

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

А.И. СЕРДЮК, Р.Р. РАХМАТУЛЛИН, А.П. ЗЕЛЕНИН

**МЕТОД ЦИКЛОГРАММ
В ИССЛЕДОВАНИИ
ГИБКИХ ПРОИЗВОДСТВЕННЫХ
ЯЧЕЕК.
МОДЕЛИ И АЛГОРИТМЫ**

МОНОГРАФИЯ

Рекомендовано к изданию Ученым советом
государственного образовательного учреждения
высшего профессионального образования
«Оренбургский государственный университет»

Оренбург 2009

УДК 621:681.5
ББК 34.4 – 05
С 19

Рецензенты

доктор технических наук, профессор А.Н. Поляков,

доктор технических наук, профессор В.Н. Тарасов

С 19 **Сердюк, А.И.**
Метод циклограмм в исследовании гибких производственных ячеек.
Модели и алгоритмы : монография / А.И. Сердюк, , Р.Р. Рахматуллин, А.П. Зеленин. – Оренбург : ГОУ ОГУ, 2009. – 208 с.
ISBN

Рассмотрена проблема создания инструментальных средств компьютерного моделирования высокоавтоматизированного машиностроительного производства.

Разработан комплекс логических, алгоритмических, аналитических и имитационных моделей системы компьютерного моделирования, адекватно отражающий иерархическое взаимодействие узлов, агрегатов, станков и станочных комплексов на уровнях технологической операции и технологического перехода. Особое внимание уделено глубине и универсальности проработки алгоритма моделирования, что дает возможность конфигурировать модели гибких производственных ячеек для сквозного использования на различных этапах их жизненного цикла – от предпроектных исследований до варьирующихся режимов эксплуатации.

В качестве входных данных используются технические характеристики элементов производственной системы, что позволяет отслеживать в выходных показателях эффективности влияние работы каждого отдельного агрегата.

Для научных работников и инженеров, занятых вопросами создания и внедрения компьютерно управляемых комплексов оборудования. Книга может быть полезна студентам старших курсов технических специальностей

ББК 34.4 – 05

ISBN

© Сердюк А.И.,
Рахматуллин Р.Р.
Зеленин А.П., 2009
© ГОУ ОГУ, 2009

Содержание

Введение	6
1 Состояние проблемы. Цель и задачи исследования	7
1.1 Актуальность гибкой автоматизации машиностроительного производства.....	7
1.1.1 Историческая справка	9
1.1.2 Современное состояние проблемы в мире	12
1.1.3 Состояние и прогноз развития гибкой автоматизации машиностроения России и стран СНГ	13
1.2 Аппарат для расчета ГПЯ	16
1.2.1 Аналитические расчеты	16
1.2.2 Теория массового обслуживания	18
1.2.2.1 Аппарат теории массового обслуживания	19
1.2.2.2 ГПЯ как система массового обслуживания	20
1.3 Теория расписаний	22
1.4 Приоритетные правила в АСУ ГПЯ	24
Выводы	28
2 Оценка эффективности создания новой техники	30
2.1 Зависимость срока окупаемости ГПЯ от технических, экономических и организационных решений	30
2.2 Влияние стоимости изготавливаемой продукции на срок окупаемости ГПЯ	32
2.3 Влияние величины календарных простоев на срок окупаемости ГПЯ.....	34
2.4 Влияние сменности работы оборудования на срок окупаемости ГПЯ.....	35
2.5 Влияние внутрисменных простоев оборудования на срок окупаемости ГПЯ	37
2.6 Влияние загрузки оборудования ГПЯ на срок окупаемости капитальных затрат	39
2.7 Влияние производительности ГПЯ на срок ее окупаемости	40
2.8 Анализ эффективности различных путей сокращения срока окупаемости ГПЯ	41
Выводы	45
3 Пути повышения производительности и загрузки оборудования ГПЯ	46
3.1 Циклограммы работы ГПЯ на уровне технологической операции	46
3.2 Циклограммы работы ГПЯ на уровне технологического перехода	49
3.2.1 Циклограммы автоматической смены инструментов	50
3.2.2 Циклограммы автоматической замены инструментов	55
3.2.3 Циклограммы автоматической доставки режущих инструментов	58
3.3 Режимы резания и производительность ГПЯ	61
Выводы	66
4 Метод циклограмм в компьютерном моделировании ГПЯ	69
4.1 Необходимость комплексного подхода к моделированию	69
4.2 Метод циклограмм	73
4.3 Общий подход к разработке компьютерной модели ГПЯ	79

4.3.1 Постановка задачи	79
4.3.2 Построение алгоритма моделирования	79
4.3.3 Организация моделирования	81
4.4 Критерии оценки результатов моделирования	82
4.4.1 Средство визуализации показателей эффективности ГПЯ	82
4.4.2 Вывод зависимости для расчета срока окупаемости ГПЯ	84
4.5 Особенности статистических исследований ГПЯ в методе циклограмм...	89
4.5.1 Алгоритм формирования вариантов сменного задания	90
4.5.2 Статистическая обработка результатов моделирования	92
Выводы	95
5 Модели ГПЯ на разных уровнях иерархии	96
5.1 Модели ГПЯ на уровне технологической операции	96
5.1.1 Формализованное описание сменного задания	96
5.1.2 Формализованное описание технологических процессов	103
5.1.3 Исходное состояние системы	105
5.1.4 Правила обслуживания станков.....	108
5.1.4.1 Правило «с первой заявки в очереди»	109
5.1.4.2 Правило «по приоритетам станков»	111
5.1.4.3 Правило «по минимальной трудоемкости заготовок в накопителе»...	112
5.1.4.4 Правило «по min остаточному времени обработки заготовок в рабочей зоне»	113
5.1.4.5 Правило «по min числу заготовок в пристаночном накопителе»	114
5.1.4.6 Правило «по минимальному времени транспортной операции»	115
5.1.5 Аналитические модели длительности транспортных операций	117
5.2 Модели ГПЯ на уровне технологического перехода	121
5.2.1 Схема обобщенного алгоритма моделирования	121
5.2.2 Схема алгоритма моделирования многоцелевого станка ГПЯ	124
5.2.3 Логика работы устройств АСИ и АЗИ станка	125
5.2.4 Формальное описание процесса моделирования	127
5.2.5 Функциональные процедуры	131
Выводы	132
6 Программные продукты, основанные на методе циклограмм	134
6.1 Интегрированная система расчета и моделирования ГПЯ механообработки «Каскад»	134
6.1.1 Уровни задач, решаемых с помощью системы «Каскад»	134
6.1.2 Области использования системы	138
6.2 Программа многокритериальной оценки проектных решений в гибких производственных ячейках механообработки «Fania»	142
6.2.1 Входные данные	142
6.2.2 Результаты моделирования	145
6.2.3 Пример применения программы	147
6.3 Моделирование работы ГПЯ с несколькими транспортными средствами. Программа «PolyTrans»	151

6.3.1 Ограничения и допущения на моделирование работы нескольких робокар	152
6.3.2 Подход к моделированию ГПЯ с несколькими транспортными средствами	153
6.3.3 Результаты тестирования программы «PolyTrans»	156
6.4 Моделирование работы участков традиционного производства.	
Программа «Расписание»	159
Заключение	171
Список использованных источников	172
Приложение А Структурные и функциональные схемы системы «Каскад»	178
Приложение Б Фрагмент исходного кода программы «Fania». Правила обслуживания	198
Приложение В Фрагмент исходного кода программы «Расписание».	
Описание работы оборудования	208

Введение

Одним из путей ликвидации технологического отставания страны служит техническое перевооружение предприятий базовых отраслей промышленности на основе компьютерно управляемых технологических комплексов нового поколения. Согласно ГОСТ 26228-90 такие комплексы называются гибкими производственными ячейками (ГПЯ) и системами (ГПС).

Показателен опыт США, где к 2003 году ГПЯ используются на 73 % фирм с числом работающих от 100 человек. Появился термин – «экономное производство», или «экономная автоматизация» – обозначающий ГПЯ, работающие в течение 140 ч в неделю или по 20 ч в сутки в безлюдном режиме. За счет интенсивной работы системы высокие первоначальные затраты на создание окупаются за короткий промежуток времени.

В отечественной практике накоплен опыт создания ГПЯ, когда в 80-х годах XX века на государственном уровне был предпринят масштабный эксперимент по модернизации отечественного производства в соответствии с мировыми тенденциями промышленного развития. Были созданы образцы систем, опередившие по уровню гибкости и степени автоматизации зарубежные аналоги: например, АЛПЗ-1 и АЛПЗ-2. Был налажен выпуск производственных систем серий АСК, АСВ, «Талка». ГПЯ создавались и на предприятиях Оренбуржья (в Оренбурге - на ПО «Стрела», в Орске - на ЮУМЗ и ОЗТП).

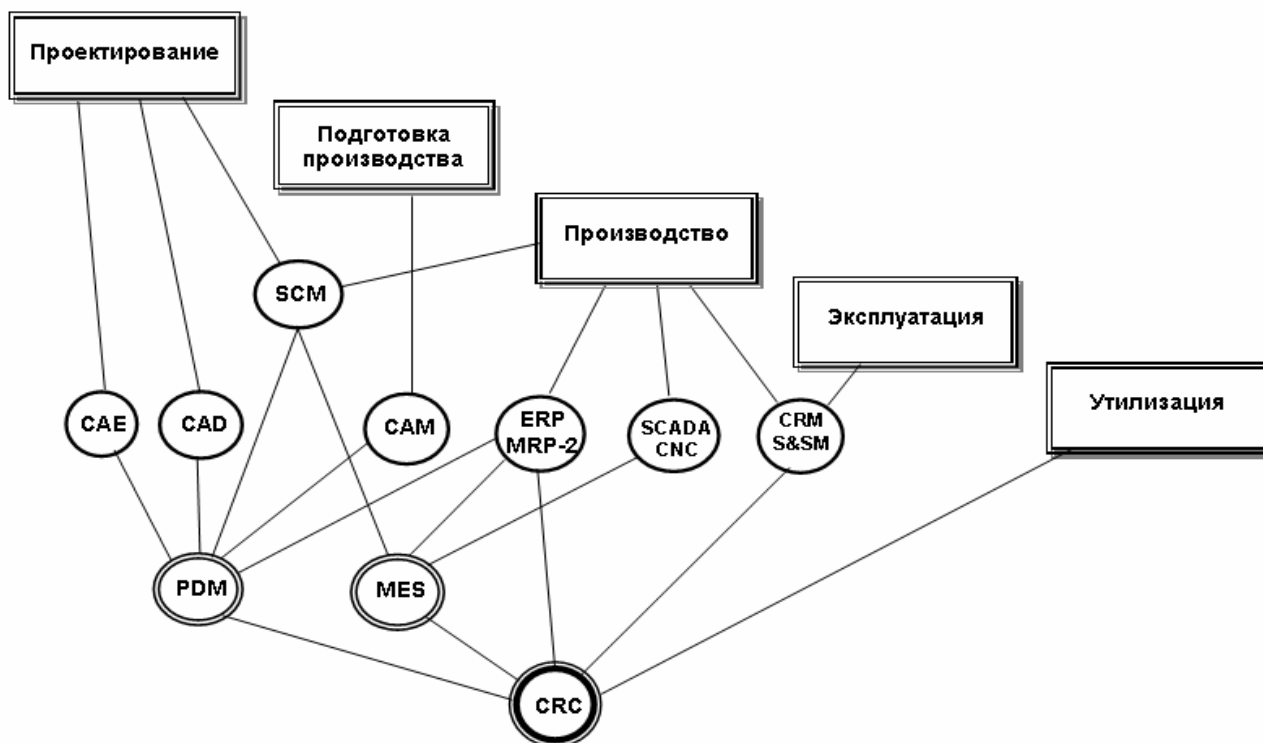
В немалой степени накопленный опыт использования ГПЯ оказался отрицательным: сказалась низкая надежность отечественных ЭВМ, используемых в системах управления, неподготовленность персонала, другие причины. Вместе с тем открылся пласт научных проблем, требующих принципиально нового решения. Причем большая часть проблем связана не с техническими или технологическими решениями, а с отсутствием общей методологии проектирования и системной поддержки производственных систем нового поколения.

Актуальность проблемы подтверждается финансированием работы в данном направлении в рамках выполнения госбюджетной НИР № 1.4.06 «Разработка методологии создания высокоэффективных производственных систем нового поколения с заданными свойствами», а также поддержкой работы грантом № 613 «Исследование механизма и закономерностей перехода от технического задания к техническому предложению на создание гибких производственных ячеек» аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2009-2010 годы)».

1 Состояние проблемы. Цель и задачи исследования

1.1 Актуальность гибкой автоматизации машиностроительного производства

Любое изделие современного производства сопровождается сложным переплетением информационных потоков, поддерживаемых компьютерными системами автоматизации проектирования, подготовки производства, изготовления, эксплуатации и утилизации (рисунок 1.1) [30].



SCM (Supply Chain Management) – управление цепочками поставок

CAE (Computer Aided Engineering) – автоматизированные расчеты и анализ

CAD (Computer Aided Design) – автоматизированное проектирование

CAM (Computer Aided Manufacturing) – автоматизированная технологическая подготовка производства

ERP (Enterprise Resource Planning) – планирование и управление предприятием

MRP-2 (Manufacturing Requirement Planning) – планирование производства

SCADA (Supervisory Control And Data Acquisition) – диспетчерское управление производственными процессами

CNC (Computer Numerical Control) – компьютерное числовое управление

CRM (Customer Relationship Management) – управление взаимоотношениями с заказчиками

S&SM (Sales and Service Management) – управление продажами и обслуживанием

PDM (Product Data Management) – управление проектными данными

MES (Manufacturing Execution System) – производственная исполнительная система

CPC (Collaborative Product Commerce) – совместный электронный бизнес

Рисунок 1.1 - Этапы жизненного цикла производственных изделий и системы их автоматизации

Информационная интеграция и системная поддержка жизненного цикла продукции отнесены к критическим технологиям Российской Федерации [2,3,4]. Появились вузовские специальности подготовки инженеров в области ИПИ (CALS) технологий [31].

Востребованность систем автоматизации разных этапов и уровней иерархии предполагает использование производственного оборудования с числовым программным управлением (ЧПУ). Таким оборудованием служат роботизированные технологические комплексы (РТК) и гибкие производственные модули (ГПМ), способные обеспечить практически полный цикл изготовления изделий. Управляющие программы (УП) для систем ЧПУ становятся продуктом последовательного использования систем автоматизированного проектирования (САД) и автоматизированной технологической подготовки производства (САМ), интегрируемых в иерархические системы управления проектными данными (РДМ).

Передача автоматически сгенерированных УП от компьютера в системы ЧПУ станков становится выгодной для предприятий: можно экономить на работе соответствующих подразделений, обслуживающих многочисленные системы ЧПУ разных поколений. РТК и ГПМ становятся, по сути, таким же периферийным оборудованием компьютеров, как сканеры, принтеры, плоттеры. Включение в состав компьютерно управляемых комплексов оборудования транспортных средств и складов (рисунок 1.2) означает появление качественно новых средств производства, согласно ГОСТ 26228-90 определяемых как гибкие производственные системы (ГПС) [13].

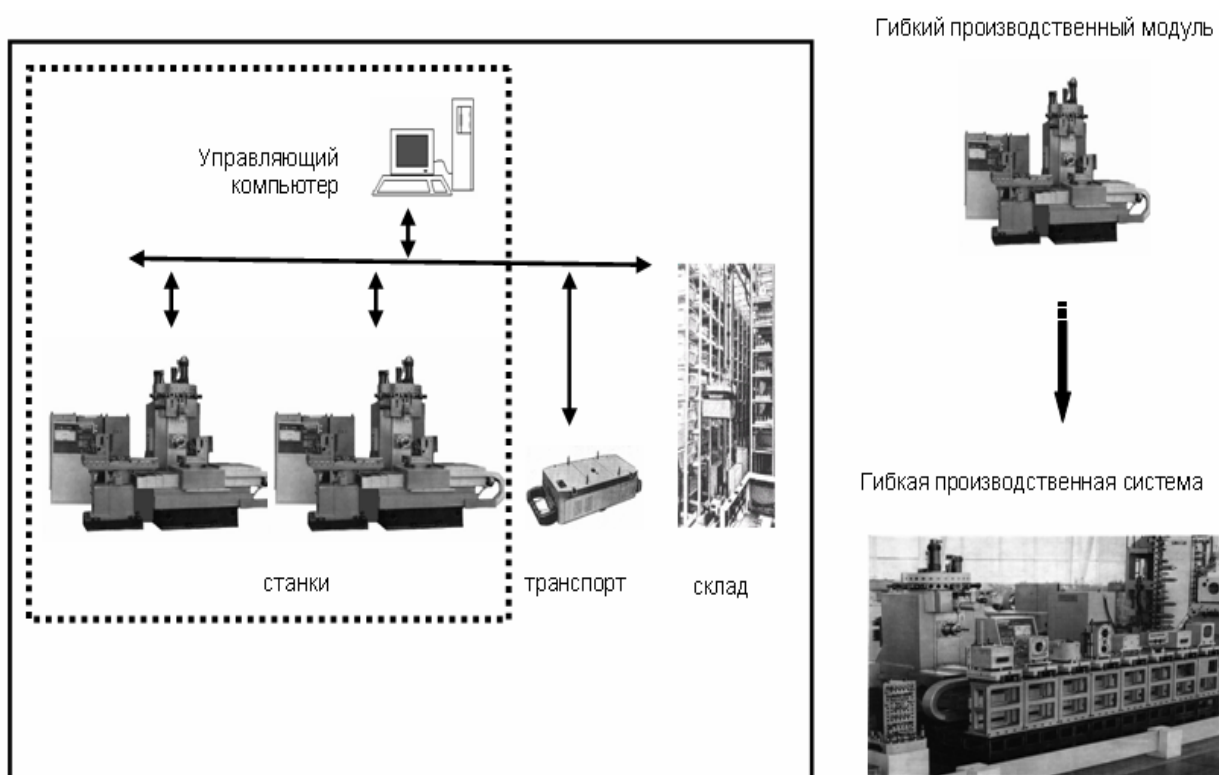


Рисунок 1.2 – Схема поэтапного формирования гибкой производственной системы

Современные ГПЯ позволяют выпускать продукцию под конкретные заказы в течение 20 ч в сутки в безлюдном режиме и 140 ч в неделю [86].

Производство перестает быть серийным со всеми свойственными ему недостатками – «незавершенкой», замораживающей оборотные средства, неритмичной сборкой, мало способствующей стабильности качества продукции, непроизводительными простоями оборудования. Сегодня нет технических проблем создания подобных производств: даже отечественные производители предлагают создание ГПЯ «под ключ» [44].

Вместе с тем последние 15 лет практически отсутствуют отечественные публикации о разработках в области гибкой автоматизации машиностроительного производства. С 1996 г. ГПЯ не представляются в списке приоритетных направлений научно-технического развития [37]. В чем дело? Может быть, есть более эффективное направление развития средств производства, чем ГПЯ?

1.1.1 Историческая справка

Гибкая производственная система – «управляемая средствами вычислительной техники совокупность технологического оборудования, состоящая из разных сочетаний гибких производственных модулей и (или) гибких производственных ячеек, автоматизированной системы технологической подготовки производства и систем обеспечения функционирования, обладающая свойством автоматизированной переналадки при изменении программы производства изделий, разновидности которых ограничены технологическими возможностями оборудования» [13].

Один из первых производственных комплексов оборудования, удовлетворяющих данному определению, появился в 1969 г. На одной из английских фирм был пущен в эксплуатацию управляемый от ЭВМ автоматизированный технологический комплекс «Система-24». Комплекс предназначался для изготовления деталей самолета «Торнадо» в условиях серийного производства, состоял из 8 станков типа «обрабатывающий центр», включал склад паллет с заготовками и автоматический склад режущих инструментов конвейерного типа [8, 18]. Свое название «Система-24» получила благодаря возможности работать 24 часа в сутки – две смены при участии операторов, обеспечивающих загрузку накопителей и настройку оборудования, и третью смену – в автоматическом цикле под наблюдением дежурного оператора.

В 1972 г. появились первые отечественные ГПЯ – АП1 и АУ1 [18], предназначенные для автоматической обработки валов и корпусных деталей в условиях серийного производства.

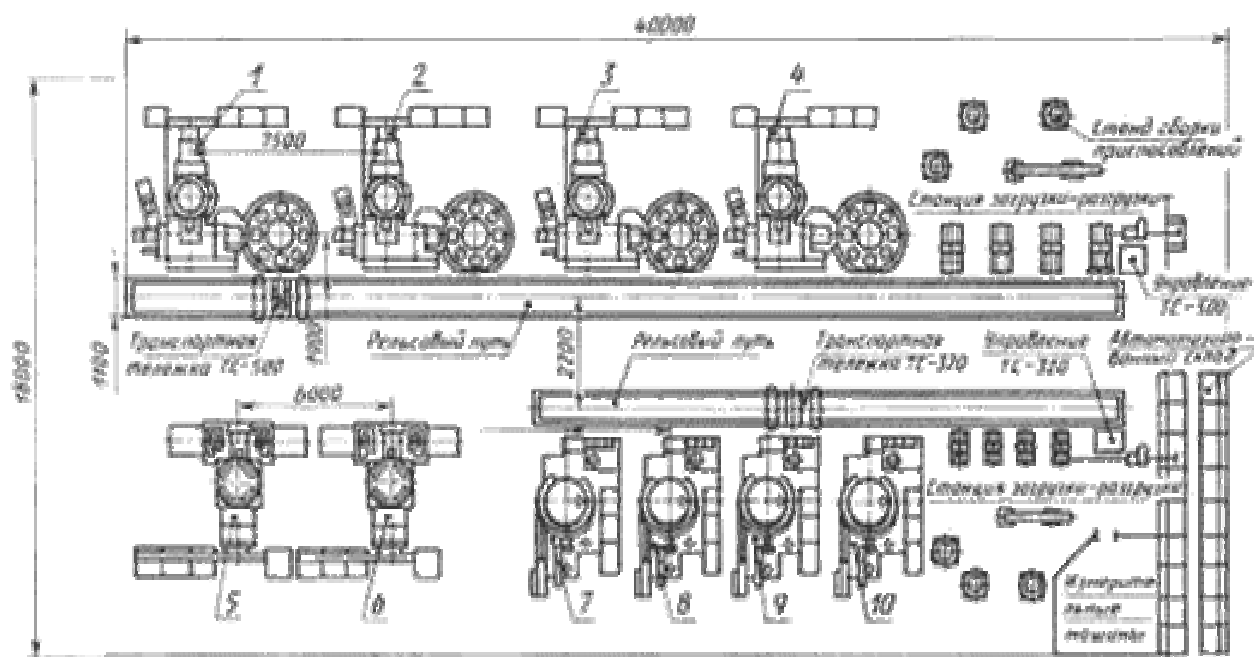
К началу 80-х годов XX века в НПО ЭНИМС уже разработаны две серии ГПЯ – серия АСК для обработки корпусных деталей, и серия АСВ – для обработки деталей типа тел вращения.

В 1981 г. в литературе появляется описание АЛПЗ-1 и АЛП 3-2 – первых отечественных ГПЯ, в которых реализована автоматизированная система инструментального обеспечения с индивидуальной заменой инструментов в магазинах станков [8]. Впервые в отечественной практике разработана и применена система кодирования режущих инструментов на оправке.

К середине 80-х годов за рубежом начинается выполнение национальных и

межгосударственных программ по гибкой автоматизации производства: программы ICAM (США), MUM, FMC (Япония), CAM-I (страны Северной Америки, США, Япония), ESPRIT, ISTEEL (страны ЕЭС) и другие [2]. Создаются научные центры по ГПЯ, например, южнокорейский институт автоматизации и системных исследований (ASRI) [29].

В середине 80-х годов в нашей стране в рамках кооперации и интеграции со странами СЭВ (совет экономической взаимопомощи социалистических стран) разворачивается широкомасштабная программа гибкой автоматизации производства. Начинается выпуск высокоавтоматизированного оборудования для гибкой автоматизации, например, гибких производственных модулей ИС500ПМФ4 и ИС800ПМФ4 («Иваново-София») на базе обрабатывающих центров ИР500ПМФ4 и ИР800ПМФ4 Ивановского станкостроительного завода, оснащенных приводами из Болгарии и системами ЧПУ, производимыми в ГДР [52, 53]. Создаются новые производственные мощности по выпуску ГПЯ на Рязанском станкостроительном заводе. На Ивановском станкостроительном заводе начинается выпуск ГПЯ серии «Талка» (рисунок 1.3).



1,2,3,4 – гибкий производственный модуль ИР500ПМ1Ф4; 5,6 – станок ИР800ПМФ4; 7,8,9,10 – станок ИР320ПМФ4

Рисунок 1.3 - Один из вариантов ГПЯ серии «Талка»

Появился ГОСТ 26228-90, регламентирующий термины и определения в области ГПЯ [13].

На ведущих машиностроительных предприятиях начинается создание пробных ГПЯ. В Оренбуржье ГПЯ создаются на Орском заводе тракторных прицепов (ГПЯ «Автопром-2»), на ЮУМЗе на базе тяжелых станков фирмы «Шкода». На Оренбургском машиностроительном заводе (ПО «Стрела») создаются две ГПЯ: для обработки тел вращения и корпусных деталей.

При технических вузах страны начинается создание опытных полигонов и центров (ИЦ) по проблемам ГПЯ [35]. Массовыми тиражами издается литература,

обобщающая накопленный отечественный и зарубежный опыт создания ГПЯ.

К 1985 г. в мире насчитывается около 1000 ГПЯ [5, 10, 19, 73].

Девяностые годы характеризуются дальнейшим расширением использования производственных систем и созданием первых гибких автоматизированных заводов (ГАЗ). Появился термин «компьютерная интеграция производства», трактуемый как компьютерная автоматизация этапов, предшествующих изготовлению изделий и объединение их с ГПЯ различного целевого назначения в единый информационный комплекс. Это позволяет полностью перейти на безбумажный цикл передачи информации и существенно сократить сроки подготовки и производства новых изделий [4]. Объектом автоматизации становится цикл подготовки и производства продукции – от получения заказа на изготовление до выпуска готового изделия.

Одним из первых стал проект ГАЗ, реализованный в Японии фирмой Mazak, для производства деталей металлорежущих станков – «Система-21» (система XXI века) [5, 7]. Завод включал в себя: комплекс гибких производственных модулей (ГПМ) и ГПЯ, автоматизированные склады, робокарную транспортную систему. Предусматривалось использование компьютерных сетей для сервисной и технической поддержки филиалов, а также взаимодействия с предприятиями-поставщиками комплектующих изделий.

В рамках Государственной научно-технической программы «Технологии, машины и производства будущего» в СССР началась реализация комплекса проектов по созданию ГАЗ «Красный пролетарий» по производству металлорежущих станков и «Тверского завода штампов» [7].

Всего в эти годы в разных странах было создано около 20 компьютерно интегрированных производств (КИП) с различным уровнем автоматизации [7].

Начало XXI века характеризуется дальнейшим расширением сферы использования информационных технологий для нужд производства: объектом автоматизации (точнее - объектом информационной поддержки) становится жизненный цикл продукции (рисунок 1.1). По определению, приведенному в стандарте ISO 9004-1, «жизненный цикл (ЖЦ) продукции это - совокупность процессов, выполняемых от момента выявления потребностей общества в определенной продукции, до момента удовлетворения этих потребностей и утилизации продукции» [7].

Стремительное развитие информационных технологий, появление все новых аспектов их применения для производства, а, следовательно, и новых научных проблем способствовало утрате интереса отечественных ученых к проблемам создания гибких производственных систем. В немалой степени этому содействовали и социально-экономические преобразования в стране, в процессе которых разработки в области ГПЯ не находили практического применения. «К сожалению, этот этап так и остается незавершенным... Не подлежит сомнению необходимость комплексного подхода к проблеме проектирования ... с позиций системного подхода с учетом очень больших особенностей проектирования, функционирования и управления ГПЯ и ее компонентов по сравнению с «традиционным» металлорежущим оборудованием» [32].

1.1.2 Современное состояние проблемы в мире

Эксплуатация ГПЯ экономически выгодна для малых и больших предприятий с числом работающих от 45 [17] до 10000 человек.

В США к 2003 г. гибкие производственные системы и ячейки используются на 73 % фирм с числом работающих от 100 человек [68] – т.е. фирма, не эксплуатирующая ГПЯ – скорее, исключение, чем правило.

ГПЯ оказываются выгодными и на крупных предприятиях. Так, в 2003 году компания Ford Motor сократила 9-тысячный рабочий коллектив на треть и потратила около 700 млн. евро на создание новой гибкой производственной системы.

В зарубежной печати появился новый термин – «экономное производство», или «экономная автоматизация» – обозначающий ГПЯ, работающие в течение 140 ч в неделю или по 20 ч в сутки в безлюдном режиме [86].

Наметился следующий уровень автоматизации – создание реконфигурируемых производственных систем (РПС) с регулируемой производственной мощностью [75]. Цель создания реконфигурируемых производственных систем — иметь производственное оборудование и системы управления для экономичного оперативного реагирования на изменяющийся спрос на продукцию на рынке. Методология системного проектирования и необходимость оперативно создавать РПС являются основой этого нового принципа организации производства. Новый принцип реконфигурируемого производства позволяет иметь оборудование с необходимыми функциями и мощностью точно в нужное время.

Определение «изменение архитектуры» («реконфигурация») означает способность регулировать производственную мощность и функциональность производственной системы в соответствии с новыми условиями, путем изменения компоновки оборудования или компонентов системы. «Компонентами» могут быть станки и конвейеры во всей системе, механизмы в отдельных станках, новые датчики или новые алгоритмы работы контроллера. «Новыми условиями» могут быть изменения спроса на продукцию, производство нового изделия на существующей системе, или интегрирование нового технологического процесса в существующие производственные системы.

Реконфигурируемые системы будут открытыми, т.е. их можно усовершенствовать и повышать их уровень вместо того, чтобы заменять. Они будут гибкими не только в смысле изготовления разных деталей, но также в смысле изменения характера функционирования самой системы. Ключевым условием является то, что система этого типа изначально должна быть запроектирована с возможностью изменения конфигурации, и должна состоять из основных аппаратных и математических модулей, которые можно соответственно размещать быстро и надежно.

Научно-исследовательский центр при Мичиганском университете (ERC) по реконфигурируемым производственным системам разработал новый развивающийся завод, в котором работают такие системы.

«Новые системы с изменяемой архитектурой переведут обрабатывающие фирмы в положение, соответствующее XXI веку, и окажут такое же влияние на промышленность, как массовое и мелкосерийное производство в XX веке. А может быть и больше»[75].

1.1.3 Состояние и прогноз развития гибкой автоматизации машиностроения России и стран СНГ

В настоящее время сырье является основной составляющей Российского экспорта.

Вот как оценивают продажу сырья украинские ученые [40]. Прибыль, получаемая от реализации одной тонны металла в виде проката, составляет одну условную единицу. Переработка 1 тонны металла в автомобиль даёт уже прибыль в 10 таких единиц, а передел в металлорежущее оборудование - от 10 до 50 условных единиц. Отсюда напрашивается вывод, что повысить эффективность металлургического производства, а, следовательно, и всей промышленности возможно за счёт глубокого передела металла. Для этого необходимо современное машиностроительное производство, способное производить конкурентоспособную продукцию [40].

Для белорусских предприятий, чей технологический парк зачастую представляет собой “зверинец” из станков, приобретенных по случаю у разных фирм и в разные годы, вопрос унификации и создания гибких производственных систем весьма актуален. Тем более что модернизация оборудования стоит от 20 до 40 % суммы, которую пришлось бы выложить за новый станок. А эффект получается тот же. В Белоруссии активно создаются совместные белорусско-германские предприятия, в которых немецкие специалисты передают свой опыт в области создания гибких производственных и управленческих технологий.

В работе Кетова А. В., Давыдова В.М. отмечается, что, «в условиях рыночной экономики производственная система машиностроительного предприятия должна быть автоматизированной гибкой производственной системой (ГПЯ); исключением, только подтверждающим правило, являются узко специализированные производства с высокой серийностью (например, производства шарикоподшипников, крепежных изделий и т. д.)» [32].

В работе «Белая книга спецслужб» А.И. Подберезкина [47] отмечается, что «направление технического перевооружения оборонных предприятий должно быть направлено на рациональную диверсификацию производства, на использование преимущественно гибких, легко перестраиваемых производств, исходя из того, что государственный оборонный заказ и в перспективе будет ориентирован на выпуск вооружения и военной техники, в основном малыми сериями и не обеспечит определяющей загрузки производственных мощностей ОПК».

Согласно прогнозу мирового развития экономики в ближайшем будущем приоритетно будут развиваться: космическая техника, авиастроение, автомобилестроение, судостроение, энергомашиностроение, транспортное машиностроение, сельхозмашиностроение, лесопереработка, производство бытовой и хозяйственной техники [69].

Организация производства высокоэффективного оборудования потребует значительных финансовых затрат на разработку и выпуск новых образцов и еще больших - на модернизацию, реконструкцию существующего производства и основных фондов. Собственных средств предприятий станкостроения для этого явно недостаточно. Окупаемость инвестиционных проектов в станкостроении составляет в среднем 3-4 года, что значительно снижает возможность использования заемных

средств. Без государственной поддержки и создания особых условий для производства новой техники станкостроителям не обойтись.

Согласно прогнозу к 2010 году ежегодное потребление металлообрабатывающего оборудования предприятиями машиностроительного комплекса, в том числе ВПК составит 45,0 тыс. единиц (металлорежущие станки - 35,0 тыс. единиц, КПО - 10,0 тыс. единиц) общей стоимостью до 3 млрд. долларов США, 80 % потребления будет покрываться за счет отечественного производства, 20 % - за счет импорта. Из 45,0 тысяч единиц отечественного оборудования станки с ЧПУ составят от 18,0 до 20,0 тыс. штук, в том числе обрабатывающие центры от 7,0 до 8,0 тыс. штук, ГПЯ необходимо поставлять в количестве 25 - 50 систем в год [69].

Одним из путей технического перевооружения промышленности служит закупка импортного оборудования и технологий. Однако это путь постоянного технологического отставания, так как научные результаты, положенные в основу новой техники и технологии, держатся в секрете. Примером может служить производство авиационных газотурбинных двигателей (ГТД): «конгресс запрещает передавать за пределы США программы термогазодинамического моделирования ГТД (типа GECAT, JGTS или Java Simulator)...Казалось бы, времена КОКОМ прошли, но специализированные версии для авиации CAD/CAM/CAE/PDM/ERP-систем, большая часть которых разрабатывается фирмами США (EDS, SDRC, PTC, SAP, MSC и др.) в Россию также не продаются» [36].

Другой путь – развитие собственных научных исследований в данной области. Примером может служить создание в интересах Минатома отраслевого учебно-научного центра МИФИ по CALS-технологиям. Одним из структурных подразделений центра является лаборатория ГПЯ [74]. Другой пример – разработка ГПЯ в рамках плановых работ АВИАПРОМа (рисунок 1.4).

Подводя итог, можно сделать следующие выводы:

1) Назрела необходимость коренной реконструкции отечественного машиностроения как базовой отрасли создания средств производства для всех отраслей промышленности: для ракетно- и авиастроения, автомобиле- и судостроения, и т.д. Реконструкция должна быть направлена на выпуск современных средств производства, обеспечивающих экономичное изготовление конкурентоспособной продукции.

2) Практика стран дальнего зарубежья показывает, что современными средствами производства являются гибкие производственные системы, экономически выгодные практически для любых объемов производства.

3) Создание гибких производственных систем представляет собой динамично развивающуюся область научных и практических разработок, выполнение которых предполагает наличие нерешенных научных проблем.

4) Исследования в направлении создания ГПЯ следует признать актуальным научным направлением.

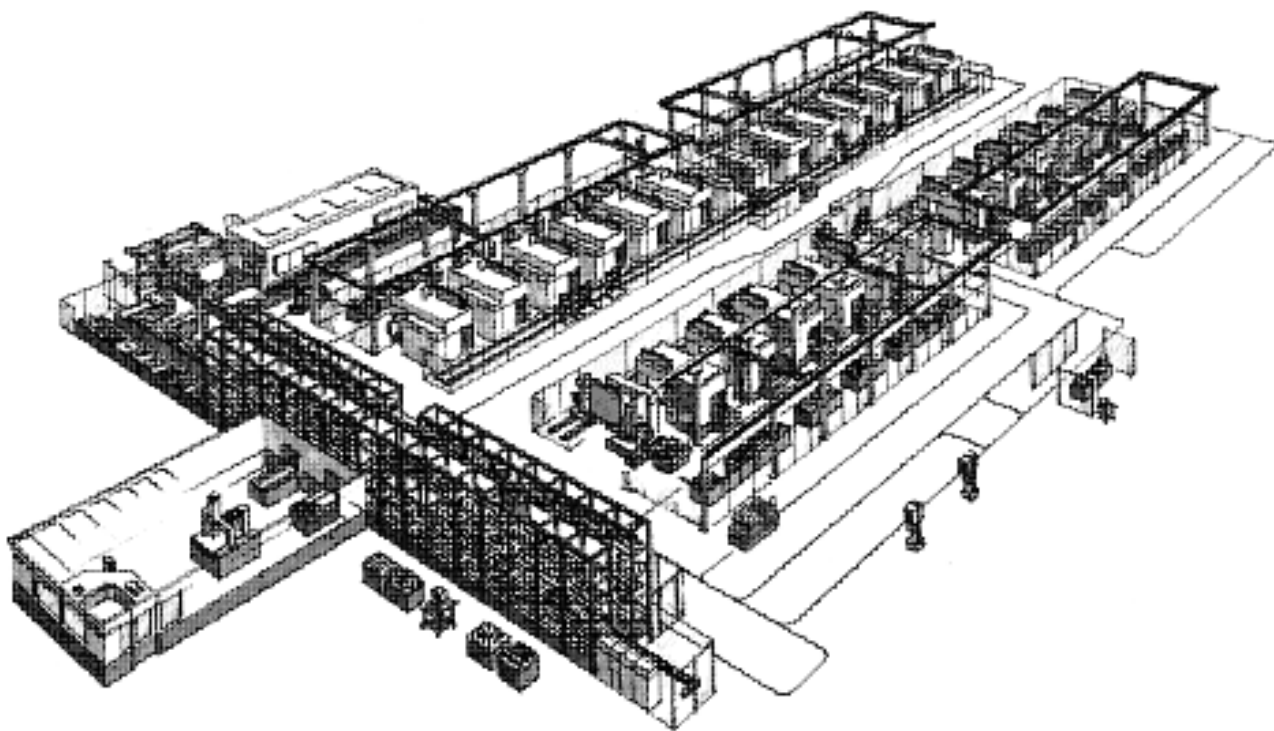


Рисунок 1.4 – Общий вид ГПЯ АВИАПРОМа

Отечественный опыт «директивного» создания гибких производственных систем в 80-х годах прошлого века не прошел даром. На государственном уровне был предпринят масштабный эксперимент по модернизации отечественного производства в соответствии с мировыми тенденциями промышленного развития. Были созданы образцы ГПЯ, опередившие по уровню гибкости и степени автоматизации зарубежные аналоги.

Вместе с тем «открылся пласт научных проблем, требующих принципиального решения. Причем большая часть проблем связана не с техническими или технологическими решениями, а с созданием современных методов проектирования и эксплуатации гибкого производства» [32].

1.2 Аппарат для расчета ГПЯ

1.2.1 Аналитические расчеты

Отличительный признак ГПЯ - возможность работы в «безлюдном» режиме от 8 ч и более [13]. Данная возможность обеспечивается за счет использования средств автоматической доставки и смены заготовок в рабочих зонах станков, а так же управления от компьютера всем комплексом оборудования.

Работа станка ГПЯ может быть организована следующим образом (рисунок 1.5). В момент подачи паллеты с очередной заготовкой в рабочую зону считывается код заготовки на паллете. Компьютер системы управления ищет в библиотеке необходимую управляющую программу (УП) и передает ее в систему ЧПУ. Тем временем паллета передается в рабочую зону, и начинается процесс работы станка по УП.

Таким образом, время работы станка включает время на смену заготовки $t_{см}$ и время работы по УП $t_{ум}$. Соответственно, производительность станка $P_{ОЦ}$ можно рассчитать как

$$P_{ОЦ} = \frac{1}{t_{см} + t_{ум}} \quad (1.1)$$

Однако наличие в ГПЯ средств автоматической доставки заготовок требует дополнительных затрат времени на выполнение транспортных операций $t_{мп}$. В частном случае фрагмент ГПЯ включает обрабатывающий центр 1 и робокару 2 для передачи заготовок и изготовленных деталей между станком и передаточной позицией накопителя (рисунок 1.5). Производительность станка при условии бесперебойной доставки заготовок в этом случае составит

$$P_{ОЦ} = \frac{1}{t_{см} + t_{ум} + t_{мп}} \quad (1.2)$$

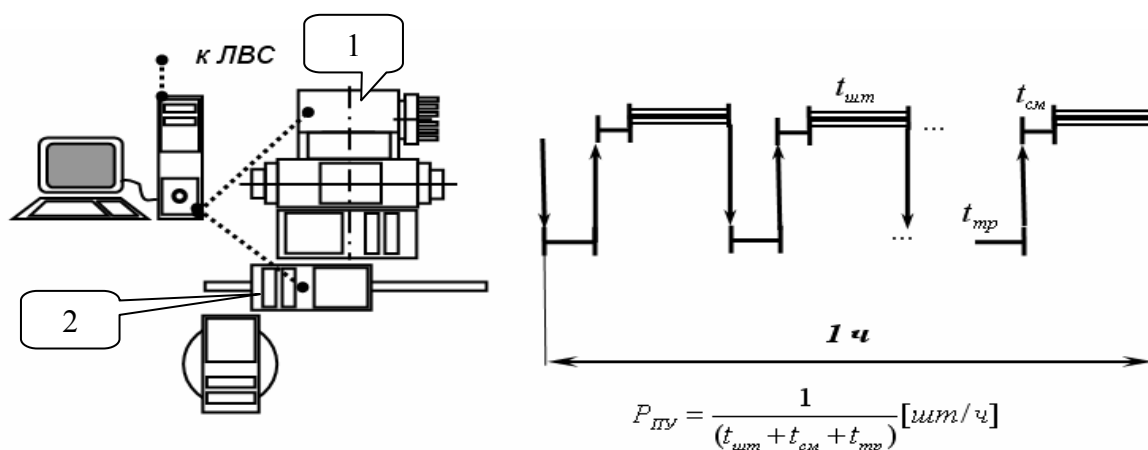


Рисунок 1.5 – Схема к расчету производительности станка, обслуживаемого робокарой

При создании ГПЯ из 2-х станков и более, обслуживаемых транспортным роботом, циклограмма работы оборудования включает простои станков и транспортного средства (рисунок 1.6). Наличие данных простоев делает невозможным аналитический расчет производительности системы, исходя из станкоемкости изделий.

Невозможность аналитического расчета производительности системы, в свою очередь, ставит под сомнение возможность аналитического расчета числа станков в ГПЯ. Однако оценочный расчет работы ГПЯ с различным числом станков можно выполнить на основе построения циклограммы работы оборудования. Рассмотрим пример.

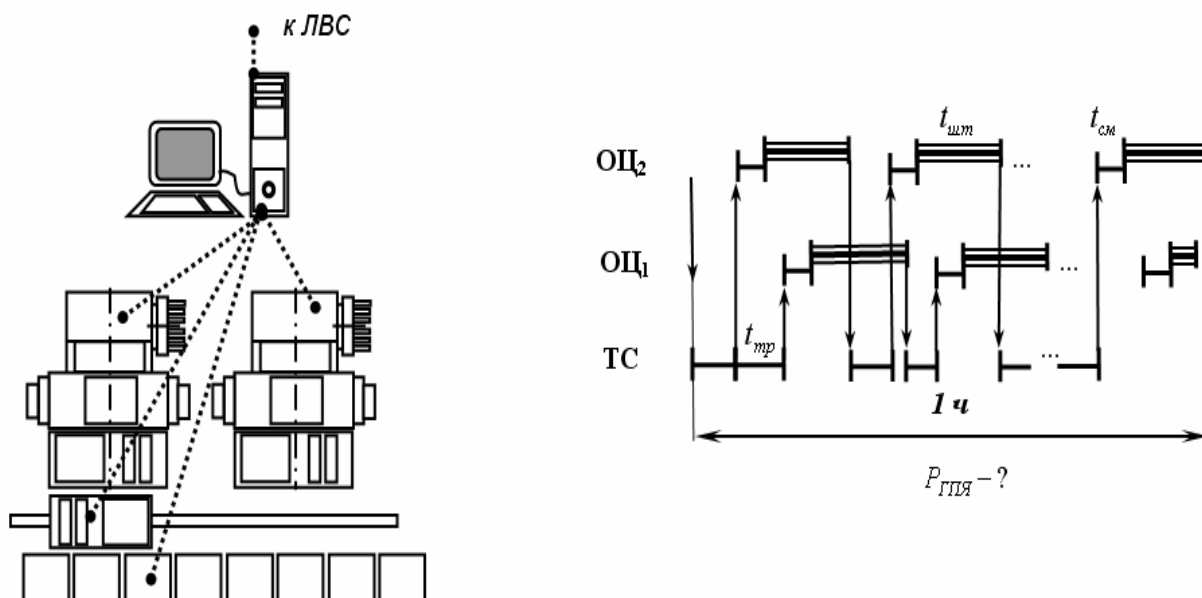


Рисунок 1.6 – Схема к расчету производительности гибкой производственной ячейки

Необходимо изготовить $N=10$ изделий, время изготовления каждого из которых $t_{шт}=3,0$ мин. Для изготовления может использоваться одна из двух ГПЯ – ГПЯ №1, состоящая из 5 станков или ГПЯ №2, состоящая из 10 станков. Пусть в обоих случаях время смены паллеты $t_{см}=60$ с, а время транспортной операции $t_{тр}=2,0$ мин.

