции каждой технологической карты, обеспечивающие выполнения (11);

длительность производственного цикла  $E$  и такта для предложенных вариантов планирования.

Такт производства будет уникальным для каждой производственной линии и будет равен:

$$tact_j = \frac{\sum_{j=1}^y dur_j}{r}. \quad (13)$$

Пусть производственной единице доступно количество сотрудников, позволяющее распределить их так, что  $s_j = x$  для всех  $j$  операций каждой технологической карты. Тогда задачу (10) можно представить в виде:

$$E = \sum_{i=1}^z l_i * D_i \rightarrow \min, \quad (14)$$

где  $D_i$  – суммарная длительность выполнения операций  $i$ -й технологической карты. Также план можно представить как совокупность технологических карт и операций, которая может быть вычислена с использованием имитационного моделирования:

$$f(OS) = duration, \quad (15)$$

где *duration* – длительность выполнения плана с конкретным распределением сотрудников по постам. Тогда задача сводится к

$$f(OS) \rightarrow \min. \quad (16)$$

Заметим, что полученная формула (14) аналогична (2), описывающей задачу SALBP. Таким образом, с учетом введения описанных упрощений, поставленная задача аналогична проблеме, известной как NP трудная [16]. Соответственно, так как задача, решаемая в данной работе, характеризуется бóльшим числом параметров и, в наиболее благоприятном случае ( $s_j = x$  для всех  $j$  операций каждой ТК), решить необходимо задачу типа SALBP, тогда решаемая задача в каждом из возможных случаев будет не легче, чем NP трудная.

## **2.2. ОЦЕНКА ПРИМЕНИМОСТИ ПОДХОДОВ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ К РЕШЕНИЮ ПОСТАВЛЕННОЙ ЗАДАЧИ**

Для того, чтобы оценить возможность применения подходов и методов, распространенных при решении некоторых задач теории принятия решений и теории расписаний, рассмотрим упрощенный пример поставленной в работе задачи и сведем его к линейному функционалу.

Рассмотрим цех, производящий 2 типа продукции. Пусть на цех выпущено производственное задание на выпуск  $A$  штук изделий первого типа и  $B$  штук изделий второго типа. Технологические карты выпуска этих изделий

предполагают, что на производство одного изделия первого типа необходимо  $h_1$  человеко-часов, соответственно на изделие второго типа – необходимо  $h_2$  человеко-часов.

В доступе цеха имеются  $s$  сотрудников и  $l$  сборочных линий. Каждая линия имеет одинаковое число постов, длительность работы за которыми обратно пропорционально зависит от числа привлеченных сотрудников:

$$l_n = \{p_1, p_2, \dots, p_k\}, \quad (17)$$

$$duration(l_n) = \sum_{i=1}^k \frac{t(p_i)}{x_i}, \quad (18)$$

где  $x_i$  – число сотрудников, вовлеченных в  $i$  –ую операцию. Заметим, что  $x_i$  зависит от времени и определяется для каждой производственной линии цеха и каждого поста линии:

$$x_i(t)_n = \begin{cases} x_i = a_i, 0 < t \leq z \\ x_i = b_i, z < t \leq v \\ \dots \\ x_i = f_i, y < t \leq t(p_i) \end{cases}. \quad (19)$$

При этом длительность выполнения параллельных операций разбивается на некоторые интервалы, в течение которых численность вовлеченных сотрудников остается постоянной для каждой отдельной операции. Важно учитывать, что для каждого такого интервала и на всем промежутке времени действует следующее ограничение на одновременное вовлечение сотрудников:

$$\forall t > 0 \exists \{x_i(t)_n, \forall 1 \leq n \leq l\}: \sum_{i=1}^n x_i(t)_n \leq s. \quad (20)$$

Тогда глобальная задача оптимизации имеет вид:

$$\left\{ \begin{array}{l} A * \frac{h_1}{duration(l_n)} \rightarrow \min; \\ B * \frac{h_2}{duration(l_n)} \rightarrow \min; \\ \forall t > 0 \exists \{x_i(t)_n, \forall 1 \leq n \leq l\}: \sum_{i=1}^n x_i(t)_n \leq s. \end{array} \right. \quad (21)$$

Из системы уравнений 21 явно следует, что целевая функция определяется отношением трудоемкости операции к длительности работы производственной линии, переменной на временном интервале  $[0; t]$ , что не позволяет линеаризовать систему и подходить к ее решению с использованием инструментов линейного программирования, в том числе симплекс-метод, который успешно применяется при решении задач с фиксированным числом сотрудников.

Отметим, что в рассмотренном примере было сделано несколько упрощающих допущений – число возможных заказов на производство может достигать десятков типов изделий, что увеличивает количество целевых функций, которые необходимо минимизировать. Также необходимо обратить внимание, что, даже при рассмотрении фиксированных длительностей выполнения операций, необходимо поддерживать инвариант, описанный условием 20 и поддерживать ограниченное число сотрудников, вовлеченных в работу в каждый момент времени, что также не позволяет свести задачу к линейной.

Рассмотренные проблемы сведения задачи к линейному программированию решаются с использованием более широкого метода, который называется программированием в ограничениях (Constraint Programming). На первый взгляд этот метод может оказаться эффективным. Его суть заключается в том, что для задачи формируют ограничения – отношения между переменными. Решение, которое удовлетворяет этим ограничениям определяется алгоритмом.

В статье [22] было проведено сравнение применимости программирования в ограничениях для задачи SABLР. Результаты этого сравнения показывают, что для задачи из 1000 операций за ограниченное время решение, даже субоптимальное, в более чем половине случаев не было найдено. При этом сложность рассматриваемой в этой работе задачи выше, так как используются дополнительные ограничения по сравнению с оригинальной задачей SALBP, что только ухудшит результаты такого решения. Также стоит

упомянуть и расширяемость метода программирования в ограничениях. Для некоторых специфических операций, например с использованием станков механообработки, которые требуют дополнительного времени для пусконаладки в некоторых условиях разработать модель с системой ограничений по меньшей мере нетривиально. Также дополнительные сложности возникнут и с изменением целевой функции, для которой потребуются придумывать новую модель в каждом конкретном случае.

### **2.3. ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ ПЛАНИРОВАНИЯ**

Системы планирования производственной деятельности занимают определенное положение в инфраструктуре предприятия. Наиболее распространенными являются MES и ERP-системы, часто в совокупности с ними используются и цифровые двойники, имитационные модели.

Имитационное моделирование может применяться на предприятии по разным назначениям в зависимости от специфики решаемой задачи. В частности, при формировании ОП, ведутся расчеты для совокупности операций (любая ТК описывается набором операций, а заказ, в свою очередь, описывается перечнем технологических карт и их количеством). Имитационные вычисления ведутся также и для вычисления данных о занятости сотрудников и вовлеченности в работу технологических ресурсов.

Целью применения имитационной модели чаще всего является построение оперативного плана для некоторого набор производственных задач без применения сложных оптимизаций. При планировании предполагается, что запуск продукции в производство осуществляется в заданном набором задач порядке, что и моделируется посредством имитационной модели. На основе результатов ее работы оперативный план - такое планирование предполагает в качестве результата некоторое число, обозначающее совокупную длительность выполнения всех необходимых операций в отсчете от нуля и без учета каких-либо перерывов в рабочем времени.

Также посредством имитационной модели может быть оценено рассогласование фактических и плановых производственных показателей, оценена потребность в материалах на некоторый плановый период. Таким образом, применение на многих предприятиях имитационной модели сегодня объясняется гибкостью и многофункциональностью этого инструмента. Однако вычислительная сложность процессов планирования для крупных предприятий столь высока, что применение имитационной модели без алгоритмов оптимизации зачастую невозможно.



Рисунок 7 – Обобщенная схема взаимодействия блоков планирования производства

Входные данные для имитационного моделирования вносятся в систему планирования в основном сотрудниками, занимающими руководящие должности на производственных единицах. Это может быть руководитель производственной площадки или цеха, который владеет информацией о приоритетности заказов и может корректно формировать набор и последовательность входных данных, а также вносить ряд изменений в режиме ручной корректировки плана, основываясь на субъективной информации.

Учитывая результаты проведенных исследований предметной области, оценки применимости различных подходов к решению задачи, определено направление, в рамках которого будет решена поставленная задача.

Многие исследователи, как и авторы [20], подходят к задаче оптимизации, объединив различные методы решения. На основе задач ALBP и теории расписаний формализуются целевые функции для некоторых частных

задач, после чего решение ищется генетическим перебором, а не напрямую. Такой подход является наиболее перспективным, поскольку его можно применить к широкому спектру частных задач, он обобщает достоинства всех рассмотренных подходов и предоставляет свободу в выборе инструментов и математического аппарата. Серьезным недостатком такого подхода является трудоемкость процесса формализации целевой функции, однако практическая значимость такого решения соотносима с трудозатратами.

Превосходство генетических алгоритмов, как инструмента реализации, состоит в том, что они показывают эффективность при весьма сложном рельефе функции приспособленности, а результаты работы представлены не единственным решением, но набором субоптимальных решений, имеют больший шанс сойтись к локальному оптимуму. Поэтому именно этот подход будет использован в работе.

## **ВЫВОДЫ ПО ГЛАВЕ 2**

1. Классической задача SALBP, известная как NP-трудная, сводится к частному случаю рассматриваемой задачи, если для этой задачи сделаны несколько допущений, упрощающих условия. Это позволяет утверждать, что рассматриваемая задача – не легче чем NP-трудная.
2. На конкретном примере решаемой задачи была рассмотрена возможность решения задачи методами линейного программирования с целью дальнейшего использования симплекс-метода. Исследование показала, что рассматриваемая задача не может быть линеаризована, что не позволяет применять к ней подходы, распространенные для задач с фиксированной длительностью операции.
3. Рассмотрена архитектура системы планирования и место разрабатываемого алгоритма в ней. Определены источники входных данных, а также потенциально заинтересованные в рассчитанной информации блоки системы управления предприятием.

4. Определен подход к решению поставленных задач – для разработки алгоритма сборочной линии будут использоваться совокупность идей из различных методик, опорным и основным подходом станет применение генетических алгоритмов.



### **ГЛАВА 3. РАЗРАБОТКА АЛГОРИТМА ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ**

В качестве основного инструмента был выбран язык программирования Golang, разработанный компанией Google как язык со строгой типизацией и компиляцией в машинный код под конкретную платформу, предназначенный для создания высокоэффективных программ, работающих на современных распределённых системах. Его достоинствами, ставшими при выборе решающими, стали встроенные средства параллельного программирования (потoki Go routines), а также высокая скорость компиляции в совокупности с простотой синтаксиса – эти качества позволили обеспечить высокий темп разработки.

Несколько подробнее остановимся на технологии Go routines – это особая технология, реализуемая в рамках языка программирования Go, которая обеспечивает простоту написания программного кода многопоточно, что необходимо для реализации ряда тяжёлых расчетов в алгоритме – выполнение расчетов для нескольких субоптимальных вариантов плана, подобранных генетическим алгоритмом, например, возможно организовать параллельно, что существенно ускоряет получение готового результата пользователем. Суть технологии состоит в том, что она не требует физического формирования потока – он создается виртуально, что значительно проще и быстрее.

Для хранения больших массивов входных и выходных данных необходимо организовать базу данных. Разработка хранилища данных со сложными взаимосвязями не входит напрямую в задачи данной работы. Поэтому для тестирования генетического алгоритма и обеспечения формирования производственных планов была использована система управления базами данных (СУБД) SQLite. Достоинством взаимодействия с данной СУБД стала возможность создать версию базы данных, которая не будет задействовать много памяти и не отразится на темпе разработки.

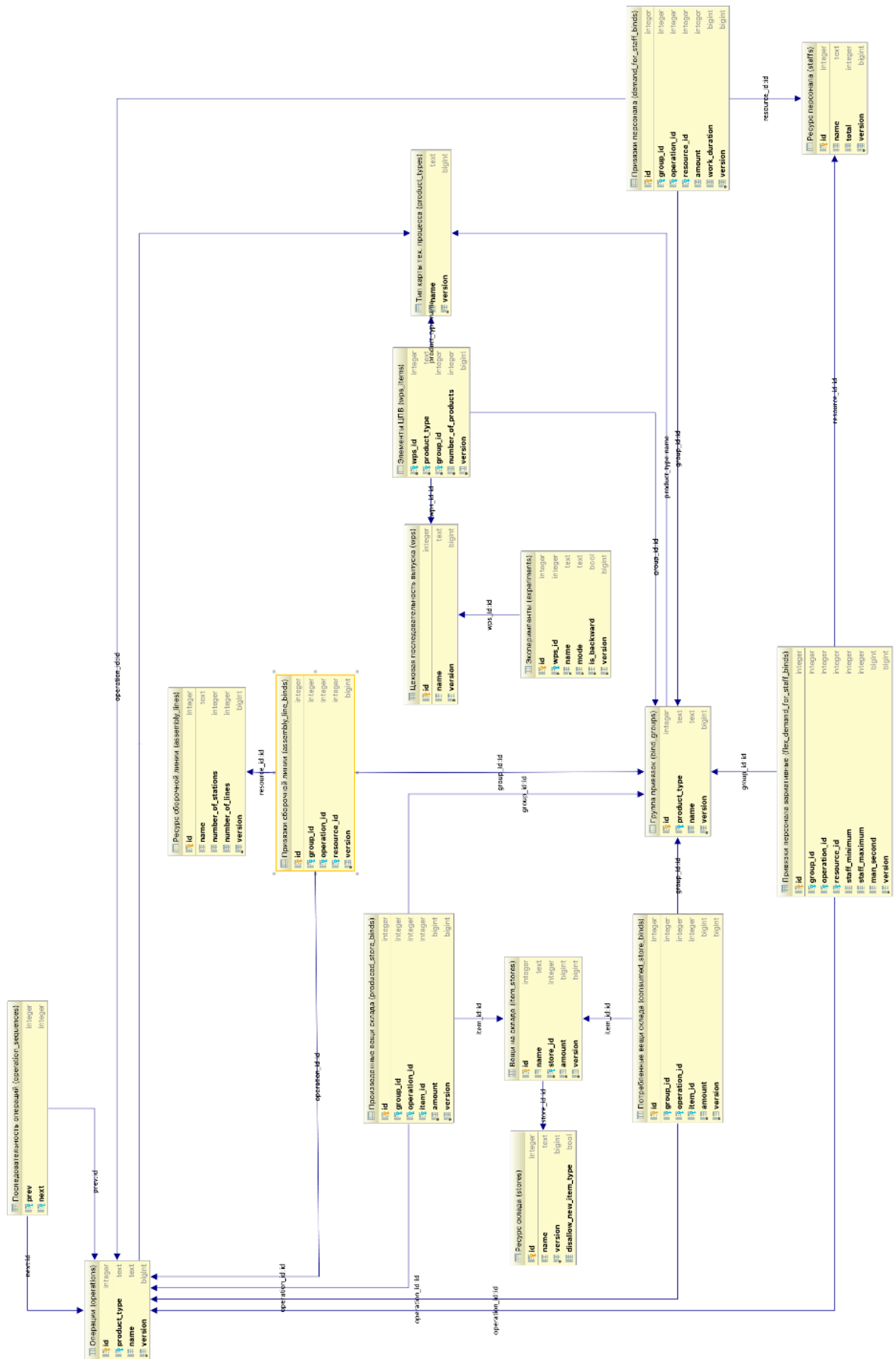


Рисунок 8 – Схема БД для хранения входных и выходных данных

Схема БД, которая была разработана в рамках работы над алгоритмом приведена на рисунке 8. Подробности её внутреннего устройства не имеют прямого отношения к проделанной работе, поэтому будем рассматривать ее исключительно как готовое решения для хранения входных и выходных данных алгоритма.

### **3.1. РАЗРАБОТКА ГЕНЕТИЧЕСКОГО АЛГОРИТМА ОПТИМИЗАЦИИ СБОРОЧНОЙ ЛИНИИ**

Генетический алгоритм представляет собой эвристический алгоритм поиска решения, используемый для задач оптимизации и моделирования с помощью комбинирования и вариативности значений исходных параметров.

Разработанный генетический алгоритм направлен на решение задачи оптимизации плана работы сборочной линии. Формирование плана сборочной линии происходит посредством определения двух групп параметров:

- оптимизация привязки трудовых ресурсов к постам (*DemandForStaff* – потребность в персонале и *FlexDemandForStaff* – переменная потребность в персонале, заданная неравенством);
- оптимизация привязки операций технологической карты к постам сборочной линии.

В зависимости от переданных на производство заказов, определяется потребность в технологических и трудовых производственных ресурсов. Планирование происходит посредством имитационного моделирования ресурсов предприятия. Оптимизация осуществляется по заданным параметрам, основной из которых это длительность производственного плана.

Применение генетического алгоритма направлено на оптимизацию процесса формирования краткосрочного плана производственной единицы – оперативного плана (ОП) уже на этапе получения входных данных. После первичной обработки данные обрабатываются алгоритмом: производятся вычисления, необходимые для поиска нескольких наборов субоптимальных

значений входных данных (количество вовлеченных в работу сотрудников и их распределение по рабочим постам) для создания нескольких оперативных планов на один и тот же период.

Возможность вносить содержательные изменения входных данных является важным этапом их обработки и позволяет осуществить подготовительные работы для дальнейших расчетов, а также гибко менять параметрические величины, чем и достигается большая вариативность, необходимая для поиска оптимального значения.

Входными данными для алгоритма являются:

- технологическая карта – структура данных, детально описывающая перечень операций для производства единицы продукции. Включает в себя: операции, последовательность, трудоемкость, перечень требований и привязок к ресурсам и т. д.;
- производственные ресурсы – структура цеха (сборочные линии, их описание), запасы на складе производственных единиц, сотрудники;
- конфигурация для алгоритма – структура данных, описывающая входные параметры для алгоритма (подробнее будет описана далее).

Работа алгоритма направлена на создание оперативного плана и его оптимизации по заданным параметрам, проходит следующие этапы:

- создание и оптимизация шаблона продукта – структуры данных, которая является технологической картой одной единицы конкретного изделия;
- пошаговый расчет оптимизированного оперативного плана производственной единицы с учетом ограничений количества ресурсов, обновления информации о ресурсах каждый раз при пересчете переменной.

Процесс планирования начинается после подготовки данных. На вход поступает список производственных задач на единицу продукции. Алгоритму известны места, которые он должен оптимизировать, то есть если, например существуют ограничения, не учтенные в модели, например операция под номером три должна производиться на второй станции из-за особенностей

оборудования, во входных данных может быть указана жесткая привязка третьей операции ко второй станции. Алгоритм оптимизирует входные данные для имитационной модели и непосредственно взаимодействует с моделью для создания лучшего и/или субоптимального оперативного плана. Входные данные изменяются в зависимости от гена, происходит привязка операций к постам и людей к операциям. На вход алгоритма может поступить только те ТК, которые прошли через процесс предобработки. Он заключается в формализации данных, которые в дальнейшем поступают в подсистему имитационного моделирования.

Алгоритм работает циклически, весь процесс подготовки ТК и генерации оперативного плана происходит  $N$  поколений, для размера популяции  $k^3$ .

Следующий этап – формирование начальной популяции. Популяция содержит в себе несколько особей, каждая особь состоит из определенного набора генов. Первая группа генов — это привязки операций к станциям на сборочной линии. Вторая группа – это привязки каждого типа персонала к операциям. Третья группа генов — это суммарное количество персонала данного типа, доступное для работы. Разберем отличие второй и третьей группы генов. Пусть у нас есть две операции с требованием по персоналу по два человека на операцию. Если общее доступное количество равняется четырем, то эти операции будут произведены параллельно. При этом если общее доступное количество людей будет два, то эти операции будут произведены последовательно.

Для начала работы генетического алгоритма случайным образом создается начальная популяция, которая может оказаться неконкурентоспособной, но алгоритм на следующих этапах произведет ее улучшения. Преимущества случайного распределения особей заключается в том, что алгоритм будет сходиться к нескольким локальным минимумам, и уже при постепенном отборе сможет найти лучший из них. Но на случайно созданную популяцию накладываются некоторые ограничения. Первое – это ограничения, которые нам задаются входными данными, например, то, что операцию должно выполнять не больше, чем  $n$  человек и не меньше, чем  $k$ . Второе – это

ограничение, которое нам задается технологической картой. Если в технологической карте операция под номером два, стоит после операции под номером один, то алгоритм не может назначить на первую операцию пост два, а на вторую пост один. В этом случае процесс моделирования не сможет завершиться корректно. Последнее исправление, которое производится в рамках гена это соотношение максимально доступного персонала и требования персонала на одну операцию. Первое должно быть строго не меньше второго. Заметим, что в процессе мутации, нам снова приходится проводить эту валидацию.

Таким образом, на первом шаге создается любая популяция, соответствующая формату особей популяции, для которой можно почитать функцию приспособленности. Итогом первого шага является популяция  $N_1$ , состоящая из  $k$  особей. Каждая особь соответствует привязкам для технологической карты, для которой будет сгенерирован свой оперативный план.

Далее начинается процесс формирования ОП для каждой особи. Общий алгоритм работы подсистемы имитационного моделирования выполняет привязку каждой операции обработки каждого изделия к временным интервалам, конкретному работнику и конкретным технологическим средствам. Для каждого оперативного плана мы получаем статистику, такую как длительность плана и максимальный такт линии.