Построим циклограммы работы обеих ГПЯ (рисунок 1.7). Судя по полученным результатам, время изготовления 10 изделий в обеих ГПЯ одинаково и составляет $t_{ц}=24,0$ мин. Соответственно, производительность ГПЯ в том и другом случае будет одинакова

$$P_{ГПЯ} = n_{ср} = \frac{N}{t_{ц}} = \frac{10}{24} \times 60 = 25 \text{ штук/ч.} \quad (1.3)$$

Однако в первом случае используется станков в два раза меньше, чем во втором. Соответственно, примерно также будет различаться себестоимость обработки. Рассмотренный пример намеренно упрощен: не рассматривается более длительный цикл работы ГПЯ, не учитывается варьирование станкоемкости изделий и изменение длительности транспортных операций: пример служит лишь для иллюстрации возможных ошибок при расчете основного технологического оборудования ГПЯ. Однако из сказанного следует вопрос: какие методы применяются для расчета гибких производственных ячеек?

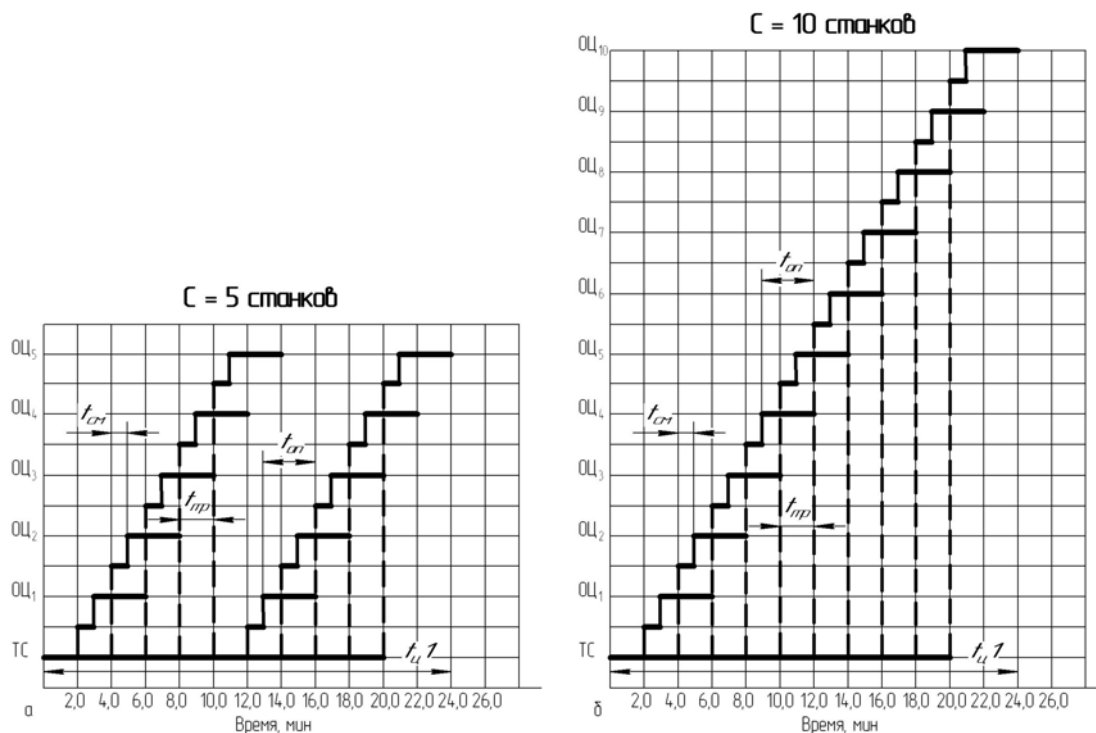


Рисунок 1.7 - Возможные циклограммы изготовления 10 изделий в ГПЯ из 5 (а) и 10 многоцелевых станков (б)

1.2.2 Теория массового обслуживания

Ситуации, связанные с наличием очередей на обслуживание, характерны для многих сфер человеческой деятельности: очереди в билетных кассах, в крупных аэропортах, на телефонных станциях в ожидании освобождения линии абонента, и т.д. Во всех перечисленных случаях имеем дело с массовостью и обслуживанием. Изучением таких ситуаций занимается теория систем массового обслуживания (СМО), или теория очередей [27, 33].

Основной задачей теории СМО является изучение режима функционирования обслуживающей системы и исследование явлений, возникающих в процессе обслуживания. Так, одной из характеристик обслуживающей системы является время пребывания требования в очереди. Очевидно, что это время можно сократить за счет увеличения количества обслуживающих устройств. Однако каждое дополнительное устройство требует определенных материальных затрат, при этом увеличивается время бездействия обслуживающего устройства из-за отсутствия требований на обслуживание, что также является негативным явлением. Следовательно, в теории СМО возникают задачи оптимизации: каким образом достичь определенного уровня обслуживания (максимального сокращения очереди или потерь требований) при минимальных затратах, связанных с простоем обслуживающих устройств.

В теории СМО рассматриваются такие случаи, когда поступление требований происходит через случайные промежутки времени, а продолжительность обслуживания требований не является постоянной, т.е. носит случайный характер. В силу этих причин одним из основных методов математического описания СМО является аппарат теории случайных процессов.

1.2.2.1 Аппарат теории массового обслуживания

В математических моделях (ММ) сложных объектов, представленных в виде систем массового обслуживания СМО, фигурируют средства обслуживания, называемые обслуживающими аппаратами (ОА), и обслуживаемые заявки, называемые транзактами. Так, в модели производственной линии ОА отображают рабочие места, а транзакты - поступающие на обработку детали, материалы, инструмент.

Состояние СМО характеризуется состояниями ОА, транзактов и очередей к ОА. Состояние ОА описывается двоичной переменной, которая может принимать значения «занят» или «свободен». Переменная, характеризующая состояние транзакта, может иметь значения «обслуживания» или «ожидания».

Состояние очереди характеризуется количеством находящихся в ней транзактов.

Имитационная модель СМО представляет собой алгоритм, отражающий поведение СМО, т.е. отражающий изменения состояния СМО во времени при заданных потоках заявок, поступающих на входы системы. Параметры входных потоков заявок - внешние параметры СМО. Выходными параметрами являются величины, характеризующие свойства системы - качество ее функционирования. Примеры выходных параметров: производительность СМО - среднее число заявок, обслуживаемых в единицу времени; коэффициенты загрузки оборудования - отношение времен обслуживания к общему времени в каждом ОА; среднее время обслуживания одной заявки. Основное свойство ОА, учитываемое в модели СМО, - это затраты времени на обслуживание, поэтому внутренними параметрами в модели СМО являются величины, характеризующие это свойство ОА. Обычно время обслуживания рассматривается как случайная величина, и в качестве внутренних параметров фигурируют параметры законов распределения этой величины.

Имитационное моделирование позволяет исследовать СМО при различных типах входных потоков и интенсивностях поступления заявок на входы, при вариациях параметров ОА, при различных дисциплинах обслуживания заявок. Дисциплина обслуживания - правило, по которому заявки поступают из очередей на обслуживание. Величина, характеризующее право на первоочередное обслуживание, называется приоритетом. В моделях СМО заявки, приходящие на вход занятого ОА, образуют очереди, отдельные для заявок каждого приоритета. При освобождении ОА на обслуживание принимается заявка из непустой очереди с наиболее высоким приоритетом.

Основной тип ОА - устройства, именно в них происходит обработка транзактов с затратами времени. К ОА относятся также накопители (памяти), отображающие средства хранения обрабатываемых деталей в производственных линиях или обрабатываемых данных в вычислительных системах. Накопители характеризуются не временами обслуживания заявок, а емкостью - максимально возможным количеством одновременно находящихся в накопителе заявок.

К элементам имитационных моделей СМО кроме ОА относят также узлы и источники заявок. Связи ОА между собой реализуют узлы, т.е. характеризуют правила, по которым заявки направляются к тому или иному ОА.

Для описания моделей СМО при их исследовании на ЭВМ разработаны специальные языки имитационного моделирования. Существуют общецелевые языки, ориентированные на описание широкого класса СМО в различных предметных областях, и специализированные языки, предназначенные для анализа систем определенного типа. Примером общецелевых языков служит широко распространенный язык GPSS [83].

1.2.2.2 ГПЯ как система массового обслуживания

Применение методов теории массового обслуживания применительно к анализу ГПЯ описано в [2, 42, 85].

В работе [48] подробно рассматриваются способы представления станочных систем с позиций СМО. Сформулированы общие заключения:

а) продолжительность ожидания для робокары существенно меньше продолжительности ожидания станка;

б) при малом времени ожидания манипулятора имеют место состояния, при которых два или более станка одновременно требуют новых деталей;

в) подобные состояния с двумя и более совпадающими запросами наступают сравнительно редко, и общий простой невелик;

г) частота заявок зависит от ассортимента деталей, особенно от времени цикла обработки отдельных деталей;

д) время ожидания при многостаночном обслуживании не зависит от частоты заявок и от отдельного простоя станка.

е) заказы на обслуживание носят случайный характер.

Средняя частота или интенсивность заказов на обслуживание в единицу времени λ описывается как

$$\lambda = \frac{1}{T_a}, \quad (1.4)$$

где T_a - среднее значение случайного периода T_A времени между двумя последовательными не принятыми заказами на обслуживание и время ожидания.

Для ГПЯ $T_A = t_{um}^{cp}$, где t_{um}^{cp} – среднее время обработки заготовки.

Распределение заявок близко к закону распределения Пуассона. В этом случае функция вероятности для заявок на обслуживание

$$P_k = \frac{\lambda^k \times e^{-\lambda}}{k!} \quad (1.5)$$

для $k = 0, 1 - m$.

Вероятность P_0 при $k=0$ означает отсутствие заявок на обслуживание, P_1 означает вероятность появления единичной заявки и т.д. В предельном случае P_m означает одновременное требование на обслуживание от всех m объектов ГПЯ.

Для расчета вероятности используется рекуррентная формула

$$P_{k+1} = \frac{\lambda}{k+1} \times P_k \quad (1.6)$$

Интенсивность обслуживания станков определяется как

$$\nu = \frac{1}{t_{mp}}, \quad (1.7)$$

где t_{mp} – среднее время выполнения транспортной операции.

Таким образом, теория массового обслуживания на стадии эскизного проектирования ГПЯ позволяет произвести предварительную оценку проекта. В качестве исходных данных должны быть известны:

- а) интенсивность заявок на обслуживание λ ;
- б) число станков в системе M ;
- в) интенсивность обслуживания ν .

Следует отметить, что для создания моделей, адекватных моделируемым объектам, необходимо знать достоверные количественные характеристики случайных возмущений. В очень редких случаях данные о параметрических возмущениях типа отказов и отклонений параметров технологического оборудования от расчетных приводятся в паспортных данных на это оборудование. Но фактические значения этих данных значительно отличаются от паспортных. Поэтому любому расчету или анализу ГПЯ должно предшествовать накопление статистических данных о действующих возмущениях и обработка их с использованием методов теории вероятностей и математической статистики [41]. «Получается замкнутый круг: чтобы создать имитационную модель, адекватную проектируемой производственной системе, нужно, как минимум, эту систему построить и накопить статистику по ее работе» [58].

1.3 Теория расписаний

Задачи упорядочения и выбора из очередей решаются теорией расписаний [34, 70, 71], которая является частью исследования операций. Теория расписаний исследует задачи, в которых необходимо упорядочить или, другими словами, определить последовательность выполнения совокупности работ, использования каких-либо средств и т.д.

Задачи упорядочения носят самый общий характер. Они возникают там, где существует возможность выбора той или иной очередности выполнения работ: при распределении работ на производстве, составлении расписания приземления самолетов, составлении расписания движения поездов, обслуживании клиентов в обслуживающих системах и т.д.

Результаты, к которым приводит то или иное упорядочение, существенно от-

личаются. В ряде практических случаев эти различия принимают стоимостный характер или определяются какой-либо другой величиной.

Существует определенное различие между терминами “упорядочение” и “составление расписания”. Упорядочение подразумевает формирование очередности операций, выполняемых одной машиной, в то время как составление расписания означает задание последовательности действий нескольким машинам.

Почти вся теория, разработанная в настоящее время, относится к ограниченному числу моделей простого процесса обслуживания. Под последним понимается процесс, для которого существенны следующие ограничения [81]:

1) Каждая машина может быть назначена в любой момент времени, т.е. запрещены перерывы в работе машины. Каждая машина формально представляет собой интервал $(0, T)$, где T есть произвольно большое число.

2) Работы представляют собой строго упорядоченные последовательности операций. Для заданной операции X существует не более одной операции Y такой, что $Y \ll X$, и одной операции Z такой, что $X \ll Z$.

3) Каждая операция выполняется только одной машиной.

4) Существует только по одной машине каждого типа. Между номерами машин и вторыми индексами операций, указывающими номер выполняющей их машины, существует взаимно однозначное соответствие.

5) Отсутствуют прерывания операций. Для каждой операции задан единственный интервал (a, b) , причем длительность операции равна $(b - a)$.

6) Интервалы выполнения последовательных операций одной и той же работы не пересекаются. Величина a , соответствующая назначению операции X , не может быть меньше b для любой операции Y такой, что $Y < X$.

7) В каждый момент времени машина может выполнять не более одной операции.

Перечисленные ограничения, с одной стороны, упрощают формализацию, а с другой, делают ее более абстрактной. Последнее приводит к тому, что модель становится неадекватной практическим случаям, где требуется ослабление одного или нескольких из приведенных ограничений. Тем не менее, такая модель сохраняет в основном структуру большинства практических задач, а при ее исследовании вырабатывается интуиция, полезная в многочисленных приложениях. Во всяком случае, в настоящее время описанная формализация применяется в большинстве исследований.

Порядок выполнения машинами операций одной работы определяет, является ли система машин:

- а) конвейерной;
- б) со случайным порядком выполнения работ;
- в) системой произвольного типа.

Для классификации задач теории расписаний в дальнейшем используется запись $A/B/C/D$, где

- A - характеризует процесс поступления работ. Для динамических задач A представляет собой функцию распределения времени между поступлениями. Для статических задач A соответствует числу одновременно поступивших работ, если по

этому поводу ничего специально не оговорено. Если на месте A стоит n то это означает произвольное, но конечное число работ в статическом случае.

- B - характеризует число машин в системе. Если на месте B стоит m то это означает произвольное число машин.

- C - характеризует порядок (дисциплину) выполнения работ машинами. Если на месте C находится F , то это соответствует конвейерной системе; если R - то случайной, и если G - произвольной. Для системы, состоящей из одной машины, указанные дисциплины теряют смысл и поэтому для такой системы третий параметр опускается.

- D - характеризует оценку (критерий) расписания.

Задачи теории расписаний возникают при изучении многостадийных обслуживающих систем, т.е. таких систем, в которых процесс обслуживания каждого требования состоит из нескольких последовательных стадий, на каждой из которых это требование обслуживается тем или иным прибором либо совокупностью приборов.

В любом случае имеется конечное множество $N = \{1, 2, \dots, n\}$ требований и конечное множество $M = \{1, 2, \dots, m\}$ приборов. Процесс обслуживания требования $i \in N$ включает r_i стадий. При этом каждому требованию $i \in N$ и каждой стадии q , $1 \leq q \leq r_i$ его обслуживания сопоставляется некоторое множество приборов $M_q^i \subseteq M$. В зависимости от типа обслуживающей системы требование i на стадии q либо может быть обслужено любым из приборов $L \in M_q^i$ (но не более, чем одним одновременно), либо должно одновременно обслуживаться всеми приборами из множества M_q^i .

Процесс функционирования обслуживающей системы может быть описан путем задания расписания (календарного плана, временного графика), т.е. некоторой совокупности указаний относительно того, какие именно требования какими именно приборами обслуживаются в каждый момент времени. Обычно предполагается, что каждое требование не может одновременно обслуживаться двумя и более приборами, и каждый прибор не может одновременно обслуживать более одного требования.

«В настоящее время в ней (в теории расписаний) реализован достаточно узкий класс формальных моделей и методов планирования простого процесса обслуживания» [54].

«В общем виде задача теории расписаний не решена. Оптимальные решения получены лишь для простейших случаев, имеющих чисто теоретическое значение» [49].

«Теоретически хорошо разработаны методы выбора дисциплины обслуживания только для одиночной системы массового обслуживания, отображающей один станок или одну группу оборудования. Имеющая практическое значение более сложная задача выбора дисциплин обслуживания для ГПК в целом, не решена» [2].

1.4 Приоритетные правила в АСУ ГПЯ

При отсутствии строгого математического решения и невозможности применения полного перебора из-за большой размерности задачи, единственным путем получения результата является использование эвристических методов. Из них для указанного круга задач наиболее разработан метод эвристических приоритетных правил

Введем понятие приоритета. Это числовая характеристика элемента множества, позволяющая провести его выбор из всего множества подобных элементов. В частности, при выборе какого-либо действия на выполнение из множества возможных он может осуществляться по возрастанию или убыванию его приоритета, вычисляемого по заранее оговоренному правилу или алгоритму.

Приоритет задает порядок выбора из очереди в освободившийся прибор заявок одинакового приоритета. Известно 5 правил присвоения приоритетов заявкам на обслуживание: а) первый пришел - первый обслужен (FCFS- First Come First Serve); высший приоритет присваивается самой ранней заявке; б) наименьшее время обработки (SPT - Shortest Processing Time); в первую очередь транспортируется заготовок с минимальной станкоемкостью; в) наименьшее остаточное время обработки (LWKR-Least Work Remaining) - обслуживается станок, на котором суммарное время всех переходов, которые осталось выполнить, минимально; г) суммарное время обработки (TWK - Total Work) - учитывается сумма длительностей всех технологических переходов обработки заготовки; д) наименьшая очередь на станке для выполнения последующей операции (NXQL - Next Queue Length). Первым обслуживается станок, имеющий минимальное количество заготовок в накопителе. Рассмотренный перечень может быть расширен такими правилами, как присвоение высшего приоритета заявке от более дорогого станка, первоочередное выполнение заявки с минимальным временем транспортной операции и т.д.

В теории массового обслуживания [11, 27, 33] приоритеты называются дисциплинами обслуживания, которые задает порядок выбора заявок из очереди. Различают следующие дисциплины: FIFO (First Input - First Output): первым пришел - первым обслужен, LIFO (Last Input - First Output): последним пришел - первым обслужен, RAND (Random): случайный выбор из очереди. В быту обычно действует дисциплина FIFO. Дисциплина LIFO реализуется в буфере, организованном по принципу стека. Такая дисциплина может оказаться целесообразной, например, в системах передачи информации, если ее ценность быстро падает со временем.

В работе [72] рассматриваются следующие правила выбора операции из $\{S_{so}\}$:

RANDROM – равновероятный выбор;

MOPNR – выбор операции, входящей в работу с максимальным числом неупорядоченных операций (maximum number of operations remaining);

MWKR-P – выбор операции, входящей в работу, для которой длительность выполнения всех остальных неупорядоченных операций максимальна (most processingtime on subsequent operations);

MWKR/P – выбор операции, входящей в работу, для которой максимально отношение суммарной длительности всех неупорядоченных операций к длительности операции, включаемой в расписание ($\max \{(\text{work remaining})/ (\text{processing-time})\}$);

SPT – выбор кратчайшей операции, т.е. операции с минимальной длительностью (shortest processing-time);

MWKR – выбор операции, входящей в работу, для которой оставшаяся длительность всех операций максимальна (most work remaining);

LWKR – выбор операции, входящей в работу, для которой оставшаяся длительность выполнения всех операций минимальна (least work remaining);

Полученные по этим правилам расписания сравнивались с незадерживающими расписаниями, полученными с помощью соответствующих диспетчеризаций следующим образом: сначала выбиралась машина, которая может начать выполнение очередной операции раньше других, а затем из множества ожидающих операций выбиралась операция по одному из правил:

RANDROM – равновероятный выбор;

SPT – выбор операции с минимальной длительностью выполнения (shortest processing-time);

MWKR – выбор операции, входящей в работу с максимальной длительностью выполнения всех неупорядоченных операций (most work remaining);

LWKR – выбор операции, входящей в работу с минимальной длительностью выполнения всех неупорядоченных операций (least work remaining);

LPT – выбор наиболее длинной операции (longest processing time);

FCFS – выбор операции, попавшей в $\{S_{so}\}$ раньше всех, т.е. выбор в порядке поступления (first-come, first-served).

Хотя на первый взгляд правила из этого списка совпадают с правилами из предыдущего списка, различие между выбором из всего множества $\{S_{so}\}$ и из подмножества операций, ожидающих выполнения на той машине, которая окажется доступной раньше других, весьма существенно и приводит к совершенно разным расписаниям.

Система приоритетов должна быть полной в том смысле, чтобы обеспечивать возможность однозначного выбора действия. Если однозначный выбор невозможен, то необходимо либо перейти на другое правило назначения приоритета, либо после выбора по первому правилу нескольких действий осуществить выбор среди них по дополнительному (вторичному) правилу и т. д. В качестве правила назначения приоритета, в которое можно использовать как некоторую «точку отсчета» для сравнения эффективности различных правил, наиболее часто используется назначение приоритета случайным образом. В этом случае значение приоритета генерируется датчиком случайных чисел.

Таким образом, решение задачи выбора при диспетчировании и планировании предусматривает необходимость встраивания в систему управления какого-либо правила назначения приоритета (или системы правил).

Приоритетные правила можно разделить на два класса - простые и комбинированные (составные). К простым относят правила, которые нельзя представить в виде некоторой совокупности других. Такие правила для назначения приоритета обычно используют какую-либо одну характеристику действия. Примером простого приоритетного правила является правило кратчайшей операции, когда из некоторого множества действий выбирается действие с наименьшей длительностью реализуе-

мой им операции, или правило планового срока, когда выбирается действие, которое необходимо закончить раньше других.

Комбинированные приоритетные правила применяются в том случае, когда невозможен однозначный выбор по простому правилу или когда необходимо расширить номенклатуру учитываемых параметров действия:

- приоритетное правило представляет собой взвешенную сумму простых правил;

- приоритетное правило представляет собой произведение простых правил, взятых с показателями степени;

- приоритетное правило представляет собой последовательность простых правил, применяемых друг за другом до тех пор, пока каждое действие из множества не будет иметь значение приоритета, отличное от остальных действий, т. е. будет возможен однозначный выбор.

Чтобы получить множество простых приоритетных правил, из которых впоследствии можно строить комбинированные, необходимо рассмотреть основные характеристики ГПЯ и для каждой из этих характеристик сформулировать простое приоритетное правило. Так как известно большое число приоритетных правил и многие из них достаточно исследованы для различных целей, то при формировании множества простых правил необходимо учитывать этот накопленный опыт. Поэтому, выделив характеристики действий и ресурсов в ГПЯ, следует подобрать из множества правил -подмножество, свойственное данной ГПЯ. Такой выбор осуществляется путем сопоставления характеристик, учитываемых в приоритетных правилах и полученных из анализа ГПЯ, и выполняемых ею действий [24]:

- а) правило случайного выбора. Приоритет передаточной партии вычисляется при поступлении ее в систему и равен случайному числу;

- б) правило первым пришел - первым обслужен;

- в) правило является разновидностью предыдущего правила. Приоритет передаточной партии остается постоянным в течение всего планового периода и равен времени поступления ее в ГПЯ;

- г) Правило кратчайшей операции. По очереди выбирается операция, имеющая минимальную длительность. Использование данного правила позволяет увеличить загрузку оборудования за счет уменьшения длин очередей, однако за счет этого время изготовления передаточных партий деталей с большими длительностями операций обработки увеличивается. Наряду с предыдущим правилом данное правило является наиболее распространенным и дает в большинстве случаев хорошие результаты;

- д) правило максимально длинной операции;

- е) выбор операции над передаточной партией деталей, для которой длительность всех оставшихся минимальна;

- ж) выбор операции над передаточной партией деталей с максимальной длительностью оставшихся операций. Данное правило противоположно предыдущему;

- и) приоритет равен суммарной длительности всех операций обработки передаточной партии деталей. В этом приоритетном правиле, как и во многих рассмотренных выше, необходимо иметь данные о длительностях будущих операций обработки. На этапах диспетчирования, а тем более планирования ГПЯ, эти данные

известны лишь с определенной степенью точности. Поэтому многие приоритетные правила работают с прогнозной информацией и соответственно решения принимаются в условиях неполной информации, что должно учитываться при сравнении эффективности приоритетных правил;

к) правило минимального числа оставшихся невыполненных операций.

Все рассмотренные выше правила, так или иначе, основываются на длительностях обслуживания (выполнения операций). Среди других показателей, наиболее часто используемых на практике, следует отметить характеристики, связанные с плановыми сроками выполнения заданий (изготовлением передаточной партии деталей). Это объясняется тем, что выполнение заданий в срок во многом является оценкой качества системы оперативного управления. Поэтому плановые сроки нужно задавать, исходя из возможности производства, и назначать извне, следовательно, плановые сроки являются для системы оперативного управления точно известными (при отсутствии внешнего вмешательства) неизменными на плановом периоде. Основные приоритетные правила здесь следующие:

а) правило плановых сроков. Приоритет равен плановому сроку выпуска передаточной партии с ГПЯ;

б) правило поэтапных плановых сроков. Плановый срок определяется для каждой операции путем деления планового срока для всей партии пропорционально длительностям выполнения операций;

в) приоритет основан на временном резерве. Данное правило увеличивает приоритет передаточных партий, для которых приближается срок выпуска.

Существует ряд алгоритмов назначения приоритетов, использующих характеристики состояния ГПЯ, например, связанных с величинами очередей на обслуживание при транспортировании или обработке. Как и в случае с использованием длительностей обслуживания, данные о загрузенности очередей не являются точными. Данный алгоритм назначения приоритета приводит к выравниванию объемов очередей в ГПЯ и уменьшению их объемов. Сложность здесь, заключается в трудности получения информации в процессе функционирования ГПЯ и принятии решений в реальном масштабе времени. Перечисленные приоритетные правила далеко не исчерпывают множества простых правил, но уже здесь можно увидеть ряд проблем возникающих в процессе использования их при управлении. Первой из них является выбор правила для использования при построении конкретной АСУ ГПЯ. Как было видно, каждое правило работает с какими-либо параметрами решаемой задачи, передаточной партии, ГПЯ и ориентируется на достижение различных целей. Поэтому выбор правила представляет сложную задачу, а если учесть эвристический характер большинства из них, то эта задача еще и слабо формализованная. Второй проблемой можно считать вопрос об оптимальности получаемых решений. Приоритетное правило, используемое при планировании или диспетчировании, обеспечивает выбор одного из вариантов функционирования ГПЯ и отбрасывание остальных вариантов, возможных в рассматриваемой ситуации. Из всех возможных путей протекания производственного процесса отдельное приоритетное правило реализует лишь один. В процессе решения критериальная функция, оценивающая эффективность работы ГПЯ, приоритетным правилом не используется. Выбор приоритетного правила лишь опосредовано оказывает влияние на критерий. Поэтому

использование метода приоритетов принципиально не позволяет говорить об оптимальности функционирования ГПЯ. В этом заключается одно из основных отличий эвристических методов от методов математического программирования, в которых целевая функция играет важнейшую роль в решении и результат получается оптимальным [24, 84, 85].

Вопрос отыскания способа оценки эффективности того или иного приоритетного правила может решаться на основе сравнения отдельных правил между собой по выбранному критерию эффективности или по степени приближения даваемого им решения к оптимальному. На практике используется исключительно сравнительный метод оценки эффективности компьютерным моделированием функционирования ГПЯ [11, 22].

Выводы

Обеспечение конкурентоспособности продукции отечественного машиностроения на внутреннем и внешнем рынках требует применения современных технологий и средств производства. Как показывает мировая практика, современными средствами производства служат компьютерно управляемые комплексы производственного оборудования – гибкие производственные ячейки и системы. Создание ГПЯ «под ключ» не является проблемой и для отечественных станкостроительных предприятий. Проблема заключается в отсутствии инженерных методов расчета проектных параметров и эксплуатационных режимов этих сложных технических систем. Практически единственным способом анализа и синтеза гибких производственных ячеек служит компьютерное моделирование их работы при различных характеристиках оборудования и алгоритмах его взаимодействия. Однако поиск математических методов, оптимально приспособленных для построения компьютерных моделей ГПЯ, похоже, еще далек от завершения. Возникает проблема разработки нового подхода к формализованному описанию функционирования гибких производственных ячеек в виде информационного, математического алгоритмического и программного обеспечения автоматизированной системы научных исследований гибких производственных систем - АСНИ ГПЯ. Таким подходом может стать метод автоматизированного построения и анализа циклограмм работы оборудования.

2 Оценка эффективности создания новой техники

Создание ГПЯ, как и любой другой новой техники, тем выгоднее, чем меньшим окажется срок окупаемости затрат. Существует ряд примеров быстрой окупаемости ГПЯ: 2.5-3 года против 5 - 7 лет для универсального оборудования [5, 10, 19, 20, 73]. Вместе с тем из более чем 1000 ГПЯ, внедренных в промышленно развитых странах к 1985 году, лишь 230 соответствовали данному показателю [5].

Прослеживается необходимость разработки путей и методов оценки эффективности создания ГПЯ. Для этого попытаемся разобраться в механизме их формирования.

2.1 Зависимость срока окупаемости ГПЯ от технических, экономических и организационных решений

Основными показателями эффективности капитальных вложений в создание новой техники являются требуемый объем капитальных вложений и время оборота капитала. Важную роль имеют процент прибыли, получаемый с суммы капитальных затрат в единицу времени (рентабельность), и объем прибыли за весь срок эксплуатации новой техники.

В общем виде объем прибыли P (руб.) за начальный срок T_0 (час) эксплуатации ГПЯ можно определить как разность между стоимостью реализованной продукцией и затратами на ее изготовление

$$P = (C - S) \times N - K, \quad (2.1)$$

где C - рыночная цена единицы продукции, руб./шт.;

S - расходы, связанные с изготовлением и сбытом единицы продукции (себестоимость), включая обслуживание капитальных вложений (кредиты, амортизация оборудования и др.), руб./шт.;

N - объем реализации продукции, шт.;

K - капитальные вложения в создание новой техники для выпуска данной продукции, руб.

Очевидно, что создание ГПЯ окажется целесообразным при выполнении условия $P \geq 0$.

Знак равенства прибыли нулю означает условие неубыточности создания ГПЯ, когда создание новой техники не приносит заметной материальной выгоды, однако позволяет повысить уровень автоматизации оборудования и накопить опыт его использования, повысить качество выпускаемой продукции, улучшить культуру производства и т.д. С учетом сказанного выражение (2.1) примет вид

$$(C - S) \times N - K \geq 0, \text{ откуда}$$

$$N \geq \frac{K}{(C - S)}. \quad (2.2)$$

Выражение (2.2) позволяет при прочих равных условиях определить объем продукции, начиная с выпуска которого целесообразно применение ГПЯ. Поскольку заданным значениям K и S соответствует и достигаемый уровень производительности оборудования, определим N как

$$N = \frac{T_0}{t_{cp}}, \quad (2.3)$$

где T_0 – рассматриваемый период эксплуатации ГПЯ, час;
 t_{cp} – среднее время изготовления единицы продукции, час.

Из (2.2) с учетом (2.3) определим T_0 как срок окупаемости капитальных затрат

$$T_0 = \frac{K \times t_{cp}}{(C - S)}. \quad (2.4)$$

Выразим через T_0 (час) количество лет работы ГПЯ L_O (год) до полной окупаемости

$$L_O = \frac{T_0}{(365 - C_{np}) \times (T_{cm} - T_{np}) \times K_{ГПС} \times n}, \quad (2.5)$$

где C_{np} – число суток простоя ГПЯ в году, шт.;
 T_{cm} – продолжительность одной смены работы ГПЯ, час;
 T_{np} – внутрисменные внецикловые простои оборудования, час;
 $K_{ГПС}$ – коэффициент использования оборудования ГПЯ внутри цикла;
 n – коэффициент сменности работы оборудования ГПЯ.

Из выражения (2.5) с учетом выражения (2.4) получим формулу для расчета количества лет окупаемости затрат на создание ГПЯ:

$$L_O = \frac{K \times t_{cp}}{(C - S) \times (365 - C_{np}) \times (T_{cm} - T_{np}) \times K_{ГПС} \times n} \quad (2.6)$$

Полученное выражение позволяет оценить срок окупаемости капитальных затрат на создание ГПЯ, для которой известны ее технические ($k_{ГПС}$, t_{cp}), а также организационные (условия эксплуатации) и экономические параметры (C , S).

Можно также при заданном сроке окупаемости капиталовложений выявить необходимые условия эксплуатации ГПЯ или допустимый объем затрат на изготовление и реализацию продукции. Рассмотрим возможные пути сокращения срока окупаемости капитальных затрат на создание ГПЯ.

2.2 Влияние рыночной стоимости изготавливаемой продукции на срок окупаемости ГПЯ

В условиях рынка цена формируется с учетом спроса и предложения на данный вид продукции. Стремление увеличить цену на выпускаемую продукцию может быть достигнуто несколькими способами, известными и законными из которых являются: маркетинговые исследования, позволяющие выявить продукцию повышенного спроса и создать производство по ее выпуску; реклама выпускаемой продукции; научные исследования, на основе которых рождается новый вид продукции, отсутствующий на рынке. Каждый из перечисленных способов требует определенных затрат на рекламу, исследования рынка или на опытно-конструкторские разработки, что в свою очередь ведет к увеличению затрат на реализацию продукции S либо требуемых капиталовложений K .

Рассмотрим, как изменение цены на изготавливаемую продукцию влияет на срок окупаемости капитальных вложений. Пусть начальные значения выглядят следующим образом: $K = 500000,0$ \$; $C_{np} = 65,0$ суток; $C = 100,0$ \$; $T_{np} = 1,0$ час; $t_{cp} = 1,0$ час; $T_{cm} = 8,0$ час; $S = 50,0$ \$; $k_{ГПС} = 0,5$; $m = 2$ смены. Знаком «\$» отмечена условная денежная единица.

Расчеты, выполненные для представленных данных с использованием полученных зависимостей (2.3 – 2.6) показывают (рисунок 2.1), что при увеличении цены на единицу выпускаемой продукции от 100 до 280 \$, т.е. в 2,8 раза, срок окупаемости капитальных затрат сокращается с 4,76 лет до 1,04 года - в 4,58 раза (Кривая $\Delta S = 0$).

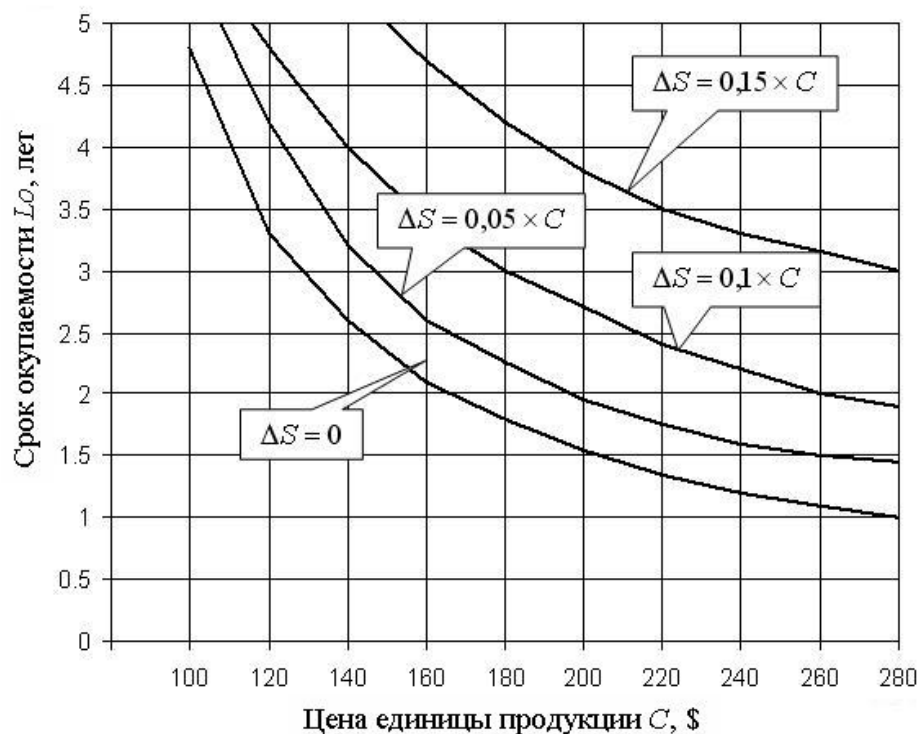


Рисунок 2.1 - Влияние цены C и себестоимости S продукции на величину срока окупаемости капитальных затрат в ГПЯ

Кривые на рисунке 2.2 иллюстрируют зависимость срока окупаемости ГПЯ от величины приращения издержек ΔS на реализацию выпускаемой продукции при ее различной рыночной стоимости.

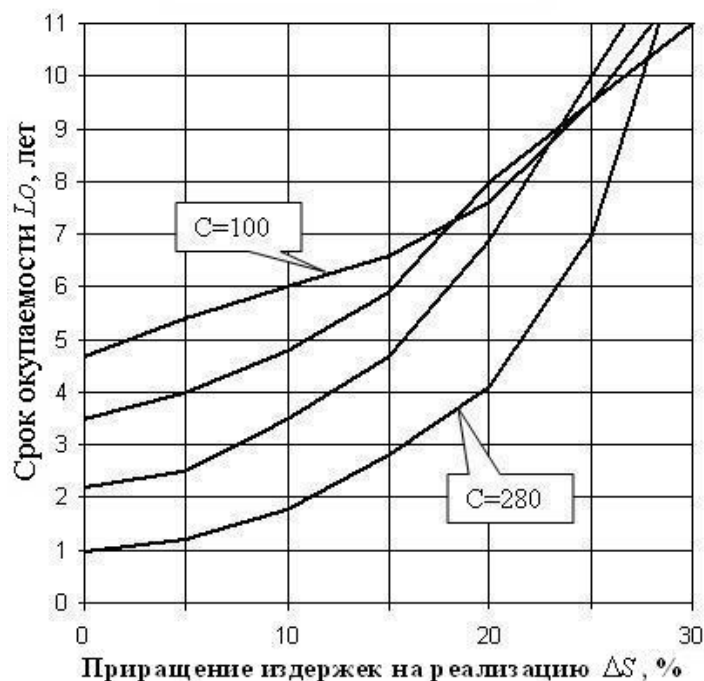


Рисунок 2.2 - Зависимость срока окупаемости ГПЯ от цены единицы продукции и приращения издержек на реализацию ΔS

Увеличение рыночной стоимости продукции возможно на основе улучшения ее потребительских свойств (надежности, долговечности, экологических показателей и др.) на основе проводимых научных исследований и конструкторских разработок. Данный путь предполагает использование дополнительных капиталовложений на выполнение указанных разработок.

На рисунке 2.3 показано влияние роста цены единицы продукции и приращения суммы капитальных вложений ΔK на срок окупаемости затрат. Полученные данные иллюстрируют, что повышение цены продукции за счет совершенствования ее потребительских свойств целесообразно до определенного соотношения (равновесия) между ростом цены и требуемым приростом капитальных затрат ΔK : в рассмотренном примере критическим является значение $\Delta K = 0,004 \% K$, при котором дальнейшее повышение затрат не компенсируется ростом цены. В результате срок окупаемости может увеличиваться.

Таким образом, эффективность создания гибких производственных систем, оцениваемая как продолжительность срока окупаемости капитальных затрат, в условиях рыночной экономики может обеспечиваться экономическими решениями.

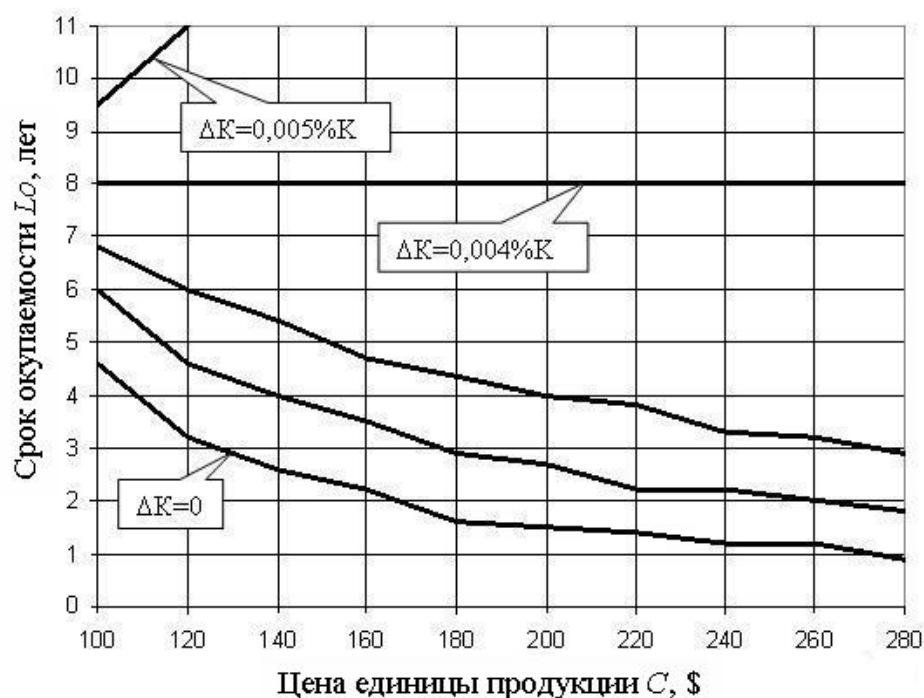


Рисунок 2.3 - Зависимость срока окупаемости ГПЯ от цены единицы продукции и приращения капитальных затрат

2.3 Влияние величины календарных простоев на срок окупаемости ГПЯ

К календарным простоям обычно относятся простои оборудования в выходные и праздничные дни, а также простои, связанные с плановым профилактическим ремонтом (ППР) оборудования. Согласно[5] доля календарных простоев может составлять 30 % годового фонда времени.

Сокращение простоев данной группы можно обеспечить за счет организации работы ГПЯ в выходные и праздники, а также за счет сокращения продолжительности ППР. В обоих случаях необходимы дополнительные затраты, связанные либо с увеличенной оплатой труда, либо с приобретением более дорогостоящего и долговечного оборудования. На рисунке 2.4 представлено влияние величины календарных простоев на продолжительность срока окупаемости капитальных затрат.

В случае, если сокращение календарных простоев оборудования не требует дополнительных затрат (кривая $\Delta K = 0$), то для использованных данных 25 %-ое сокращение календарных простоев позволяет примерно на столько же (23,2 %) сократить срок окупаемости, снизив его с 5,39 до 4,14 лет. Необходимость использования в данном случае дополнительных капитальных вложений, как свидетельствуют остальные кривые рисунка 2.4, приводит к отрицательному эффекту. Следует отметить, что календарные простои на отметке 31 - 35 % годового фонда времени соответствуют наименьшим колебаниям срока окупаемости при различной величине капитальных затрат. Видимо, это оптимальное, исторически сложившееся соотношение.