Следующий этап работы алгоритма – выбор лучших результатов. В процессе реализации этапа осуществляется проверка улучшения результата на требуемую величину. Величина выбирается из статистики в зависимости от целевой функции. Целевая функция может быть не только длительностью плана или такта, но и загруженность линии или персонала. На следующем этапе происходит выбор родителей для особей следующей популяции. Выбор родителей происходит по заданной целевой функции и параметру, например это  $n$  особей, которые получили лучший план. Непосредственно происходят этапы отбора (селекции) и выбора родителей:

- на этапе отбора из всей популяции выбирается определённая доля, которая будет функционировать на данном этапе; количество выживших особей является параметром генетического алгоритма. Количество родителей, выбранных на этом этапе, влияет на изменяемость и стабильность результатов. Чем больше родителей, тем больше разных особей мы получим в следующем поколении, но тем менее стабильные будут уже достигнутые результаты;
- выбор родителей производится с помощью заданного метода, для производства потомка требуется не менее двух родителей.

Далее происходит скрещивание родителей для создания новых особей. Основное требование в процессе скрещивания – новая особь должна унаследовать черты обоих родителей, «смешав» их.

Скрещивания происходит по определённому алгоритму, в данной реализации был выбран алгоритм равномерного скрещивания (*uniform crossover*). В данном алгоритме, как правило, каждый ген выбирается из любого родителя с равной вероятностью. Иногда используются другие соотношения для смешивания, в результате чего потомство наследует больше генетической информации от одного родителя.

Далее происходит процесс мутации. Мутации схожи с процессом скрещивания, из популяции выбирают некое количество особей и изменяют их в соответствии с заранее определёнными операциями. В каждой особи может мутировать любой ген. Следующий этап – формирование нового поколения. Далее происходит процесс построения ОП для новой каждой особи, который рассматривался выше.

Завершение работы алгоритма происходит в следующих случаях:

- отработало количество поколений, которое указано в конфигурации;
- результат не улучшался последние 10% поколений от общего количества поколений.

После завершения работы алгоритма, проведения вычислений и составления ОП, осуществляется выбор лучшего ОП для всех особей популяции.

На основе вычислений лучший по показателям оперативный план выводится непосредственно пользователю и записывается в базу данных.

Следующим этапом разработки стала интеграция алгоритма с имитационной моделью сотрудников и технологических ресурсов – в работе рассматривается производственная единица, оснащённая только поточными производственными линиями и некоторым количеством сотрудников разной категории и квалификации.

### **3.1.1. Имитационная модель производственных ресурсов и производственное планирование**

Входными данными для имитационной модели являются:

- упорядоченный набор данных, вида:
  - наименование продукции каждого вида, необходимой для реализации в текущем плановом периоде;
  - карта технологического процесса для каждого вида продукции с конфигурациями ресурсов<sup>2</sup>, которые используются для производства соответствующего изделия;
  - количество продукции, необходимой для реализации в текущем плановом периоде.
- структура производственной единицы (площадки, цеха, участка, линии), для которого строится оперативный план;
- перечень производственных ресурсов, которые могут быть задействованы в рамках выполнения поставленных задач;
- свойства производственных ресурсов;
- привязки операций и ресурсов к участкам, линиям или рабочим местам производственного подразделения.

---

<sup>2</sup> Конфигурация ресурса - набор свойств ресурса, сконфигурированных для выполнения конкретной производственной задачи. Примером конфигурации ресурса может являться длительность выполнения операции на основе количества персонала, связанного с операциями.



На первом этапе работы компонента формирования оперативного плана пользователь задает последовательность выпуска. Результат работы имитационной модели - длительность работ в логическом времени - сохраняется в БД. После этого пользователь сохраняет отчет или, в случае неудовлетворения результатом - перезапускает вычисление оперативного плана, изменив последовательность выпуска. Отметим, что порядок работ в последовательности формируется пользователем на основе как объективных данных (даты отгрузки, маржинальная стоимость работ и пр.), так и субъективных суждений (приоритетность, репутационные риски и пр.).

Одним из важнейших этапов работы алгоритма имитационного моделирования является подготовка входных данных, полученных из БД. Она подразумевает в том числе предварительную обработку данных, одной из функций которого является расчет длительности выполнения операций как одной из конфигураций ресурсов. Так как длительность выполнения операций является переменной величиной, зависимой от количества сотрудников, выполняющих операцию, существуют четыре свойства этого ресурса (они описываются параметрами *DemandForStaff* и *FlexDemandForStaff*):

- минимум - на операцию назначается минимальное количество сотрудников, длительность выполнения операции становится максимальной;
- максимум - на операцию назначается максимальное количество сотрудников, длительность выполнения операции становится минимальной;
- среднее - на операцию назначается среднее количество сотрудников, длительность выполнения операции становится усредненной;
- случайная, при этом количество персонала на операцию выбирается случайным образом.

В процессе создания оперативного плана, для получения корректной оценки длительности выполнения операции или набора операций, необходимо

ввести систему ограничений, которая будет отражать как ресурс, участвующий в операции, может влиять на её время выполнения.

Это привело к созданию модели ресурсов накладывающей ограничения на выбор операции для вычислений имитационной модели.

Под ресурсом подразумевается любое техническое устройство, деталь, инструмент или средство, за исключением сырьевого материала и промежуточного продукта, находящиеся в распоряжении предприятия для производства товаров и услуг.

В соответствии с данным определением к ресурсам относятся в том числе и человеческие ресурсы, которые в данной системе не рассматриваются с точки зрения поведения или других аспектов человеческой жизни, а лишь с точки зрения возможности выполнить конкретную задачу.

Также необходимо обозначить, что в данной работе под моделью ресурса будет пониматься упрощенная модель реального ресурса, отражающая его основные (в рамках выполняемых операций) характеристики.

Каждая модель ресурса представляет из себя структуру данных, которая должна реализовывать три метода:

- метод привязки операции к модели ресурса;
- метод, осуществляющий проверку возможности выполнения данной операции моделью ресурса;
- метод, осуществляющий логику работы и в котором происходит изменение состояния данной модели.

Под привязкой операции к модели подразумевается добавление операции в очередь на выполнение и, если это первая привязанная для данного продукта операция, то добавление продукта в очередь на распределение. Привязка осуществляется в начале работы системы, что позволяет ресурсам манипулировать процессом имитационного моделирования, разрешая или запрещая выбирать привязанные к ним операции для расчета, что может повлечь за собой изменение последовательности выполнения операций и, соответственно, расчетного времени выполнения операций ТК.

Проверка привязки ресурса к операции производится во время работы системы и именно на этом этапе происходит отбор операций в соответствии с внутренним состоянием модели. Логика осуществляется при выборке операции имитационной моделью и для каждой операции вызывается два раза: чтобы отметить состояние модели в начале и в конце вычислений для очередной операции.

Одной из моделей ресурсов, которые были разработаны в рамках формирования окружения для алгоритма оптимизации, является модель рабочих. Ранее уже было сказано, что ресурс рабочих рассматривается с точки зрения необходимого элемента функционирования предприятия, при этом упускаются особенности жизнедеятельности рабочих. Эта особенность ведет к тому, что возникает необходимость ввести соответствующие погрешности, регулирующие различные ситуации - состояние работника, внеплановые перерывы и т.д. - их отсутствие может на порядок уменьшить точность модели.

Ресурс работника характеризуется общим количеством сотрудников одной профессии (квалификации). Также в виде структуры данных реализовано состояние ресурса, которое включает следующие элементы:

- операции, которые предстоит выполнить рабочим;
- операции, которые уже выполнены;
- количество людей данной профессии на конкретную операцию, конкретного продукта, включая серийный номер;
- привязку каждого конкретного работника к выполняемой операции;
- последняя временная метка, на которой остановились работы.

Другой ресурс, модель которого была проработана – непосредственно сборочная линия. Он характеризуется количеством линий некоторого типа и количеством рабочих постов.

### **3.1.2. Повышение эффективности работы имитационной модели**

При проведении первичных тестовых испытаний было замечено, что скорость вычислений плана разработанной имитационной моделью

производственной единицы была крайне низкая, что мешало оценить эффективность алгоритма оптимизации для больших наборов входных данных. Необходимость быстрой работы имитационной модели обусловлена прямой зависимостью от нее скорости работы генетического алгоритма – в процессе работы генетический алгоритм обращается в имитационной модели множество раз (для нескольких сотен операций количество обращений достигает 5000 раз). Поэтому было принято решение о необходимости повышения эффективности и использован профилировщик для поиска узких мест моделирования.

Было выявлено, что наибольшее время занимал поиск доступного интервала времени в графике сотрудника. Интервалы изначально хранились в виде списка рабочих интервалов. Для того, чтобы получить доступный интервал, начиная с нужного нам времени, требуется найти свободный промежуток. Для этого требуется  $O(N)$  времени на каждого сотрудника, где  $N$  – количество интервалов. Для последующей модификации свободных интервалов нужно было разделить модифицируемый интервал или удалить его, если он будет полностью занят. Для этого также требуется  $O(N)$  времени.

Те же самые операции вставки и поиска доступных людей на интервале можно реализовать с использованием дерева отрезков. Суть в том, что эта структура данных хранит общий интервал в виде дерева, и позволяет посчитать результат быстрее, если запрос на промежутке полностью содержится в детях какой-то вершины. Модификация такого дерева происходит также только на двух вершинах, которые содержат в себе нужный интервал.

Такая структура позволяет выполнять поиск свободных людей на интервале и модификацию этого интервала за  $O(\log T)$ , где  $T$  – это временной промежуток, на котором нам нужно провести модификацию. При этом одно дерево содержит информацию про всех взаимозаменяемых людей одной профессии, то есть, например на всех инженеров третьего разряда нужна одна такая структура.

Такое дерево содержит  $2T$  вершин, и так как время одного планирования может быть достаточно долгим, например на год вперед, нужно было уменьшить

константу памяти и времени. Для этого было применено еще две техники. Первая это реализация так называемого ленивого доступа. Если в дереве не проводилась модификация в какой-то вершине или в потомке, то структура хранит вместо инстанса этой вершины ссылку на пустой указатель. При поиске, для пустых вершин алгоритм представляет, будто там находится стандартное поддереву, которое еще не было модифицировано, и считает результат для такого пустого поддереву. Модификация же создает необходимые вершины. Это позволяет снизить расходы по памяти с  $O(T)$  до  $O(N * \log T)$ , так как запросов модификации у нас столько же, сколько и занятых интервалов, а при одном запросе создается не более чем  $2 * \log T$  вершин.