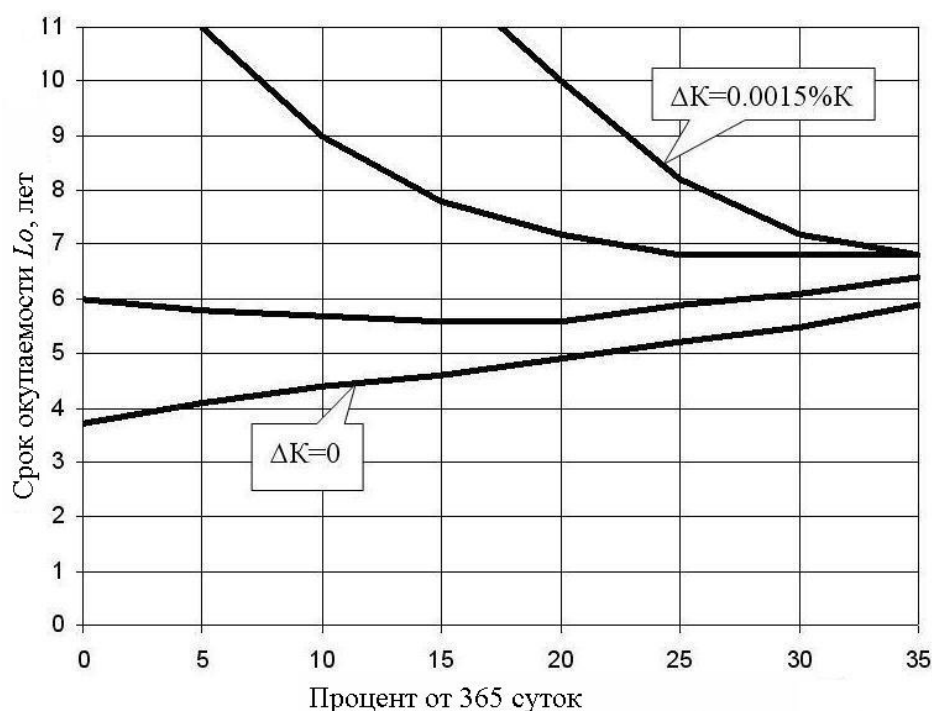


Рисунок 2.4 - Зависимость срока окупаемости ГПЯ от величины календарных простоев оборудования

2.4 Влияние сменности работы оборудования ГПЯ на срок окупаемости капитальных затрат

Основным свойством ГПЯ является ее способность «автономно функционировать в течение заданного интервала времени» [13]. Технически данное свойство обеспечивается выбором емкости накопителей заготовок и инструмента, с которыми станки связаны автоматическим транспортом. Возможность автономно функционировать позволяет вести обработку заготовок в выходные и праздничные дни, во вторую и третью смены. «Даже если обеспечить работу ГПЯ только в рабочие дни и по 20 часов в сутки, общий годовой фонд времени использования оборудования возрастет не менее чем в 5 раз» [13]. Следует добавить, что работа ГПЯ, имеющих короткий срок окупаемости капитальных затрат, организована именно в 2,5 - 3 смены [19, 20, 73]. Таким образом, увеличение годового фонда времени работы ГПЯ за счет увеличения коэффициента сменности работы оборудования, является одним из путей повышения их эффективности. Увеличение коэффициента сменности может сопровождаться дополнительными капитальными затратами и издержками на производство единицы продукции. На рисунках 2.5 и 2.6 представлены результаты соответствующих расчетов срока окупаемости ГПЯ.

Результаты показывают, что увеличение коэффициента сменности работы ГПЯ, не сопровождаемое увеличением капитальных затрат или издержек на единицу продукции, позволяет увеличить годовой фонд времени работы ГПЯ и за счет этого сократить календарный срок окупаемости затрат.

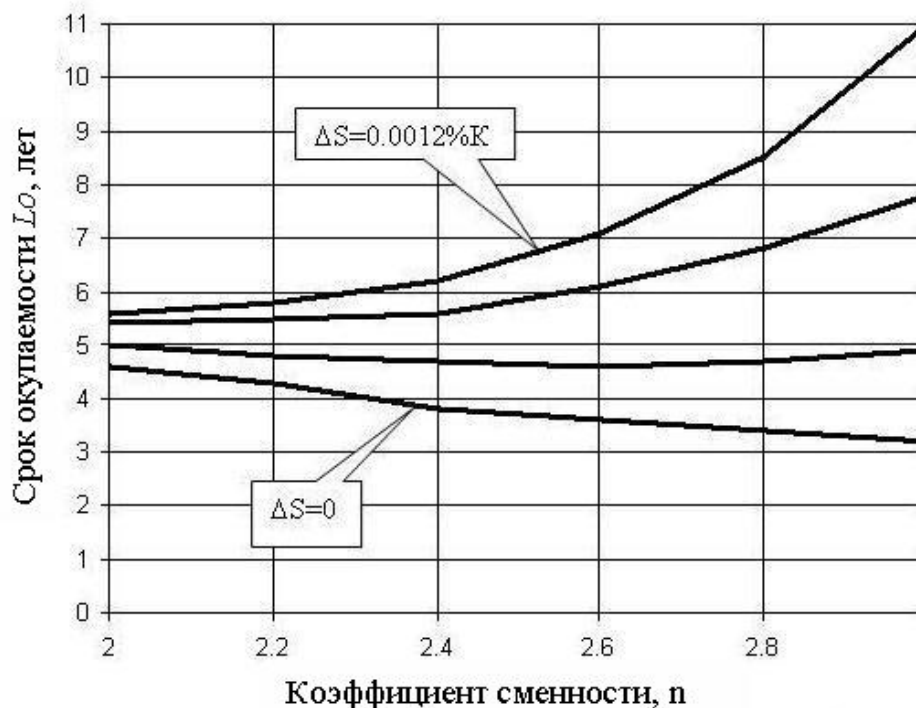


Рисунок 2.5 – Срок окупаемости ГПЯ в зависимости от коэффициента сменности n и приращения затрат на единицу продукции ΔS

Для рассмотренных данных (кривые $\Delta S = 0$ и $\Delta K = 0$ на рисунках 2.5 и 2.6) максимальное сокращение срока окупаемости составляет 33,4 %. Однако дополнительные капитальные вложения или издержки на единицу продукции могут привести к прямо противоположному результату.

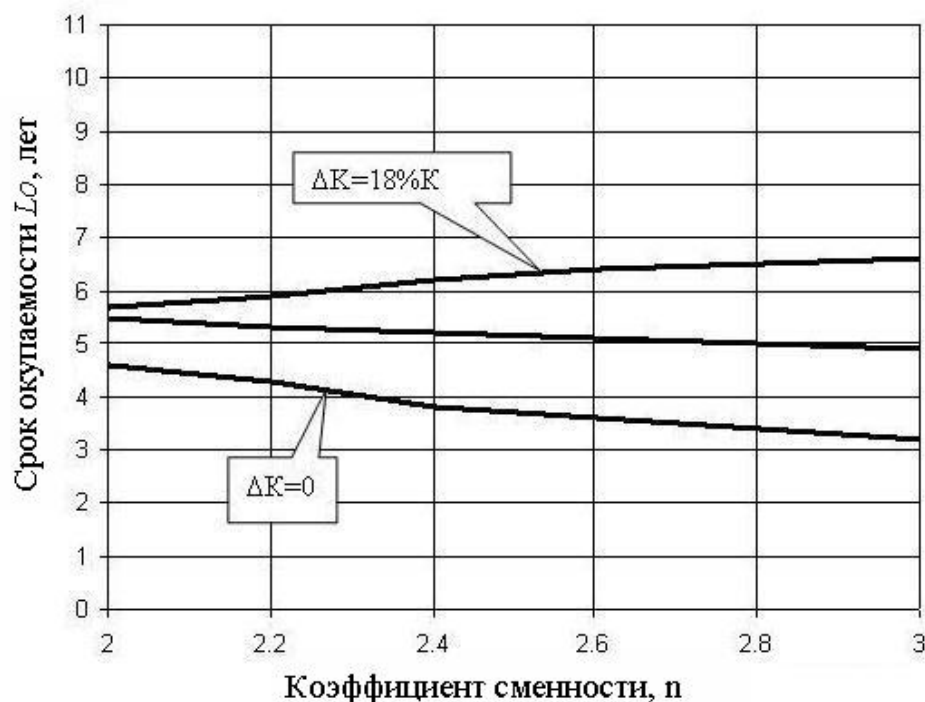


Рисунок 2.6 – Срок окупаемости ГПЯ в зависимости от коэффициента сменности n и приращения капитальных затрат ΔK

2.5 Влияние величины внутрисменных простоев оборудования на срок окупаемости ГПЯ

Существуют различные подходы к дифференциации внутрисменных простоев оборудования. В работе [5] на основе заводской практики внутрисменные простои ГПЯ подразделены на две группы: 1 группа - цикловые потери рабочего времени. Сюда относятся холостые ходы машины-автомата в общем цикле производства одной детали, а также подача материалов, фиксация, зажим заготовки и инструмента, подводы и отводы рабочих органов, переключения отдельных механизмов и т.д.; 2 группа - внецикловые потери. Сюда относятся простои ГПЯ из-за отказов инструмента, оборудования, по организационным причинам, из-за предупреждения брака, из-за переналадки ГПЯ и прочие. В работе представлены результаты анализа потерь времени участка из 4-х станков с ЧПУ до и после объединения в ГПЯ (таблица 2.1).

Данные по работе ГПЯ свидетельствуют, что время резания и цикловые потери составляют лишь 43,4 %, т.е. коэффициент загрузки ГПЯ $k_{ГПЯ}$ составляет порядка 0,43.

По данным французских авторов [73] время резания и цикловые потери, включая время на смену заготовок и инструмента, составляют в ГПЯ 48,5 % календарного фонда времени. В работе [67] показано, что за счет устранения простоев по оргпричинам и по вине персонала (прочие- по терминологии [5]) можно повысить действительный фонд времени на 37 %, т.е. довести его долю в календарном фонде времени до 80,4 %.

Таблица 2.1 - Структура потерь времени 4-х станков типа ОЦ ([5])

Структура времени работы станков	В составе участка	В составе ГПЯ
Время резания	26,8	35,3
Цикловые потери	11,6	8,1
Внецикловые потери по:	61,6	56,6
- инструменту	11,7	8,2
- оборудованию	2,9	10,9
- оргпричинам	24,5	17,1
- браку	3,1	2,2
- переналадке	5,7	6,3
- прочие	13,7	11,9
Итого	100	100

В работе [86] описаны результаты исследования условий работы двух ГПЯ, эксплуатируемых на предприятиях ФРГ, в условиях крупно- и среднесерийного производства. Исследования проводились Исследовательским институтом по рационализации (FIR) Высшей технической школы в Аахене (RWTH). За время исследования (20 дней) действительный фонд времени работы ГПЯ составил 84,8 % для одной и 84,9 % для другой системы. В выводах к работе указано, что

недоиспользование фонда времени (15,2 % и 15,1 %, соответственно) во многом предопределилось недостаточной квалификацией и компетентностью обслуживающего персонала.

Таким образом, прослеживается тенденция увеличения доли времени работы в календарном фонде времени ГПЯ за счет сокращения простоев станков из-за отказов инструмента и оборудования, совершенствования организации производства, повышения квалификации обслуживающего персонала. Сокращение внутрисменных простоев оборудования сопряжено с дополнительными капитальными затратами, связанными с использованием высоконадежного, а, следовательно, и более дорогого оборудования и инструмента, с применением автоматизированной переналаживаемой технологической оснастки, с тщательной профессиональной подготовкой обслуживающего персонала и совершенствованием организации производства.

На рисунке 2.7 представлены результаты расчета срока окупаемости ГПЯ при разной величине внутрисменных простоев оборудования T_{np} и при разной величине сопутствующих дополнительных капитальных затрат $\Delta K = f(K)$.

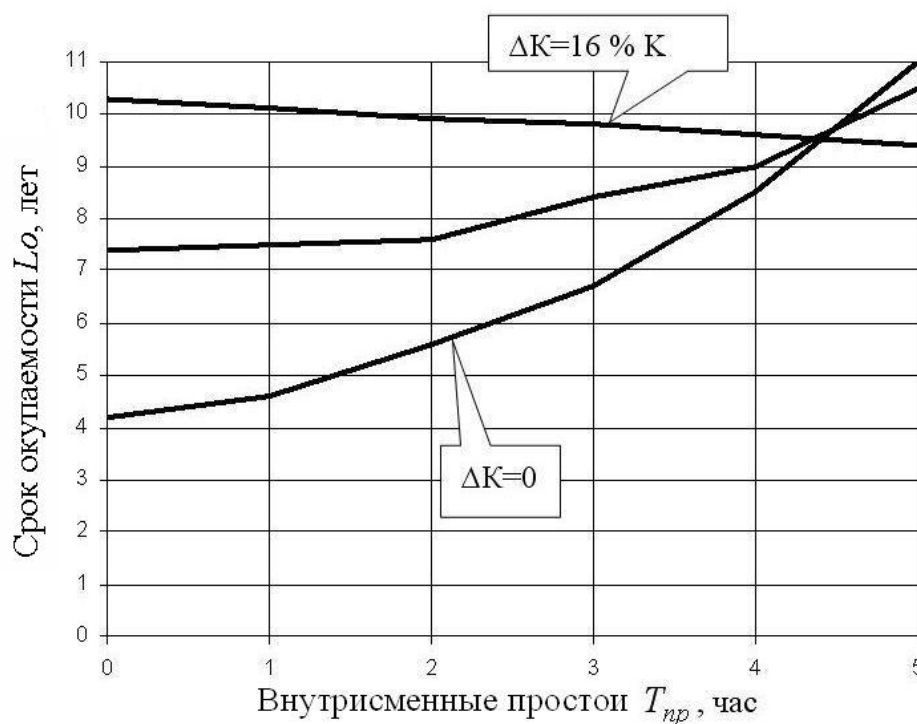


Рисунок 2.7 - Срок окупаемости ГПЯ в зависимости от величины внутрисменных простоев T_{np} и дополнительных капитальных затрат ΔK

Как следует из представленного примера, за счет сокращения внецикловых внутрисменных простоев оборудования можно обеспечить существенное сокращение срока окупаемости ГПЯ.

При этом выигрыш существенно зависит от уже достигнутого уровня: так, при сокращении T_{np} с 4 до 3 часов (на 25 % в смену при отсутствии дополнительных капитальных затрат срок окупаемости уменьшится с 8,33 до 6,67 лет (на 20,0 %)). Сокращение же с 1 до 0 часов ведет к сокращению срока окупаемости с 4,76 до 4,17 лет, т.е. на 14,2 %. Как показывают расчеты, дополнительные капитальные

вложения могут привести и к обратному результату. Следует отметить, что дополнительные капитальные затраты в сокращение внутрисменных простоев, при которых может увеличиться срок окупаемости ГПЯ, весьма существенны и составляют в рассмотренном примере порядка 15 %.

2.6 Влияние коэффициента загрузки оборудования ГПЯ на срок окупаемости капитальных затрат

Определим действительный фонд времени работы ГПЯ как продолжительность рабочей смены за вычетом внутрисменных внецикловых простоев оборудования: $T_{см} - T_{пр}$. Тогда коэффициент загрузки ГПЯ $k_{ГПЯ}$ можно представить как отношение действительного фонда времени за вычетом внутрицикловых простоев оборудования $T_{ц}$ к действительному фонду времени ГПЯ

$$k_{ГПЯ} = \frac{(T_{см} - T_{пр} - T_{ц})}{(T_{см} - T_{пр})}. \quad (2.7)$$

Коэффициент загрузки $k_{ГПЯ}$ и время изготовления единицы продукции $t_{шт}$ характеризуют эффективность использования действительного фонда времени работы ГПЯ. Большинство современных ГПЯ имеют $k_{ГПЯ}=0,35 - 0,45$, наиболее удачные образцы – $0,75 - 0,8$ [5].

На рисунке 2.8 представлены результаты расчета срока окупаемости ГПЯ при разных значениях $k_{ГПЯ}$ и разной величине приращения капитальных затрат.

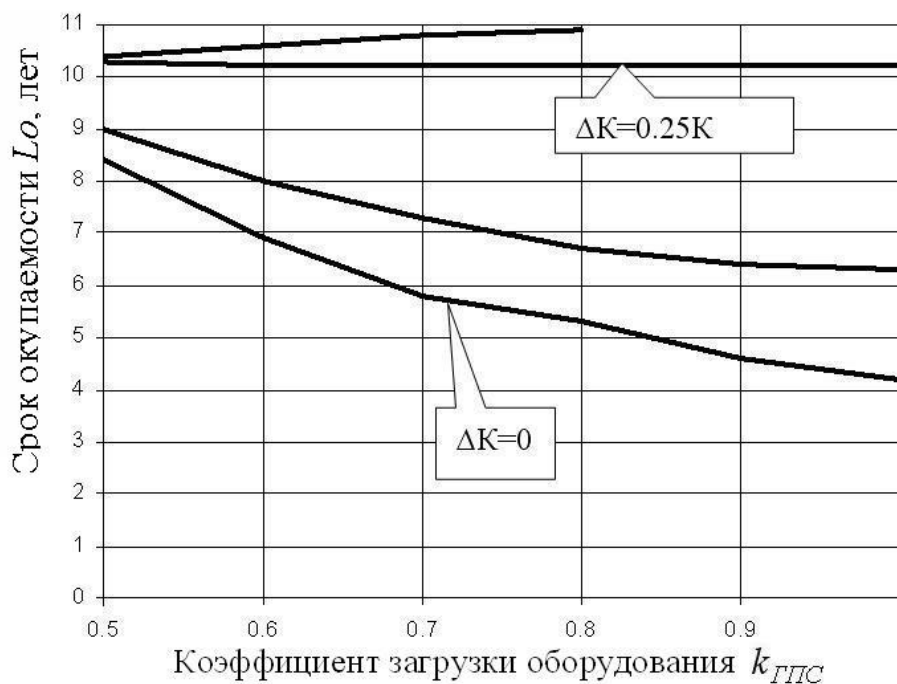


Рисунок 2.8 - Срок окупаемости ГПЯ в зависимости от коэффициента загрузки оборудования $k_{ГПЯ}$ и дополнительных капитальных затрат ΔK

Как следует из представленного примера, увеличение $k_{ГПЯ}$ с 0,5 до 0,8 при отсутствии дополнительных капитальных затрат позволяет сократить срок окупаемости ГПЯ с 8,33 до 5,21 лет, т.е. на 60 %. Сопутствующие капитальные затраты, при вложении которых увеличение $k_{ГПЯ}$ становится невыгодно, соответствуют примерно 25 % общей суммы капитальных вложений.

Таким образом, повышение коэффициента загрузки основного оборудования следует рассматривать как перспективное направление совершенствования гибких производственных ячеек.

2.7 Влияние производительности ГПЯ на срок ее окупаемости

При анализе работы производственных систем различают производительность сменную, штучную, минутную и цикловую [63]. Цикловая производительность системы оценивается затратами действительного фонда времени (за исключением цикловых простоев) t_{um} на изготовление единицы продукции. Для металлорежущего оборудования цикловая производительность может быть увеличена за счет форсирования режимов обработки. Возрастающий при этом расход режущих инструментов может быть отнесен к увеличению производственных издержек на единицу продукции.

Результаты расчета срока окупаемости ГПЯ при увеличении цикловой производительности оборудования и росте соответствующих издержек производства представлены на рисунке 2.9.

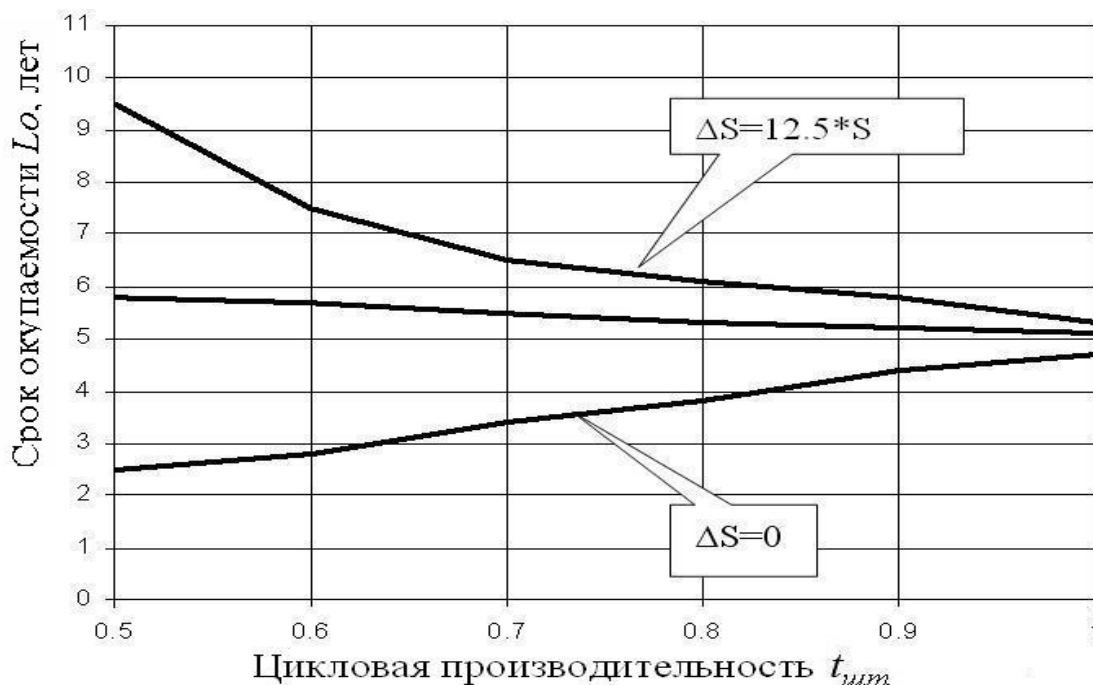


Рисунок 2.9 - Срок окупаемости ГПЯ в зависимости от цикловой производительности t_{um} и приращения издержек производства ΔS

Результаты показывают, что сокращение времени изготовления единицы продукции при достаточно малом (до 10 %) приросте ее стоимости сокращение срока окупаемости капитальных затрат может быть весьма существенным: так, для кривой $\Delta S = 0$ на рисунке 2.9 50 %-ое сокращение времени изготовления также составляет 50 %, т.е. имеет место линейная зависимость. Наличие приращения издержек производства в размере 12,5 % и более от стоимости единицы продукции ведет к отрицательному результату.

2.8 Анализ эффективности различных путей сокращения срока окупаемости ГПЯ

Рассмотренные пути сокращения срока окупаемости капитальных затрат на создание ГПЯ позволяют обеспечить требуемую эффективность новой техники за счет принятия рациональных экономических, организационных и технических решений.

Экономические решения, связанные с изучением спроса и предложения, назначением цены на продукцию и объемов выпуска требуют специальных исследований, на основании которых и формируется производственная программа ГПЯ.

В то же время возникает вопрос: а насколько эффективны организационно-технические решения? Другими словами, за счет чего можно, вложив одну и ту же сумму затрат, получить наибольшее сокращение срока окупаемости затрат на ГПЯ? За счет сокращения календарных простоев C_{yt} , коэффициента сменности n , внутрисменных простоев T_{np} , коэффициента загрузки $k_{ГПЯ}$ или цикловой производительности t_{um} ?

Поскольку изменение вышеперечисленных показателей вызывают изменения суммы капитальных затрат K либо издержек на производство единицы продукции, установим соотношение между величинами приращения ΔK и δS . Пусть приращению суммы капитальных затрат ΔK соответствует приращение издержек на производство единицы продукции δS . Из условия постоянства срока окупаемости и объема выпуска (2.2)

$$\frac{K}{C - S} = \frac{\Delta \times K}{C - \delta \times S}, \quad (2.8)$$

откуда

$$C - \delta \times S = \Delta \times (C - S), \quad (2.9)$$

окончательно

$$\delta = \frac{(C - \Delta \times (C - S))}{S}. \quad (2.10)$$

Рассмотрим сокращение срока окупаемости затрат при пропорциональном улучшении исследуемых показателей. Для этого в выражение (2.6) будем поочередно подставлять значения показателей, соответствующие 10 %, 20 %, ..., 100 % от их а) минимального (максимального) значения и б) достигнутого (заданного) значения.

Приращения показателей от предельного допустимого значения

$$\left. \begin{aligned} \Delta t_{ум} &= 0,1 \times 10 \times i \\ C_{ум} &= 365/100 \times 10 \times i \\ n &= 3/100 \times 10 \times i \\ T_{пр} &= 8/100 \times 10 \times i \\ k_{ГПЯ} &= 1/100 \times 10 \times i \end{aligned} \right\} i \in \{1, 2, 3, \dots, 10\} \quad (2.11)$$

Приращения показателей от достигнутого значения

$$\left. \begin{aligned} \Delta t_{ум} &= 0,1 \times 10 \times i \\ C_{ум} &= 65/100 \times 10 \times i \\ n &= 2 + 1/100 \times 10 \times i \\ T_{пр} &= 4/100 \times 10 \times i \\ k_{ГПЯ} &= 0,5 + 0,5/100 \times 10 \times i \end{aligned} \right\} i \in \{1, 2, 3, \dots, 10\} \quad (2.12)$$

Результаты расчетов представлены на рисунках 2.10, 2.11. Представленные зависимости позволяют оценить степень влияния различных факторов на срок окупаемости капитальных затрат при условии, что приращение значений показателей не требует дополнительных капитальных затрат ($\Delta K = 0$). На рисунках 2.12, 2.13 представлены те же зависимости, однако в предположении, что приращение значений показателей требует увеличения затрат в 1,5 раза - $\Delta = 1,5$ и, соответственно,

$$\delta = \frac{(C - 1,5 \times (C - S))}{S}.$$

В соответствии с полученными результатами расположим рассматриваемые показатели в порядке убывания их влияния на срок окупаемости ГПЯ:

- цикловая производительность,
- коэффициент загрузки оборудования,
- коэффициент сменности работы,
- календарные простои,
- внутрисменные простои ГПЯ.

Вышеизложенное позволяет сделать вывод, что наименьший срок окупаемости ГПЯ может быть достигнут за счет повышения производительности и ко-

эффициента загрузки основного оборудования, т.е. за счет принятия рациональных технических решений. Дополнительное сокращение срока окупаемости обеспечивается рациональным увеличением коэффициента сменности работы оборудования, а также сокращением календарных и внутрисменных простоев системы.

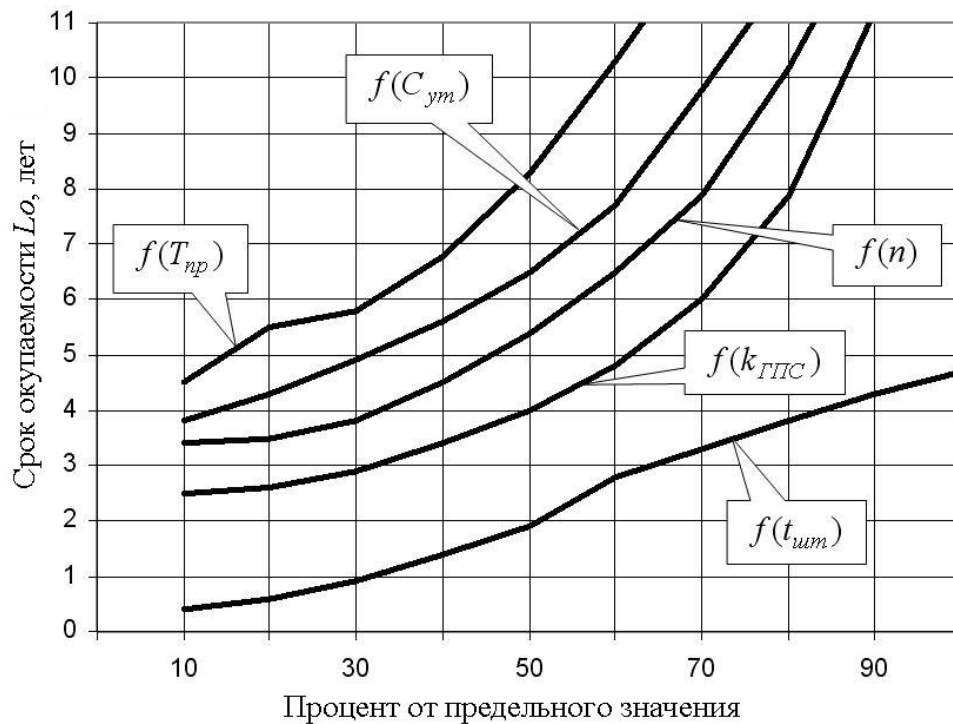


Рисунок 2.10 – Закономерности изменения срока окупаемости ГПЯ при приращении параметров от предельного допустимого значения ($\Delta K = 0$)

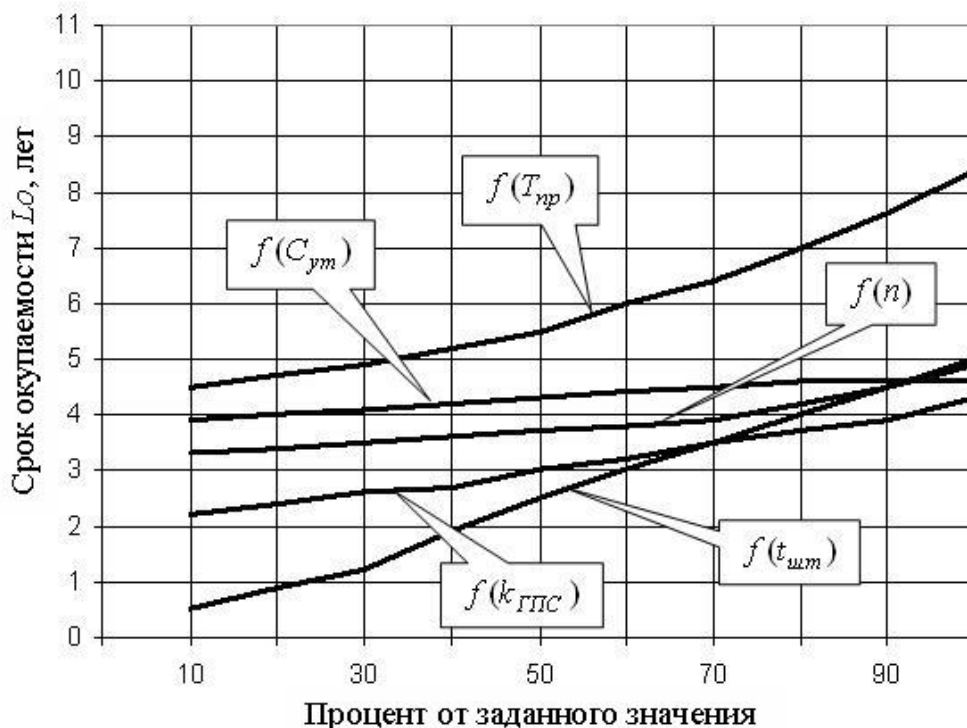


Рисунок 2.11– Закономерности изменения срока окупаемости ГПЯ при приращении параметров от заданного значения ($\Delta K = 0$)

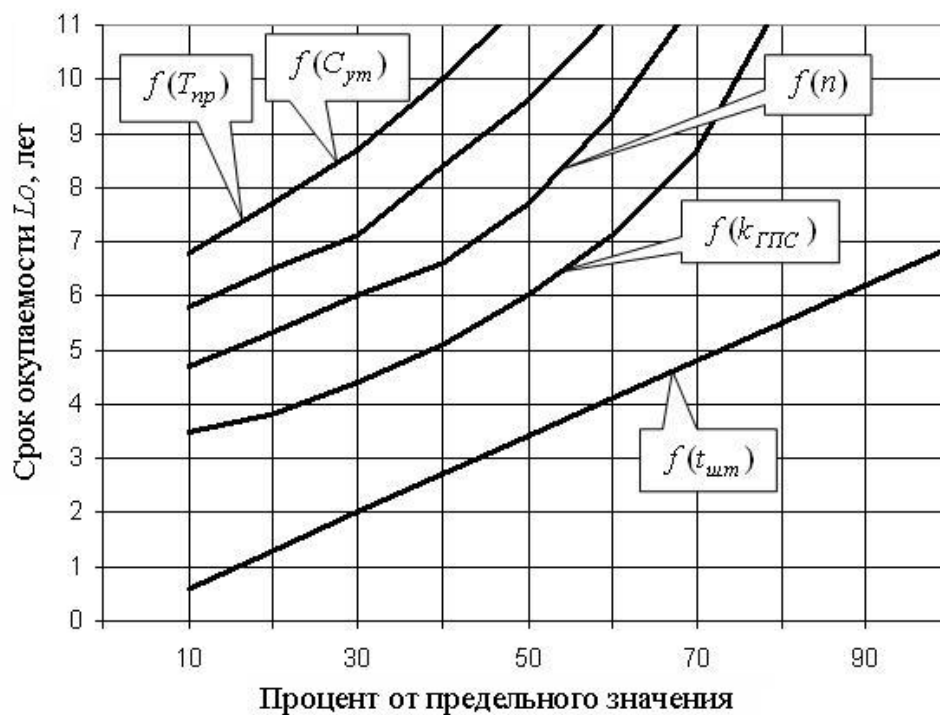


Рисунок 2.12– Закономерности изменения срока окупаемости ГПЯ при приращении параметров от предельного допустимого значения ($\Delta = 1,5$)

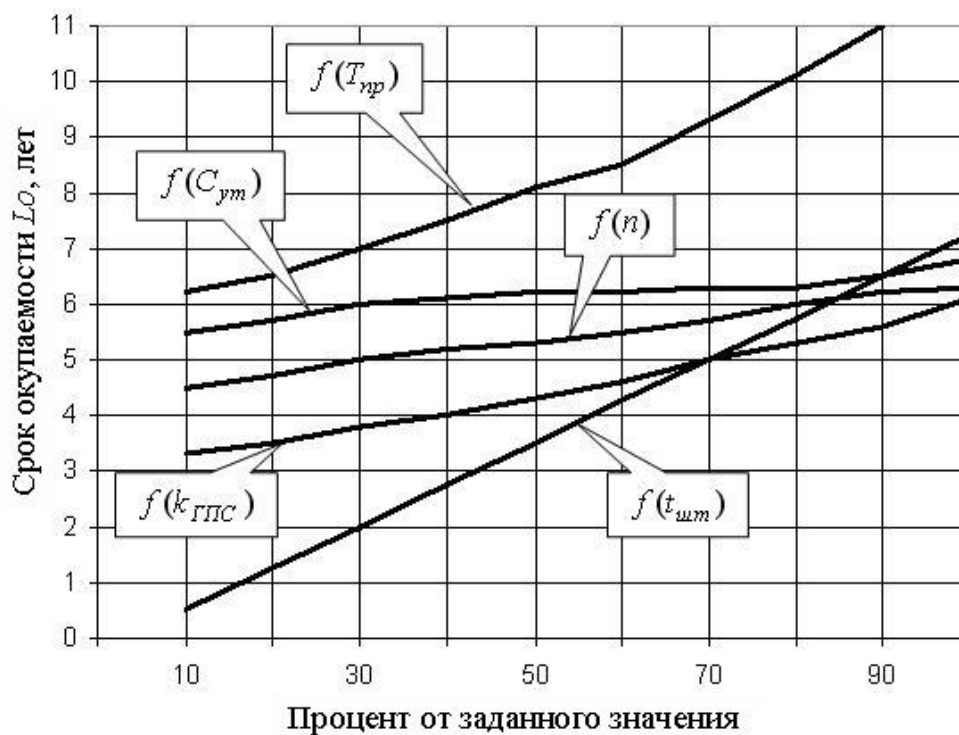


Рисунок 2.13– Закономерности изменения срока окупаемости ГПЯ при приращении параметров от заданного значения ($\Delta = 1,5$)

Выводы

Получено аналитическое выражение, описывающее закономерности изменения срока окупаемости затрат на создание ГПЯ в зависимости от экономических, технических и организационных решений.

Установлено, что наименьший срок окупаемости ГПЯ может быть достигнут за счет повышения производительности и коэффициента загрузки основного оборудования, т.е. за счет принятия рациональных технических решений.

Резервом повышения производительности и коэффициента загрузки основного оборудования служит сокращение внутрисменных простоев оборудования.

Сокращение внутрисменных простоев оборудования сопряжено с дополнительными капитальными затратами, связанными с использованием высоконадежного, а, следовательно, и более дорогого оборудования и инструмента, с применением автоматизированной переналаживаемой технологической оснастки, с тщательной профессиональной подготовкой обслуживающего персонала и совершенствованием организации производства.

Для количественной оценки внутрисменных простоев ГПЯ необходимо провести детальный анализ величины и причин потерь времени при работе основного технологического оборудования.

3 Пути повышения производительности и загрузки оборудования ГПЯ

Как показано выше, основное условие быстрой окупаемости ГПЯ – это как можно более полное использование их во времени. Этот и ряд других факторов ужесточили требования к временной и пространственной организации автоматизированного производственного процесса, и, как следствие, усложнили его проектирование.

Графической иллюстрацией производственного процесса являются временные диаграммы (циклограммы) работы ГПЯ, включающие циклограммы взаимосвязанной работы технологических и транспортных модулей и их агрегатов. На циклограммах в выбранном масштабе изображаются отрезки времени, символизирующие моменты начала, продолжительность и моменты завершения циклов работы устройств. По циклограммам можно судить и о продолжительности простоев устройств по различным причинам.

Степень детализации циклограмм зависит от целей их построения и решаемых с их помощью задач.

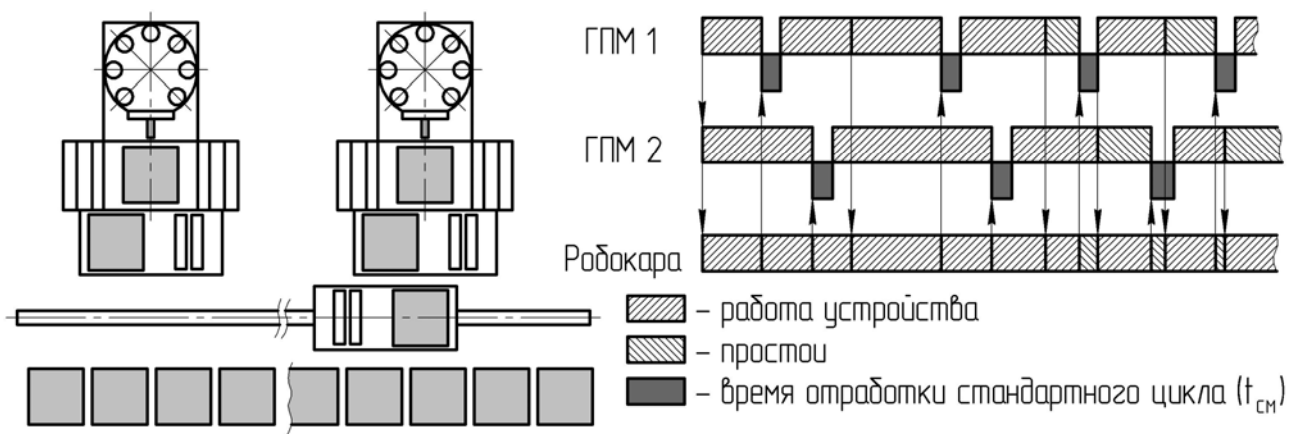
Рассмотрим циклограммы, иллюстрирующие состав и закономерности образования простоев основного оборудования ГПЯ.

3.1 Циклограммы работы ГПЯ на уровне технологической операции

Рассмотрим работу ГПЯ как процесс взаимодействия технологических и транспортных модулей, выполняющих, соответственно, технологические и транспортные операции. Пусть имеется ГПЯ из двух станков с линейным накопителем столов-спутников и двухместной транспортной тележкой (рисунок 3.1 а). Циклограмма (рисунок 3.1 б) отображает процесс работы участка на уровне технологической операции (без дифференциации длительности отдельных технологических переходов) при следующих исходных условиях. Перед началом цикла работы ГПЯ все заготовки на паллетах размещены в позициях линейного накопителя. Заявка от станка на доставку очередной заготовки из накопителя выдается в момент, когда заготовка из приемо-передающей позиции станка будет установлена на стол станка.

Работа ГПЯ начинается с опроса состояния станков: поскольку на обоих отсутствуют заготовки, то практически мгновенно выдаются заявки тележке на доставку их из накопителя. Тележка начинает выполнение заявок с обслуживания 1-го станка, простои которого от начала цикла работы участка до момента начала обработки первой заготовки составят $t_{np} = t_{mp} + t_{cm}$, где t_{mp} - время выполнения тележкой 1-ой транспортной операции, t_{cm} - время смены паллеты с заготовкой на столе 1-го станка.

Время выполнения робокарой транспортных операций пропорционально маршевой скорости, проходимым расстояниям и может быть различным в зависимости от местоположения позиции накопителя с нужной заготовкой и от местоположения обслуживаемого станка (предельные значения длительности транспортной операции можно выявить из планировки ГПЯ). С другой стороны, из-за широкой номенклатуры обрабатываемых заготовок время их обработки также различно.



а б
Рисунок 3.1 - Схема (а) и фрагмент циклограммы работы (б) ГПЯ из двух станков с линейным накопителем паллет

Различная продолжительность транспортных и технологических операций приводят к появлению простоев тележки и станков. Простой станков ведут к ухудшению использования действительного фонда времени. Избыточные простои тележки свидетельствуют о ее излишней производительности, что, скорее всего, потребовало неоправданных затрат на ее приобретение. Результатом является увеличение срока окупаемости затрат на ГПЯ.

Величина простоев станков зависит и от принятого способа управления очередностью выполнения транспортных операций (раздел 1.4).

Итак, взаимодействие станков и транспортных средств автоматического участка ведет к появлению простоев станков двух видов:

а) простои станков из-за несвоевременной доставки заготовок, проявляющиеся в процессе взаимодействия станков и транспортных средств и неподдающиеся расчету аналитическими методами;

б) простои, связанные с отработкой стандартных циклов смены заготовок в рабочих зонах станков. Продолжительность стандартных циклов смены паллет является паспортной характеристикой устройства смены, зависящей от его конструкции.

Отдельные схемы устройств автоматической смены паллет представлены на рисунке 3.2.