Чтобы снизить временную константу, так как  $T$  может быть достаточно большим, было применено расширение дерева. Дерево не сразу создавалось на большой промежуток, а увеличивалось, как только занятые интервалы переходили дальше старого размера дерева. При этом запрос свободного персонала работает на любом промежутке, по аналогии с пустыми, не пересекающимися интервалами. Как только происходит модификация на участке, который не покрывается старым размером, создается новый корень, и текущая вершина подвешивается как левая вершина нового корня. Размер дерева умножается на два, пока текущий правый край не будет больше правого края интервала. Возникает проблема: как поддерживать исходное количество свободных людей на всем интервале? Для этого используется параметр глобальной модификации, это число, которое обозначает за сколько человек отвечает это дерево. Оно эквивалентно модификации на промежутке от нуля до бесконечности, и не пропадает при расширении дерева. Эта модификация позволила снизить временную константу, особенно для небольших планов.

Также особенности языка разработки Golang позволили запускать моделирование в несколько потоков, что в разы ускорило работу как модели, так и генетического алгоритма. Особенностью реализации этой модификации заключается в том, что нужно уметь завершать процессы, которые делятся

слишком долго, при этом не теряя результат. Для этого был написан специальный обработчик контекстов, куда потоки складывали лучший результат на текущий момент и слушали сигналы об остановке работы.

На рисунке 9 представлено сравнение оптимизаций и исходного имитационного моделирования для сравнения на нескольких разных размерах экспериментов на компьютере с четырьмя физическими потоками:

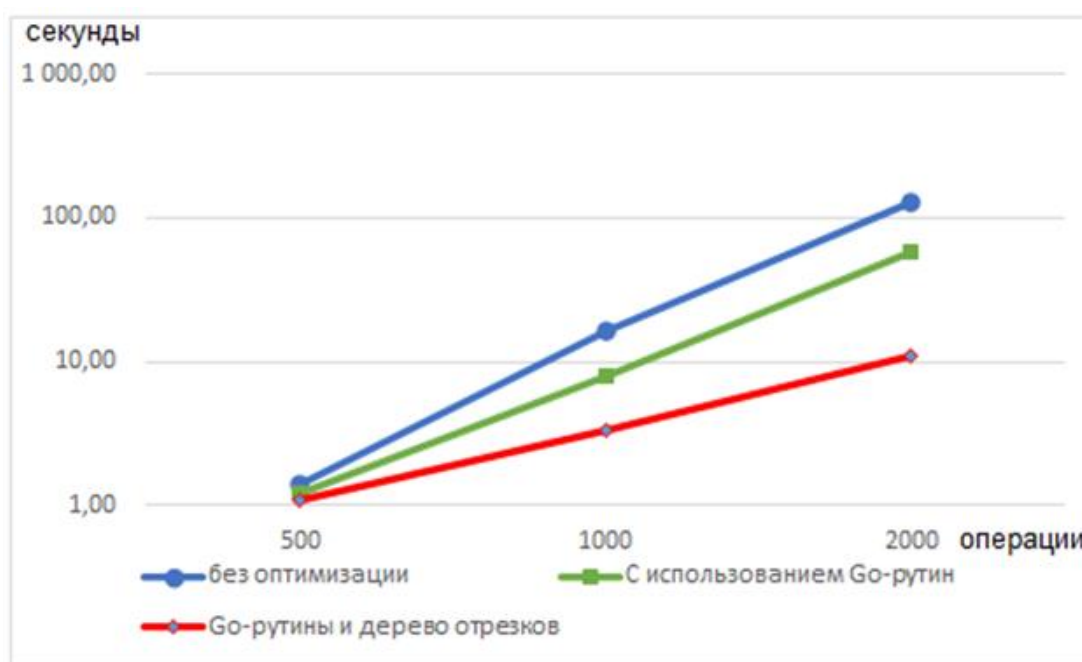


Рисунок 9 – Скорость работы имитационной модели до и после оптимизации

### 3.1.3. Выбор гиперпараметров генетического алгоритма

Выбор параметров был реализован с помощью генератора случайных технологических карт. Такой генератор создавал планы с низким числом операций и небольшим возможным количеством сотрудников. Такие планы позволяют просчитать лучший вариант методом полного перебора. С помощью такого метода можно найти оптимальную длительность плана и сравнить с выходом генетического алгоритма.

К параметрам, которые необходимо было определить таким способом были вероятность мутации, процент родителей и коэффициент скрещивания. Для их формирования были сгенерированы 10 случайных производственных планов, характеризующихся различным числом операций и взаимосвязей между

ними. Наилучший сценарий реализации каждого из них был получен сначала путем перебора значений, затем – разработанным генетическим алгоритмом. Таким образом были получены среднеквадратичные отклонения и на их основании выбраны наилучшие значения для гиперпараметров генетического алгоритма.

В первую очередь были рассмотрены среднеквадратичные отклонения для ближайших и наиболее далеких родителей с условием скрещивания половины особей в зависимости от вероятности мутации. Полученные результаты приведены на рисунках 10 и 11.

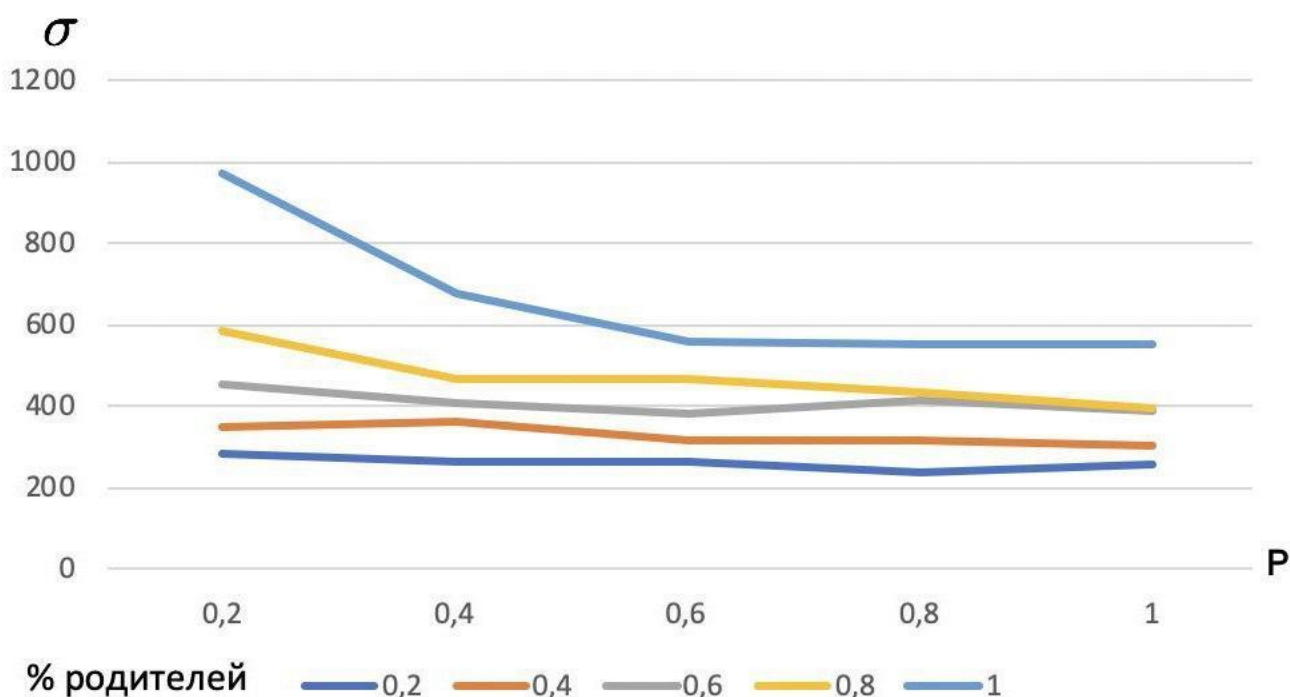


Рисунок 10 – Среднеквадратичные отклонения наиболее далеких родителей при скрещивании половины особей, где  $P$  – вероятность мутации,  $\sigma$  – среднеквадратичное отклонение

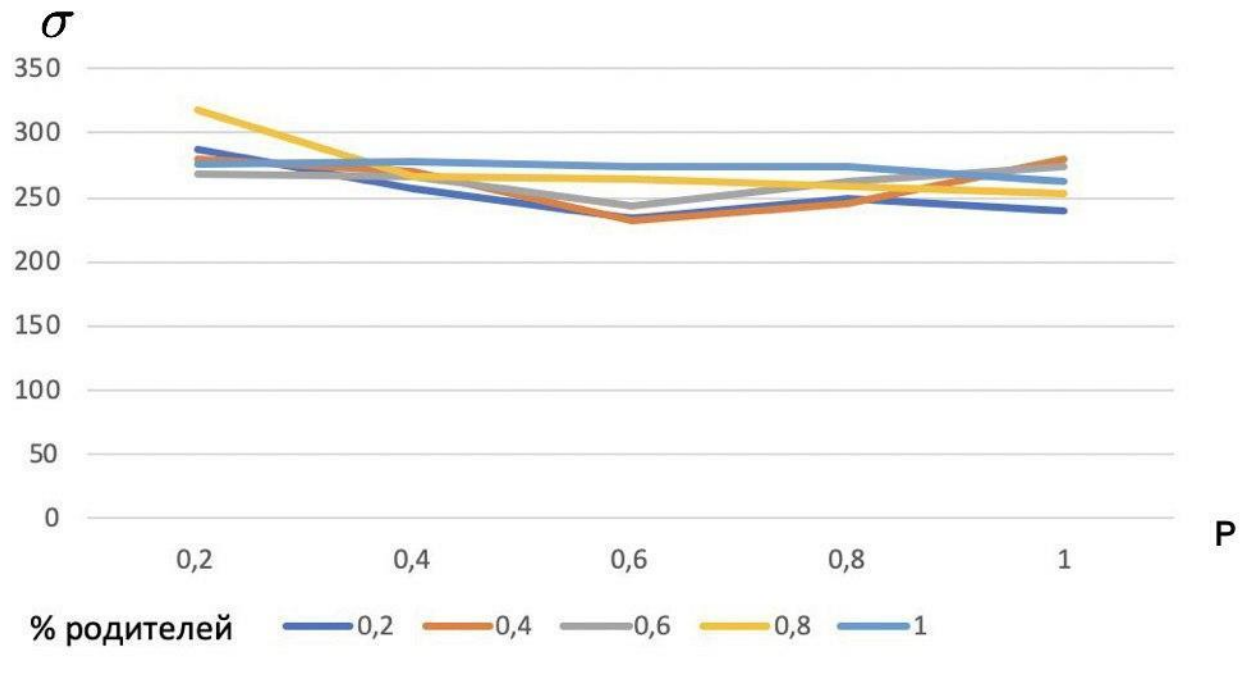


Рисунок 11 – Среднеквадратичные отклонения ближайших родителей при скрещивании половины особей, где  $P$  – вероятность мутации,  $\sigma$  – среднеквадратичное отклонение

После чего были рассмотрены среднеквадратичные отклонения для ближайших и наиболее далеких родителей с условием скрещивания всех особей в зависимости от вероятности мутации. Полученные результаты приведены на рисунках 12 и 13.