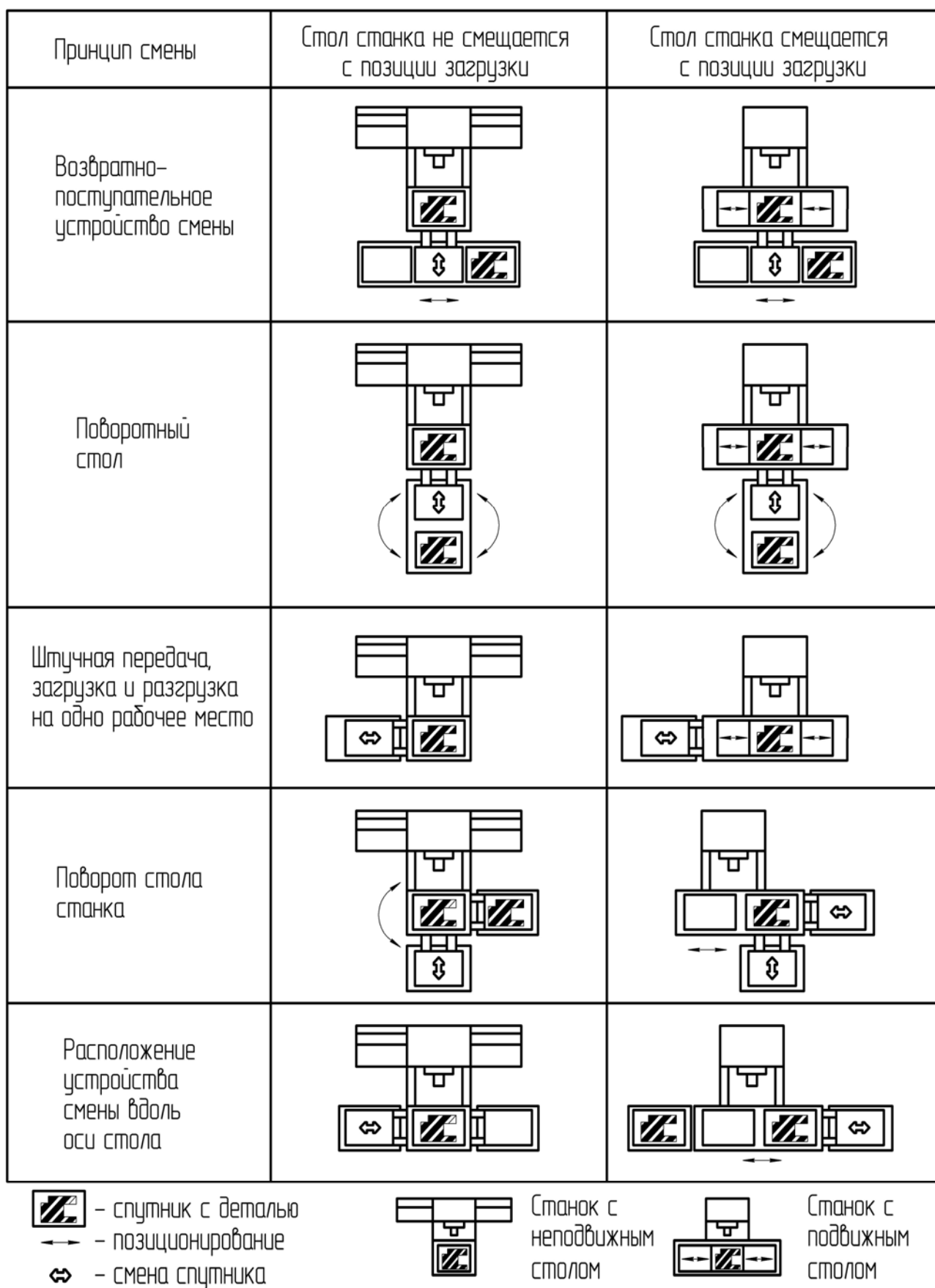


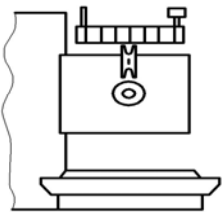
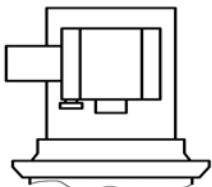
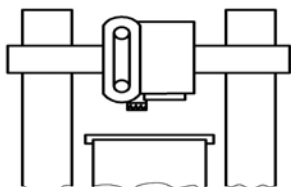
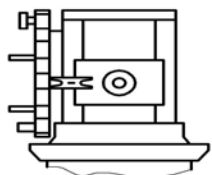
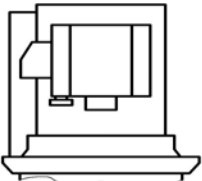
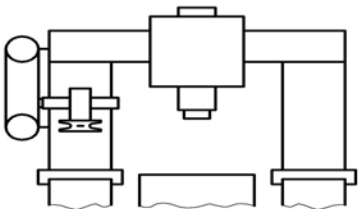
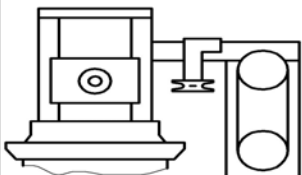
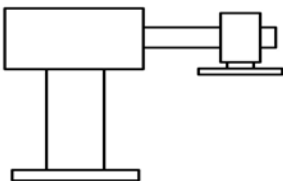
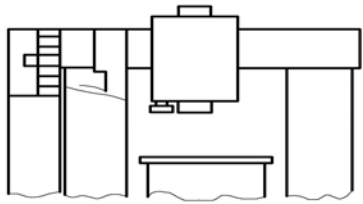
Рисунок 3.2 – Схемы устройств автоматической смены паллет на многоцелевых станках

3.2 Циклограммы работы ГПЯ на уровне технологического перехода

Как показывает практика, подавляющее большинство ГПЯ создается на базе станков типа «обрабатывающий центр» (ОЦ), обеспечивающих обработку заготовки с одной установки с 4 - 5 сторон. Использование станков типа ОЦ позволяет существенно сократить количество основного оборудования в ГПЯ, а также уменьшить число технологических и транспортных операций при обработке одной заготовки. Вместе с тем концентрация обработки на одном рабочем месте требует использования на станке широкой номенклатуры режущих инструментов. Форсирование режимов резания еще более увеличивает необходимый запас инструментов. Обычно ОЦ оснащаются инструментальными магазинами на 20 - 30 инструментов, однако в гибких производственных модулях число гнезд ИМ может достигать 200 и более гнезд.

Инструментальные магазины бывают дисковые, барабанные, цепные, с вертикальной, горизонтальной и наклонной осью вращения. Могут устанавливаться на шпиндельной бабке станка, на станине либо на отдельной стойке (таблица 3.1).

Таблица 3.1 – Разновидности схем устройств АСИ

Место ИМ	Станки с горизонтальным шпинделем	Станки с вертикальным шпинделем	Станки двухстоечные
На шпиндельной бабке			
На стойке сверху		—	—
На стойке сбоку			
Рядом со станком			

В последние годы все более широкое применение находят ИМ цепного типа, конструкции которых позволяют варьировать их вместимость (например, при включении станка в состав ГПЯ) и которым можно придать любую удобную конфигурацию [58]. Гистограмма распределения вместимости ИМ (по данным [62]) представлена в соответствии с рисунком 3.3.

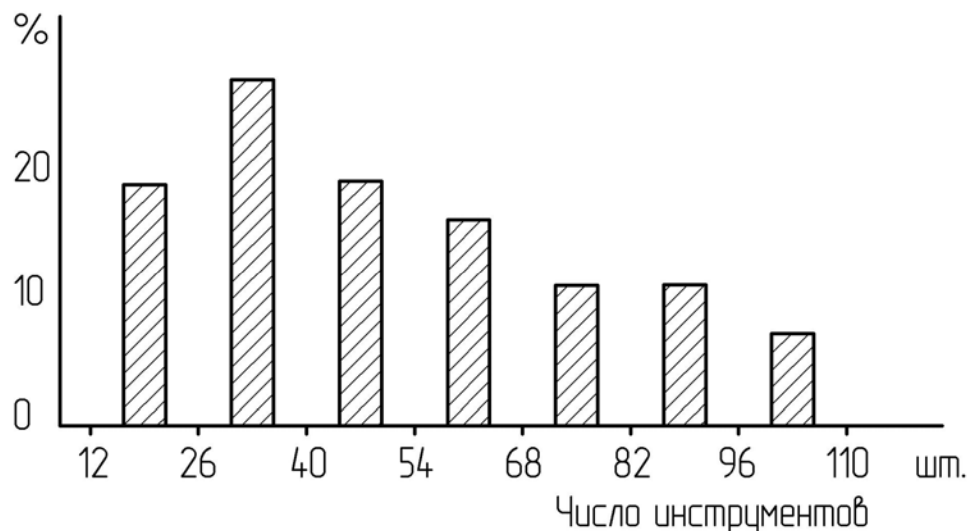


Рисунок 3.3 – Распределение емкости магазинов на многоцелевых станках

Рассмотрим возможные причины простоев станков при работе устройств АСИ, а также причины простоев при взаимодействии станков со средствами автоматической доставки и замены инструментов.

Простои оборудования будут рассмотрены, исходя из предположения, что на технологических переходах заложены форсированные режимы резания, вследствие чего оперативное время t_{on} переходов соизмеримо со временем смены РИ в рабочей зоне и с ресурсом стойкости РИ. Данное предположение, реально отображающее современные тенденции механообработки в ГПЯ, позволяет рельефно выделить возможные внутриоперационные простои станков.

По некоторым данным простои станка типа ОЦ из-за смены инструментов составляет 15 - 20 % вспомогательного времени. При использовании станков с ЧПУ со средней частотой смены инструмента 20 шт./ч при экономии на каждую смену всего 20 с годовая экономия фонда рабочего времени станка при двухсменной работе составляет около 10 % времени его работы [9].

3.2.1 Циклограммы автоматической смены инструментов

К устройствам автоматизированной системы инструментального обеспечения (АСИО) ГПЯ относятся устройства автоматической смены инструментов (АСИ) станков, внешние накопители РИ и средства доставки РИ к станкам [13].

К устройствам автоматической смены инструментов в зоне резания относятся инструментальные магазины (ИМ) станков и автооператоры (АО). В некоторых моделях станков смена инструментов может осуществляться и без помощи АО за счет соответствующих перемещений шпинделя.

На многоцелевом станке могут использоваться разные способы идентификации режущих инструментов: кодирование гнезд ИМ, самих РИ или применение системы «кодовый ключ», когда код РИ сообщается гнезду ИМ (способ расположения РИ в гнездах сменных ИМ в технологической последовательности, применяемый на гибких автоматических линиях со сменными ИМ, в настоящей работе не рассматривается).

Особенность кодирования гнезд ИМ - в том, что отработавший в шпинделе инструмент возвращается в «штатное» гнездо магазина. Следовательно, поиск и смена следующего по технологическому процессу инструмента может начаться лишь после возврата в ИМ отработавшего РИ. Вместе с тем система ЧПУ станка по заданному коду гнезда определяет кратчайший путь поворота магазина (при наличии в его приводе механизма реверса), что сокращает время поиска. В зависимости от конструкции автооператора смены РИ АО отработка циклов смены РИ может быть организована по-разному.

Если на станке используется однозахватный АО (рисунок 3.4 а), то отработка цикла АСИ выполняется в следующей последовательности (рисунок 3.4 б): выход шпиндельной бабки (ШБ) в позицию смены, удаление отработавшего инструмента в штатное гнездо, поворот магазина по кратчайшему пути для доставки гнезда с нужным РИ в позицию смены, перенос в шпиндель нужного инструмента. Далее ШБ возвращается в зону обработки и начинается выполнение очередного технологического перехода.

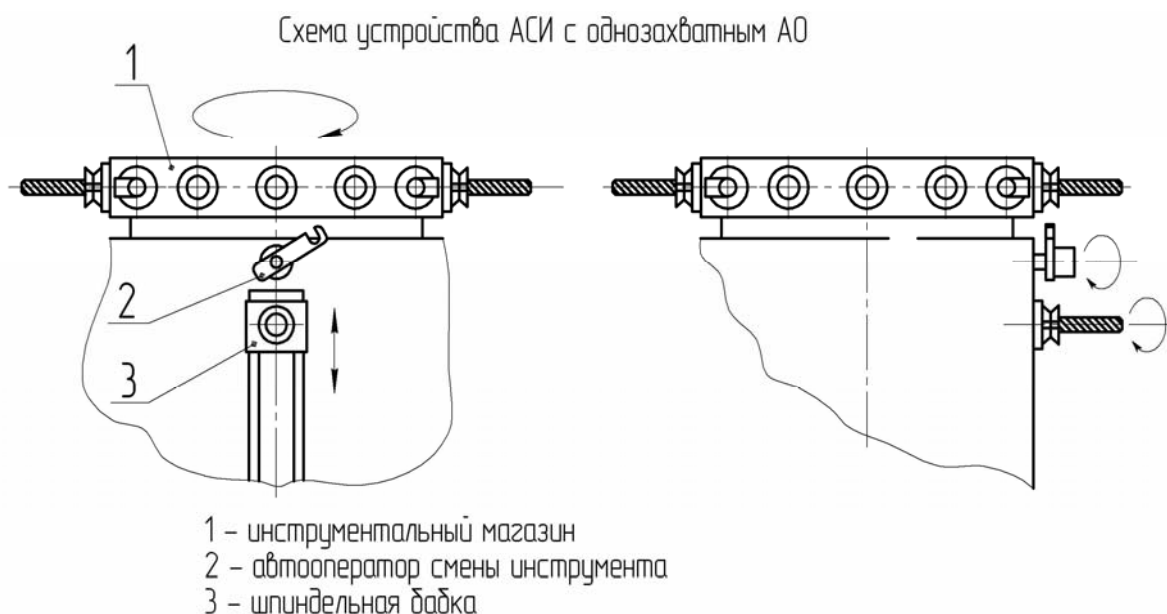


Рисунок 3.4 а – Схема устройств АСИ с однозахватным автооператором

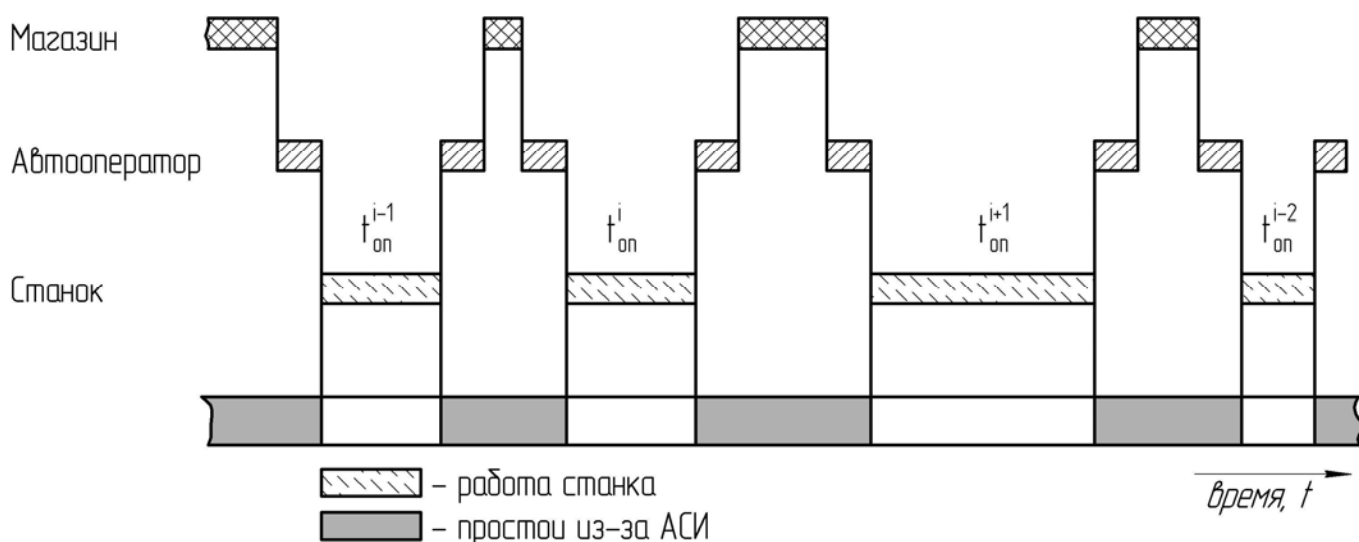


Рисунок 3.4 б – Циклограмма работы устройств АСИ с однозахватным автооператором

Если используется двухзахватный автооператор, имеющий позицию ожидания, то процесс поворота ИМ при подготовке очередного РИ может производиться в ходе выполнения текущего технологического перехода. Перед началом цикла смены в одном захвате АО может иметься очередной РИ (рисунок 3.5 а).

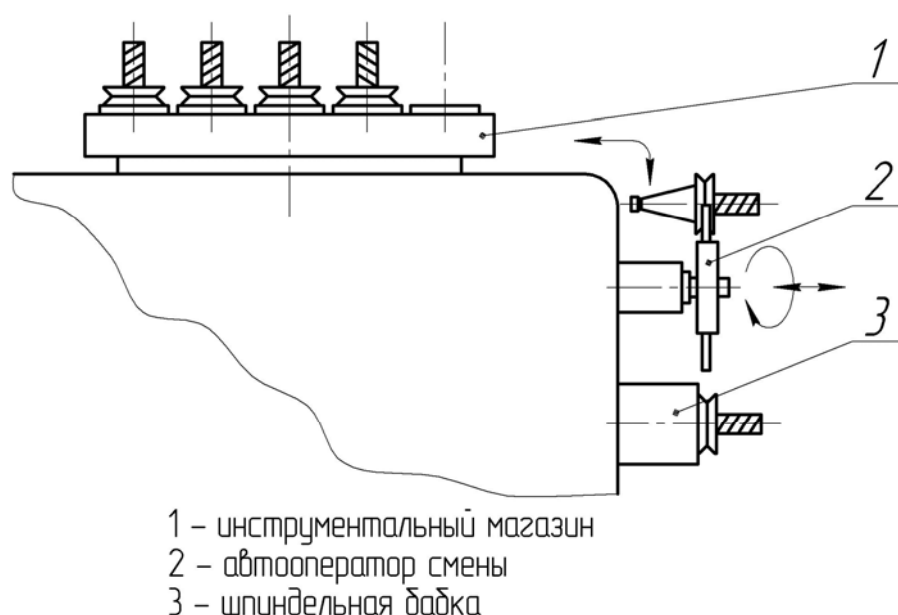


Рисунок 3.5 а - Схема работы устройств АСИ с двухзахватным АО с позицией ожидания

Цикл смены организуется следующим образом: выход ШБ в позицию смены, замена отработавшего РИ на подготовленный, возвращение ШБ в зону обработки. После смены РИ АО ждет завершения поворота ИМ для доставки в позиции смены штатного гнезда для отработавшего РИ, выгружает его и ожидает окончания поворота ИМ для захвата следующего по техпроцессу инструмента. Таким образом, при использовании двухзахватного АО имеющего позицию ожидания, при кодировании

гнезд ИМ в течение каждого технологического перехода обрабатываются два поворота ИМ: один - при доставке в позицию смены гнезда с требуемым РИ; другой - при доставке гнезда для отработавшего РИ. Такая организация АСИ (рисунок 3.5, б) позволяет сократить неперекрываемое время обработки АО стандартного цикла смены инструмента.

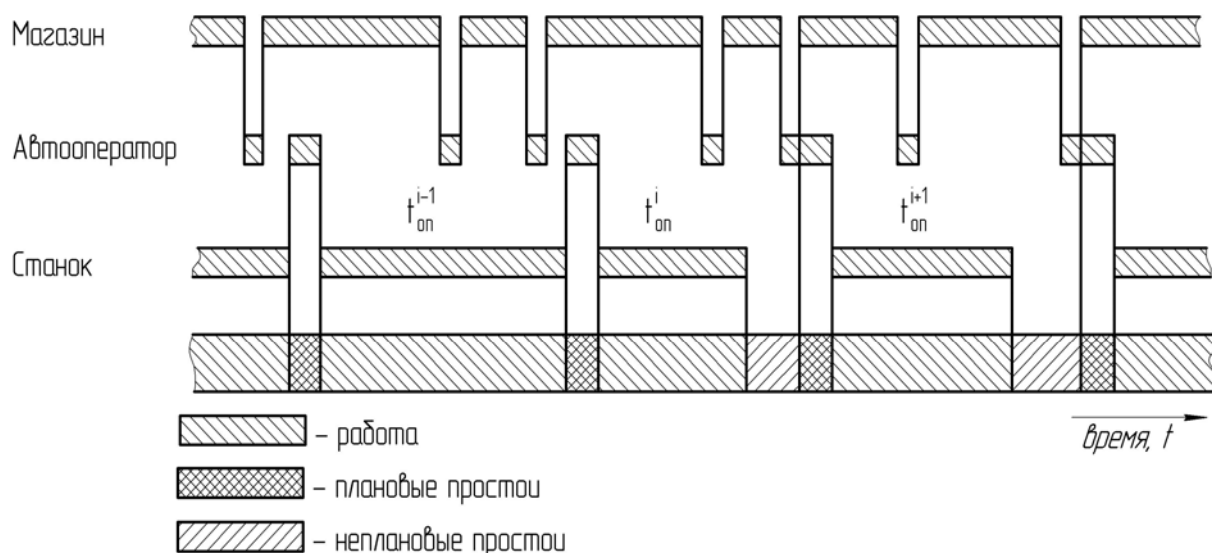


Рисунок 3.5 б - Фрагмент циклограммы работы устройств АСИ с двухзахватным АО с позицией ожидания

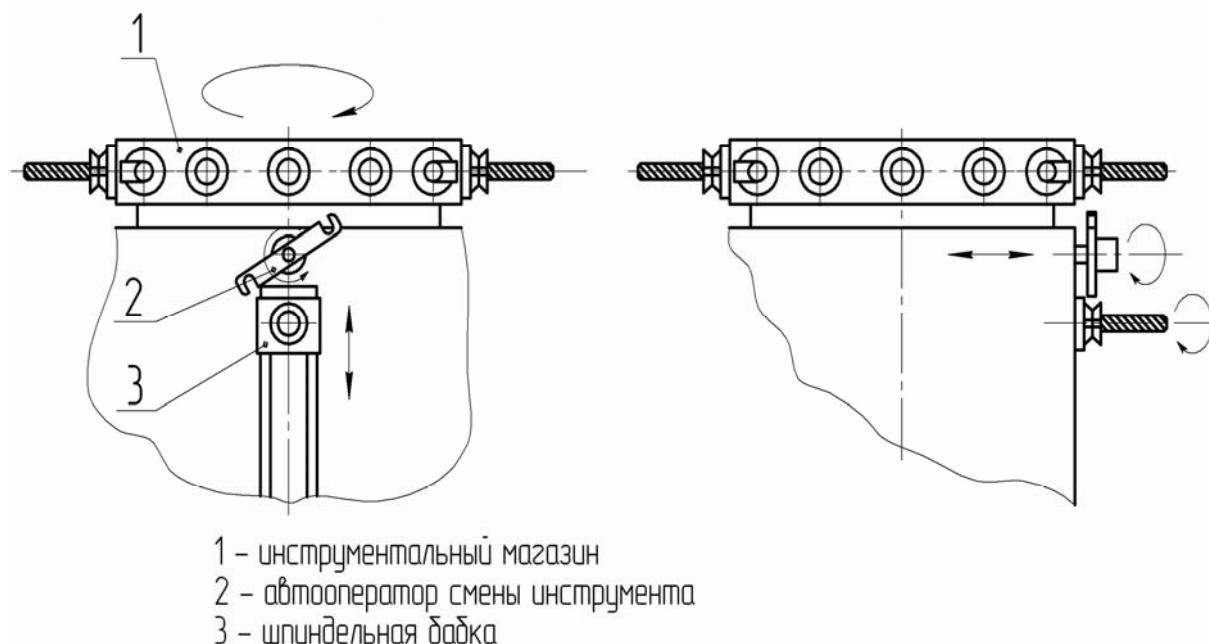
Кодирование РИ (гребенки, кодовые кольца, электронные идентификаторы и т.д.) позволяет совместить во времени процесс замены в шпинделе одного инструмента другим, так как отработавший инструмент может быть установлен на место очередного. В таких случаях используются АО с двумя захватами (рисунок 3.6, а); кроме того, поиск очередного инструмента в ИМ может быть совмещен во времени (перекрываться) с выполнением технологического перехода (рисунок 3.6, б).

Если для считывания кода РИ используется лишь позиция смены, то невозможно определить направление кратчайшего поворота ИМ, из-за чего поворот, производится лишь в одну сторону. Простои станка при смене инструментов включают неперекрываемую часть времени поворота ИМ и время обработки АО стандартного цикла смены инструментов.

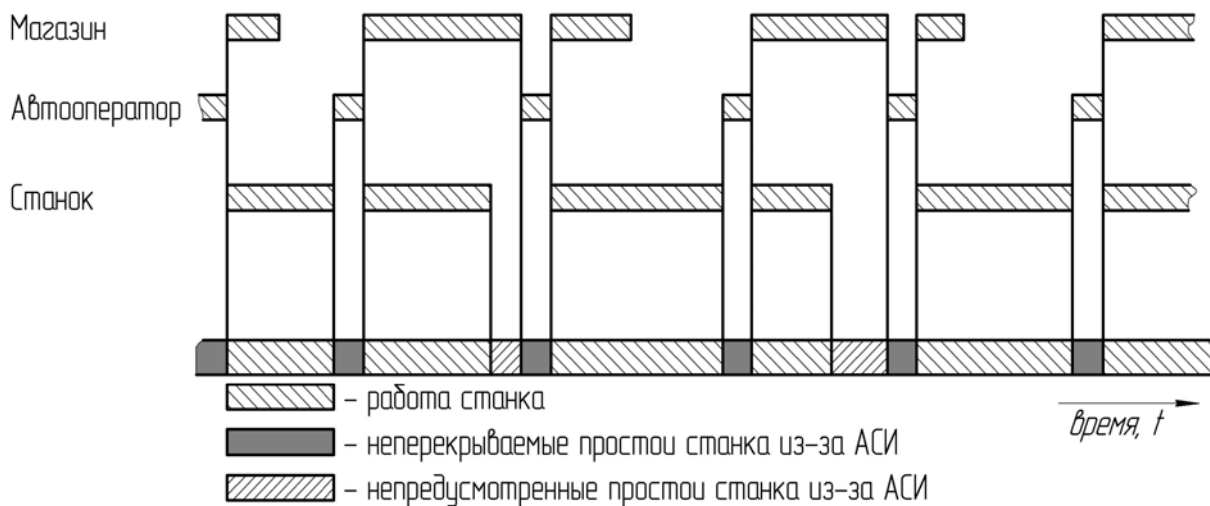
Кодирование по системе «кодовый ключ» сочетает достоинства кодирования гнезд и РИ: возможно совмещение удаления отработавшего и установки очередного РИ, а также поворот ИМ по кратчайшему пути (так как поиск РИ производится по соответствующему номеру гнезда). Циклограмма работы устройств АСИ аналогична рисунку 3.6 б, однако затраты времени на повороты ИМ вдвое меньше, чем при кодировании РИ.

Итак, автоматическая смена на станке режущих инструментов ведет к появлению простоев, связанных с поворотами инструментальных магазинов при поиске РИ, и с обработкой стандартных циклов смены инструментов автооператорами. Следует отметить, что стремление повысить производительность станка

за счет применения форсированных режимов резания ведет к сокращению не только длительности технологических переходов, но и стойкости РИ, и, соответственно, к возрастанию расхода РИ. В результате из-за увеличения доли неперекрываемых затрат времени на повороты МИ может быть получен результат, противоположный ожидаемому.



а



б

Рисунок 3.6 – Схема (а) и фрагмент циклограммы (б) работы устройств АСИ с совмещенным циклом обслуживания

3.2.2 Циклограммы автоматической замены инструментов

Автоматическая доставка РИ на станки может осуществляться несколькими способами: а) в сменных ИМ при переходе к обработке очередной партии заготовок (например, ГПЯ японской фирмы Yamazaki). Функция доставки МИ может выполнять, например транспортная тележка; б) в кассетах, содержащих набор недостающих в ИМ инструментов для изготовления очередной партии деталей (напр., система QTC (Quick Tool Change) германской фирмы Werner & Kolb, отечественная «Талка-500»); в) индивидуально в процессе работы станка (напр., отечественная ГПЯ АЛПЗ-2).

Замена кассет с инструментами (рисунок 3.7) и сменных ИМ выполняется периодически при переналадке станка на новую партию деталей и соответствующие простои станка практически не отражаются на внутриоперационных затратах времени.

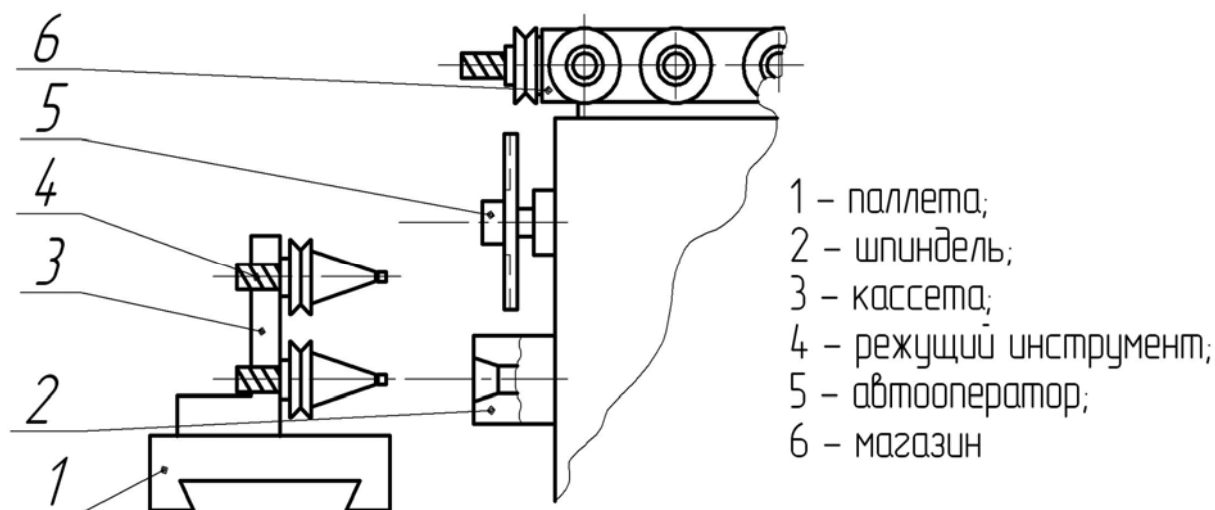


Рисунок 3.7 - Схема переагрузки магазина через шпиндель:

Особый интерес представляет индивидуальная замена РИ, т.к., во-первых, в этом случае сокращается общий расход РИ (за счет его многократного применения на разных станках), а во-вторых - отработка циклов смены РИ в рабочей зоне и замены в МИ осуществляется в ходе выполнения технологических переходов. Некоторые авторы отмечают перспективность данного способа при условии, что будет накоплен теоретический и практический опыт его реализации [5, 8].

Схемы устройств АСИ и АЗИ (автоматической замены инструментов на станке) представлены в соответствии с рисунками 3.8 и 3.9. Циклограммы работы станка с устройствами АСИ и АЗИ могут строиться с совмещенным и раздельным выполнением стандартных циклов автооператором смены.

Раздельное выполнение цикла смены РИ производится в следующей последовательности (рисунок 3.8 а, б).

циклов замены, может регламентироваться системой управления станка, например, в зависимости от продолжительности выполняемого перехода, что может свести к минимуму непредусмотренные простои станка из-за смены и замены РИ.

Совмещенная отработка стандартного цикла АО (рисунок 3.9 а, б) предполагает синхронную замену в шпинделе отработавшего РИ на требуемый. Недостатком способа является «связывание» ИМ с работой АО.

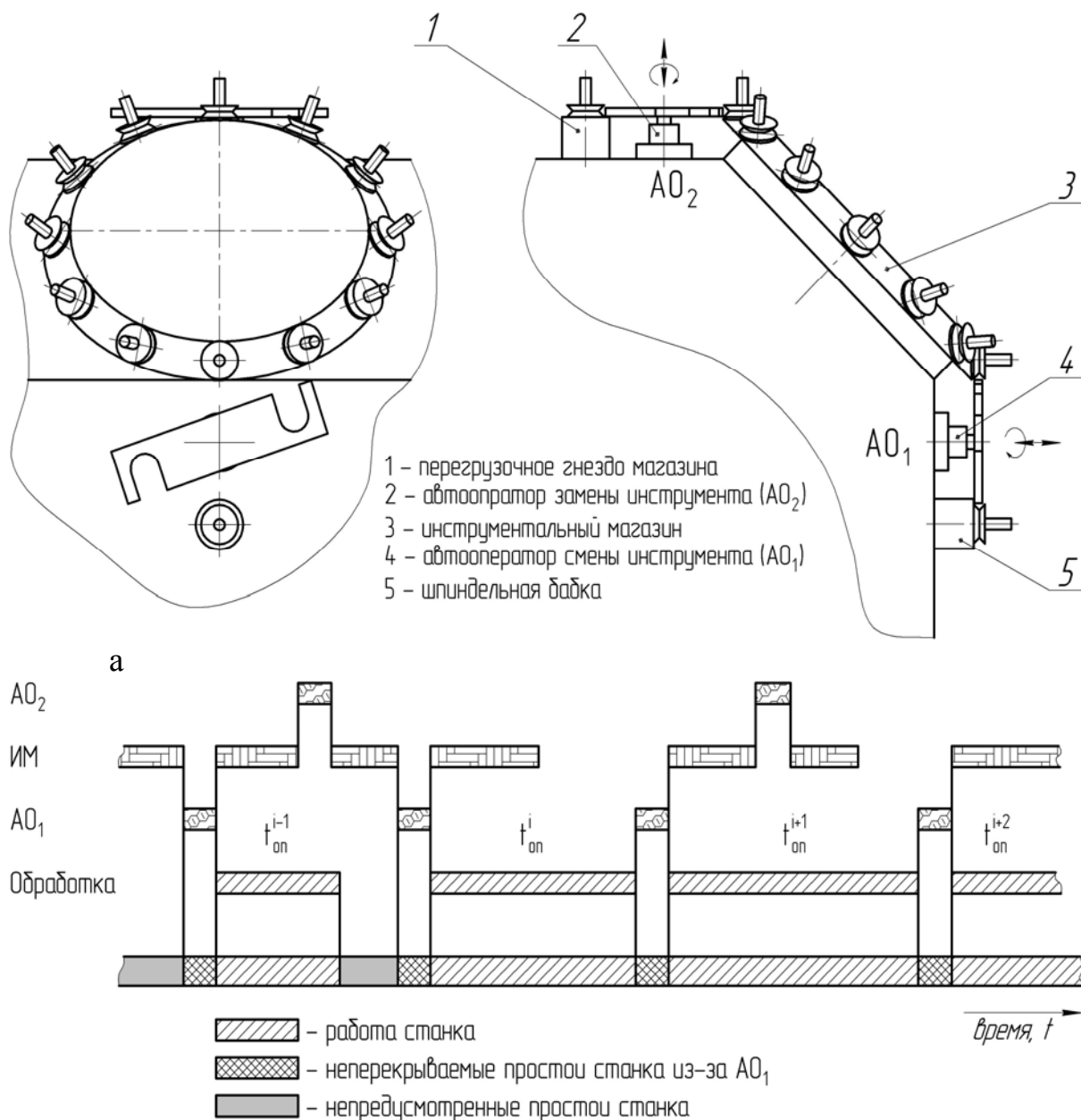


Рисунок 3.9 - Схема (а) и фрагмент циклограммы (б) работы устройств АСИ и АЗИ с совмещенными циклами работы

Циклограммы, представленные на рисунках 3.4 – 3.9, иллюстрируют следующее.

1) В ходе выполнения технологической операции на многоцелевом станке имеют место простои, связанные со сменой в рабочей зоне и заменой в инструментальном магазине режущих инструментов.

2) Указанные простои можно разделить на плановые и внеплановые. Плановые простои связаны с отработкой неперекрываемой части стандартных циклов смены РИ. Их величина постоянна, зависит от конструктивных параметров устройств АСИ и указывается в паспорте станка заводом-изготовителем. Плановые простои учитываются при нормировании техпроцесса. Внеплановые простои связаны с процессом выполнения конкретной технологической операции и определяются совокупным влиянием ряда факторов: времени выполнения текущего технологического перехода; местонахождения гнезда ИМ с требуемым РИ, относительно позиции смены (замены); характером поворота ИМ (по кратчайшему пути, или нет); очередностью отработки циклов смены и замены РИ.

3) Внеплановые простои проявляются как стохастические возмущения производственного процесса, увеличивающие суммарные простои оборудования участка и затрудняющие оперативное планирование ГПЯ.

4) Величина внеплановых простоев может быть выявлена лишь в ходе моделирования производственного процесса.

«Создалась ситуация, когда инженерные методы проектирования и расчета технико-экономических характеристик, принятые для традиционных производств, оказываются либо слишком грубы, либо вовсе неприемлемы для проектирования ГПЯ. В этих условиях особое значение приобрели методы автоматизированного синтеза технических решений, основанные на математическом моделировании» [43].

3.2.3 Циклограммы автоматической доставки режущих инструментов

В качестве средств доставки режущих инструментов к станкам могут применяться конвейеры, рольганги, транспортные тележки с установленными ПР (робокары), специализированные каретки-операторы. Паллеты с установленными на них комплектами РИ могут перемещаться из общего склада ГПЯ к станкам кранамиштабелерами, тележками единой транспортно-накопительной системы и т.д. Использование в качестве средств доставки РИ непрерывного транспорта (конвейеры, цепные транспортеры-накопители) обычно связывают с ранними этапами становления гибкой автоматизации. (английская ГПЯ «Система - 24», 1968 г., отечественная АП-1, 1972 г. [18] и др.), когда не было практического опыта создания транспортных средств других типов и использовались конструкции, «отработанные» на автоматических линиях массового производства [73]. Практически все современные ГПЯ, имеющие автоматизированные системы инструментообеспечения, используют дискретный транспорт как обладающий высокой гибкостью и способностью адаптации к быстроменяющимся условиям производства. В качестве примера в соответствии с рисунком 3.10 представлена возможная схема замены режущих инструментов в инструментальном магазине станка с помощью робокары, имеющей транспортный магазин для РИ и двухзахватный автооператор АО.

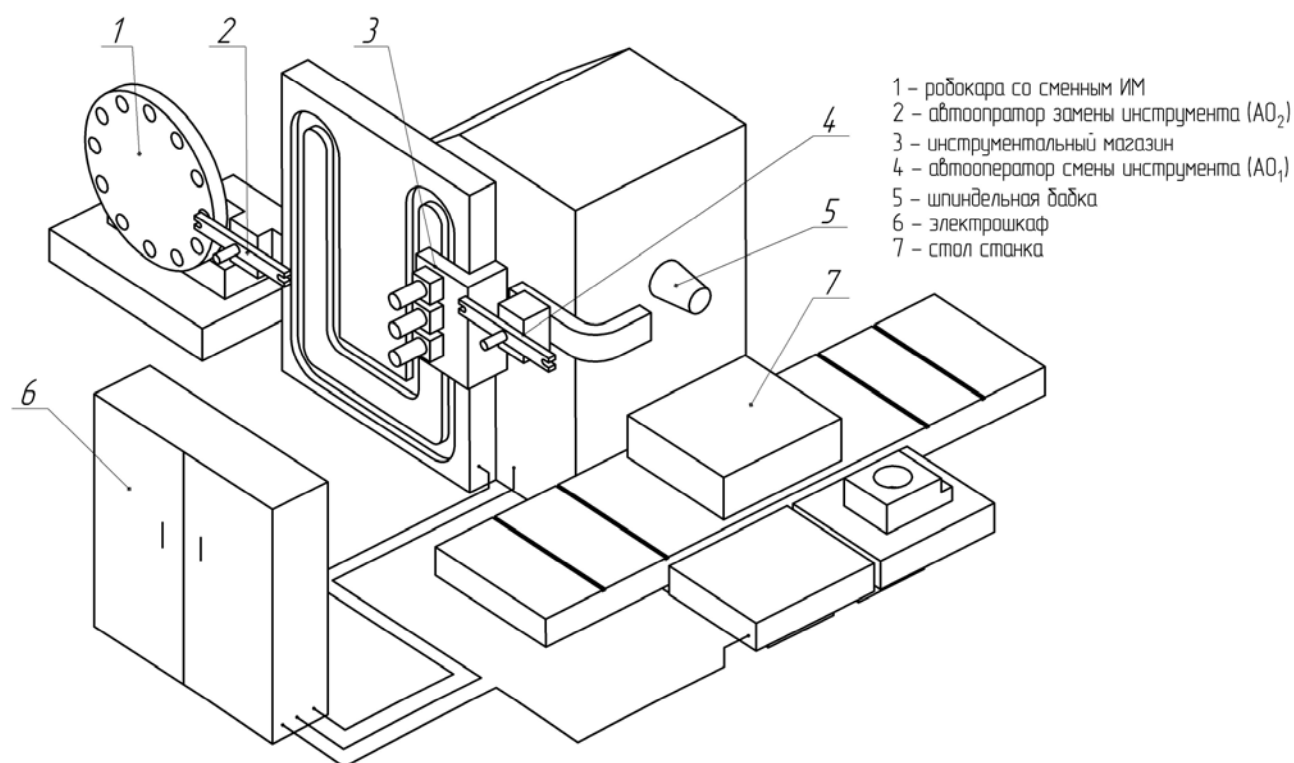
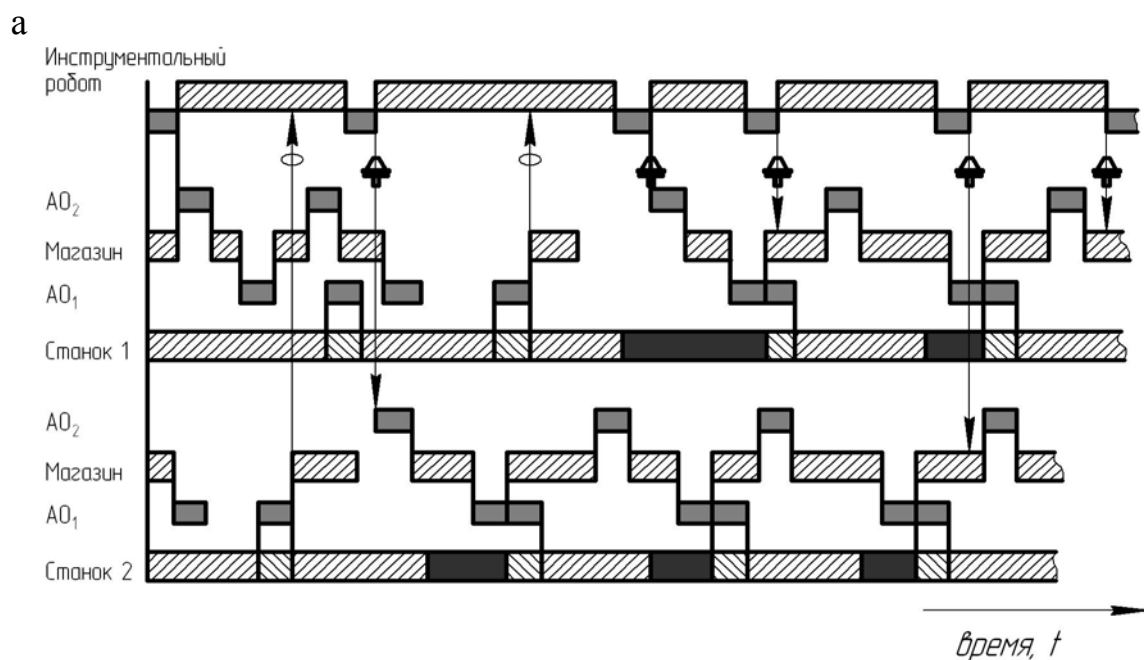
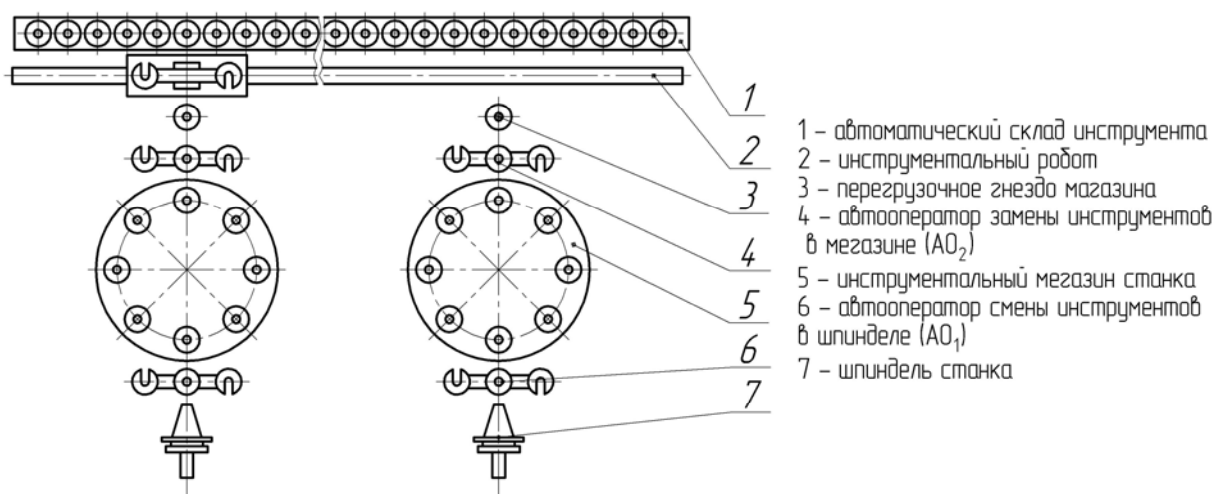


Рисунок 3.10 - Схема возможного оснащения многоцелевого станка средствами автоматической замены режущих инструментов

Анализ внутриоперационных простоев станков ГПЯ с поштучной заменой РИ в инструментальных магазинах рассмотрим на примере расчетной схемы, представленной в соответствии с рисунком 3.11 а. В момент установки очередной заготовки на стол станка система управления анализирует состав и состояние РИ, имеющихся в ИМ. На отсутствующие и выработавшие ресурс РИ выдается список запросов с указанием примерных моментов времени начала их работы (назначается по номинальной станкоемкости заготовки). На основе сопоставления поступивших списков запросов от станков формируется очередь заявок на их выполнение, и каретка-оператор приступает к ее обслуживанию.

Процедура корректировки очереди повторяется при каждой смене заготовки и при получении от какого-либо станка аварийной заявки (в случае, если заказанный в списке РИ к нужному моменту не доставлен на станок либо исчерпан ресурс имеющегося РИ). Из группы аварийных заявок выполняется самая ранняя, которая выполняется, если в перегрузочном гнезде ИМ уже находится удаляемый со станка РИ либо оно пустое. Повороты ИМ для отработки циклов замены РИ выполняются после установки в захват АО очередного инструмента.



б

Рисунок 3.11 – Схема (а) и фрагмент циклограммы (б) автоматической доставки режущих инструментов к станкам

Циклограмма работы подобной АСИО, представленная в соответствии с рисунком 3.11 б, иллюстрирует наличие непредусмотренных простоев станков из-за несвоевременной доставки режущих инструментов.

Организация работы АСИО может быть, и другой. Например, обслуживание очереди запросов на РИ может устанавливаться по приоритетам изготавливаемых деталей, станков, по $\min$ времени транспортной операции и т.д. и т.п.

Изображать все возможные варианты циклограмм работы АСИО достаточно трудоемко и, по-видимому, лишено особого смысла. В конечном итоге любая циклограмма может быть получена как результат работы программы моделирования производственного процесса с заданными параметрами и при определенных правилах взаимоотношений модулей и их агрегатов.

Целью же проводимого исследования является показать наличие внутри-операционных простоев станков, которые не учитываются при использовании существующих методов расчета ГПЯ.

3.3 Режимы резания и производительность ГПЯ

Цикл безлюдной работы оборудования ГПЯ механической обработки включает время резания и цикловые потери времени. Следовательно, при минимизации внутрицикловых простоев дальнейшее повышение эффективности ГПЯ может осуществляться по пути интенсификации режимов резания.

Применительно к станку с ЧПУ время рабочего цикла выполнения технологического перехода (или оперативное время, t_{on}) включает машинное основное время $t_{маш}$ и машинное вспомогательное время $t_{в}$

$$t_{on} = t_{маш} + t_{в} \quad (3.1)$$

Вспомогательное время связано с отработкой холостых и установочных перемещений исполнительными органами станка. Величина основного времени (в механообработке еще называется время резания) определяется соотношением длины и скорости относительного перемещения инструмента вдоль обрабатываемой поверхности. В соответствии с рисунком 3.12

$$t_{маш} = \frac{L_{рез}}{S_m}, \quad (3.2)$$

где $L_{рез}$ - длина резания с учетом врезания $l_{вр}$ и перебега $l_{пер}$ инструмента, мм;
 S_m - минутная подача инструмента, мм/мин.

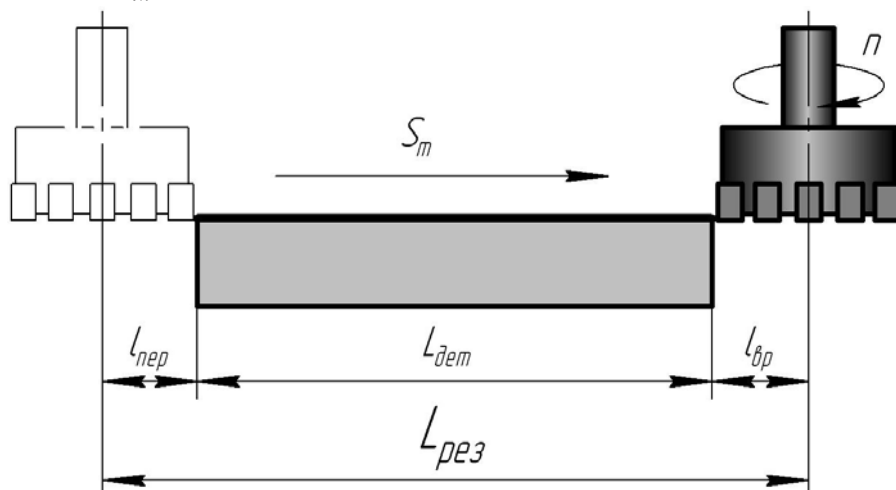


Рисунок 3.12 – Схема к расчету машинного основного времени перехода

Следовательно, одним из путей сокращения основного технологического времени $t_{маш}$ и, соответственно, повышения производительности обработки является увеличение минутной подачи.

Минутная подача напрямую зависит от принятых на переходе режимов резания. Согласно рекомендациям по нормированию работ на станках типа «об-

рабатывающий центр» назначение режимов резания ведётся в следующей последовательности.

Шаг 1. В зависимости от глубины резания или требований к качеству поверхности назначается подача на оборот S_o , мм/об, или для фрезерования - подача на зуб фрезы S_z , мм/зуб.

Шаг 2. Рассчитывается экономический период стойкости используемого режущего инструмента $T_э$

$$T_э = \left(\frac{1}{m} - 1\right) \times (t_{ци} + \frac{Q_1 + Q_2}{E}), \quad (3.3)$$

где m - показатель относительной стойкости (для твердосплавного инструмента $m = 0.2$);

$t_{ци}$ - время цикла смены инструмента в шпинделе;

Q_1, Q_2 - затраты, связанные с работой инструмента в течение одного периода стойкости, а так же с его разборкой, переточкой, сборкой и настройкой, соответственно;

E - стоимость одной станко-минуты работы станка.

Шаг 3. В зависимости от $T_э$ назначается скорость резания $V_{рез}$

$$V_{рез} = \frac{C_v}{T_э^m}, \quad (3.4)$$

где C_v - произведение поправочных коэффициентов на скорость резания.

Шаг 4. Рассчитывается частота вращения шпинделя n

$$n = 1000 \times \frac{V_{рез}}{\pi \times d}, \quad (3.5)$$

где d - диаметр обработки (инструмента), мм.

Шаг 5. При уточненном по паспорту станка значении n рассчитывается величина минутной подачи S_m

$$S_m = \begin{cases} S_z \times z \times n & \text{— для переходов фрезерования} \\ S_o \times n & \text{— для прочих технологических переходов} \end{cases} \quad (3.6)$$

где z - число зубьев фрезы.

«Слабым» звеном в описанной методике является назначение периода стойкости РИ. Одним из путей повышения производительности обработки является сокращение периода экономической стойкости. Некоторые исследователи считают рациональным его уменьшение от 30 - 40 до 20 - 25 минут и меньше. Исходят при

этом из того, что затраты Q_1, Q_2 пренебрежимо малы по сравнению со стоимостью станко-минуты E работы многоцелевого станка и тем более ГПЯ: $E \gg (Q_1 + Q_2)$.

Следовательно, в условиях ГПЯ формула (3.3) трансформируется в формулу расчета стойкости по критерию максимальной производительности

$$T_9 = \left(\frac{1}{m} - 1\right) \times t_{cu}. \quad (3.7)$$

Однако, подставив в формулу $m = 0,2$ и $t_{cu} = 16$ секунд (время цикла смены инструмента на станке ИР500ПМФ4) получим $T_9 = 64$ с!

Отсюда можно сделать единственный вывод - о том, что формула, полученная для станков с ручным управлением, некорректна для станков с автоматической сменой инструментов.

Если допустить, что современные станки имеют достаточный запас мощности привода, жесткости и быстроходности, а режущие инструменты – запас прочности к хрупкому разрушению, то возникает вопрос: как же на практике задать стойкости РИ?

Для поиска ответа изменим процедуру нормирования следующим образом.

Шаг 1. В зависимости от глубины резания или требований к качеству поверхности по нормативам назначается S_z или S_o .

Шаг 2. Задается минимальное значение минутной подачи S_m .

Шаг 3. Рассчитывается частота вращения шпинделя и корректируется по станку

$$n = \frac{S_m}{S_o} \text{ или } n = \frac{S_m}{S_z \times z}. \quad (3.8)$$

Шаг 4. Рассчитывается соответствующее значение скорости резания

$$V_{рез} = \frac{\pi \times d \times n}{1000}. \quad (3.9)$$

Шаг 5. Рассчитывается стойкость инструмента

$$T = \left(\frac{C_v}{V_{рез}}\right)^{\frac{1}{m}}. \quad (3.10)$$

Шаг 6. Рассчитывается основное время перехода

$$t_{маш} = \frac{L_{рез}}{S_m}. \quad (3.11)$$

Шаг 7. Задается приращение $S_m = S_m + \Delta$, и шаги 3 - 6 повторяются.

В качестве примера выполнен расчет режимов резания для чистового фрезерования торцевой фрезой: $d_{фрезы} = 100$ мм; $z = 10$ зубьев; $S_z = 0,10$ мм/зуб; материал режущей части - Т15К6; материал заготовки – сталь 45. Минутная подача S_m варьировалась от 100 до 700 мм/мин. Результаты расчетов представлены в таблице 3.2 и на рисунке 3.13.

В соответствии с рисунком 3.13 следует, что при увеличении минутной подачи от 100 до 500 мм/мин основное время перехода $t_{маш}$ сокращается с 30 до 6 минут, т.е. производительность обработки возрастает в 5 раз. Однако, стойкость фрезы уменьшается более чем в 20 раз, соответственно увеличивая потребность в инструментах-дублерах.

Увеличивающийся расход РИ требует повышенной вместимости инструментальных магазинов станков либо увеличения вместимости автоматического инструментального склада.

Таблица 3.2 - Сводная ведомость режимов резания технологического перехода

S_m , мм/мин	n , об/мин	$V_{рез}$, м /мин	m , мин	$t_{маш}$, мин
100	100	31	442	30,0
150	150	47	148	20,0
200	200	63	68	15,0
250	250	79	37	12,0
300	300	94	23	10,0
350	350	110	15	8,6
400	400	126	10	7,5
500	500	157	6	6,0
600	600	189	3	5,0
700	700	220	2	4,3

В результате возрастает вспомогательное время, связанной с частой сменой РИ в рабочей зоне, простоями из-за подготовки и доставки инструментов к станку. В результате может быть получен результат, противоположный ожидаемому: простои станков в ожидании РИ «съедают» всю экономию, полученную от сокращения машинного времени.

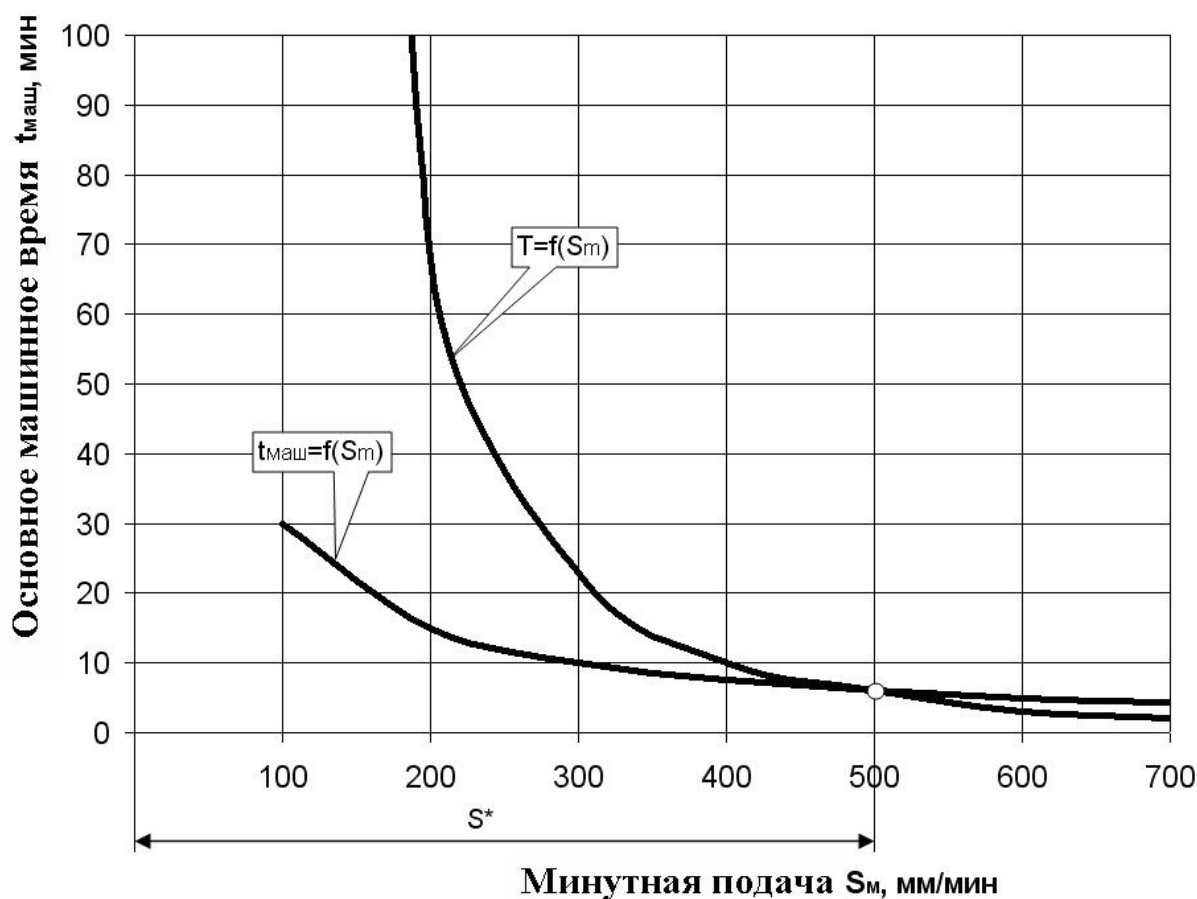


Рисунок 3.13 – Влияние минутной подачи S_m на величину основного машинного времени $t_{\text{маш}}$ и стойкость режущего инструмента T (S^* - диапазон регулирования подачи при выборе производительности перехода)

Оптимальный вариант назначения режимов резания может быть выявлен, если нормирование производить по критерию себестоимости изготовления одной детали C_∂

$$C_\partial = \frac{(C_{c-m} \times T_{cз} + C_{PI} \times N_{PI})}{N_\partial}, \quad (3.12)$$

где C_{c-m} - стоимость станко-минуты работы ГПЯ, руб./мин;
 $T_{cз}$ - время выполнения сменного задания, мин;
 C_{PI} - средняя стоимость одного инструмента, руб.;
 N_{PI} - требуемое число инструментов для выполнения СЗ;
 N_∂ - число деталей, изготовленных за время $T_{cз}$.

Результаты расчета себестоимости обработки при $C_{c-m}=50$ руб./мин и различных соотношениях C_{PI}/C_{c-m} представлены в соответствии с рисунком 3.14.

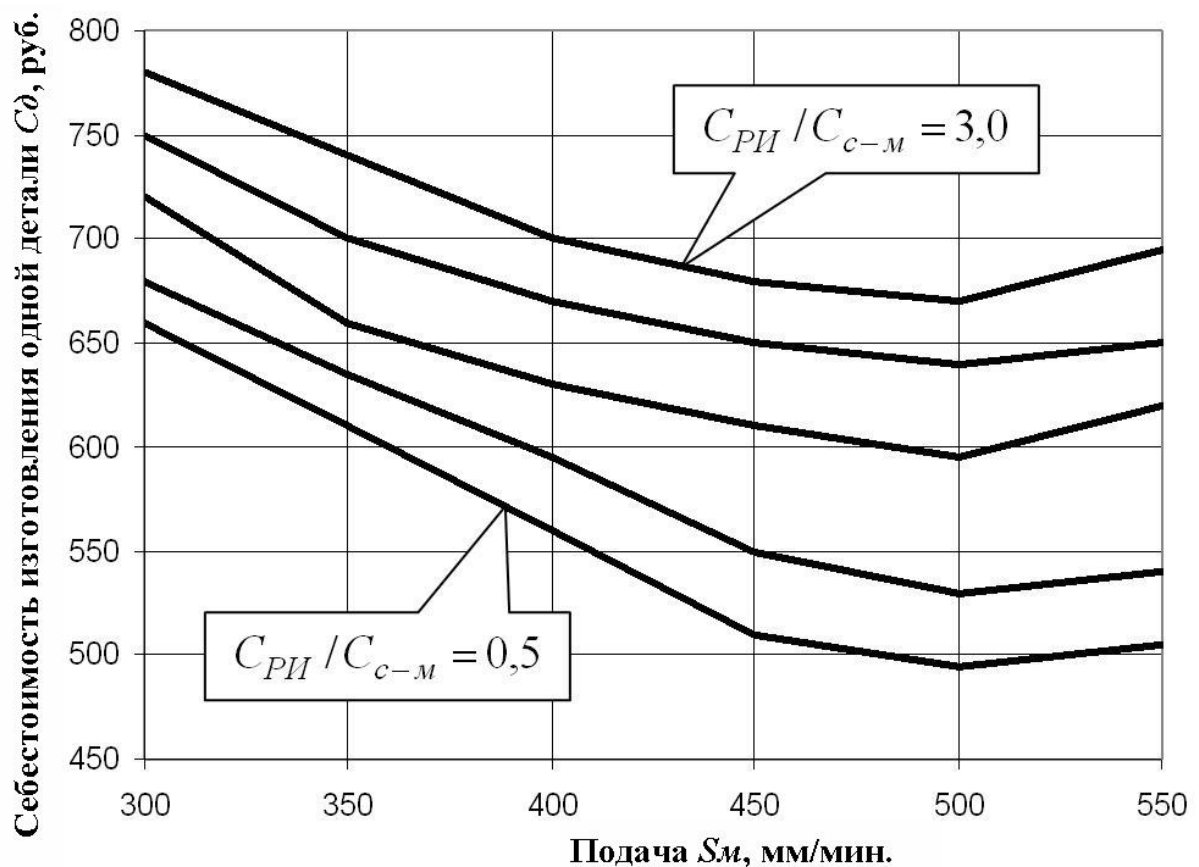


Рисунок 3.14 - Зависимость себестоимости изготовления одной детали C_d от подачи S_m

Результаты показывают, что 80 %-ное увеличение S_m позволяет на 12-26 % сократить себестоимость обработки; причем экономия тем больше, чем меньше значение C_{PI} / C_{c-m} . С увеличением C_{PI} / C_{c-m} , например, из-за использования на станке более дорогих инструментов, на графике более явно выделяется экстремум, соответствующий минимальной себестоимости обработки, т.е. оптимальному варианту нормирования режимов резания.

В рассмотренном примере минимальная себестоимость обработки обнаружилась при средней стойкости инструментов, соответствующей 10 % от нормативной.

Выводы

Проведенный анализ работы оборудования гибких производственных систем позволяет сделать следующие выводы.

В процессе автономной работы ГПЯ имеют место простои технологического оборудования, снижающие эффективность работы системы в целом. Особенностью данных простоев является их проявление при гарантированной надежности всех модулей системы и их агрегатов.

Указанные простои делятся на плановые и неплановые. Плановые простои связаны с отработкой стандартных циклов обслуживания агрегатами смены заготовок и режущих инструментов. Время отработки стандартного цикла

постоянно, является паспортной характеристикой агрегата и используется при нормировании и диспетчировании производственного процесса.

Неплановые (непредусмотренные) простои станков возникают лишь в процессе функционирования производственной системы и проявляются как стохастические возмущения производственного процесса, увеличивающие суммарные простои станков и затрудняющие оперативное планирование ГПЯ.

Неплановые простои станков проявляются на уровнях технологической операции (межоперационные) и технологического перехода (внутриоперационные). Причинами неплановых межоперационных простоев является взаимодействие технологических (станков, координатно-измерительных машин) и транспортных модулей, осуществляемое на фоне рассеяния времени технологических и транспортных операций.

Рассеяние длительности технологических операций возникает из-за широкой номенклатуры выполняемых технологических процессов, вследствие чего в одном цикле автономной работы ГПЯ могут изготавливаться различные детали. Причиной рассеяния длительности операций являются также непредусмотренные внутриоперационные простои станка, вследствие чего фактическая станкоемкость одинаковых заготовок может быть различной.

Рассеяние времени выполнения транспортных операций возникает из-за различных расстояний, проходимых транспортным устройством в процессе выполнения заявок станков.

Причинами неплановых внутриоперационных простоев является несогласованность действий в процессе выполнения технологических переходов исполнительных органов станка с агрегатами доставки, замены, подготовки и смены РИ. По причине рассеяния оперативного времени технологических переходов и времени на подготовку и смену РИ возникают ситуации, когда к моменту завершения очередного перехода РИ для следующего еще не подготовлен, в результате чего станок простаивает в ожидании инструмента. Величина неплановых внутриоперационных простоев станка зависит от целого ряда факторов: от параметров, устройств АСИ и АЗИ, алгоритмов их работы, местонахождения запрашиваемых РИ и их стойкости от длительности технологических переходов и т.д.

Предвидеть неплановые простои станков, а также предпринять, необходимые меры по устранению либо уменьшению их длительности можно посредством моделирования производственного процесса с детализацией алгоритма моделирования до уровня технологического (инструментального) перехода.

Таким образом, основными путями повышения загрузки и производительности ГПЯ являются сокращение внутрицикловых простоев оборудования и форсирование режимов резания. Вместе с тем, как это показано в разделе 3.3, существуют режимы резания, при дальнейшем ужесточении которых производительность обработки практически стабилизируется, а себестоимость возрастает из-за увеличивающегося расхода инструментов. Появляется задача синтеза ГПЯ, загрузка оборудования и производительность которой соответствовали бы наименьшим издержкам производства на изготовление продукции.

Создалась ситуация, когда инженерные методы проектирования и расчета технико-экономических характеристик, принятые для традиционных производств, оказываются либо слишком грубы, либо вовсе неприемлемы для проектирования ГПЯ. В этих условиях особое значение приобрели методы автоматизированного синтеза технических решений, основанные на компьютерном моделировании производственного процесса.

4 Метод циклограмм в компьютерном моделировании ГПЯ

4.1 Необходимость комплексного подхода к моделированию

Качество производственной системы предопределяется совокупным влиянием нескольких групп факторов, проявляющихся на различных этапах ее жизненного цикла.

В соответствии с рисунком 4.1 схематично представлено влияние на качество работы ГПЯ решений, обосновывающих выбор деталей и технологических процессов для проектируемой системы (Т); структуры ГПЯ и компоновкой входящего в ее состав оборудования (S); технических параметров основного и вспомогательного оборудования (Р); алгоритмов взаимодействия оборудования (А); принятых в системе методов планирования и диспетчирования производства (D).

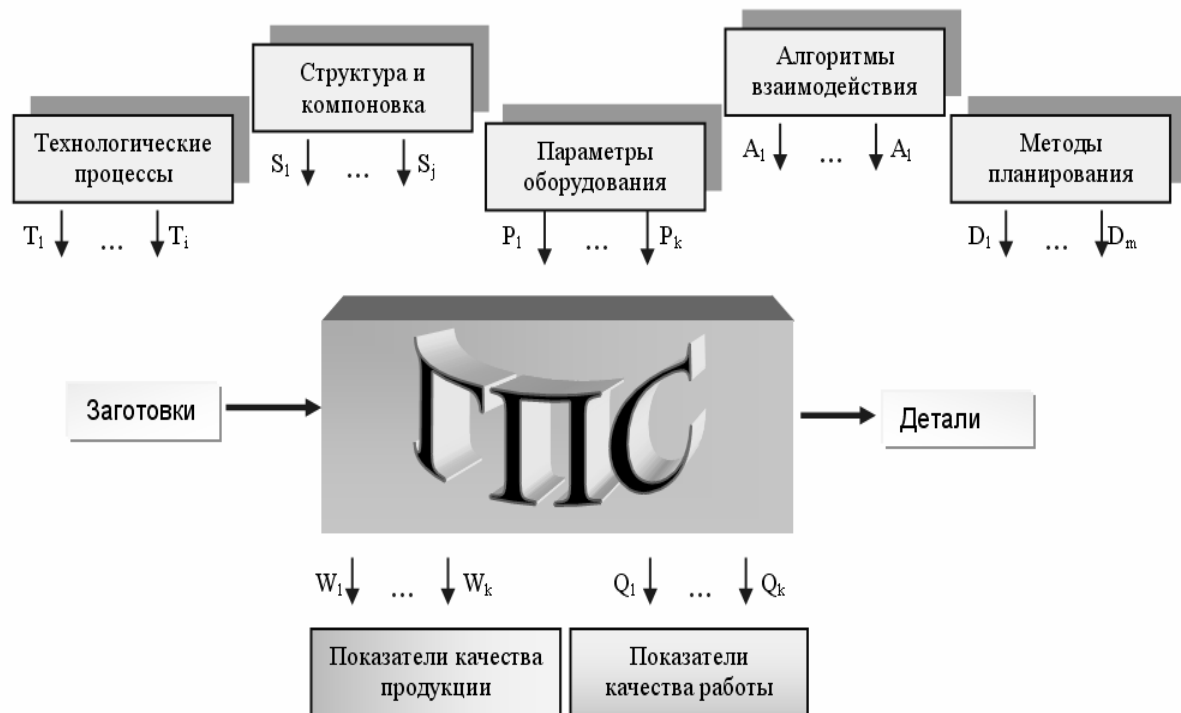


Рисунок 4.1 – Схема влияния различных групп решений на качество работы ГПЯ

Схема иллюстрирует, что качество работы производственной системы, выраженное рядом показателей качества, есть функция целого ряда аргументов

$$Q = Q\{T, S, P, A, D\} \quad (4.1)$$

Каждый из аргументов, в свою очередь, есть функция аргументов следующего уровня детализации

$$T = T\{T_1, T_2, \dots, T_i\},$$
$$S = S\{S_1, S_2, \dots, S_i\},$$

$$P = P\{P_1, P_2, \dots, P_i\},$$

$$A = A\{A_1, A_2, \dots, A_i\},$$

$$D = D\{D_1, D_2, \dots, D_i\}.$$

Как следует из представленных выражений, существует множество комбинаций (кортежей) значений аргументов T, S, P, A, D , при которых достигается требуемое качество Q работы ГПЯ.

Изложенное иллюстрирует мысль о том, что высокое качество производственной системы достигается лишь единым комплексом решений, принимаемых при выборе деталей для проектируемой ГПЯ, разработке технологических процессов их изготовления, проектировании самой ГПЯ, диспетчировании и управлении ходом производственного процесса.

Поиск оптимальных комплексов решений возможен, по-видимому, только с помощью компьютера путем моделирования производственного процесса при различных значениях аргументов из интервалов их допустимых значений и оценки получаемых вариантов решений по выбранным критериям эффективности (загрузка системы, простои оборудования, производительность и т.д.).

К настоящему времени в мире насчитывается число компьютерных моделей ГПЯ, соизмеримое с числом созданных ГПЯ [39]. Также существуют универсальные средства моделирования - GPSS World (рисунок 4.2, [83]), AnyLogic [80], Arena Enterprise Suite [81], язык моделирования UML [79].

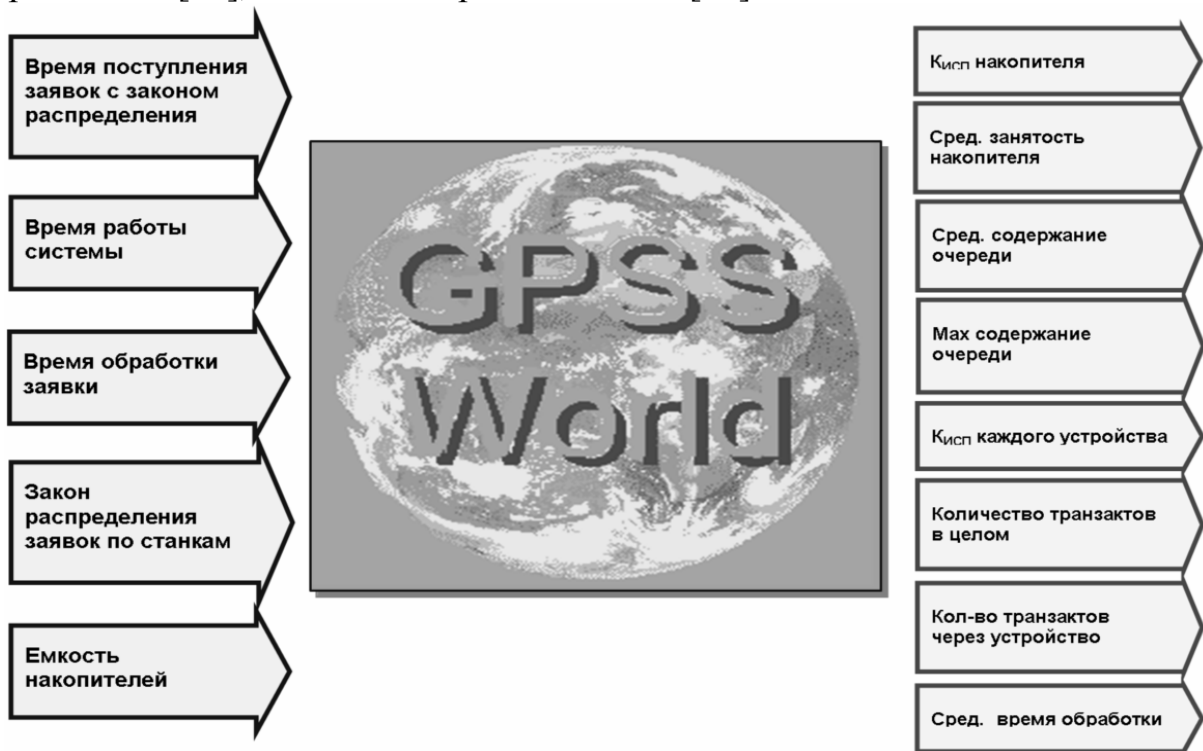


Рисунок 4.2 – Схема входных и выходных данных о ГПЯ в среде моделирования GPSS World

Однако «при всем уважении к качеству, универсальности и высокому профессиональному уровню подобных систем, нельзя не отметить следующее: использование универсальных систем и средств моделирования ... дает неполное представление об изучаемом объекте, подменяет изучение ГПЯ изучением не вполне неадекватного аппарата имитационного моделирования» [64].

Подход, реализуемый в универсальных средствах моделирования, описывает ГПЯ как сложную стохастическую систему, процессы функционирования которой характеризуются случайными величинами – времени поступления и интенсивности потоков заявок на обслуживание, времени обработки заготовок, использования накопителей, простоев оборудования и т.д. Используемые входные данные весьма условно и поверхностно интерпретируют функционирование ГПЯ, не давая ответы на многие вопросы технических специалистов [55]. В работе [65] сформулированы общие недостатки универсальных средств моделирования:

а) оценочный характер моделирования, невозможность автоматизированного синтеза оптимальных проектных параметров оборудования системы;

б) абстрагирование от некоторых особенностей производственного процесса, ограничивающее число исследуемых параметров;

в) необходимость специальной подготовки пользователей для формирования моделей, интерпретации данных и результатов, т.е. доступность ограниченному кругу специалистов;

г) ориентация на конкретные этапы жизнедеятельности ГПЯ либо на решение конкретных задач, что не обеспечивает комплексный учет совокупности проектных и эксплуатационных решений;

д) ограниченность моделирования уровнем технологической операции из-за высокой размерности моделирования на уровне перехода. Не рассматривается уровень технологического перехода, на котором организуется работа многоцелевых станков – основных «кирпичиков» ГПЯ;

е) высокая стоимость универсальных систем моделирования не гарантирующая наличие требуемых возможностей (например, стоимость системы моделирования Arena составляет \$20400,0 [81]).

«Ошибочна сама концепция представления ГПЯ как стохастических систем - автоматический процесс функционирования ГПЯ имеет не стохастический, а детерминированный, упорядоченный характер» [64].

Цикл безлюдной работы системы представляет собой цепочки взаимосвязанных циклов работы отдельных устройств, агрегатов, производственных модулей и системы в целом. Взаимосвязи циклов работы устройств предопределяются алгоритмами и правилами их взаимодействия, а также логикой эволюции процесса функционирования системы.

Продолжительность циклов зависит от технических параметров устройств (например, скорость робокара) и проходимых расстояний (например, путь резания, угол поворота, высота подъема), которые предопределяются текущим состоянием системы.

Традиционный язык технических специалистов и средство визуализации взаимодействия оборудования – циклограммы, или временные диаграммы. Работа

ГПЯ с любой требуемой точностью может быть расписана в виде циклограмм. Возможные погрешности срабатывания устройств можно учитывать вероятностными методами подобно тому, как учитывается рассеивание размеров деталей относительно номинальных значений [61].

Циклограммы позволяют вскрыть внутренние потери эффективности ГПЯ, связанные с взаимодействием ее элементов, и внешние потери, обусловленные варьированием номенклатуры выпускаемой продукции. Сокращение или полное устранение внутренних потерь позволят оптимизировать проектные параметры ГПЯ, а сокращение внешних потерь - минимизировать эксплуатационные издержки.

По сути, циклограммы работы оборудования - это и есть модели функционирования технических систем.

Автоматизированное построение циклограмм позволяет использовать на входе функционального блока рисунка 4.3 технические параметры технологического и сервисного оборудования; информацию о технологических процессах из управляющих программ для систем ЧПУ; конкретный состав сменного задания.

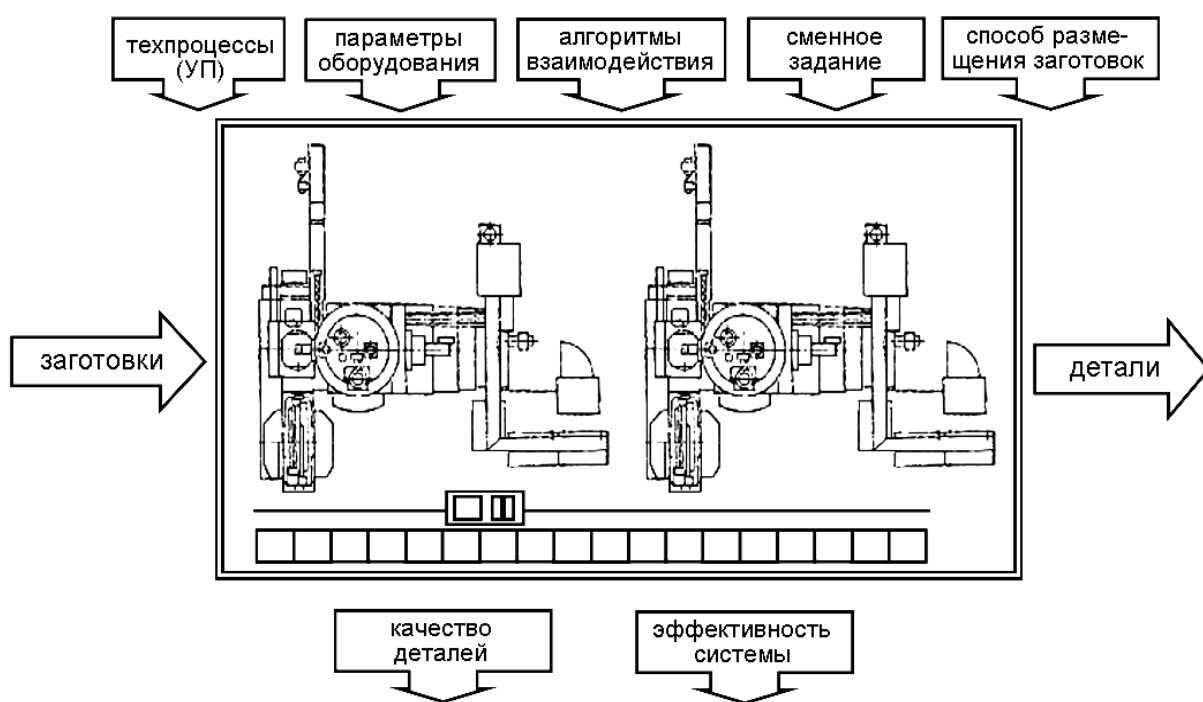


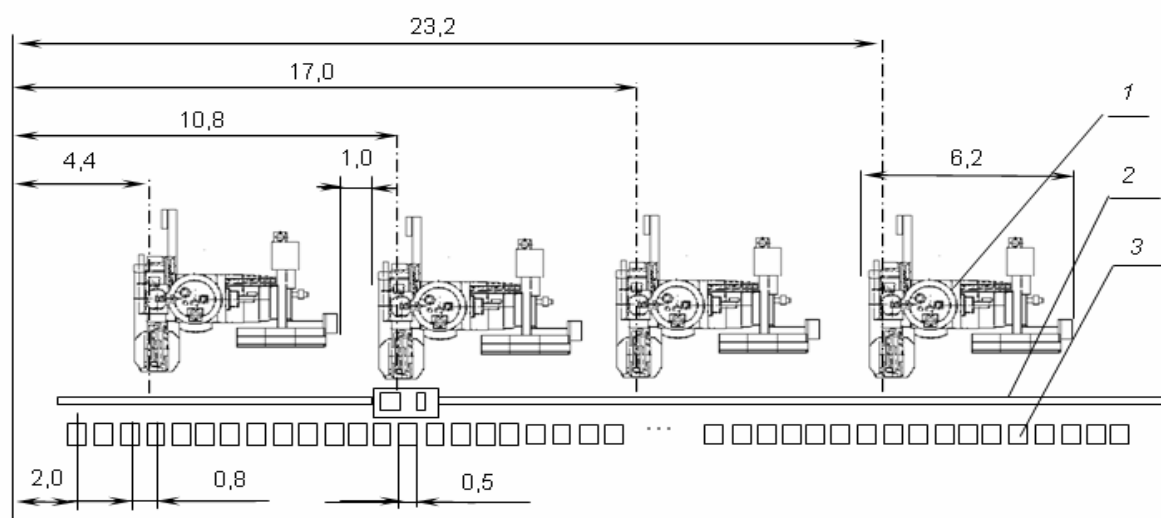
Рисунок 4.3 – Схематичное представление входных и выходных данных производственной системы в методе циклограмм

Выходами функционального блока служат а) загрузка основного технологического оборудования; б) производительность ГПЯ; в) срок окупаемости ГПЯ.

Обеспечить преобразование указанных входов в выходы позволяет метод автоматизированного построения циклограмм работы оборудования гибкой производственной ячейки [56 – 65].

4.2 Метод циклограмм

Компьютерное моделирование можно рассматривать как автоматизированное построение циклограмм работы оборудования. Степень детализации циклограмм зависит от решаемых задач. При анализе работы транспортно-накопительной системы достаточно рассмотреть работу ГПЯ как взаимодействие технологических и транспортных модулей (уровень технологической операции). При анализе работы автоматизированной системы инструментообеспечения необходимо рассматривать работу ГПЯ на уровне технологического перехода. Рассмотрим возможности метода автоматизированного построения циклограмм на примере модели ГПЯ, состоящей из 4 станков типа ОЦ, линейного накопителя паллет и двухместной робокары (рисунок 4.4).



1 - станочный модуль; 2 - транспортная тележка; 3 - накопитель паллет на 42 позиции

1) Параметры оборудования

Число станков на участке, шт 4
 Координаты станков, м 4,4; 10,8; 17,0; 23,2
 Время смены стола-спутника на станке, с 45
 Число позиций на складе заготовок, шт 42
 Координата склада, м 2,0
 Шаг ячеек склада, м 0,8
 Маршевая скорость робокары, м/с 1,0
 Цикл смены паллеты на робокаре, с 30
 Правило обслуживания станков по очереди поступления заявок

2) Производственная программа ГПЯ

корпусные детали 10 наименований, время обработки представлено в таблице:

Порядковый номер детали	1	2	3	4	5	6	7	8	9	10
Станкоёмкость, мин	1,4	2,8	14,6	1,0	0,5	12,0	2,3	4,7	2,7	4,1

3) Сменное задание на предстоящий цикл моделирования

40 деталей согласно очереди (порядковые номера в производственной программе):
 1,1,4,4,4,4,9,9,2,2,2,3,10,10,10,10,10,10,8,8,3,3,5,5,5,7,7,6,6, 1,1,3,3,2,2,10,10,8,8

4) Размещение заготовок в накопителе

(порядковые номера расположены слева направо, начиная с первой позиции накопителя):
 1,1,1,1,2,2,3,3,3,3,0,0,9,9,7,7,6,6,10,10,10,10,10,10,10,10,10,4,4,4,4,8,8,8,8,5,5,5,2,2,2

Рисунок 4.4 - Расчетная схема и исходные данные к построению циклограммы работы ГПЯ

Моделирование ведется при следующих исходных условиях:

- а) известен состав модулей ГПЯ (структура) и их параметры;
- б) сменное задание указано в виде последовательности запуска в обработку заготовок различных наименований;
- в) известны технологические процессы обработки заготовок, каждый из которых задан временем их выполнения;
- г) перед началом цикла известно размещение заготовок в ячейках склада и в рабочих зонах станков.

Работа ГПЯ рассматривается на уровне технологической операции на протяжении автоматического цикла выполнения сменного задания (рисунок 4.5).

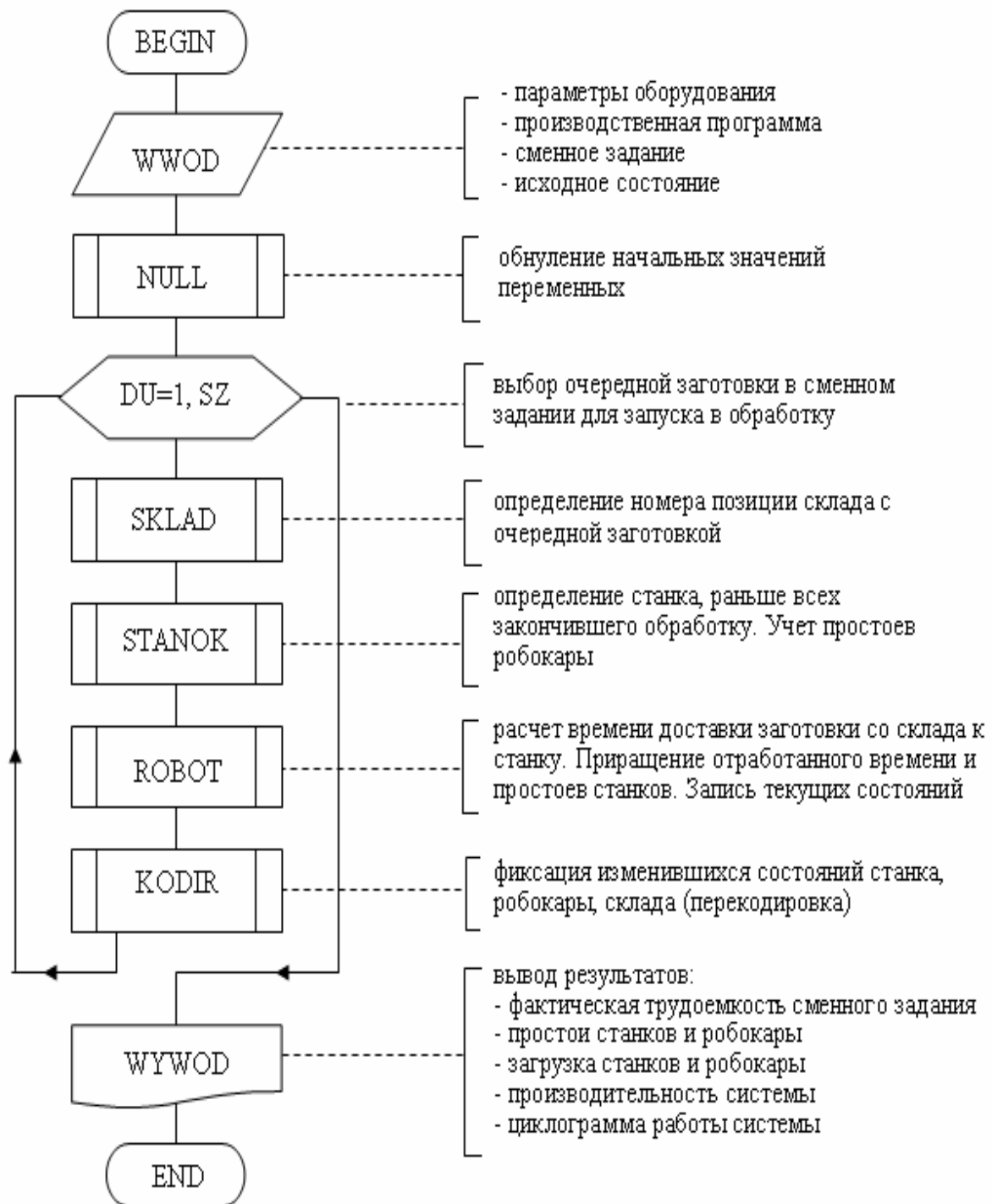


Рисунок 4.5 - Блок-схема программы построения циклограмм

Все заготовки на паллетах установлены в ячейках склада, в заданной последовательности поступают в обработку на любой свободный станок и после обработки возвращаются робокарой на склад. Состояние станков анализируется в моменты завершения робокарой транспортных операций. Время транспортных операций рассчитывают с учетом расстояний, варьирующихся в зависимости от местонахождения точек отправления и прибытия. Текущее состояние модулей записывается в файл, содержимое которого и используется для построения циклограмм работы оборудования. Текст программы моделирования, построенной в соответствии с рисунком 4.5, не превышает 150 строк операторов на языке Паскаль.

Результат выполнения программы в виде циклограммы работы станков и робокары представлен на рисунке 4.6.