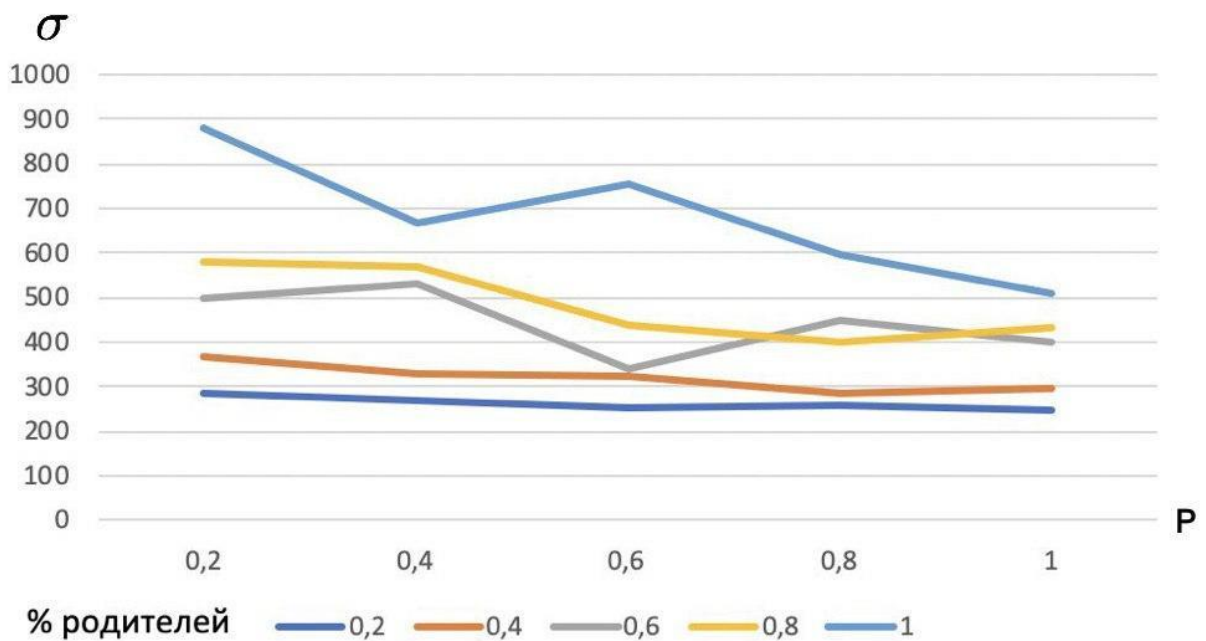




Рисунок 12 – Среднеквадратичные отклонения наиболее далеких родителей при скрещивании всех особей, где  $P$  – вероятность мутации,  $\sigma$  – среднеквадратичное отклонение

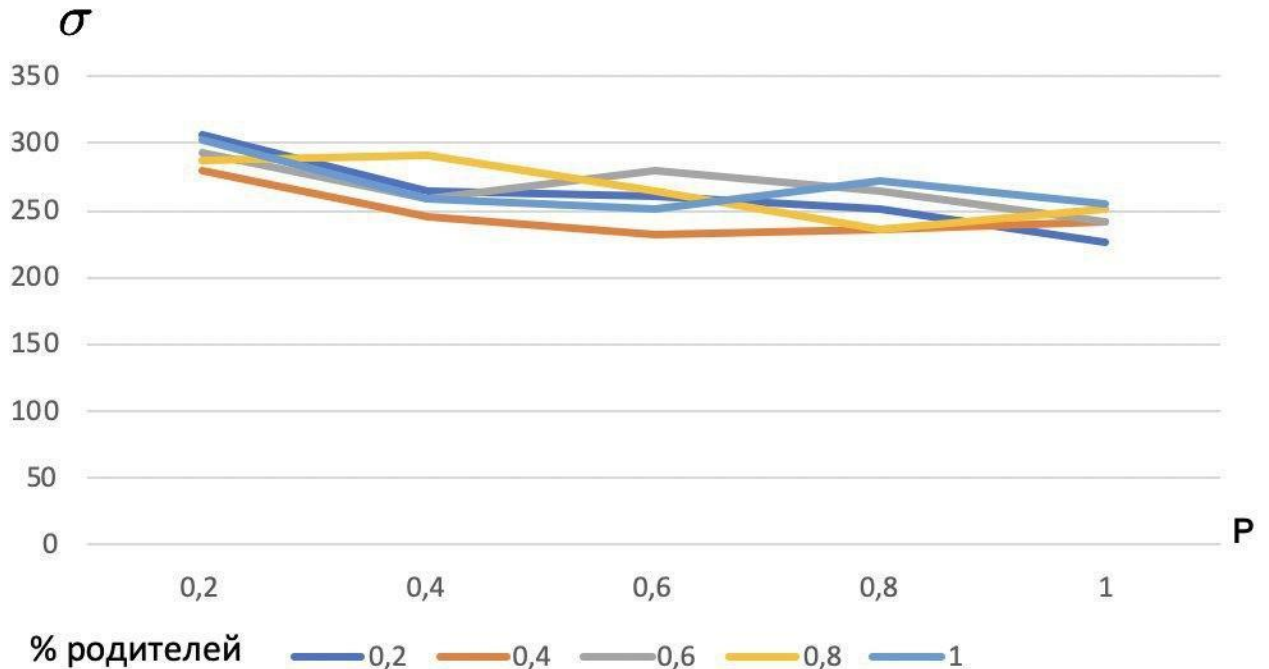


Рисунок 13 – Среднеквадратичные отклонения ближайших родителей при скрещивании всех особей, где  $P$  – вероятность мутации,  $\sigma$  – среднеквадратичное отклонение

Подход к подбору вероятности мутации и процента родителей, а также определение объема популяции и количества особей в ней объясняется таблицей 1. Длительности экспериментов были нормализованы, чтобы каждый эксперимент давал одинаковый вклад в конечный результат.

Таблица 1 – Среднеквадратические отклонения значения длительности выполнения плана в зависимости от значений гиперпараметров.

Поколений: 15. Особей в популяции: 15.				
Вероятность мутации \ Процент родителей	0.2	0.4	0.6	0.8
0.2	39,066	35,742	22,81	54,021
0.4	41,931	38,062	45,97	58,187
0.6	50,992	39,745	47,071	50,228
Поколений: 20. Особей в популяции: 50.				
Вероятность мутации \ Процент родителей	0.2	0.4	0.6	0.8
0.2	10,296	11,096	25,31	35,118
0.4	6,469	11,096	11,225	32,22
0.6	6,469	11,225	17,513	33,994
Поколений: 25. Особей в популяции: 20.				
Вероятность мутации \ Процент родителей	0.2	0.4	0.6	0.8
0.2	17,925	26,338	53,781	42,682
0.4	25,972	44,61	45,127	50,228
0.6	17,925	30,088	44,539	42,682
Поколений: 50. Особей в популяции: 50.				
Вероятность мутации \ Процент родителей	0.2	0.4	0.6	0.8
0.2	0	6,469	25,192	42,58
0.4	0	11,225	25,192	33,994
0.6	0	5,823	25,192	32,132
Поколений: 50. Особей в популяции: 25.				
Вероятность мутации \ Процент родителей	0.2	0.4	0.6	0.8
0.2	0	33,194	55,824	49,76
0.4	6,296	35,253	44,249	57,012
0.6	6,469	27,751	40,79	58,131
Поколений: 50. Особей в популяции: 100.				
Вероятность мутации \ Процент родителей	0.2	0.4	0.6	0.8
0.2	0	0	16,901	44,789
0.4	0	6,469	25,31	38,181
0.6	0	5,823	30,63	28,638

Для получения приведенных в таблице параметров были сгенерированы случайные технологические карты (рассмотрено 10 экспериментов, где необходимо выполнить от 5 до 35 операций).

После чего был проведен поиск оптимального решения методом перебора, составлен план для реализации каждой из сгенерированных технологических карт. Оценив те же самые технологические карты разработанным генетическим алгоритмом с различными гиперпараметрами, были собран значения среднеквадратического отклонения длительности выполнения плана в зависимости от значения параметров. Эти значения были занесены в таблицу 1. Из-за небольшого количества операций в плане, подбор проходил примерно со скоростью 25 особей в секунду, при этом для теста с наибольшими количеством поколений и размером популяции тест занимал 20 минут.

После проведения сравнения полученных результатов были получены следующие параметры: было принято решение скрещивать все особи (коэффициент скрещивания = 1) с вероятностью мутации равной 0,4 и процентом родителей равным 0,2.

### **3.2. АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ**

Для тестирования полученного алгоритма в качестве критерия оптимизации задавались различные целевые функции, такие как:

- общая плановая длительность выполнения работ;
- минимизация длительности такта сборочной линии (чем меньше длительность такта, тем короче цикл производства, а значит может быть произведено больше продукции);
- максимизация загрузки сборочной линии (избавление от простоев и количественное уменьшение незавершенного производства);
- максимизация загрузки персонала (оптимизация общей численности сотрудников производственной единицы с условием максимизации вовлеченности трудовых ресурсов).