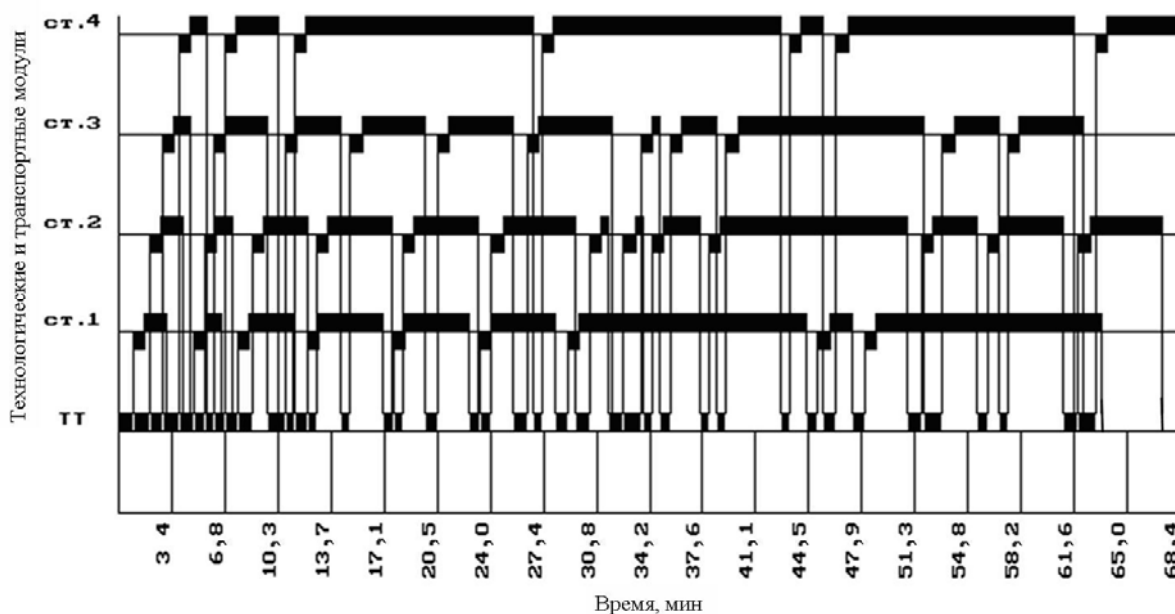


Рисунок 4.6 - Циклограмма работы оборудования ГПЯ: ст.1...ст.4 – станки с 1 по 4, соответственно; ТТ – транспортная тележка (робокара)

Циклограмма позволяет: проверить корректность работы программы; визуально оценить величину и причины простоев оборудования; рассчитать сводные показатели эффективности работы системы и представить их в табличной или графической форме (рисунок 4.7).

Таким образом, полученная компьютерная модель позволяет на уровне технологической операции вскрыть зависимости показателей эффективности ГПЯ от технических, технологических и организационных решений, использованных в качестве исходных данных (рисунок 4.8).

Использование модели в рассмотренном виде возможно для оценки диспетчерских решений: состава сменного задания (СЗ), очередности запуска заготовок в обработку, способа размещения заготовками в ячейках склада.

А вот для оценки технических параметров оборудования необходимы многократные прогоны программы при различных вариантах СЗ, содержание которых приводит к значительным колебаниям эффективности ГПЯ [61].

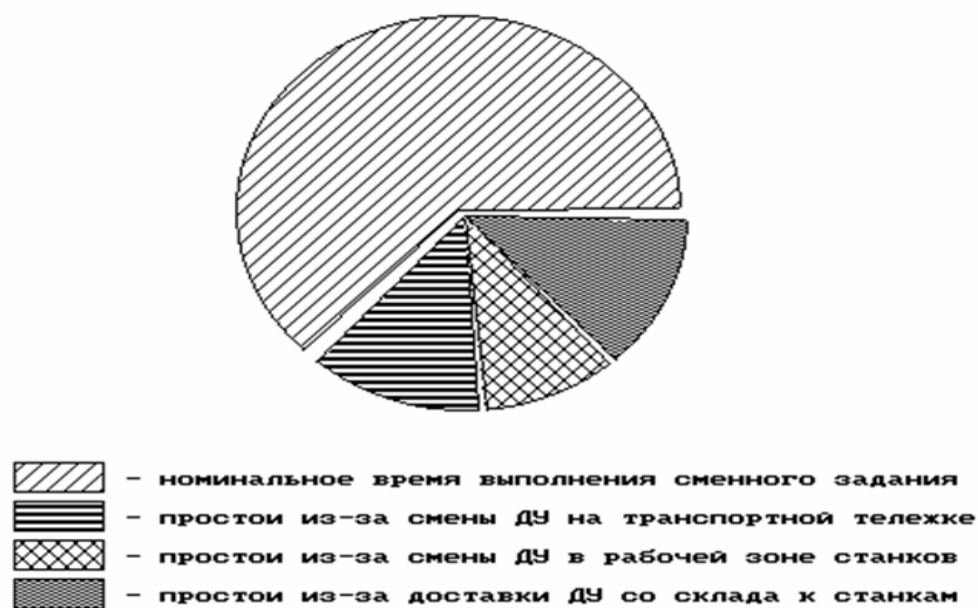


Рисунок 4.7 - Сводные результаты моделирования в виде круговой диаграммы

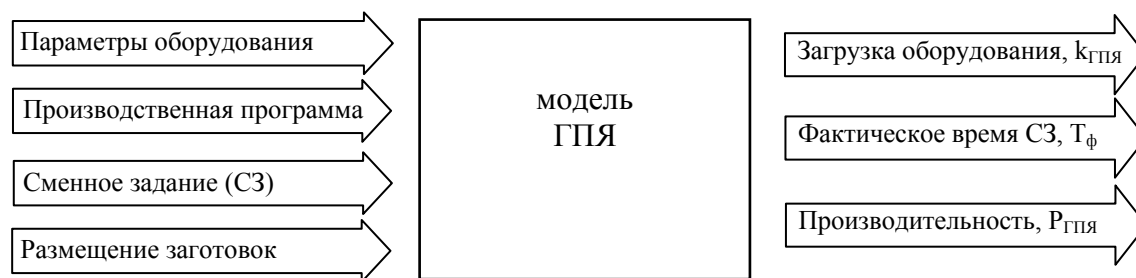


Рисунок 4.8 - Схема учета влияния технических, технологических и организационных решений на эффективность ГПЯ

Процесс многократных прогонов модели можно автоматизировать, если предварительно сформировать массив возможных вариантов СЗ; прогнать модель при каждом варианте СЗ; выполнить статистическую обработку показателей эффективности ГПЯ.

Моделирование при различных вариантах СЗ позволяет рассчитывать показатели эффективности ГПЯ с учетом их рассеяния. В таком виде модель применима для оценки технических параметров оборудования, например, для предпроектного расчета требуемой скорости робокары (рисунок 4.9).

Получаемые по результатам моделирования показатели загрузки оборудования, производительности системы и фактической длительности цикла безлюдной работы оценивают эффективность системы с отдельных сторон. Обобщенным критерием целесообразности создания или приобретения ГПЯ может служить срок окупаемости затрат L_O , выражение для расчета которого предложено в работе [60].

Из примера на рисунке 4.10 следует, что для использованных данных изменение проектной скорости робокары с 0,5 до 2,5 м/с позволяет снизить срок окупаемости ГПЯ с $3,6^{+0,8}_{-0,8}$ лет до $2,3^{+1,3}_{-0,6}$ года.

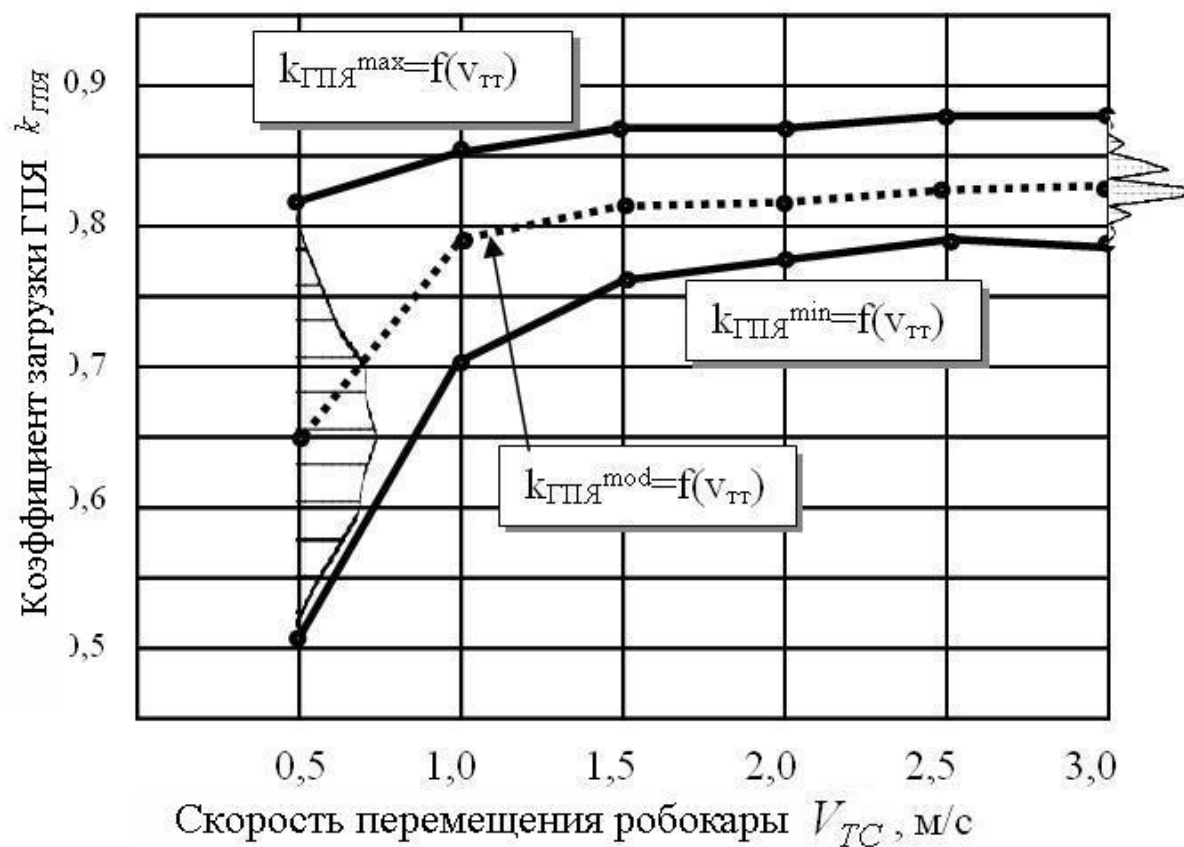


Рисунок 4.9 - Влияние скорости перемещения робокары (транспортной тележки) V_{TC} на величину коэффициента загрузки ГПЯ $k_{ГПЯ}$

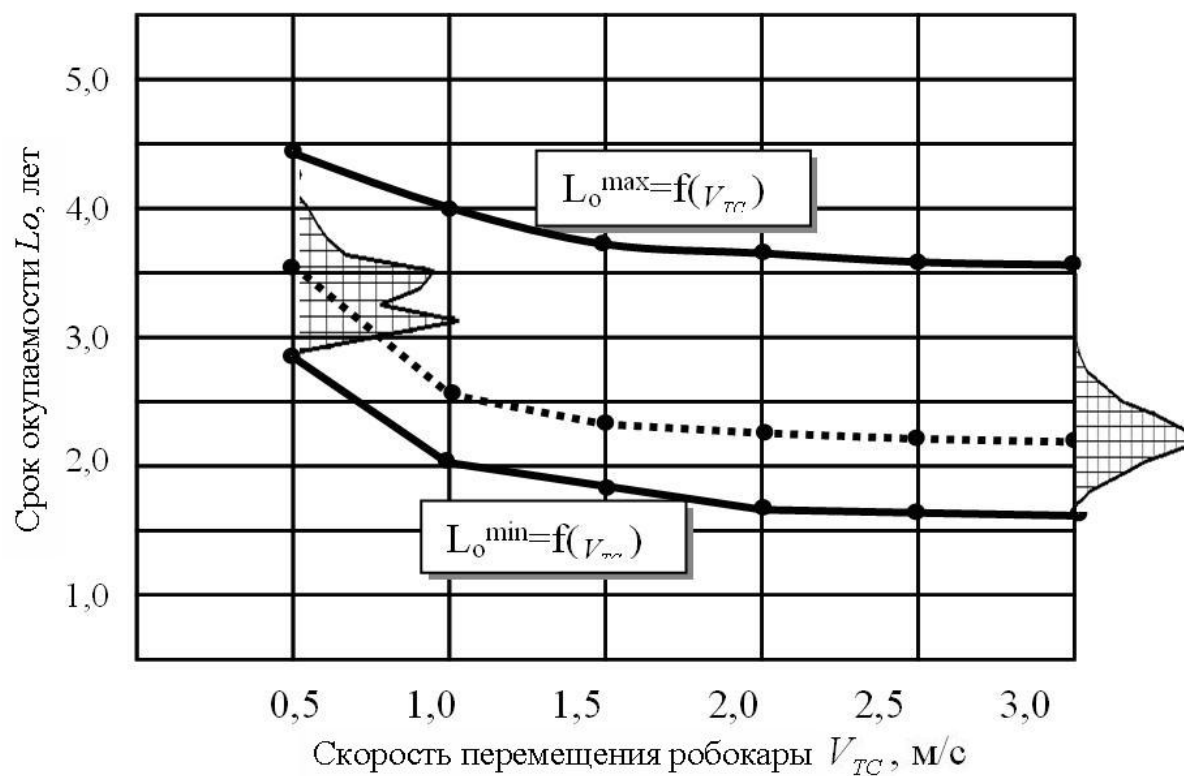


Рисунок 4.10 - Влияние скорости перемещения робокары V_{TC} на срок окупаемости ГПЯ L_o

Использование полученной модели и обобщенного показателя эффективности позволяет: произвести оптимизацию проектных параметров оборудования, минимизировав тем самым средний срок L_o окупаемости затрат на ГПЯ; по заданным ограничениям на наибольший срок окупаемости затрат $L_o^{\max}$ оптимизировать эксплуатационные режимы в виде технологических и организационных решений.

Таким образом, полученная модель пригодна для оптимизации всего комплекса технических, технологических и организационных решений, использованных в качестве исходных данных.

Моделирование основано на аналитическом описании длительности циклов работы и логическом описании многоуровневого взаимодействия модулей и их агрегатов. Длительность цикла работы каждого агрегата включает постоянные составляющие (например, цикл смены паллеты, цикл смены режущего инструмента) и переменные, продолжительность которых описана специально полученными аналитическими зависимостями.

Использованный подход позволяет учесть в совокупности практически все основные факторы реального производственного процесса - от размеров на схемах планировки оборудования до алгоритмов работы отдельных агрегатов модулей, ресурса каждого отдельного инструмента и его местонахождения в магазине или накопителе [61].

4.3 Общий подход к разработке компьютерной модели ГПЯ

4.3.1 Постановка задачи

В настоящей работе формальное описание объекта моделирования построено на следующих исходных посылах. Производственный процесс рассматривается на протяжении цикла автоматической (безлюдной) работы ГПЯ по выполнению предписанного сменного задания. Перед началом цикла известно размещение заготовок в накопителях и рабочих зонах станков, известны технологические процессы (операционная технология) изготовления деталей и стадии обработки заготовок. Известен состав технологических и сервисных модулей ГПЯ (структура), их параметры, а также алгоритмы взаимодействия модулей и их агрегатов.

Структурный состав модулей участка может включать автоматический склад режущих инструментов с транспортными средствами передачи РИ на станки; станки, оснащенные инструментальными магазинами, устройствами автоматической смены и автоматической замены инструментов, а также пристаночными накопителями заготовок; координатно-измерительную машину; автоматический - склад заготовок с транспортными средствами доставки их к станкам.

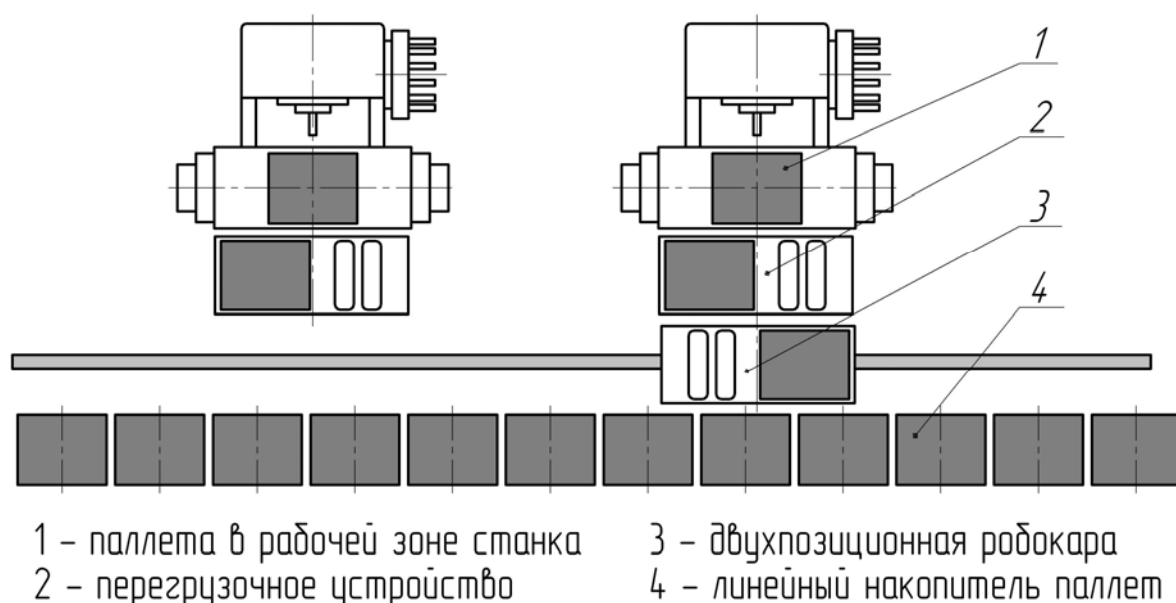
Компьютерная модель отображает протекание производственного процесса с заданным уровнем детализации: на уровне технологической операции или технологического перехода. Результаты работы компьютерной модели должны допускать возможность проверки корректности полученных результатов.

Эффективность работы ГПЯ оценивается по результатам моделирования по одному или нескольким типовым критериям: загрузка технологического оборудования, производительность и срок окупаемости системы.

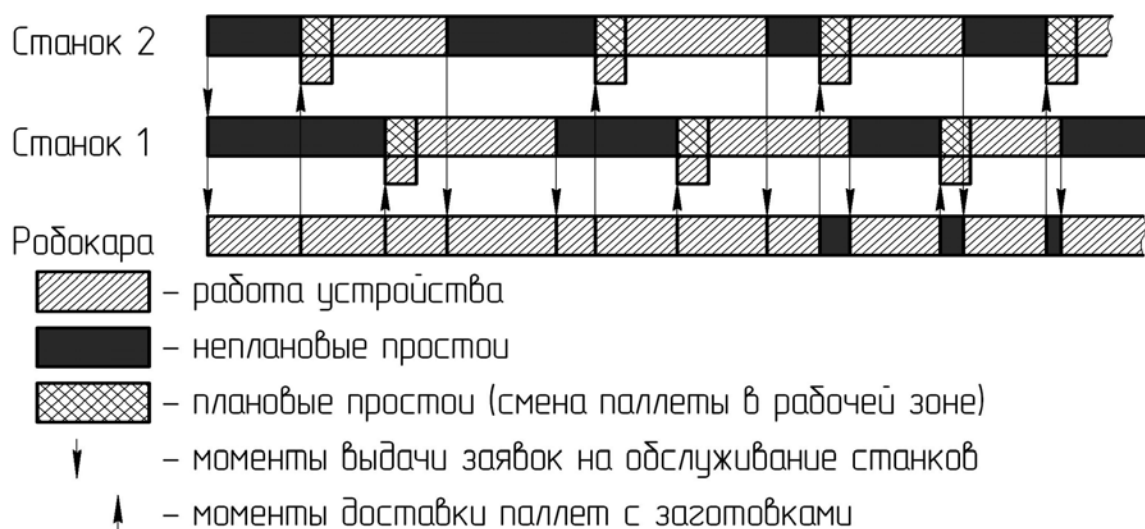
4.3.2 Построение алгоритма моделирования

Пусть имеется ГПЯ, включающая два станка, транспортную тележку (робокару) и линейный накопитель паллет с заготовками (рисунок 4.11 а). Возможная циклограмма работы такой системы представлена в соответствии с рисунком 4.11 б.

Положим, что перед началом работы системы все заготовки на паллетах размещены в позициях накопителя. В этом случае работа системы начинается с доставки робокарой заготовок из накопителя к станкам.



а



б

Рисунок 4.11 – Схема (а) и фрагмент циклограммы (б) работы ГПЯ

После доставки к станку паллета с заготовкой поступает на перегрузочное устройство и далее - в рабочую зону станка. Станок по управляющей программе производит обработку заготовки, после чего паллета выгружается в перегрузочное устройство.

Как только рабочая зона станка освободилась, выдается заявка на обслуживание станка транспортным средством.

После завершения робокарой очередной транспортной операции система управления ГПЯ анализирует поступившие заявки на обслуживание станков. Если заявок нет, то робокара простаивает до момента поступления новой заявки. Если есть одна заявка, то начинается ее выполнение. Если заявок несколько, то выбирается та, которая удовлетворяет заложенным в систему управления приоритетам (по очереди поступления заявок, по приоритетам станков и пр.). Процесс повторяется до тех пор, пока все заготовки не будут доставлены робокарой к станкам и не завершится обработка последней из них.

4.3.3 Организация моделирования

Способы организации имитационного моделирования рассмотрены в теории массового обслуживания. В памяти компьютера отводится несколько ячеек для переменных, характеризующих состояние моделируемой системы в целом и отдельных её элементов. Содержимое этих ячеек изменяется в соответствии с алгоритмом моделирования так, как это происходит в реальной системе при её функционировании. Отдельная ячейка содержит текущее системное время указывающее, к какому моменту времени относится состояние элементов системы, записанных в памяти компьютера. Содержимое указанных ячеек памяти меняется путем циклического повторения основной части алгоритма моделирования называемой шагом имитации.

За один шаг осуществляется переход к следующему значению системного времени, т.е. продвижение по времени. Попутно изменяется значение переменных, характеризующих состояние элементов системы. Таким образом, шаг за шагом, имитируется смена состояний системы, т.е. моделируется процесс функционирования ГПЯ.

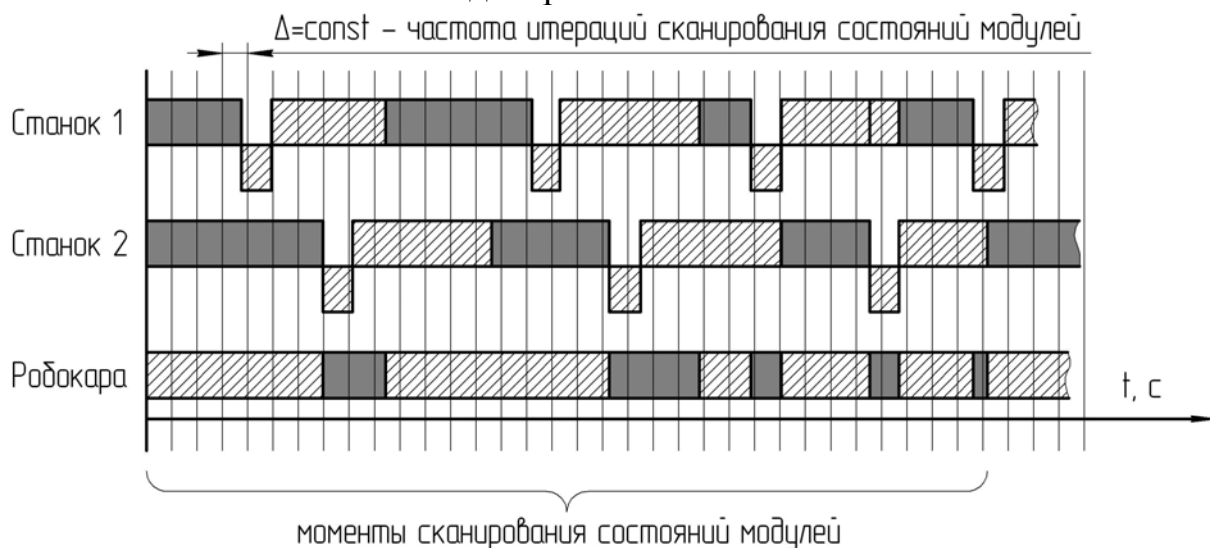
Рассмотрим способы продвижения во времени.

1) Увеличение системного времени за один шаг на постоянную величину Δ_t (принцип Δ_t). Такой способ удобен при моделировании, учитывающем, например, отказы в работе оборудования (рисунок 4.12 а). Однако при использовании этого способа возникает проблема выбора длины интервала Δ_t . К тому же чем выше частота итерации, т.е. чем меньше дискретность повторений, тем точнее получаемые результаты, но тем дольше будет работать модель. Поскольку повторять моделирование при выборе тех или иных проектных решений приходится иногда сотни раз, возникает проблема поиска компромисса между точностью модели и ее производительностью.

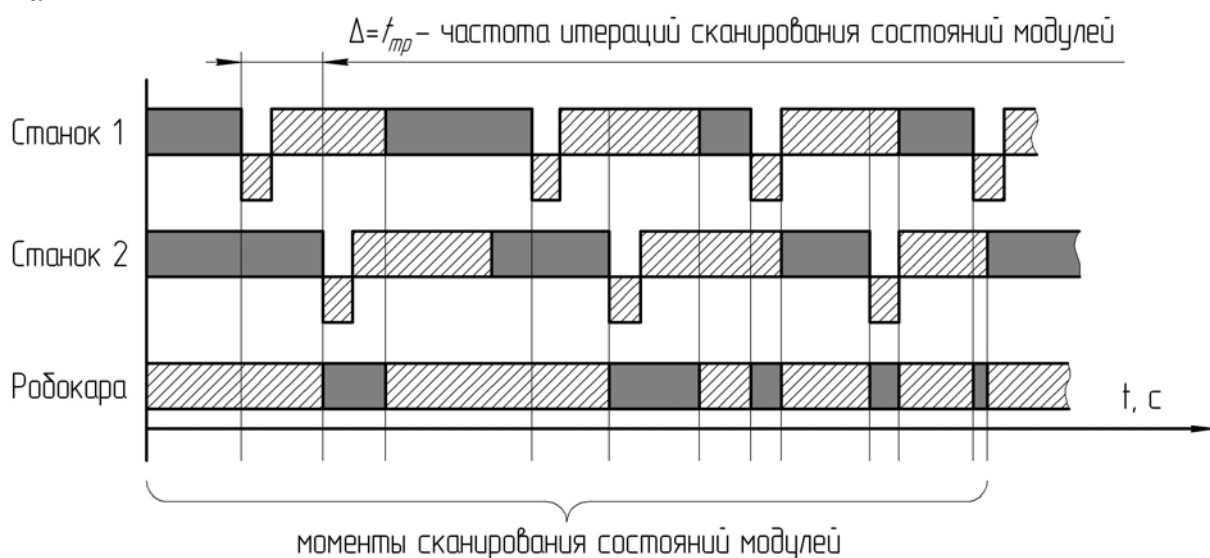
2) Принцип особых моментов. За один шаг осуществляется продвижение по времени на случайный отрезок: от одного особого момента до другого. Особым мо-

ментом принято называть такой момент времени, когда в системе происходит какое-либо изменение, иначе говоря - происходит событие. Частота итераций в таком случае связывается с устройством, работа которого предопределяет состояние других устройств. В рассматриваемом примере таким устройством является робокара, для которой моменты завершения транспортных операций являются событиями начала работы станков (рисунок 4.12 б).

Подобный подход позволяет резко повысить производительность модели при сохранении высокой точности моделирования.



а



б

Рисунок 4.12 – Моделирование ГПЯ с постоянной (а) и переменной (б) частотой сканирования

4.4 Критерии оценки результатов моделирования

Основное условие быстрой окупаемости ГПЯ - как можно более полное использование их во времени. Экономические характеристики ГПЯ ужесточили требования к временной и пространственной организации автоматизированного производственного процесса, и, как следствие, усложнили его проектирование.

4.4.1 Средство визуализации показателей эффективности ГПЯ

Средством иллюстрации организации производственного процесса служат циклограммы, в виде отрезков времени символизирующие взаимосвязанные циклы работы оборудования (рисунок 4.13).

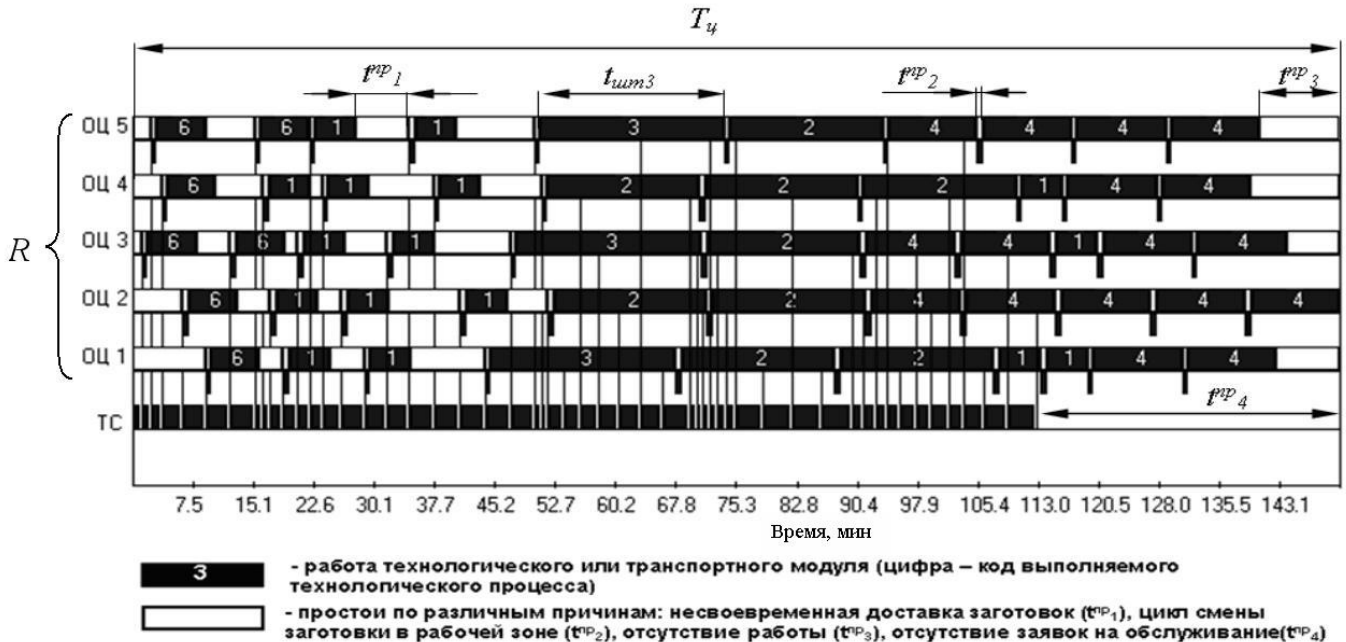


Рисунок 4.13 – Схема к иллюстрации расчета показателей эффективности ГПЯ

Одним из критериев оценки эффективности производственного процесса служит коэффициент загрузки основного технологического оборудования $k_{ГПЯ}$, который характеризует интенсивность работы системы как соотношение между длительностью работы и простоев оборудования.

Математически коэффициент загрузки ГПЯ можно выразить как

$$k_{ГПС} = \frac{(T_{Ц} \times C - P_{\Sigma})}{T_{Ц} \times C} \times 100, \quad (4.3)$$

где $T_{Ц}$ – длительность цикла безлюдной работы, мин;

C – число станков, шт.;

P_{Σ} – простои станков по различным причинам за цикл безлюдной работы, мин (рисунок 4.13)

$$P_{\Sigma} = \sum_{i=1}^C \sum_{j=1}^j t_{i,j}^{np} \quad (4.4)$$

Графически $k_{ГПЯ}$ представляется в виде сектора круговой диаграммы, соответствующего фонду времени работы оборудования.

Другим критерием эффективности служит производительность системы P_0 . Производительность характеризует интенсивность выпуска продукции и может быть выражена:

а) в виде среднего времени изготовления одного изделия, мин/шт.

$$t_{cp} = \frac{T_{Ц}}{N}; \quad (4.5)$$

б) в виде количества изделий в единицу времени, шт./мин

$$n_{cp} = \frac{N}{T_{Ц}}, \quad (4.6)$$

где N , шт. – число изделий, изготовленных за цикл работы ГПЯ.
Оба вида производительности связаны соотношением

$$t_{cp} = \frac{1}{n_{cp}} \quad (4.7)$$

Понятию «рост производительности» соответствует увеличение количества выпускаемых изделий n_{cp} – это и будем понимать в дальнейшем под производительностью ГПЯ и обозначать как P_0 .

Очевидно, что $k_{ГПЯ}$ и P_0 не зависят друг от друга и характеризуют разные стороны производственного процесса. Обобщенным критерием эффективности может служить срок окупаемости ГПЯ L_0 , учитывающий коэффициент загрузки и производительность системы.

Рассмотрим вывод уравнения для расчета L_0 на основе результатов моделирования.

4.4.2 Вывод зависимости для расчета срока окупаемости ГПЯ

В разделе 2.1 получено выражение (2.6) для расчета срока окупаемости ГПЯ. Усовершенствуем полученное выражение для его использования в практических расчетах, выполняемых на основе компьютерного моделирования. Для этого перепишем выражение (2.5) в виде

$$L_0 = \frac{K \times t_{cp}}{(C - S) \times T_{ц} \times d \times k_{ГПС}}, \quad (4.8)$$

где K - капитальные вложения в создание ГПЯ, руб.;
 t_{cp} - среднее время изготовления единицы продукции, час;

C - рыночная цена единицы продукции, руб./шт.;

S - расходы, связанные с изготовлением и сбытом единицы продукции (себестоимость), руб./шт.;

$T_{ц}$ - продолжительность одного цикла работы ГПЯ, час;

d - число циклов работы в год, шт.;

$k_{ГПС}$ - коэффициент загрузки оборудования ГПЯ внутри цикла.

Из условия равенства трудоемкости годовой производственной программы T_H и годового фонда времени работы ГПЯ ($T_H = T_{ц} \times d \times R$), выразим число циклов работы ГПЯ в год

$$d = \frac{T_H}{T_{ц} \times R}, \quad (4.9)$$

где R - число станков в ГПЯ, шт.

Подставив выражение (4.9) в выражение (4.8), получим

$$L_0 = \frac{K \times t_{cp} \times R}{(C - S) \times T_H \times k_{ГПС}}. \quad (4.10)$$

Введем коэффициент q , характеризующий соотношение капитальных затрат, прибыли на единицу продукции и трудоемкости годовой производственной программы

$$q = \frac{K}{C - S}. \quad (4.11)$$

С учетом введенных обозначений (4.10) примет вид:

$$L_0 = \frac{q \times R}{T_H} \times \frac{t_{cp}}{k_{ГПС}} \quad (4.12)$$

Выразив $t_{cp} = T_{сз} / N_{сз}$, получим

$$L_0 = \frac{q \times R}{T_H} \times \frac{T_{сз}}{N_{сз} \times k_{ГПС}}, \quad (4.13)$$

где N_{c3} – число деталей, изготовленных за цикл работы ГПЯ.

Определим примерный диапазон значений коэффициента q . Отношение T_H / R согласно (4.9) представляет собой плановый годовой фонд времени работы ГПЯ $T_H \times d$. Тогда (4.13) можно записать как

$$L_0 = \frac{q}{N_{c3} \times d} \times \frac{T_{c3}}{T_H \times k_{ГПС}}. \quad (4.14)$$

В идеальном случае, когда простои оборудования отсутствуют, $k_{ГПС} = 1$ и $T_H = T_{c3}$. Выражение (4.14) примет вид

$$L_0 = \frac{q}{N_{c3} \times d}. \quad (4.15)$$

Произведение в знаменателе есть ни что иное, как число изделий в годовой программе выпуска

$$N_{\Sigma} = N_{c3} \times d, \text{ или } N_{\Sigma} = \sum_{i=1}^L N_i, \text{ откуда } L_0 = \frac{q}{N_{\Sigma}}, \quad (4.16)$$

где i – индекс позиции номенклатуры годовой программы;

N_i – количество изделий i -го наименования в годовой программе выпуска;

L – число позиций (наименований) изделий.

Рассматривая идеальный случай дальше, примем, что номинальный срок окупаемости ГПЯ $L_H = L_0$. Тогда $L_H = \frac{q}{N_{\Sigma}}$, откуда

$$q = L_H \times N_{\Sigma}. \quad (4.17)$$

Подставив (4.17) в (4.14), получим выражение для расчета срока окупаемости ГПЯ

$$L_0 = L_H \times \frac{N_{\Sigma} \times R}{T_H} \times \frac{T_{c3}}{N_{c3} \times k_{ГПС}} \quad (4.18)$$

Учитывая, что $P_0 = \frac{N_{cз}}{T_{cз}}$, окончательно получим

$$L_0 = L_H \times \frac{N_{\Sigma} \times R}{T_H} \times \frac{1}{P_0 \times k_{ГПС}}. \quad (4.19)$$

Полученное выражение позволяет рассчитать фактический срок окупаемости ГПЯ на основе сопоставления требований Технического задания на проектирование системы (L_H , N_{Σ} , T_H) и показателей эффективности системы, обеспечиваемых Техническим предложением по ее изготовлению (R , $T_{cз}$, $N_{cз}$ и $k_{ГПС}$).

Номинальный срок окупаемости L_H , число изделий N_{Σ} и трудоемкость годовой производственной программы T_H задаются в исходных данных на проектирование. Число станков в системе R определяется расчетом и может уточняться по результатам моделирования.

Значения $T_{cз}$ и $k_{ГПС}$ получаются в результате моделирования ГПЯ при конкретном варианте сменного задания (при конкретном значении $N_{cз}$).

Таким образом, в качестве критериев оценки результатов моделирования будем использовать коэффициент загрузки системы $k_{ГПС}$, производительность P_0 и срок окупаемости системы L_0 .

Математические модели, полученные для оценки эффективности ГПЯ, представлены в сводной таблице 4.1.

Таблица 4.1 – Математические модели для оценки эффективности ГПЯ

Наименование	Обозначение	Расчетная формула	Расшифровка
Условие формирования сменного задания	-	$T_{ц} \times R \times \frac{(100-\delta)}{100} \leq \sum_{i=1}^L t_{шт}^i \times N_i \leq T_{ц} \times R$	$T_{ц}$, мин – плановое время цикла работы ГПЯ; R , шт – число станков; δ , % - допустимый процент недогрузки оборудования; L , шт – номенклатура изделий в годовой программе; $t_{шт}^i$, мин – время изготовления детали i -го наименования; N_i , шт – партия запуска деталей i -го наименования
Число заготовок в сменном задании	$N_{сз}$, шт	$N_{сз} = \sum_{i=1}^L N_i$	L , шт – номенклатура изделий в годовой программе; N_i , шт – партия запуска деталей i -го наименования
Суммарные простои станков	P_{Σ} , мин	$P_{\Sigma} = \sum_{i=1}^R \sum_{j=1}^{n_i} t_{п.ij}^j / 60$	$t_{п.ij}^j$, с – простой i -го станка, связанные с j -ой транспортной операцией; n_i , шт – число транспортных операций, выполненных для i -го станка
Суммарные простои транспортного средства	T_{Σ} , мин	$T_{\Sigma} = \sum_{j=1}^{n_{\Sigma}} t_{п.4j}^j / 60$	$t_{п.4j}^j$, с – текущие простои транспортного средства из-за отсутствия заявок на обслуживание; n_{Σ} , шт – общее число транспортных операций
Коэффициент загрузки транспортного средства	$k_{ТС}$, %	$k_{ТС} = \frac{(T_{сз} - T_{\Sigma})}{T_{сз}} \times 100$	$T_{сз}$, мин – модельное время цикла работы ГПЯ; T_{Σ} , мин – суммарные простои транспортного средства
Коэффициент загрузки ГПЯ	$k_{ГПС}$, %	$k_{ГПС} = \frac{(T_{сз} \times R - P_{\Sigma})}{T_{сз} \times R} \times 100$	$T_{сз}$, мин – модельное время цикла работы ГПЯ; P_{Σ} , мин – суммарные простои станков
Производительность ГПЯ	P_0 , шт/ч	$P_0 = \frac{N_{сз}}{T_{сз}} \times 60$	$N_{сз}$, шт – число заготовок в сменном задании
Срок окупаемости ГПЯ	L , лет	$L = L_H \times \frac{T_{ц}}{T_{сз} \times k_{ГПС}}$	L_H , лет - экономически рассчитанный срок окупаемости
Прирост срока окупаемости	L_0 , %	$L_0 = \frac{T_{ц}}{T_{сз} \times k_{ГПС}} \times 100$	процент увеличения срока окупаемости из-за технических показателей работы ГПЯ

4.5 Особенности статистических исследований ГПЯ в методе циклограмм

Результатом работы компьютерной программы служит цифровое описание работы оборудования ГПЯ за цикл выполнения сменного задания. При моделировании на уровне технологической операции, когда алгоритм моделирования привязан к работе транспортного средства, работа ГПЯ описывается состоянием станков и транспортного средства в моменты завершения транспортных операций (рисунок 4.14).

транс опер.	обсл. станок	код ДУ	прост ТС	время транс	момент достав	начало обраб.	конец обраб.
164	3	40	0.0	66.7	16579.3	20459.3	21059.3
165	1	40	69.3	70.7	16719.3	20619.3	21219.3
166	4	40	0.0	69.3	16788.7	20991.3	21591.3
167	5	40	0.0	60.0	16848.7	21048.7	21648.7
168	2	40	247.3	66.7	17162.7	21062.7	21662.7
169	3	40	0.0	73.3	17236.0	21109.3	21709.3
170	1	40	56.0	77.3	17369.3	21269.3	21869.3
171	4	40	0.0	76.0	17445.3	21636.3	22236.3
172	5	40	0.0	66.7	17512.0	21693.7	22293.7
173	2	40	227.3	73.3	17812.7	21712.7	22312.7
174	3	40	0.0	80.0	17892.7	21759.3	22359.3
175	1	40	42.7	84.0	18019.3	21919.3	22519.3
176	4	40	0.0	82.7	18102.0	22281.3	22881.3
177	5	40	0.0	73.3	18175.3	22338.7	22938.7

Рисунок 4.14 – Фрагмент файла с цифровым описанием работы ГПЯ

Визуализация полученных данных может быть представлена в виде: циклограммы работы оборудования (рисунок 4.15); в виде сводных данных об эффективности работы ГПЯ (рисунок 4.16).

Использование модели в рассмотренном виде возможно для оценки диспетчерских решений: состава сменного задания, очередности запуска заготовок в обработку, способа размещения заготовками в ячейках склада.

Для оценки эффективности различных правил обслуживания необходимы многократные прогоны модели при различных вариантах СЗ, содержание которых приводит к значительным колебаниям эффективности ГПЯ.

Процесс многократных прогонов модели можно автоматизировать, если: предварительно сформировать массив возможных вариантов СЗ; прогнать модель при каждом варианте СЗ; выполнить статистическую обработку показателей эффективности ГПЯ. Рассмотрим содержание этапов работы.