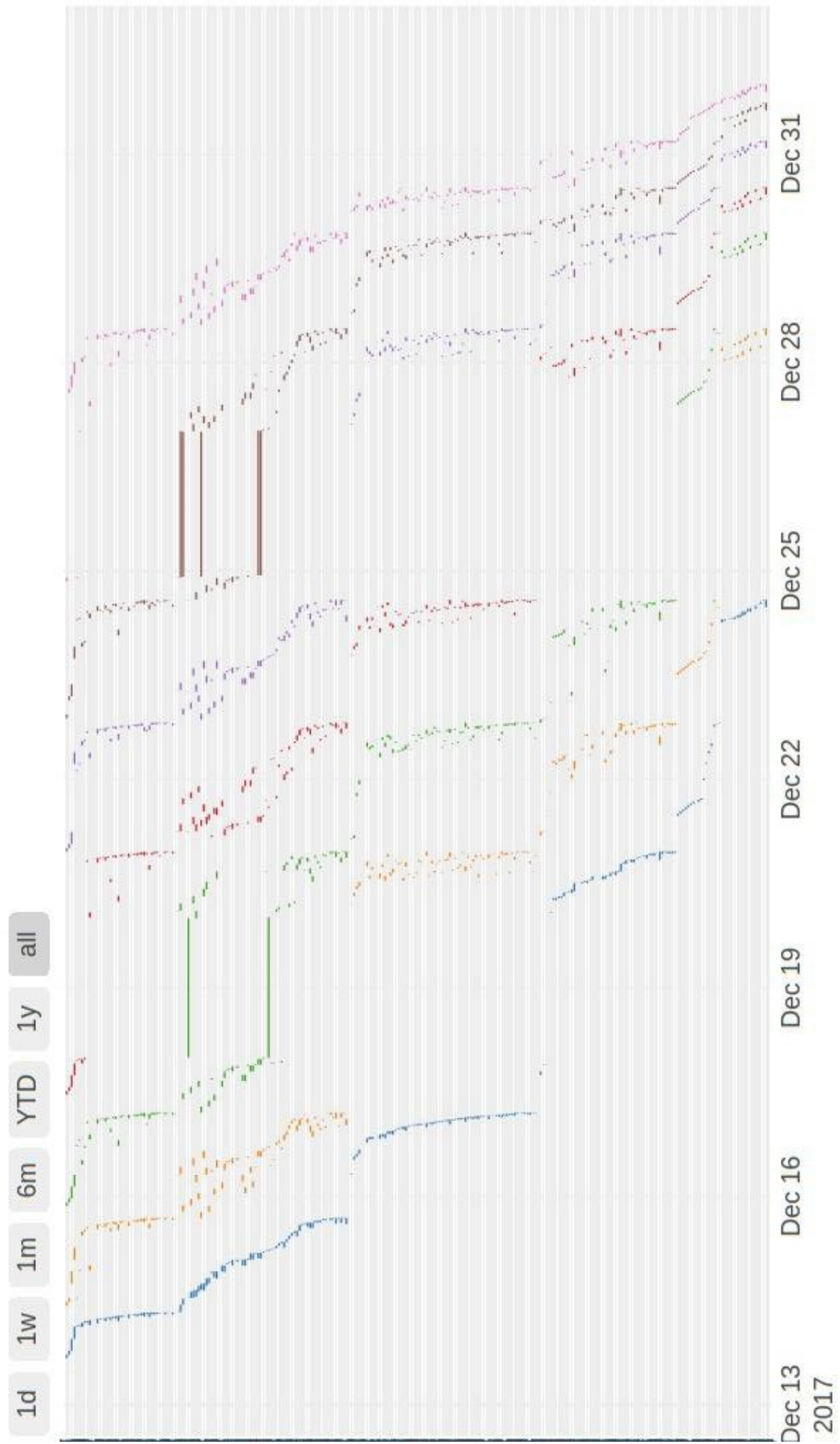


Рисунок 14 – Диаграмма Ганта с результатами работы разработанного алгоритма

Каждый из описанных целевых критериев был протестирован с использованием различных входных данных. В каждом из случаев план, сформированный имитационной моделью, был оптимизирован и длительность выполнения всего плана уменьшалась, однако варьировалась (рисунок). Таким образом, обобщенный подход к работе с входными данными позволяет использовать один алгоритм оптимизации для различных целевых функций, что, в свою очередь, расширяет возможности алгоритма и, например, рассчитать себестоимость работы производственной единицы или оценить экономически целесообразность изменения числа трудовых ресурсов.

Тестирование разработанного алгоритма проводилось в два этапа:

- 1) Анализ работы на основе тестовых данных для наиболее распространенной задачи SALBP, целью которого было обнаружение ошибок и неточностей в работе алгоритма, отладка и оценка эффективности предложенного подхода;
- 2) Тестирование с использованием данных реального предприятия, с целью последующего сравнения полученных результатов с производственным планом, разработанным технологом предприятия с использованием внутренних систем планирования.

Прежде всего возникла необходимость тестирования алгоритма на массиве тестовых данных. В качестве такого массива был использован набор данных из открытого источника [23], где приводится информация для алгоритмов, направленных на решения задач SALBP.

Для проведения тестирования были выбраны данные для задачи, известной под названием Вихеу. Для них характерны следующие особенности:

- значения данных заданы построчно;
- каждая строка содержит исключительно натуральные числа и следующие разделительные символы:

- «,» – разделяет пары узлов, индексы привязки значений<sup>3</sup> объектам;
  - «:» – разделяет индекса и непосредственно значение;
  - «;» – разделяет различные значения.
- допускается использование дополнительных пробелов, разделяющих элементы.

Данные в массиве представлены блоками, список которых приведен далее:

- количество производственных задач;
- количество станций поточной линии;
- длительность выполнения работ на каждой станции;
- причинно-следственные связи между операциями;
- параллельные станции – посты поточной линии, которые дублируются и позволяют выполнять операции параллельно;
- оснащение каждой станции;
- связи операций в технологические карты;
- привязки операции к станциям;
- ряд дополнительных, необязательных параметров, которые не были задействованы в тестировании, например, себестоимость работы станции.

Таким образом, данные содержат полный набор информации, необходимой для тестирования алгоритма.

В работе [24] автор, являющийся создателем этого набора данных, указывает, что традиционный подход SALBP, протестированный на этих данных показал в качестве результата такт поточной линии, равный 34 при условии работы одной поточной линии с 10-тью постами (рисунок 15).

---

<sup>3</sup> Когда вместо имени необходимо указать значение определенного параметра  $x$ , применяется обозначение  $val(x)$ .

Название тестов	Кол-во операций/постов	Лучший известный такт	Такт решения при одной единице	Такт при пяти единицах
Buxey	29/10	34	30	34
Bowman	8/5	20	17	20

Рисунок 15 – Характеристики известных решений для выбранных тестовых данных

На основании оригинального набора данных был протестирован и разработанный алгоритм, в качестве технологических ресурсов также были выбраны 10 постов одной поточной линии. В качестве целевой функции оптимизации был выбран критерий минимизации длительности такта сборочной линии. Результатом стал такт, равный 34, что подтверждает эффективность предложенного способа планирования и метода. Генерация ответа на такую задачу на компьютере с шестью физическими ядрами занимает примерно три минуты. Это оказалось в шесть раз быстрее, чем планирование с использованием полного перебора.

Разница в длительности была достигнута за счет того, что алгоритмы по-разному распоряжаются имеющимися трудовыми ресурсами. Алгоритм, предлагаемый в данной работе, предполагает возможность перераспределения сотрудников между станциями поточной линии в процессе выполнения работ. При увеличении количества необходимых к изготовлению продуктов длительность такта возрастала и достигла целевого такта 34 при количестве продуктов, равного 10-ти.

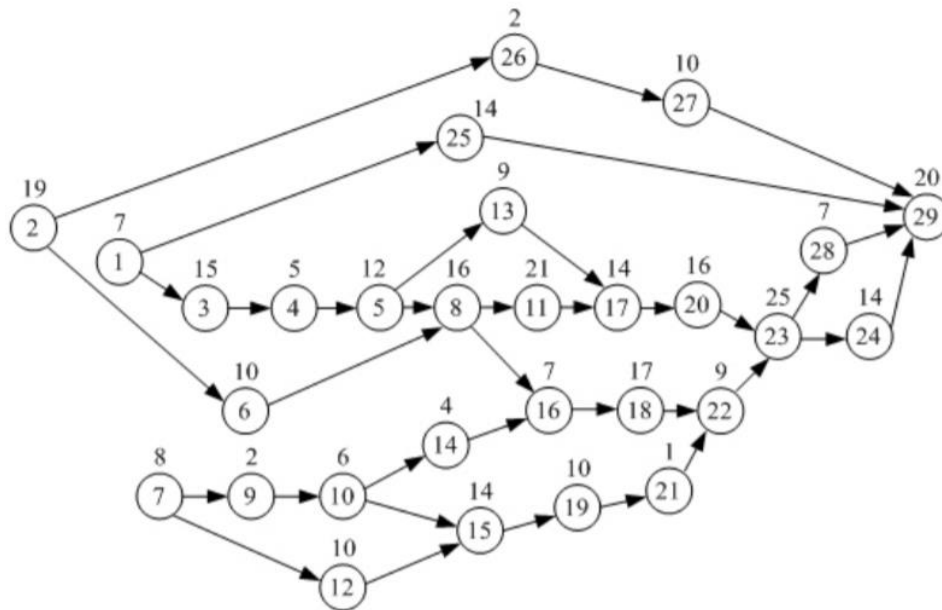


Рисунок 16 – Дерево технологической карты, на базе которой был проведен первый этап тестирования [24]

Аналогичный результат был получен и для тестовых данных, приведенных в [25].

Вторым этапом тестирования стало сравнение результатов работы алгоритма с реальным производственным планом, при условии использования одинаковых входных значений.

Данные для проведения тестирования были предоставлены предприятием АО «Диаконт». Они содержат полный перечень входных данных, необходимых для планирования разработанным алгоритмом, а также пример реального производственного плана, сформированного на основе этих данных без использования предлагаемого алгоритма оптимизации.

Первым этапом было проведено моделирование плана для предложенных входных данных без использования оптимизации, которое соответствует плану, принятому в работу на предприятии. Его результаты приведены на рисунке 17.



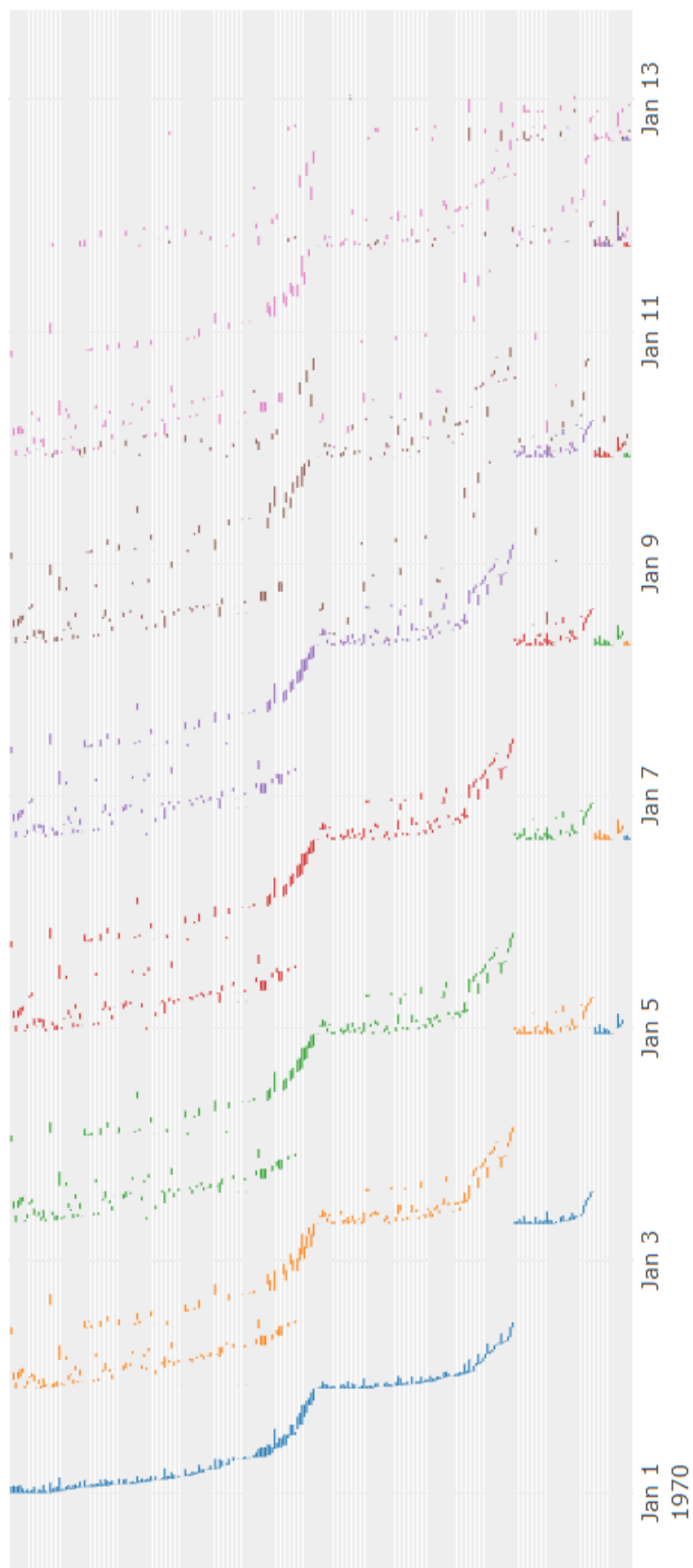


Рисунок 17 – Диаграмма Ганта производственного плана на основании реальных данных предприятия

Согласно приведенному плану, перечень работ с использованием имеющейся производственной мощности выполняется в течение 17-ти календарных дней.