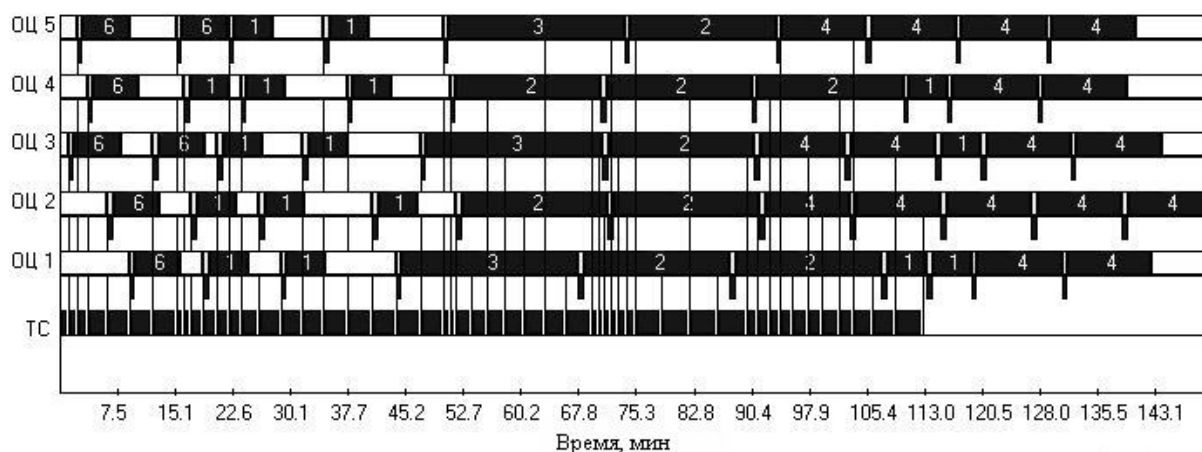


Рисунок 4.15 – Пример циклограммы, построенной по результатам цифрового описания работы ГПЯ

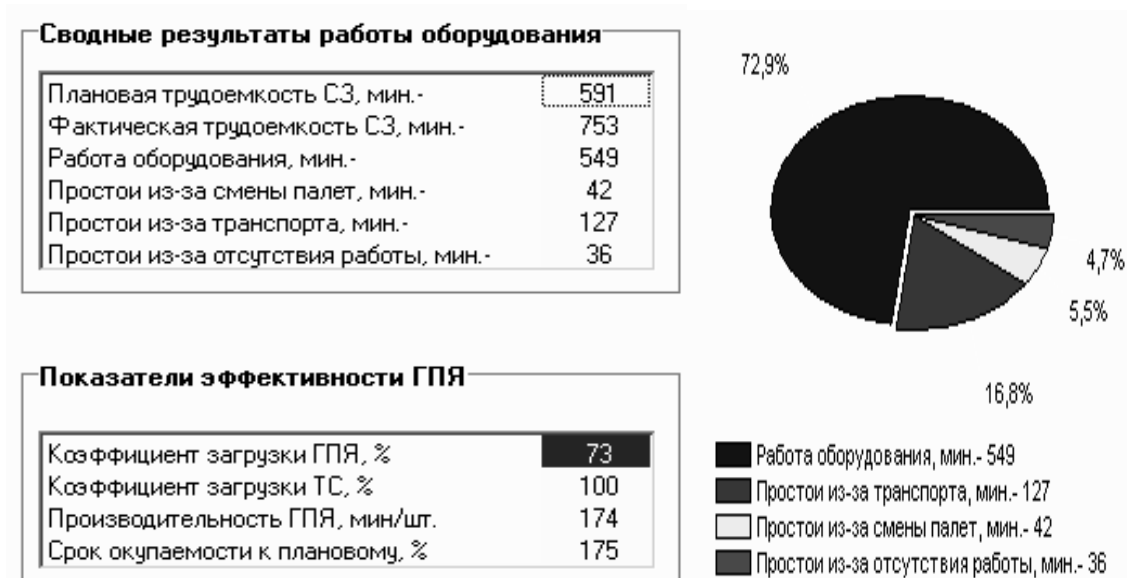


Рисунок 4.16 – Пример сводных результатов моделирования в текстовой и графической формах

4.5.1 Алгоритм формирования вариантов сменного задания

За основу взят следующий алгоритм перебора сочетаний значений нескольких переменных.

Задача. Пусть даны три переменных: А, В и С. Каждая переменная может принимать значения 0 или 1. Нужно выявить все варианты сочетаний значений переменных. Известное из комбинаторики решение в виде схемы алгоритма представлено на рисунке 4.17 а. Алгоритм достаточно прост, но неудобен, так как при изменении числа переменных требует корректировки. Соответственно нужно изменять и текст программы расчета.

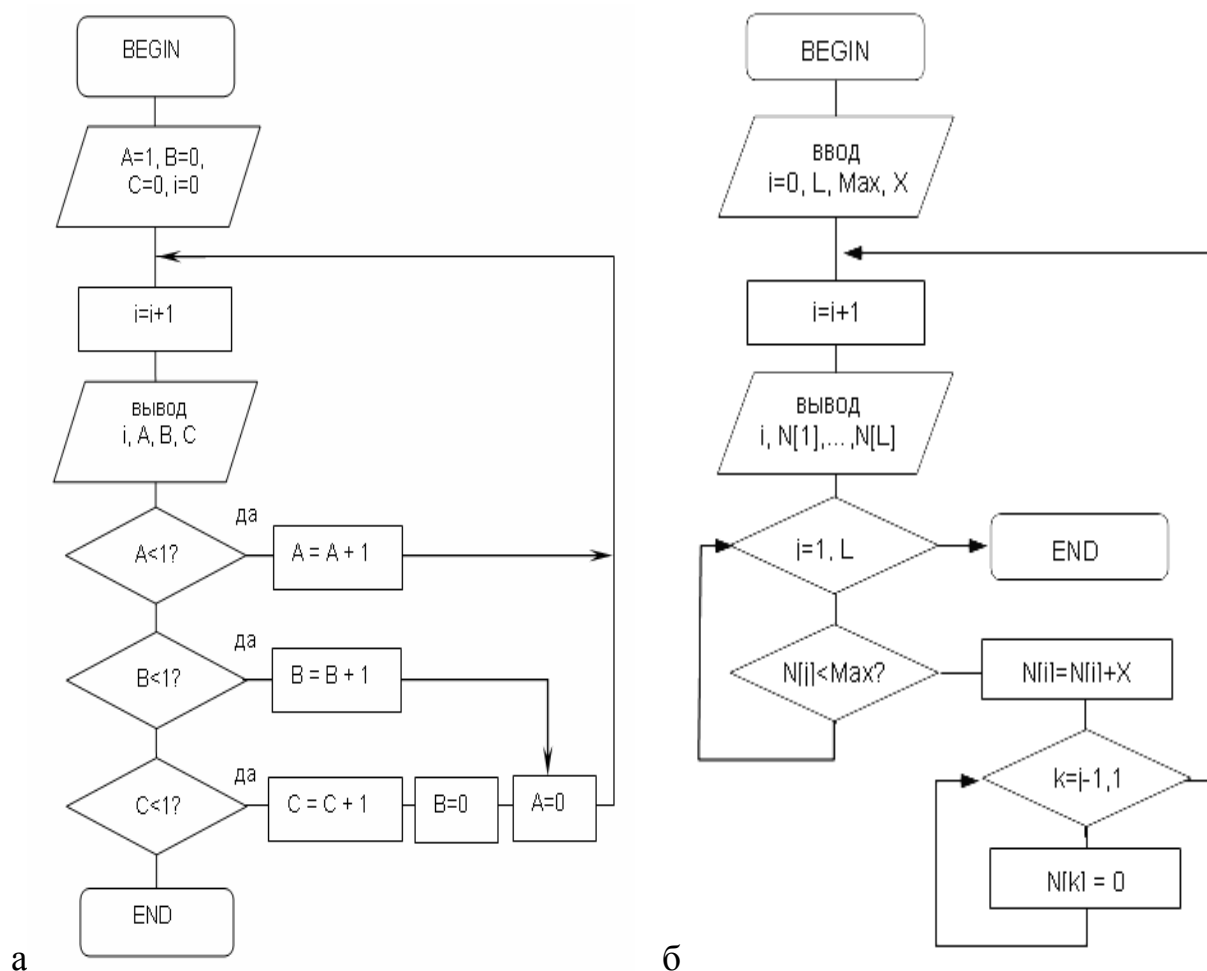


Рисунок 4.17 – Типовой (а) и усовершенствованный (б) алгоритмы итерационного перебора

Разработан универсальный алгоритм формирования сочетаний значений переменных, представленный на рисунок 4.17 б. Число переменных L и наибольшее значение каждой переменной Max дополнены дискретностью значений X , которые можно задавать в диалоговом режиме.

Суть алгоритма формирования вариантов сменного задания состоит в следующем. Известна номенклатура L деталей-установок (ДУ), изготовление которых осуществляется или планируется в ГПЯ. Для каждой ДУ известно время ее изготовления $t_1, t_2, \dots, t_L$. Задана также длительность цикла безлюдной работы ГПЯ F и допустимое время его недогрузки d_F . Сочетание заготовок $N_1 + N_2 + \dots + N_L$ считается возможным вариантом СЗ, если выполняется условие

$$(F - d_F) \times C_n \leq \sum_{i=1}^L (t_i \times N_i) \leq F \times C_n, \quad (4.20)$$

где C_n - число станков в составе ГПЯ, шт.;

t_i - оперативное время обработки заготовки по i -му техпроцессу, мин;

N_i - число заготовок в партии запуска по i -му техпроцессу, шт.;

L - номенклатура технологических процессов в производственной программе ГПЯ, шт.;

F – время цикла безлюдной работы ГПЯ, мин.

Методом итерационного перебора значений N_i из интервалов их допустимых значений выявляются возможные варианты сочетаний заготовок в сменном задании.

Если сформированный набор ДУ не удовлетворяет граничным условиям (4.20), то он отбрасывается, и формируется новый набор. Если удовлетворяет, то считается возможным вариантом СЗ и для него рассчитывается номенклатура и количество деталеустановок, номенклатура и количество требуемых режущих инструментов, а так же вероятностное распределение машинного и оперативного времени технологических переходов.

Все сформированные варианты СЗ удобно представить в виде таблицы (рисунок 4.18) для последующего использования.

NN	1	2	3	4	5	6	Деталей
Min	0	0	0	0	0	0	180
Max-300	100	100	100	100	100	100	420
84	80	60	40	40	80	20	320
85	20	40	60	60	80	20	280
86	80	0	80	60	80	20	320
87	100	80	0	80	80	20	360
88	40	60	20	100	80	20	320
89	100	20	40	100	80	20	360
90	20	40	80	0	100	20	260
91	80	0	100	0	100	20	300
92	100	80	20	20	100	20	340

Рисунок 4.18 – Схема представления сформированного множества вариантов сменного задания

4.5.2 Статистическая обработка результатов моделирования

Статистическая обработка результатов моделирования сводится к построению практических кривых вероятностного распределения контролируемого показателя эффективности. Блок-схема алгоритма статистической обработки представлена в соответствии с рисунком 4.19.

Пусть $\{1, \dots, i, \dots, n\}$ есть множество вариантов сменного задания. В результате моделирования получено множество значений коэффициента загрузки ГПЯ $\{k_{ГПС}^1, \dots, k_{ГПС}^i, \dots, k_{ГПС}^n\}$, значения которых находятся в интервале $\{k_{ГПС}^{\min}, \dots, k_{ГПС}^{\max}\}$.

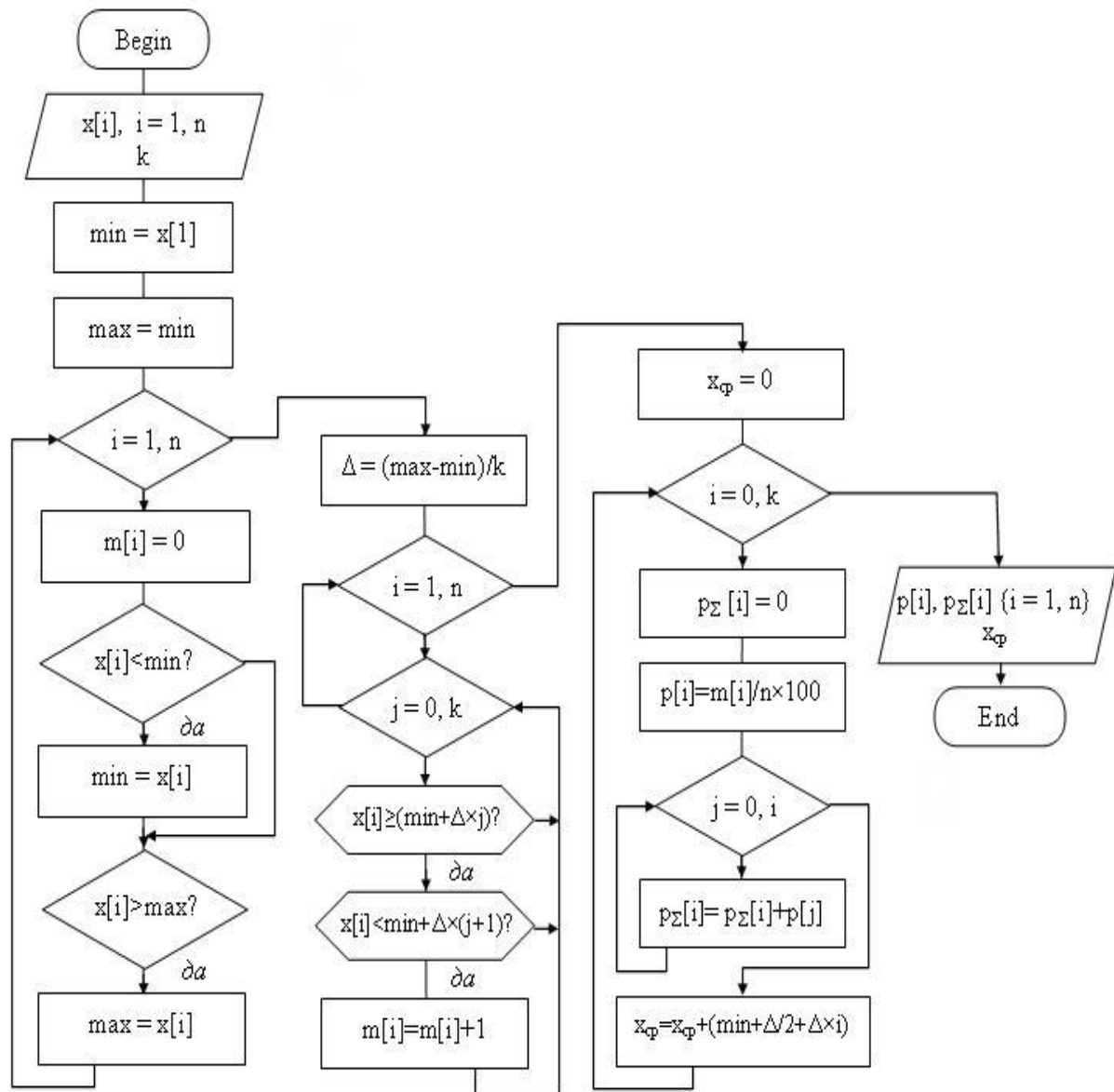
Поле рассеивания ω_k определяется как $\omega_k = k_{ГПС}^{\max} - k_{ГПС}^{\min}$. Разбив ω_k на x ин-

тервалов, определяем ширину $\Delta_x = \frac{\omega}{x}$ и границы каждого интервала.

Условие попадания значения $k_{ГПС}^i$ в j -ый интервал ($j = \{0, 1, 2, \dots, x\}$) запишем в виде:

$$(k_{ГПС}^{\min} + \Delta_x \times j) \leq k_{ГПС}^i < (k_{ГПС}^{\min} + \Delta_x \times (j + 1)). \quad (4.21)$$

При выполнении условия (4.21) отрабатывается приращение счетчика попаданий $m_j = m_j + 1$, где $m_j = \{0, \dots, \max_j\}$.



$x[i], i = 1, n$ – массив из n значений исследуемого параметра; k – число интервалов; $p[i], i = 1, n$ – вероятность попадания в i -ый интервал; $p_{\Sigma}[i], i = 1, n$ – накопленная вероятность i -го интервала; x_{cp} – среднее арифметическое значение.

Рисунок 4.19 – Схема алгоритма статистической обработки результатов моделирования

После распределения n значений $k_{ГПС}^i$ по интервалам получаем массив значений $m_j, j = \{0, 1, 2, \dots, x\}$, для которого выполняется условие

$$n = \sum_{j=0}^x m_j. \quad (4.22)$$

Для каждого значения m_j находим его относительное значение в виде вероятности (в процентах) p_j попадания в интервал, $j = \{0, 1, 2, \dots, x\}$:

$$p_j = \frac{m_j}{n} \times 100. \quad (4.23)$$

Полученные таким образом значения p_j используются для построения практической кривой вероятностного распределения 1 (рисунок 4.20), а также для расчета накопленной вероятности $p_{\Sigma}^j, j = \{0, 1, 2, \dots, x\}$:



1 – кривая распределения вероятности; 2 – кривая накопления вероятности

Рисунок 4.20 – Вероятностные кривые, получаемые при статистической обработке результатов моделирования

$$p_{\Sigma}^j = \sum_{i=1}^j p_i, \quad p_{\Sigma}^x = 100 (\%). \quad (4.24)$$

Кривая накопленной вероятности (кривая 2 на рисунке 4.20) используется в инженерных расчетах, учитывающих риск принятия решений.

Выводы

Обоснован общий подход к обеспечению высокой эффективности гибких производственных ячеек, согласно которому минимальный срок окупаемости производственной системы можно достичь лишь на основе единого комплекса решений, принимаемых специалистами разных профилей: технологами, проектировщиками, конструкторами, специалистами по АСУП, диспетчерами.

Отмечены недостатки общесистемных методов и средств компьютерного моделирования ГПЯ, не обеспечивающих возможность комплексного решения указанной проблемы.

Представлено описание метода автоматизированного построения циклограмм работы оборудования, использование которого в компьютерном моделировании ГПЯ позволяет создать универсальное средство поддержки принятия решений для специалистов указанных профилей.

Проработано методическое, математическое, информационное и логическое обеспечение метода циклограмм, дальнейшее развитие которого предполагает алгоритмическую и программную проработку компьютерных моделей на разных иерархических уровнях взаимодействия технологического и сервисного оборудования.

5 Модели ГПЯ на разных уровнях иерархии

Моделирование работы ГПЯ возможно с разной степенью детализации процесса функционирования: на уровне технологической операции и на уровне технологического перехода.

Уровень технологической (транспортной) операции предполагает анализ работы системы как взаимодействия технологических и транспортных модулей, выполняющих операции по перемещению и формообразованию объектов производства. Данный уровень позволяет количественно оценить потери в работе технологического оборудования из-за несвоевременной доставки заготовок. По результатам моделирования возможно уточнение характеристик систем основного технологического оборудования и транспортно-накопительной, что необходимо на ранних стадиях проектирования.

Уровень технологического (инструментального) перехода предполагает количественную оценку потерь при работе технологического модуля в процессе выполнения технологической операции, состоящей из множества технологических переходов. Данная задача особенно актуальна для многоцелевых станков, из которых, в основном, и создаются современные ГПЯ. По результатам моделирования на данном уровне возможно уточнение характеристик устройств автоматической смены и замены инструментов, технологических режимов обработки и стойкости инструментов, накопителей, транспортных средств и алгоритмов автоматизированной системы инструментального обеспечения. Применение моделей данного уровня целесообразно на завершающих этапах разработки технического предложения по созданию ГПЯ.

Рассмотрим особенности реализации моделей на каждом уровне детализации.

5.1 Модели ГПЯ на уровне технологической операции

Укрупненный алгоритм модели данного уровня представлен в соответствии с рисунком 5.1. Для работы алгоритма необходимо ввести исходные данные по оборудованию, производственной программе, сменному заданию и исходному состоянию ГПЯ. Если необходимые данные по оборудованию выясняются после разработки математического обеспечения и программной реализации алгоритма, то для других данных особенности их формализованного описания целесообразно выполнить на данном этапе.

5.1.1 Формализованное описание сменного задания

Сменное задание служит исходным документом, в соответствии с которым организуется работа ГПЯ в планируемый k -ый цикл безлюдной работы системы.

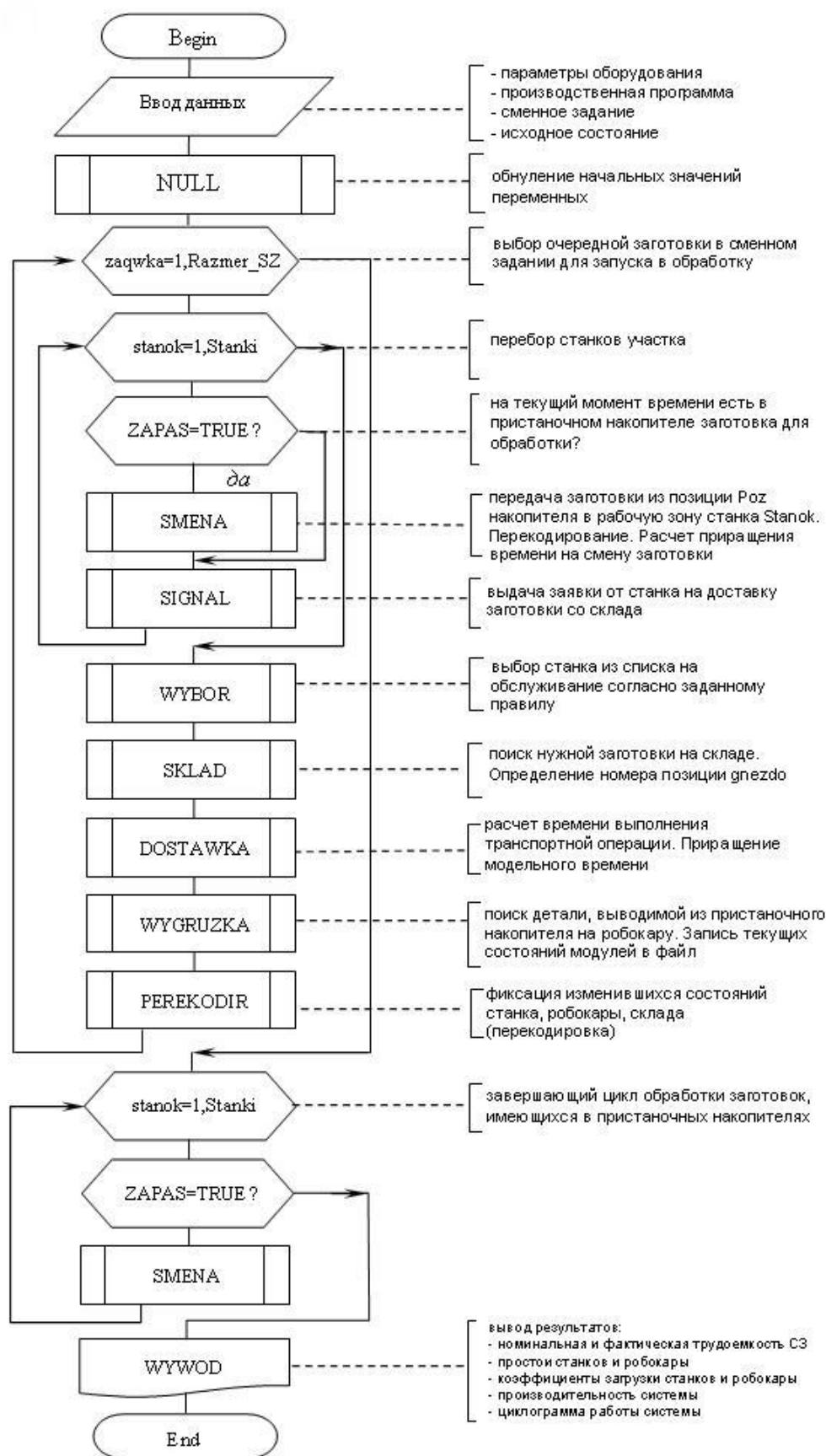


Рисунок 5.1 - Блок-схема алгоритма моделирования ГПЯ на уровне технологической операции

Состав СЗ задается числом партий запуска (ПЗ) заготовок в обработку, их очередностью и количеством заготовок в ПЗ в следующем виде

$$S_k = L_1 \cup n_1^L + \dots + L_i \cup n_i^L + \dots + L_Z \cup n_Z^L, \quad (5.1)$$

где $L_{i=1,Z} \in (l_1, l_2, \dots, l_L)$ - код наименования изделия из номенклатуры изделий L , закрепленных за данной производственной системой;
 n_i - количество одинаковых изделий в партии запуска, шт.;
 z - число партий запуска в k -м варианте сменного задания.

Общее число изделий в СЗ N_k^Σ составляет

$$N_k^\Sigma = \sum_{i=1}^z n_i^L. \quad (5.2)$$

Формой ввода данных по сменному заданию может служить матрица размерности $Z \times 3$

$$S_k = \begin{vmatrix} 1 & L_1 & n_1^L \\ \dots & \dots & \dots \\ z & L_Z & n_Z^L \end{vmatrix}. \quad (5.3)$$

Данная форма удобна для диалогового ввода и визуального контроля данных, позволяя представить сменное задание в виде таблицы 1.

Таблица 5.1 – Форма ввода сменного задания в программу моделирования ГПЯ

№ партии	Код партии запуска (ПЗ)	Размер ПЗ, шт.
1	L_1	n_1^L
...	...	...
2	L_i	n_i^L
...	...	...
z	L_Z	n_Z^L

При поштучной установке заготовок на паллеты и поочередном запуске их в обработку согласно сменному заданию на вход производственной системы подается очередь запуска заготовок вида

$$p_1, p_2, \dots, p_j, \dots = (\underbrace{L_1, L_1, \dots, L_1}_{n_1}, \underbrace{L_2, L_2, \dots, L_2}_{n_2}, \dots, \underbrace{L_i, L_i, \dots, L_i}_{n_i}, \dots, \underbrace{L_Z, L_Z, \dots, L_Z}_{n_z}). \quad (5.4)$$

Выражение (5.3) применяется для ввода сменного задания, выражение (5.4) - для его выполнения. Преобразование введенного СЗ (таблица 5.1) в очередь запуска заготовок в обработку $p_1, p_2, \dots, p_j, \dots$ выглядит следующим образом

$$\left. \begin{array}{l} p_1 = L_1 \\ \dots \\ p_{n_1} = L_1 \end{array} \right\} n_1$$

$$\left. \begin{array}{l} p_{(n_1+1)} = L_2 \\ \dots \\ p_{(n_1+n_2)} = L_2 \end{array} \right\} n_2$$

$$\left. \begin{array}{l} \dots \\ p_{(n_1+n_2+\dots+n_z)} = L_z \end{array} \right\} n_z$$

Форма таблицы 5.1 позволяет в некоторых случаях свернуть большой объем входных данных. Например, если СЗ состоит из 1870 изделий 3-го наименования, то содержание таблицы 5.1 будет иметь вид

№ пртии	Код партии запуска (ПЗ)	Размер ПЗ, шт.
1	3	1870

Сменное задание же будет представлять собой последовательность из 1780 изделий 3-го наименования.

Алгоритм процедуры преобразования сменного задания из формы ввода данных в упорядоченную последовательность запуска заготовок в обработку представлен в соответствии с рисунком 5.2.

Входом процедуры служат число партий запуска PZ , коды партий запуска $Kod[i]$, $i=1, PZ$ и число заготовок в каждой партии запуска $Kol[i]$, $i=1, PZ$. Выходными данными процедуры является число заготовок в сменном задании Sum и последовательность запуска заготовок в обработку $Posled[i]$, $i=1, Sum$ (таблица 5.2).

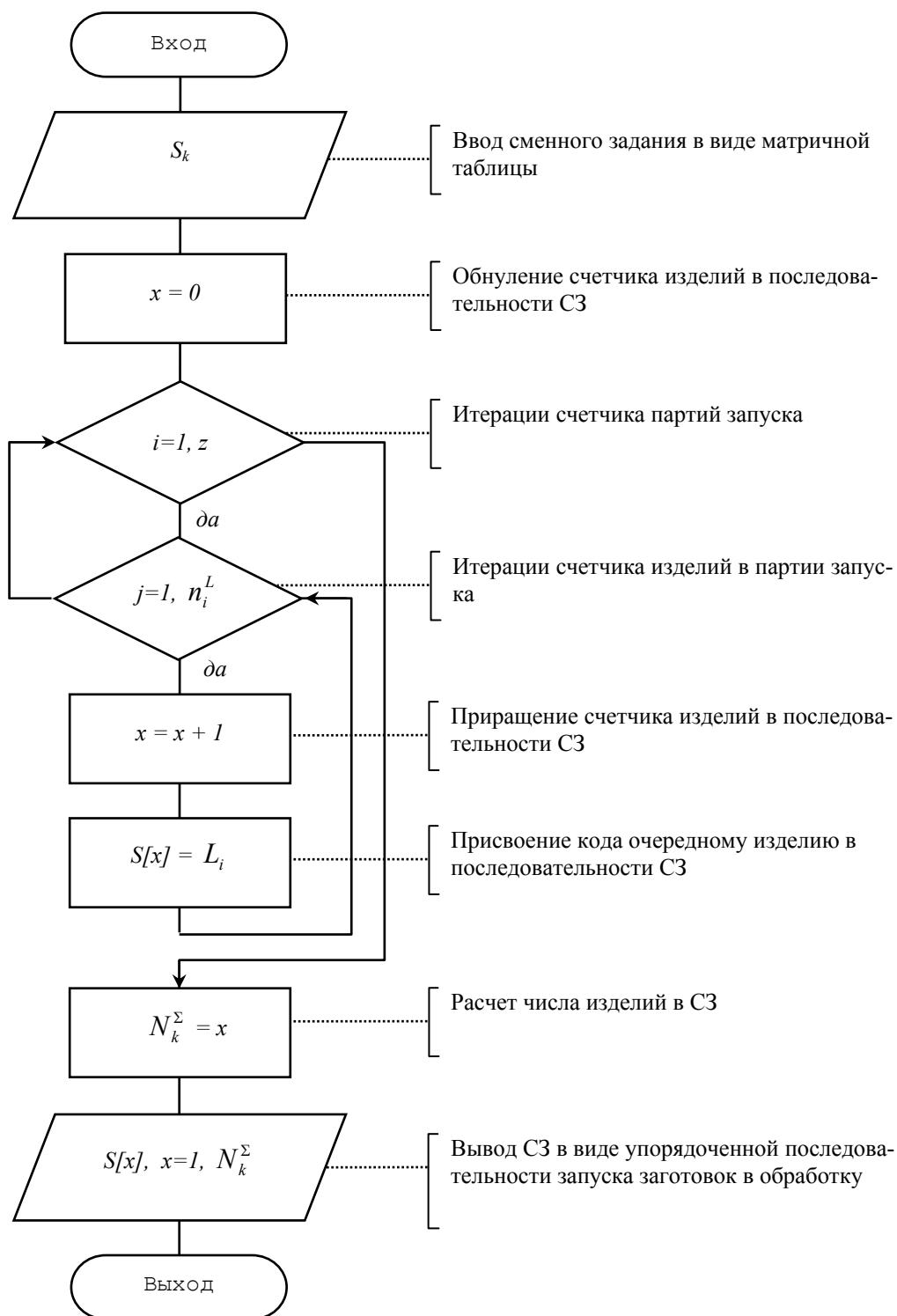


Рисунок 5.2 – Алгоритм процедуры **POSLED_ZAP**

Таблица 5.2 – Входные и выходные данные процедуры формирования очереди запуска заготовок в обработку **POSLED_ZAP**

Идентификатор	Тип идентификатора	Обозначение на схеме алгоритма	Смысловое значение идентификатора
POSLED_ZAP	Procedure	алгоритм процедуры на рисунке 5.2	преобразование сменного задания из табличной формы диалогового ввода в последовательность (очередь) запуска в обработку
i, j, k	integer	i, j	переменные цикла
Sum	integer	N_k^{Σ}	число заготовок в сменном задании
Входные данные			
PZ	integer	Z	число партий запуска в сменном задании
Kol	array[1..PZ] of integer	n_i^L	число изделий в каждой из z партий запуска
Kod	array [1..PZ] of integer	L_i	код наименования изделия i -ой партии запуска
Выходные результаты			
Posled	array[1..Sum] of integer	$S[x]$	код изделия на x месте в очереди запуска

Для автоматизации ввода и преобразования сменного задания в очередь запуска разработано приложение в среде Delphi (рисунок 5.3). Программный код процедуры представлен в соответствии с рисунком 5.4.

Преобразование сменного задания в очередь запуска

Число партий запуска, шт: 5 [Принять]

Номер партии запуска	1	2	3	4	5
Код партии запуска	1	8	1	10	10
Размер партии запуска	2	3	4	4	3

Получить очередь запуска заготовок в обработку

Порядковые номера	1	2	3	4	5	6	7
Коды заготовок	1	1	8	8	8	1	1

Рисунок 5.3 – Экранная форма процедуры ввода и преобразования сменного задания

```

procedure TForm1.Button1Click(Sender: TObject);
Var i: integer;
begin
  Randomize;
  With StringGrid1 do
    begin
      ColCount:=StrToInt(Edit1.Text)+1;
      ColWidths[0]:=150;
      Cells[0,0]:=' Номер партии запуска';
      Cells[0,1]:=' Код партии запуска';
      Cells[0,2]:=' Размер партии запуска';
      For i:=1 to ColCount-1 do Cells[i,0]:=IntToStr(i);
      For i:=1 to ColCount-1 do
        Cells[i,1]:=IntToStr(1+Random(10)); // случайный ввод партии запуска
      For i:=1 to ColCount-1 do
        Cells[i,2]:=IntToStr(1+Random(5)); // случайный ввод размера партии запуска
      end; {With}
    end; {Button 1}

procedure TForm1.Button2Click(Sender: TObject);
Var
  Sum, PZ, i, j, k: integer;
  Kol, Kod: array[1..50] of integer;
  Posled: array[1..500] of integer;
begin
  PZ:=StrToInt(Edit1.Text); // PZ - число партий запуска (ПЗ)
  Sum:=0;
  With StringGrid2 do
    begin
      For i:=1 to PZ do
        begin
          Kod[i]:=StrToInt(StringGrid1.Cells[i,1]); // Kod[i] - код i-ой ПЗ
          Kol[i]:=StrToInt(StringGrid1.Cells[i,2]); // Kol[i] - число заготовок в ПЗ
          Sum:=Sum+Kol[i]; // Sum - общее число заготовок в сменном задании
        end;
        StringGrid2.ColCount:=Sum+1;
        ColWidths[0]:=150;
        Cells[0,0]:=' Порядковые номера ';
        Cells[0,1]:=' Коды заготовок';
        For i:=1 to Sum do Cells[i,0]:=IntToStr(i);
        k:=0;
        For i:=1 to PZ do
          For j:=1 to Kol[i] do
            begin
              k:=k+1;
              Posled[k]:=Kod[i]; // Posled[k] - код k-ой заготовки в очереди запуска
              Cells[k,1]:=IntToStr(Posled[k]);
            end;
          end; {With}
        end; {Button 2}

```

Рисунок 5.4 – Программный код процедуры ввода и преобразования сменного задания

После того, как i -ая заготовка запущена в обработку (выполнена транспортная операция по ее доставке к определенному станку), значение элемента $\text{Posled}[i]$ массива обнуляется. Таким образом, проверка наличия ДУ в очереди на доставку к станкам сводится к проверке наличия ненулевых значений в массиве Posled .

Следует отметить, что формируемая таким образом очередь заготовок в обработку может изменяться согласно задаваемым правилам диспетчирования, о чем речь пойдет ниже.

5.1.2 Формализованное описание технологических процессов

Изготовление изделия каждого наименования $a_i \in \{1, 2, \dots, i, \dots, A\}$ из номенклатуры A , закрепленной за данной производственной системой, выполняется по определенному технологическому процессу $P(a_i)$. В общем случае технологический процесс представляет собой упорядоченную последовательность и содержание технологических операций на определенном наборе групп оборудования. В зависимости от принятого в модели уровня детализации функционирования ГПЯ технологический процесс может рассматриваться на уровнях технологической операции или технологического перехода.

На уровне технологической операции каждый технологический процесс $P(a_i)$ может быть представлен в виде матрицы $3 \times M$

$$P(a_i) = \begin{bmatrix} m_1 & \dots & m_M \\ r_{a1} & \dots & r_{aM} \\ t_{a1} & \dots & t_{aM} \end{bmatrix},$$

где M - множество групп технологического оборудования, на котором выполняются технологические операции: $M = (m_1, m_2, \dots, m_j, \dots, m_M)$;

m_j - модель оборудования j -ой технологической группы. В каждую группу m_j входит множество единиц технологического оборудования (рабочих мест) $N(m_j) = (n_1, n_2, \dots, n_N)$;

$r_{i=1,M}$ - ранг (приоритет) технологической операции, выполняемой на соответствующей группе оборудования. $r_{i=1,M} = 0$ означает отсутствие операции. Операция с рангом $r_{i=1,M} = 2$ выполняется раньше операции с рангом $= 1$. Наличие двух и более операций с одинаковым рангом означает любую очередность их выполнения. В этом случае очередь может быть установлена по дополнительным условиям (например, первой выполняется та операция, для которой имеется свободное рабочее место);

$t_{i=1,M}$ - время выполнения операции, с.

Столбец из трех параметров (m, r, t) описывает наличие операции в технологическом процессе, ее место в очереди операций и трудоемкость выполнения.

Таким образом, матрица $P(a_i)$ по критериям технологии упорядочивает маршрут прохождения изделия a_i через множество групп технологического оборудования $M = (m_1, m_2, \dots, m_j, \dots, m_M)$.

Выбор конкретного рабочего места $(n_1, n_2, \dots, n_N)$ внутри технологической группы $N(m_j) = (n_1, n_2, \dots, n_N)$ обеспечивается в режиме реального времени путем описания логики работы оборудования в алгоритме моделирования. Схема последовательности выпуска изделий изображена на рисунке 5.5.

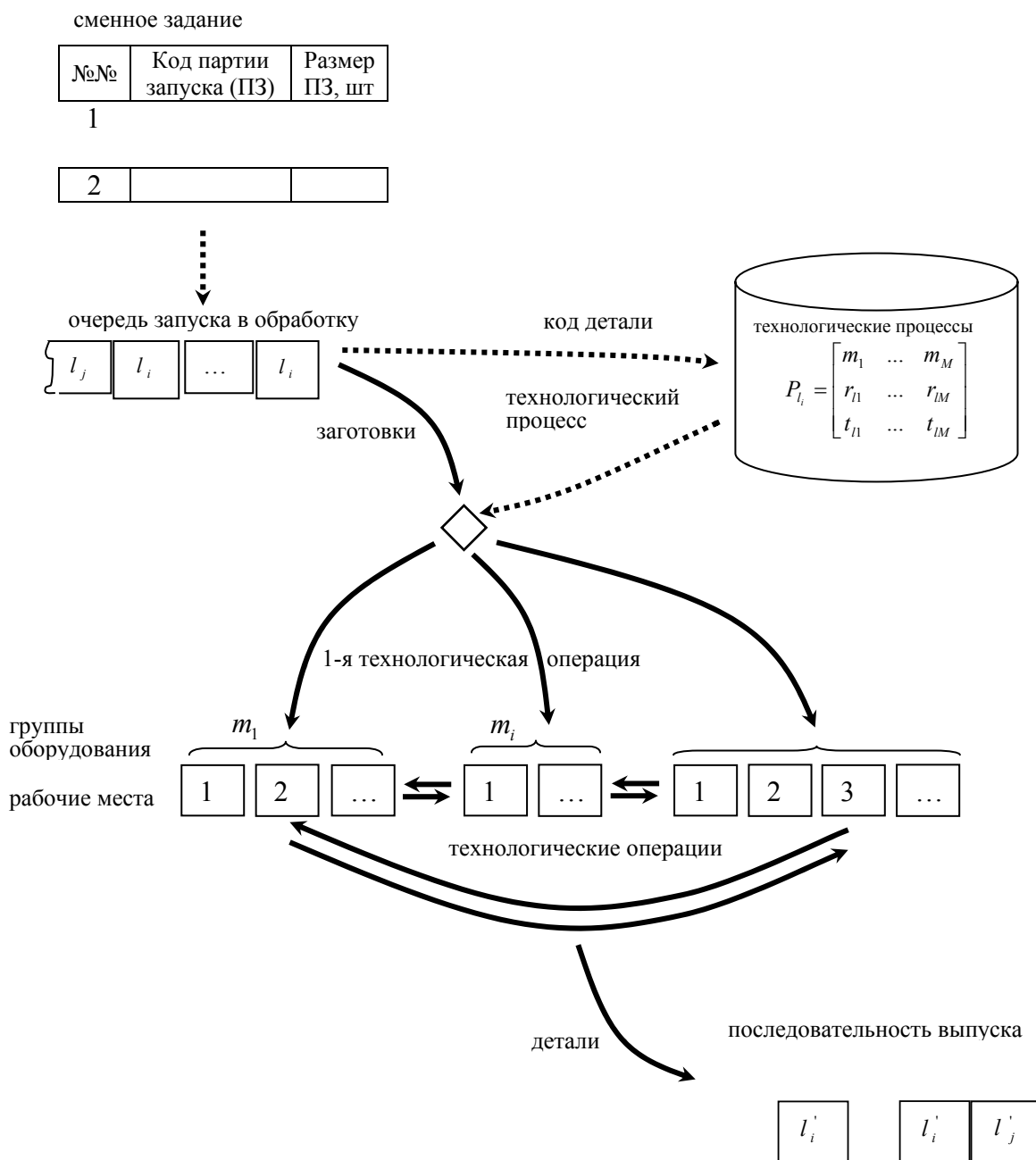


Рисунок 5.5 – Схема формирования последовательности выпуска изделий

5.1.3 Исходное состояние системы

Под исходным состоянием системы будем понимать ответы на два вопроса: а) где находится транспортное средство (ТС) перед началом цикла моделирования; б) каким образом заготовки размещены в позициях склада.

Исходное местонахождение транспортного средства сказывается лишь на длительности первой транспортной операции, а таких операций за цикл моделирования могут быть сотни. Поскольку в этом случае влияние обсуждаемой величины оказывается пренебрежимо мало, то для уменьшения числа исходных данных примем, что ТС всегда находится возле 1-го станка системы.

Способ размещения заготовок в позициях склада может оказать существенное влияние на результаты моделирования. Поэтому рассмотрим данный вопрос более подробно.

Итак, пусть в ГПЯ имеется линейный накопитель паллет на 25 позиций. Пусть сменный задание данного состава представляет собой очередь запуска заготовок в обработку вида:

$$\overbrace{1, 1, 1, 1, 1}^{1\text{-я партия}}, \overbrace{2, 2, 2, 2, 2, 2}^{2\text{-я партия}}, \overbrace{3, 3, 3, 3, 3, 3, 3, 3}^{3\text{-я партия}}, \overbrace{4, 4, 4, 4}^{4\text{-я партия}}, \overbrace{5}^{5\text{-я партия}}.$$

Варианты возможного размещения заготовок в позициях накопителя приведены в таблице 5.3.

Таблица 5.3 – Возможные варианты исходного размещения заготовок в позициях накопителя

Номера ячеек склада																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Варианты исходного размещения заготовок в позициях склада																								
1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4	4	4	5	0	0
0	1	1	1	1	1	0	5	2	2	2	2	2	2	4	4	4	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	2	2	2	2	2	2	0	1	1	1	1	1	5	0	4	4	4
...																								

Очевидно, что число вариантов n размещения заготовок на складе, обусловленное вариантами перестановок партий запуска, составляет $m = m!$. Для рассматриваемого случая при $m=5$ $n=1 \times 2 \times 3 \times 4 \times 5 = 120$ вариантов – и это при условии последовательного размещения заготовок каждой партии запуска. При размещении заготовок в произвольном порядке число вариантов становится астрономическим.

Таким образом, даже в рамках одного состава сменного задания существует множество вариантов исходного размещения заготовок в позициях накопителя. Местонахождение заготовки в накопителе учитывается при расчете длительности транспортной операции, а, следовательно, отражается на результатах моделирования. Рассмотреть (промоделировать) все варианты размещения даже для одного сменного задания практически невозможно. Поэтому обычно ограничиваются лишь

теми вариантами размещения, которые можно упорядочить по определенным условиям. Тем самым можно формализовать размещение заготовок с помощью соответствующей программной процедуры.

Условиями упорядочивания последовательности размещения могут служить:

а) рандомизация размещения с помощью генератора случайных чисел (процедуры random и randomize на языке Паскаль);

б) упорядочивание размещения по партиям запуска. Данное условие, в свою очередь, может выполняться:

- 1) в порядке возрастания или убывания номеров партий запуска;
- 2) в порядке возрастания/ убывания размеров партий запуска;
- 3) упорядочиванием в накопителе слева направо и справа налево;
- 4) упорядочиванием от центра склада;

в) в порядке, задаваемом пользователем в диалоговом режиме;

г) по минимальной длительности транспортной операции.

Блок-схема процедуры заполнения склада в случайной последовательности, представлена в соответствии с рисунком 5.6.