Далее сформирован производственный план на основе идентичных данных, однако с использованием генетического алгоритма оптимизации длительности производственного цикла. Отметим, что полученный план позволяет реализовать производственную программу за 13 календарных дней, что значительно высвобождает производственные ресурсы ввиду ускорения производства в 1,3 раза (рисунки 18, 19).

На рисунках 20, 21 приведен пример тестирования алгоритма на технологических картах с большим количеством возможных взаимосвязей параллельных операций. Стоит отметить, что план, сформированный с использованием разработанного алгоритма, предполагает выполнение работ на 11 часов быстрее (что составляет 14% от длительности до оптимизации), нежели план, применяющийся на производстве для аналогичных входных данных. Отсюда следует высокая степень эффективности анализа алгоритмом внутренних зависимостей в рамках технологической карты.



Рисунок 18 – Распределение результатов запуска алгоритма для данных технологической карты из 400 операций и большим количеством возможных взаимосвязей параллельных операций

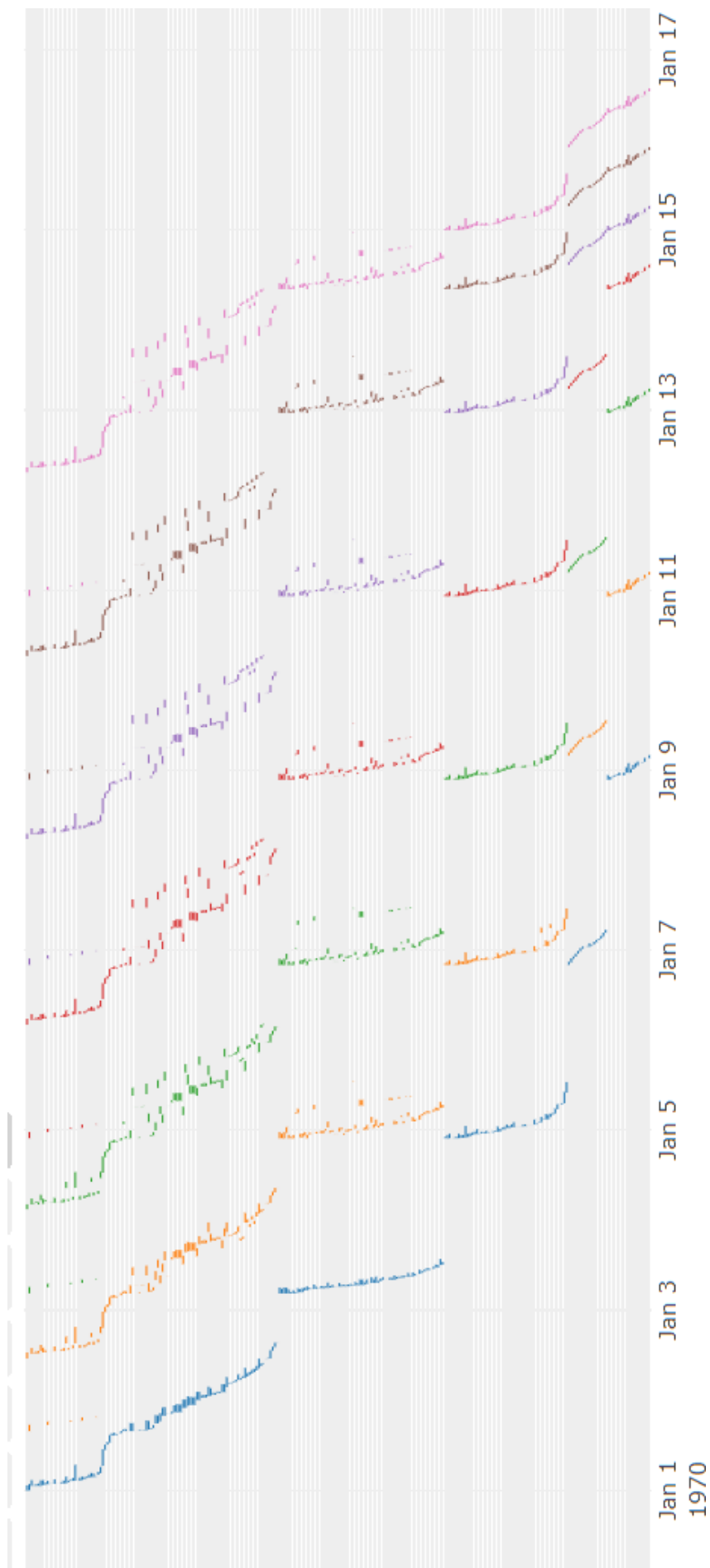


Рисунок 19 – Диаграмма Ганта производственного плана на 2000 операций, сформированного и оптимизированного с использованием разработанного алгоритма

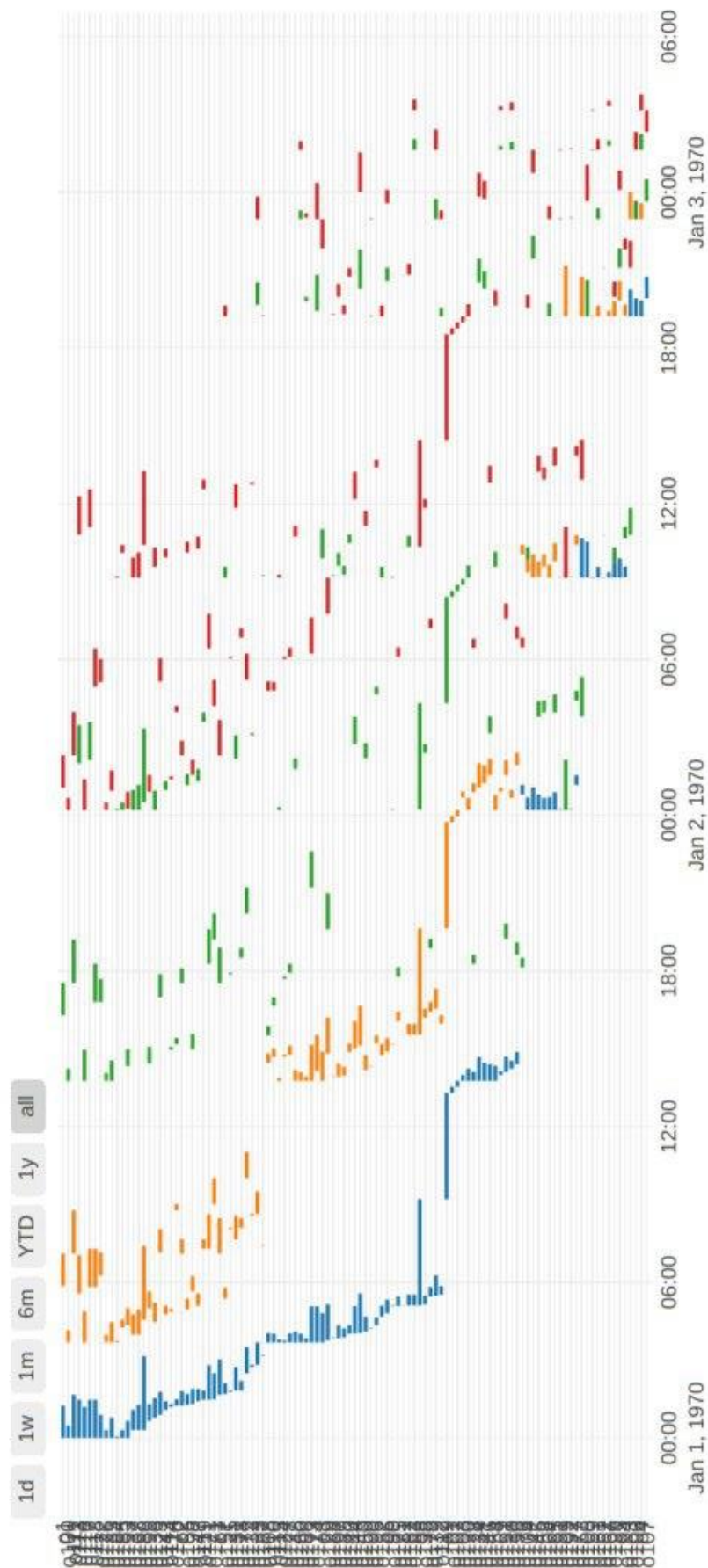


Рисунок 20 – Диаграмма Ганта производственного плана на 400 операций, сформированного и оптимизированного с использованием разработанного алгоритма

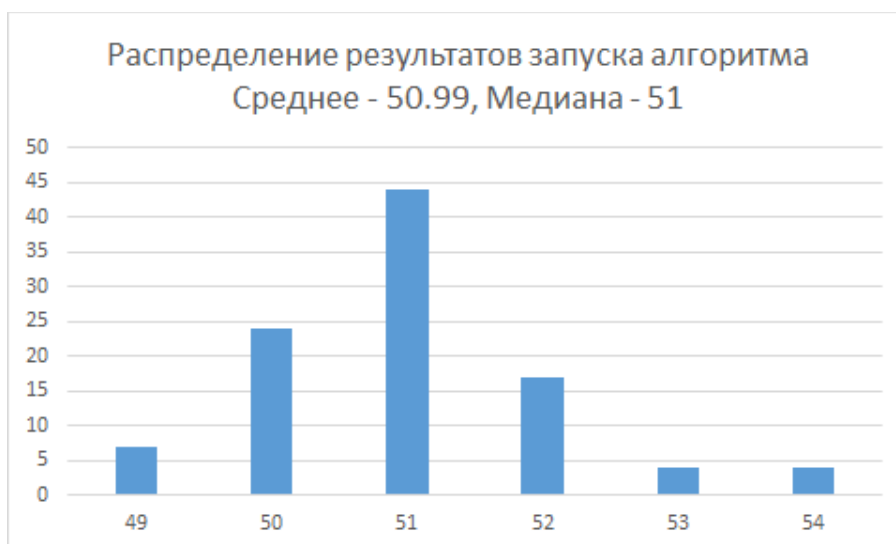


Рисунок 21– Распределение результатов запуска алгоритма для данных технологической карты из 400 операций, полученных из реального производственного плана

Проанализировав отличия полученных планов, отметим, что достичь такого расписания удалось путем иной группировки трудовых ресурсов и возможности перераспределять их в процессе обработки. Также были обнаружены операции, которые могли выполняться параллельно, но осуществлялись последовательно – алгоритм перераспределил трудовые ресурсы для обеспечения одновременного протекания работ в большинстве таких случаев.

Также алгоритм был протестирован на реальном плане для небольшой технологической карты из 8 операций, но прирост по сравнению с версией плана не наблюдалось. Это связано с тем, что для маленького плана человек способен найти оптимальный вариант самостоятельно и отсутствие в изначальных данных возможности для изменения числа людей на операцию.

Время работы для большого плана составило почти 3 часа, что несравнимо с оптимизацией для плана, которая была произведена. Таким образом, проведенное тестирование подтверждает эффективность разработанного алгоритма, а также его применимость для реальных производств.

### **ВЫВОДЫ ПО ГЛАВЕ 3**

1. Разработанный генетический алгоритм направлен на решение задачи оптимизации плана работы сборочной линии. Применение генетического алгоритма направлено на оптимизацию процесса формирования оперативного плана на этапе получения входных данных.
2. Выбор гиперпараметров генетического алгоритма был реализован с помощью генератора случайных технологических карт, который формировал планы с низким числом операций и небольшим возможным количеством сотрудников. Это позволило вычислить оптимальный вариант методом полного перебора и сравнить с выходом генетического алгоритма.
3. Тестирование, проведённое на данных для задач ALBP, а также на реальных производственных данных, показала высокую эффективность разработанного алгоритма – отметим, что улучшение результата возрастает, с ростом числа обрабатываемых операций, поскольку алгоритм направлен на работы с большим количеством взаимосвязей и предназначен для формирования планов, которые не могут быть разработаны сотрудником предприятия или более простым и универсальным методом.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения работ был разработан алгоритм оптимизации сборочных линий для крупного машиностроения. Его суть заключается в определении оптимального и субоптимальных вариантов производственных планов, где минимизирована длительности производственных процессов тяжелого машиностроения при работе со сборочными линиями с учетом реального оснащения предприятия производственными ресурсами, а также наличием причинно-следственных связей между производственными операциями. Основными задачами, решенными в ходе выполнения работы, являются:

- моделирование и оптимизация процесса распределения производственных операций по рабочим постам сборочной линии;
- моделирование и оптимизация процесса распределения трудовых ресурсов предприятия между рабочими постами сборочной линии.

Был проведен обзор существующих подходов к организации производственной деятельности, выявлены основные проблемы и актуальные задачи индустрии. Установлено, что одной из актуальных проблем в области создания ПО для предприятий тяжелого машиностроения является разработка алгоритма планирования и оптимизации для сборочных линий со свободным тактом, который позволит реализовать задачу сведения к непрерывному потоку производства.

Также был проведен обзор существующих алгоритмических решений схожих производственных задач с целью определения наиболее перспективного подхода к разработке алгоритма. В результате был выбран генетический алгоритм как инструмент реализации частной задачи, смежной с областью ALBP и теорией расписаний, как наиболее гибкий и эффективный подход к решению такого типа задач.

На основании выводов, полученных в процессе математической формализации и доказательства NP-полноты поставленной задачи, разработан генетический алгоритм, минимизирующий плановую длительность такта

сборочной линии, обеспечивающий возможность планирования и оптимизации длительности необходимых вспомогательных рабочих операций (дополнительная наладка, контрольные операции), проведения наладки сборочной линии за короткое время и при минимальных затратах, а также исключение возможности проведения избыточных вспомогательных операций. Для реализации тестирования и проведения экспериментов было создано окружение, включающее:

- базу данных, предназначенную для хранения сериализованных входных и выходных данных алгоритма;
- подсистему имитационного моделирования ресурсов производственной единицы, предназначенную непосредственно для формирования производственных планов, которые необходимо оптимизировать.

По результатам разработки было проведено тестирование алгоритма на тестовой подборке данных, а в последствии – на реальных производственных данных, предоставленных предприятием механообработки и сборки крупных технических средств.

Таким образом были решены задачи, поставленные в рамках данной работы, получены результаты, актуальные для последующего внедрения на существующие предприятия.

Преимущественное влияние на процессе разработки алгоритма оптимизации сборочных процессов оказал поиск решения проблем планирования сборки, определение подходов к созданию соответствующих задаче частных моделей, учитывающих индивидуальные особенности производств.

Предложенный алгоритм позволяет внедрить на предприятии необходимую организацию труда для реализации метода сведения к непрерывному потоку, применяемый для мелко- и среднесерийных предприятий с целью формирования прерывно-поточной сборочной линии. Это достигается следующими преимуществами алгоритма:



- стабилизация технологических процессов и формализация периодичности выпуска изделий путем контроля свободного такта линии и оптимизации длительности производственного цикла;
- минимизация и условное постоянство значений объема незавершенного производства ввиду устранения нерабочих перерывов в работе, а также контроля заделов на сборочных линиях;
- увеличить число рабочих операций, выполняемых в рамках рабочей смены и максимизировать насыщенность рабочих интервалов сотрудников путем оптимизации последовательности выпуска изделий;
- оптимизировать длительность цикла сборки и возможность уменьшить количество необходимых производственных площадей.

Таким образом, поставленные задачи были достигнуты в полном объеме, полученный результат характеризуется практической значимостью, возможностью реального применения на практике, а также имеет научную значимость как новое решение частной задачи планирования и оптимизации работы сборочной линии на производстве.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мюллер Э., Шенк М., Вирт З. Планирование и эксплуатация промышленных предприятий: Рабочие методики для адаптивного, сетевого и ресурсосберегающего предприятия / Пер. с нем. – М.: Альпина Паблишер, 2017. – 978 с.
2. О'Лири Д. ERP системы. Современное планирование и управление ресурсами предприятия. Выбор, внедрение и эксплуатация / Пер. с англ. Водянова Ю.И. – М.: Вершина, 2004. – 272 с.
3. Salveson, M.E. The assembly line balancing problem / M.E. Salveson // The J. of Industrial Engineering. – 1955. – Vol. 6, № 3. – P. 18–25.
4. Baybars, I. A survey of exact algorithms for the simple assembly line balancing problem / I. Baybars // Management Science. – 1986. – Vol. 32. – P. 909–932.
5. Scholl, A. Balancing and sequencing of assembly line / A. Scholl; Second ed. – Heidelberg : Physical-Verlag, 1999. – 532 p.
6. Kottas, J.F. A cost oriented approach to stochastic line balancing / J.F. Kottas, H.S. Lau // AIIE Transactions. – 1973. – Vol. 5. – P. 164–171.
7. Reeve, N.R. Balancing stochastic assembly lines / N.R. Reeve, W.H. Thomas // AIIE Transactions. – 1973. – Vol. 5. – P. 223–229.
8. Kottas, J.F. A cost oriented approach to stochastic line balancing / J.F. Kottas, H.S. Lau // AIIE Transactions. – 1973. – Vol. 5. – P. 164–171.
9. Silverman, F.N. A cost- based methodology for stochastic line balancing with intermittent line stoppages / F.N. Silverman, J.C. Carter // Management Science. – 1986. – Vol. 32. – P. 455–463.
10. Shin, D. An efficient heuristic for solving stochastic assembly line balancing problem / D. Shin // Computers and Industrial Engineering. – 1990. – Vol. 18. – P. 285–295.
11. Shin, D. Uniform assembly line balancing with stochastic task times in just-in-time manufacturing / D. Shin, H. Min // International J. of Operations and Production Management. – 1991. – Vol. 11, № 8. – P. 23–34.

12. Gen, M. Solving fuzzy assembly-line balancing problem with genetic algorithms / M. Gen, Y. Tsujimura, E. Kubot // Computers inc. Engineering. – 1995. – Vol. 29. – P. 543–547.
13. Gen, M. Fuzzy assembly line balancing using genetic algorithms / M. Gen, Y. Tsujimura, Y. Li // Computers inc. Engineering. – 1996. – Vol. 31. – P. 631–634.
14. Rabbani M., Considering the conveyer stoppages in sequencing mixed-model assembly lines by a new fuzzy programming approach // International J. of Advance Manufacture Technology. – 2010. – Vol. 10. – P. 170–180.
15. Brucker P., Scheduling Algorithms / Springer Science & Business Media. – 2013 г. – 367 p.
16. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи / Пер. с англ. М.: Мир, 1982.
17. Зак Ю.А. Решение обобщенной задачи Джонсона с ограничениями на сроки выполнения отдельных заданий и времена работы машин. Ч. 1. Точные методы решения // Проблемы управления. – 2010. – № 3. – С. 17–25.
18. Johnson S.M. Optimal-two-and tree stage production schedules with setup times included // Research Logistics Quarterly. – 1954. – No. 1. – P. 61–68.
19. К. Г. Розенков, С. А. Федосеев, А. А. Вдовиченко Метод настройки генетического алгоритма для задачи планирования поточного производства. Прикладная математика и вопросы управления. – 2015. – № 2. – С. 49–60.
20. Н.С. Гарколь, М.В. Гунер Применение генетических алгоритмов в решении задач планирования производства. Вестник Томского государственного университета. – 2012. – № 2. – С. 72–79.
21. Кузин Б.И., Юрьев В.Н., Шахдинаров Г.М. Методы и модели управления фирмой: учеб. пособие. СПб.: Питер, 2001. 432 с.
22. Bukchin Yossi, Raviv Tal. Constraint Programming for Solving Various Assembly Line Balancing Problems. – Omega – 2018. – №78 – P. 57–68.
23. Assembly Line Balancing. Data sets & Research topics, [Электронный ресурс] URL: <https://assembly-line-balancing.de/> (дата обращения 23.02.2020).

24. Armin Scholl Data of Assembly Line Balancing Problems. Technische Hochschule Darmstadt, 1993. 32 p.

25. Otto, A.; Otto, C.; Scholl, A. Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing. European Journal of Operational Research 228/1, 2013. P 33–45.