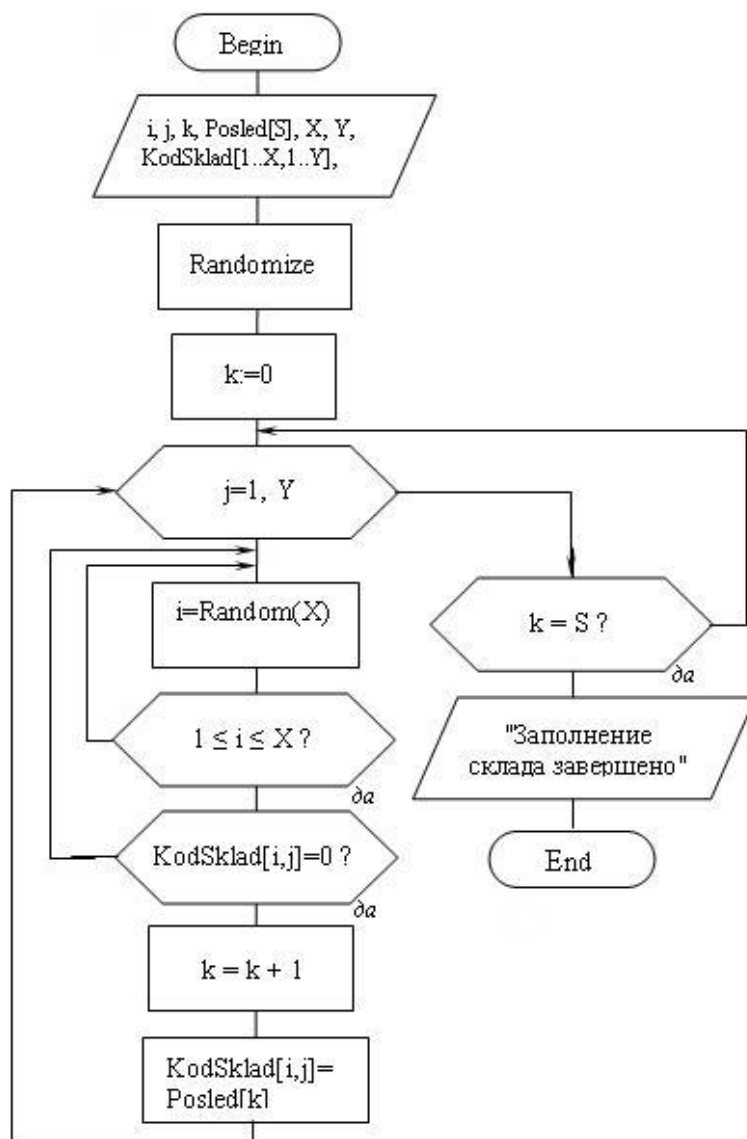
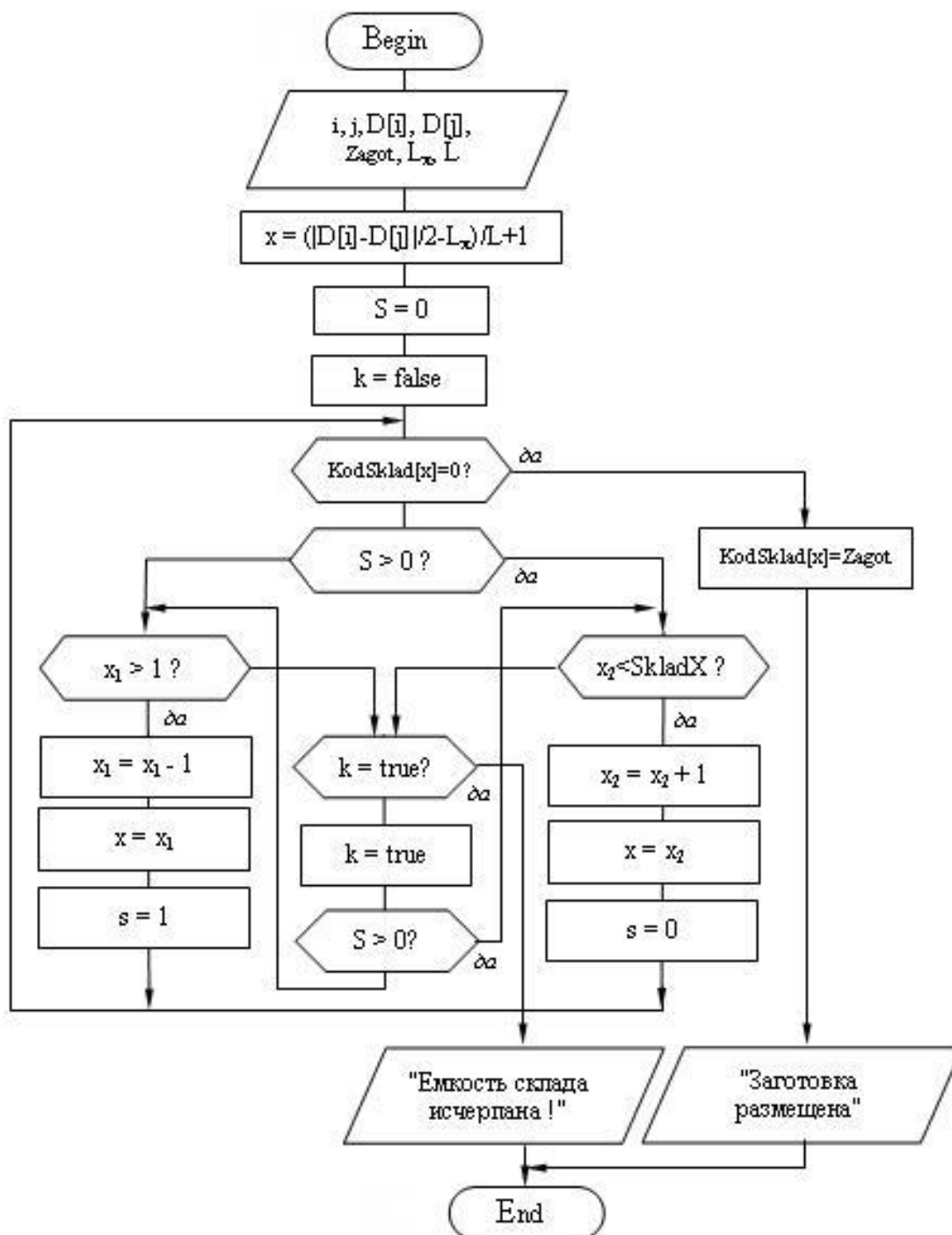


Рисунок 5.6 – Блок-схема процедуры заполнения многоярусного склада заготовками в случайной последовательности

Цель работы процедуры – разместить S заготовок из очереди сменного задания (одномерный массив данных $\text{Posled}[k]$, $k=1, S$) в позиции многоярусного накопителя (двухмерный массив $\text{KodSklad}[i, j]$, $i=1, X$, $j=1, Y$). X, Y – число позиций на одном ярусе и число ярусов склада, соответственно. Работа процедуры заканчивается после того, как все S заготовок «установлены» в ячейки склада. При использовании в ГПЯ линейного накопителя паллет переменная Y получает значение, равное 1.



В данном случае с началом работы программы моделирования массив переменных, содержащий коды заготовок в ячейках склада `KodSklad[x]` ($x=1, SkladX$), обнуляется. Заполнение массива генерируется таким образом, чтобы обеспечить появление нужных заготовок в тех позициях, время доставки из которых к обслуживаемому станку минимально.

В процедуре организован поиск пустой позиции в складе, начиная с позиции на отрезке склада между станками i и j : нахождение нужной позиции в данном диапазоне не увеличивает длительность транспортной операции. Вызов процедуры должен осуществляться перед каждой транспортной операцией.

5.1.4 Правила обслуживания станков

Взаимодействие станков и транспортного средства организуется процедурой WYBOR (рисунок 5.1), которая возвращает номер обслуживаемого станка.

В момент завершения очередной транспортной операции анализируется очередь запуска заготовок в обработку (массив `Posled`), из которой выбирается очередная заготовка.

Затем определяется позиция местонахождения заготовки на складе.

Далее составляется альтернативный список станков, сформированный процедурой `SIGNAL`, от которых поступили заявки на обслуживание (элементы массива `Zapros[stanok]=true`). Для каждой заявки в массив `Moment[stanok]` записывается время ее поступления. Принято, что заявка формируется в момент завершения обработки очередной заготовки в рабочей зоне.

Следующий шаг – выбор станка `pribyt`, на который нужно доставить данную заготовку. Рассмотрим возможные способы формализации выбора обслуживаемого станка с учетом назначенных приоритетных правил. В соответствии с разделом 1.4, в данном случае могут использоваться следующие правила:

- 1) с первой заявки в очереди (FIFO - First Input - First Output);
- 2) с последней заявки в очереди (LIFO - Last Input - First Output);
- 3) по приоритетам станков;
- 4) по минимальной трудоемкости заготовок в накопителе (TWK - Total Work);
- 5) по минимальному остаточному времени в рабочей зоне (LWKR- Least Work Remaining);
- 6) по мин. числу заготовок в накопителе (NXQL - Next Queue Length);
- 7) по мин. времени транспортной операции (SPT - shortest processing-time);

Рассмотрим особенности формализованного описания перечисленных правил обслуживания.

5.1.4.1 Правило «с первой заявки в очереди»

В этом случае из массива `Zapros[i]` выбирается заявка от станка с самым ранним временем поступления (`Moment[i]=min`). Блок-схема процедуры представлена на рисунок 5.8, текст процедуры на языке Pascal приведен ниже:

```

Procedure Prawilo_1(Var pribyt: integer);
  Var
    sum: array[1..stanki] of real;
    spisok: array[1..stanki] of integer;
    min: real;
  stanok, stankov: integer;
begin
  min:=WreOtr;
  for stanok:=1 to Stanki do
    begin
      spisok[stanok]:=0;
      If (Zapros[stanok] AND (Moment[stanok]< min then
        min:=Moment[stanok];
      end;
  stankov:=0;
  for stanok:=1 to stanki do
    If Moment[stanok]=min then
      begin
        stankov:=stankov+1;
        spisok[stanok]:=1;
        pribyt:=stanok;
      end;
  If stankov=1 then Exit;
  If stankov>1 then Dopoln(spisok)
  else begin ShowMessage('Ошибка в правиле_1'); Halt; end;
end; {Prawilo_1}

```

Достоинство данного правила в том, что обслуживание станков организуется в той последовательности, в которой они заканчивают обработку. Тем самым обеспечивается поочередное обслуживание всех станков системы.

Однако в момент начала работы ГПЯ заявки на обслуживание станков выдаются в момент времени $Moment[stanok]=0$. В данной ситуации правило не обеспечивает однозначный выбор станка. Выбор станка можно осуществить по дополнительному правилу, например, по времени транспортной операции либо по приоритету станка. Выбор дополнительного правила процедурой `Dopoln` рассмотрим позже.

К недостаткам данного правила относятся следующие:

а) момент времени появления заявки $Moment[stanok]$ не учитывает заполнение пристаночных накопителей (ПН): возможны ситуации, когда транспортное средство простаивает при наличии нескольких свободных позиций в ПН станка, ожидая момента завершения обработки в рабочей зоне;

б) невозможно обеспечить первоочередное обслуживание станков, имеющих, например, более высокую стоимость станкочаса работы.

Разновидностью данного правила является обслуживание станков с последней заявки в очереди. В этом случае из массива $Zapros[i]$ выбирается заявка от станка с самым поздним временем поступления ($Moment[i]=max$).

Данное правило не имеет логически объясняемых преимуществ, а его влияние на эффективность работы системы будет установлена на основе машинных экспериментов.

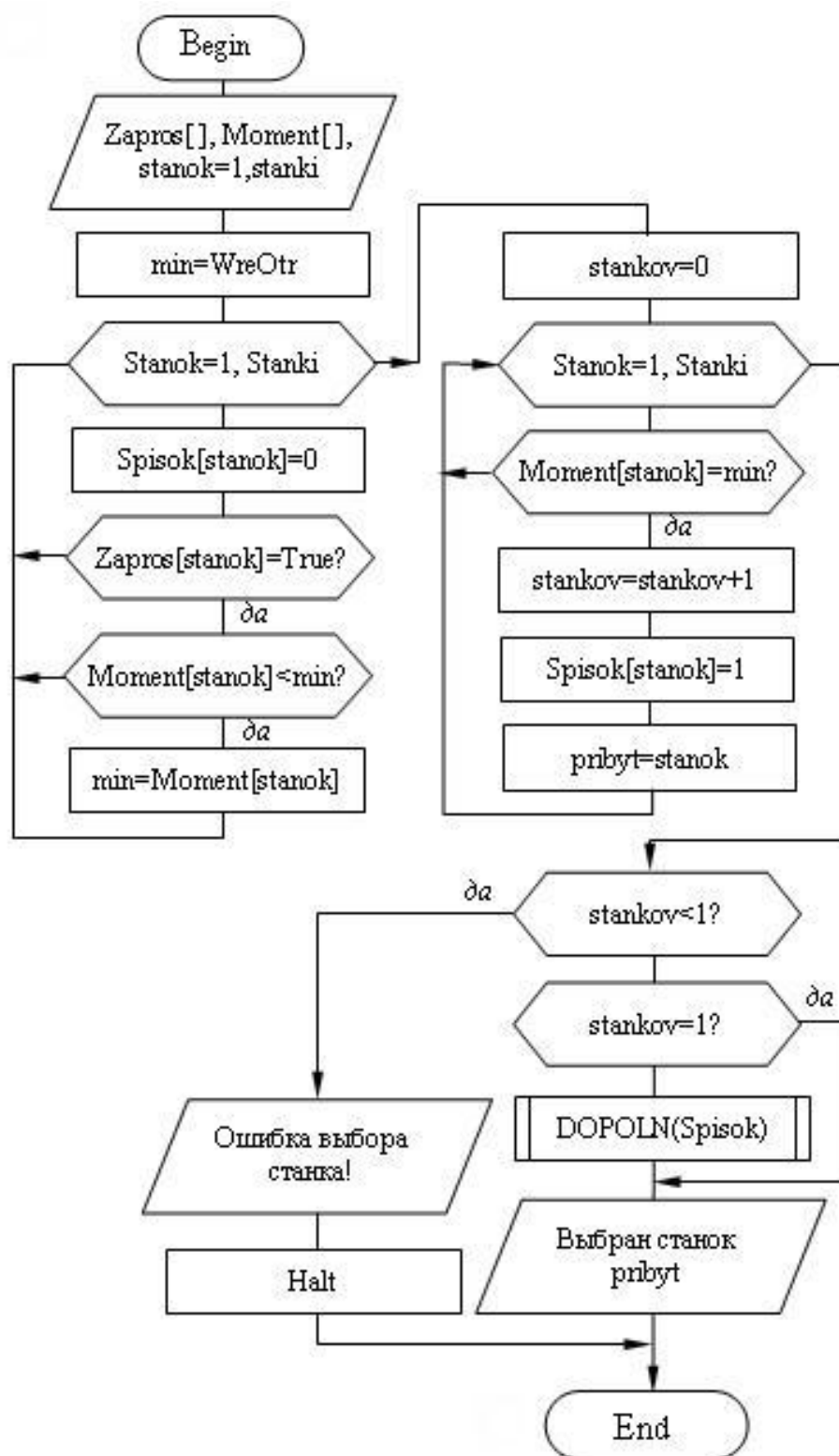


Рисунок 5.8 – Блок-схема процедуры выбора обслуживаемого станка по времени поступления заявок

5.1.4.2 Правило «по приоритетам станков»

В этом случае из массива $Zapros[i]$ $i=1, Stanki$ в соответствии с рисунком 5.9 выбирается заявка от станка с самым высоким приоритетом ($Prior[stanok] := \max$).

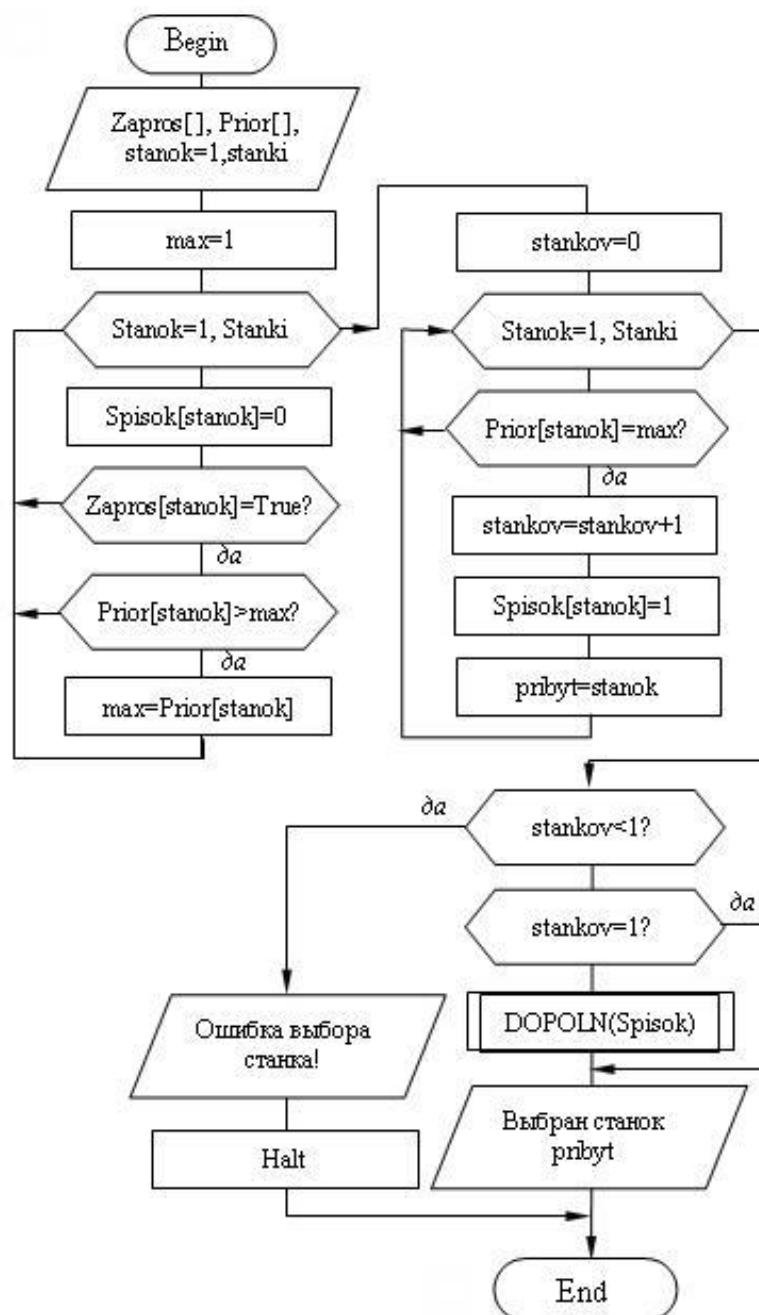


Рисунок 5.9 – Блок-схема процедуры выбора обслуживаемого станка по приоритетам станков

Достоинством правила является внеочередное обслуживание станков, для которых пользователем заданы наивысшие значения параметра $Prior[stanok]$. Например, для станков, имеющих более высокую стоимость станко-часа или лимитирующих производительность всей системы.

При наличии нескольких станков с одинаковым приоритетом дополнительно может устанавливаться, например, обслуживание по времени поступления заявок.

Недостаток правила в том, что возможны ситуации, когда транспортное средство будет успевать обслуживать лишь один или несколько станков с максимальным приоритетом, в то время как другие станки будут простаивать. Общая эффективность системы может снизиться.

5.1.4.3 Правило «по минимальной трудоемкости заготовок в накопителе»

Из списка на обслуживание выбирается станок, суммарное время обработки заготовок в накопителе которого минимально (рисунок 5.10).

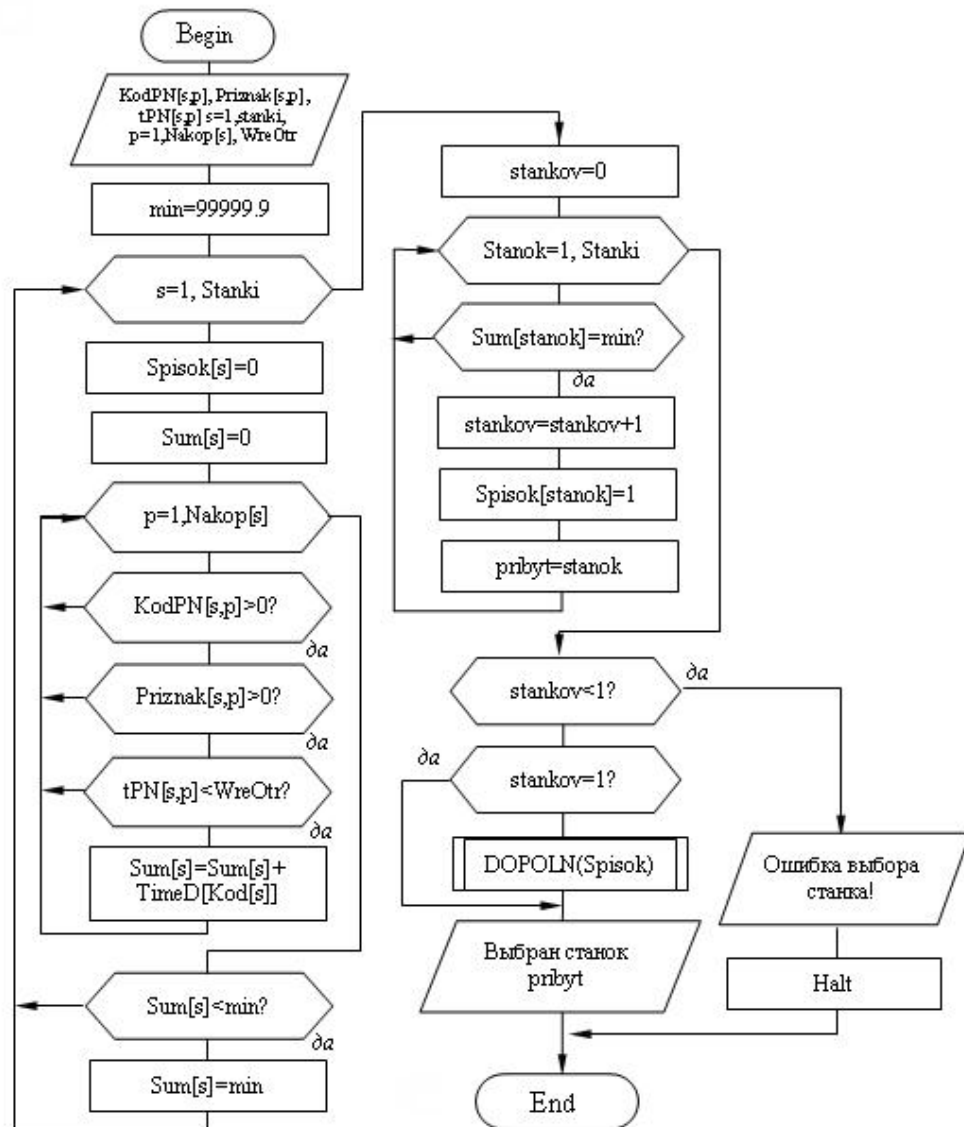


Рисунок 5.10 – Блок-схема процедуры выбора обслуживаемого станка по суммарной трудоемкости заготовок в накопителе

Правило позволяет обеспечить равномерную загрузку станков, однако в начальный момент времени работы ГПЯ при полностью пустых пристаночных накопителях ($\text{sum}[\text{stanok}] := 0$) однозначного определения обслуживаемого станка не произойдет. Будет сформирован альтернативный список $\text{spisok}[\text{stanki}]$, выбор из которого потребует использовать дополнительное правило (вызов процедуры Dopoln).

Разновидностью правила может служить учет суммарного времени заготовок в накопителе и остаточного времени обработки в рабочей зоне. Легко формализуется выбор станка по наибольшему суммарному времени обработки ($\text{sum}[\text{stanok}] := \max$), однако преимущества данного правила можно выявить лишь в процессе моделирования.

5.1.4.4 Правило «по min остаточному времени обработки заготовок в рабочей зоне»

Первым обслуживается станок s из списка $Spisok$, в рабочей зоне которого остаточное время обработки $Ostatok[s]$ минимально (рисунок 5.11).

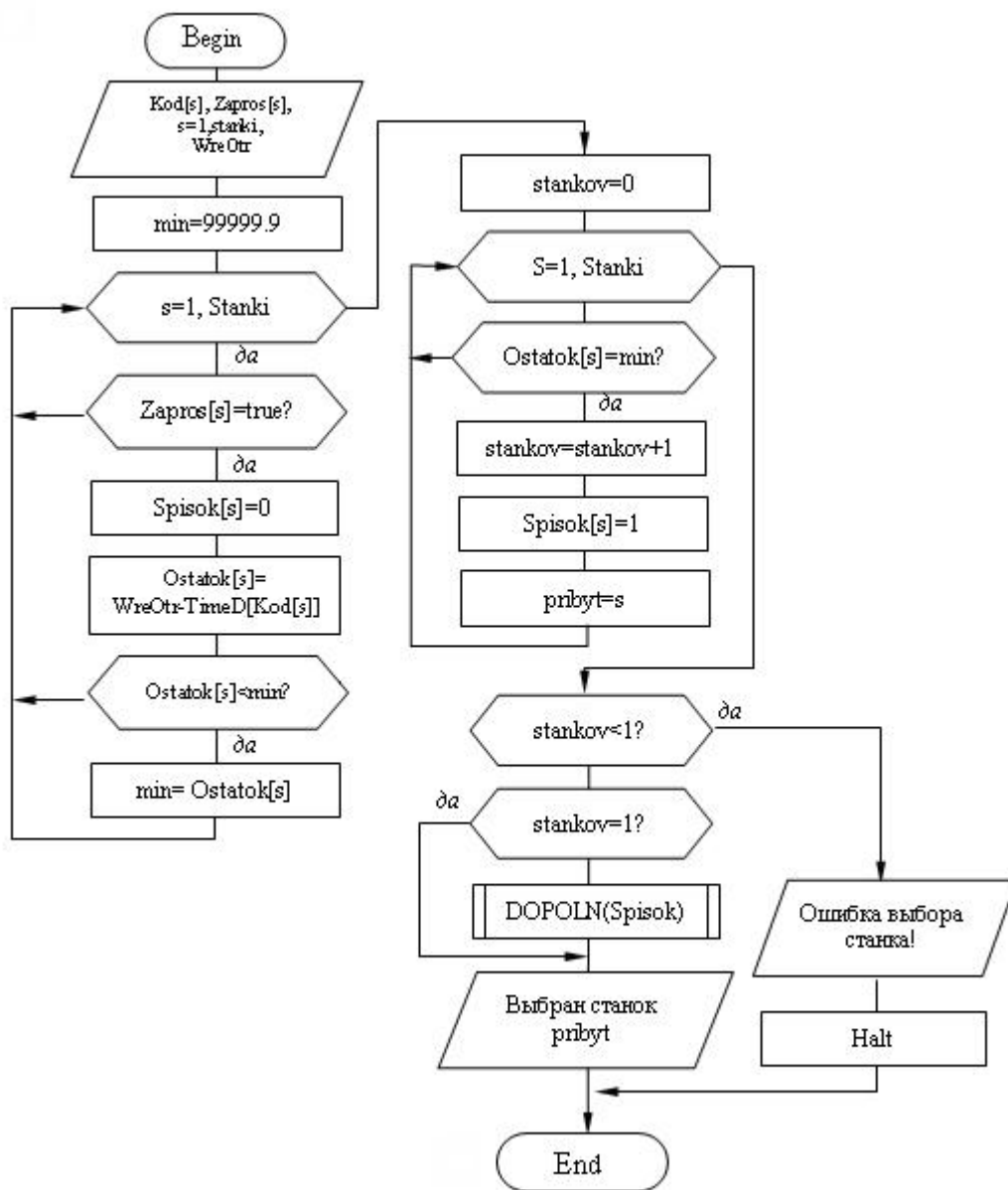


Рисунок 5.11 – Блок-схема процедуры выбора обслуживаемого станка по остаточному времени обработки в рабочей зоне

Априорно можно предположить, что данное правило позволит сократить простой станков за счет более своевременной доставки заготовок. Разновидностью данного правила может служить выбор станка по max остаточному времени обработки, однако эффективность данного выбора можно выявить лишь в результате моделирования.

5.1.4.5 Правило «по min числу заготовок в пристаночном накопителе»

Первым обслуживается станок, имеющий минимальное количество заготовок в накопителе (рисунок 5.12). В момент начала работы ГПЯ, когда на всех станках нет заготовок, правило не определяет единственный станок, что требует использовать дополнительное условие. Разновидностью правила может служить выбор станка по max числу заготовок в пристаночном накопителе.

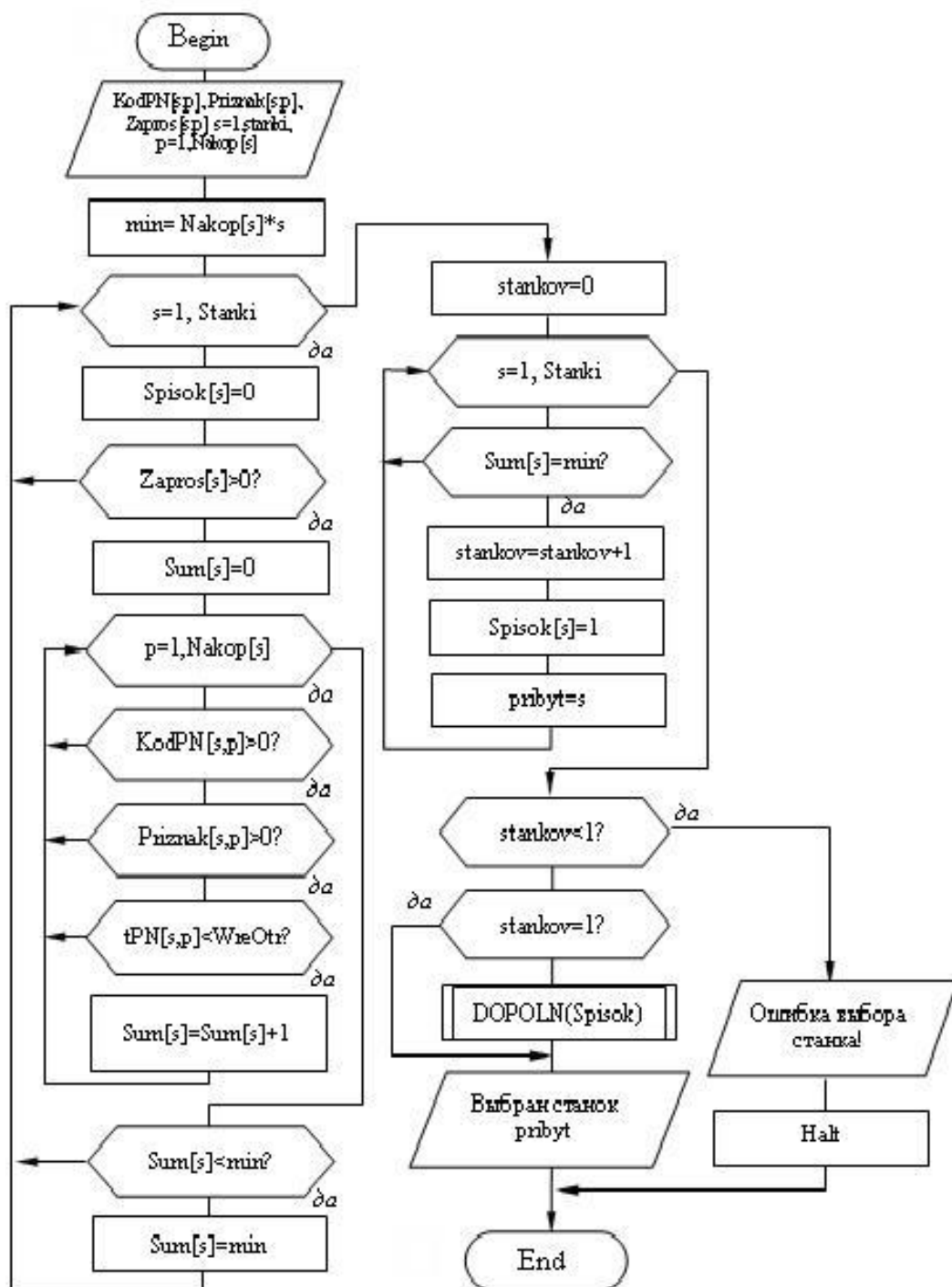


Рисунок 5.12 – Блок-схема процедуры выбора обслуживаемого станка по числу заготовок в пристаночном накопителе

5.1.4.6 Правило «по минимальному времени транспортной операции»

Первым обслуживается станок, время выполнения транспортной операции для которого окажется минимальным. Вначале для каждого из списка станков, подавших заявку на обслуживание, рассчитывается возможная длительность транспортной операции (процедура TRANSP). Затем выбирается наименьшая длительность транспортной операции и соответствующий станок (рисунок 5.13).

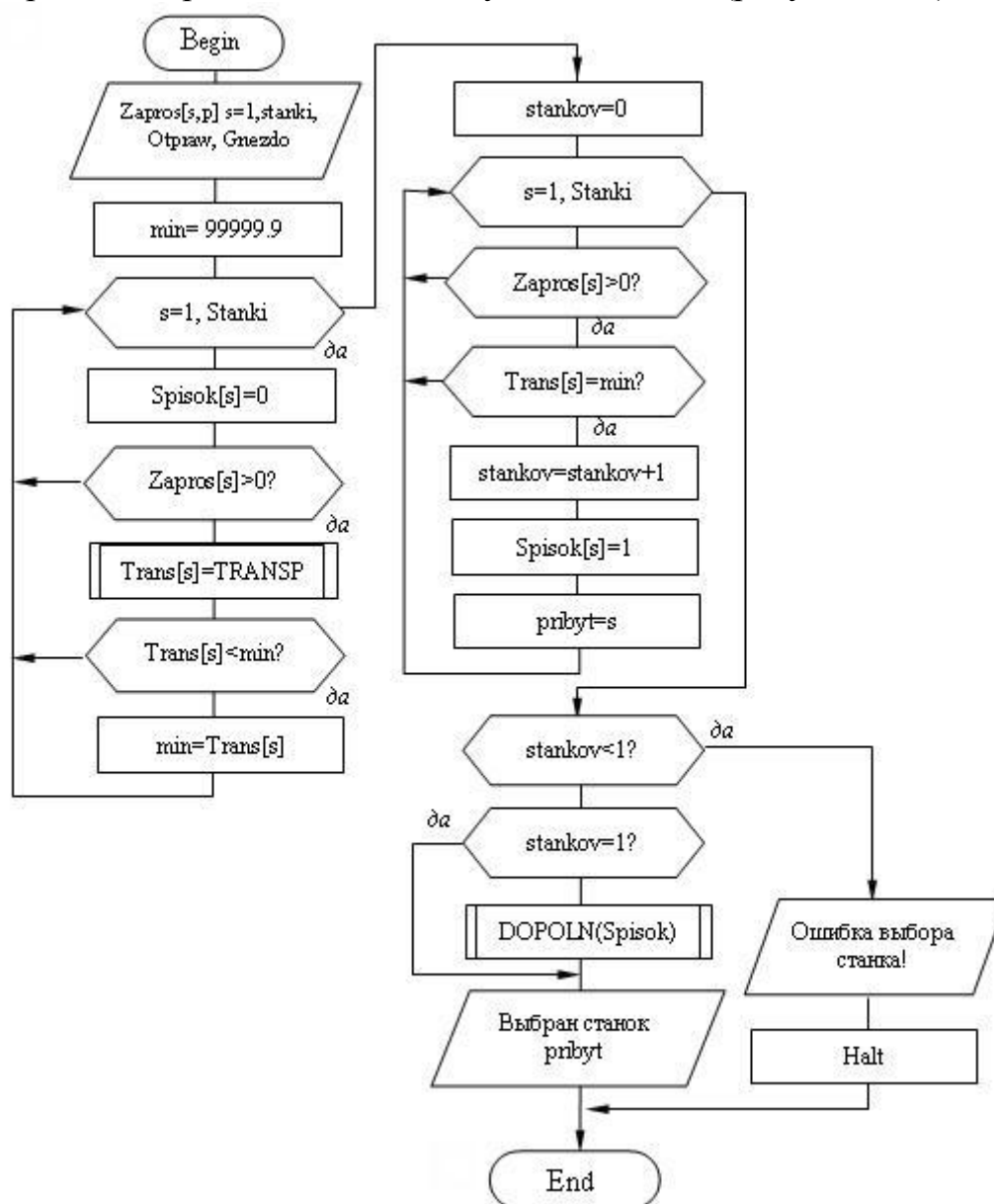


Рисунок 5.13 – Блок-схема процедуры выбора обслуживаемого станка по $\min$ длительности транспортной операции

Использование правила призвано свести к минимуму потери времени при доставке заготовок к станкам. Однако правило не учитывает состояние станков: возможно, что будет выбран станок, в накопителе которого уже имеется достаточный запас заготовок. Возможны ситуации, когда транспортное средство несколько раз подряд обслужит один станок, в то время как другие простаивают. Следовательно, данное правило должно использоваться с каким-либо дополнительным.

Подводя итог сказанному, можно отметить, что:

- а) каждое из рассмотренных правил обслуживания соответствует определенной стратегии управления работой оборудования;
- б) каждое правило характеризуется собственным набором достоинств и недостатков, сказывающихся на работе производственной системы в целом;
- в) в начальный момент времени работы системы, когда весь запас заготовок расположен на складе, большинство правил не обеспечивают равномерную загрузку оборудования;
- г) в алгоритме управления работой системы необходим комбинированный выбор правил, учитывающих реальные ситуации, складывающиеся в процессе функционирования оборудования системы.

Комбинированный выбор правил обслуживания может осуществляться в соответствии с алгоритмом на рисунок 5.14.

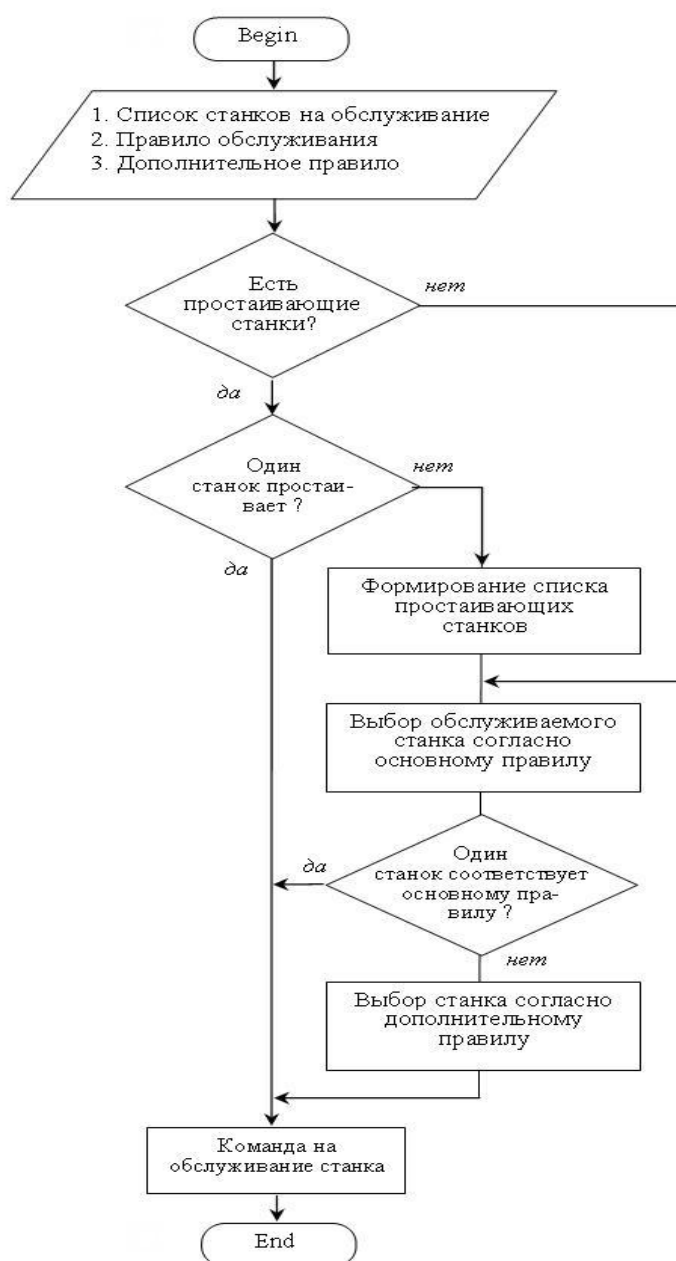


Рисунок 5.14 – Алгоритм процедуры комбинированного выбора правил обслуживания

Алгоритм учитывает следующие ситуации на момент выбора заявки на обслуживание: нет простаивающих станков; простаивает один станок; простаивают несколько станков.

При отсутствии простаивающих станков выбор заявки производится по выбранному правилу. В ситуации, когда заданному правилу удовлетворяют несколько станков (например, для правил 3, 6), составляется альтернативный список станков, из которого выбор производится по дополнительному правилу.

В качестве дополнительных назначаются правила, исключающие риск появления двух одинаковых заявок: например, правила 1, 4, 5.

При наличии одного простаивающего станка безальтернативно выполняется его заявка. В ситуациях, когда простаивают несколько станков, что характерно для начального этапа функционирования системы, выбор по основному и дополнительному правилам производится из альтернативного списка простаивающих станков.

5.1.5 Аналитические модели длительности транспортных операций

Пусть перед началом выполнения заявки от i -го станка робокара находится возле j -го станка, а нужная заготовка обнаружена в позиции x склада (рисунок 5.15).

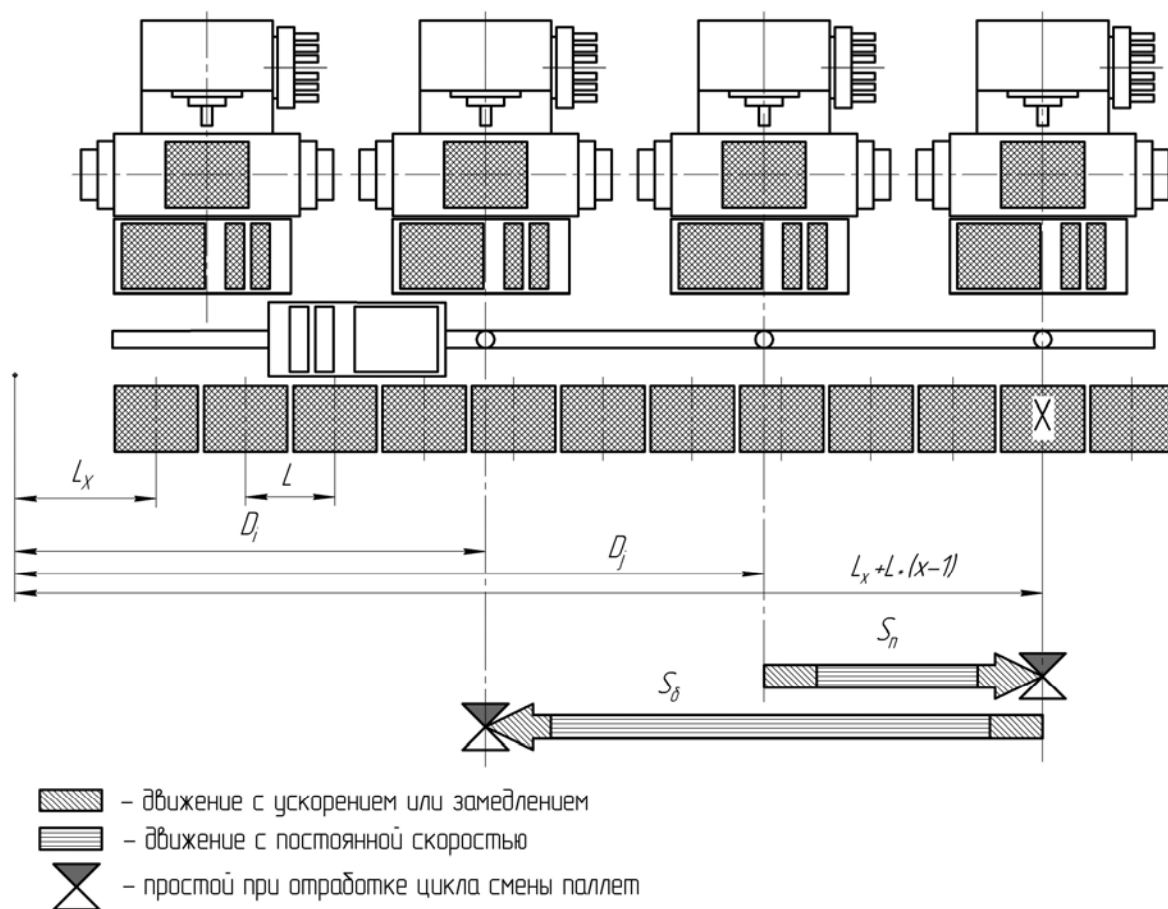


Рисунок 5.15 – Схема к выводу формулы расчета длительности транспортной операции

В одной позиции робокары может находиться паллета с деталью, удаленной после обработки с i -го станка. Цикл выполнения очередной заявки от станка складывается из следующих составляющих: а) перемещение робокары от j -го станка к позиции склада x (путь поиска S_n); б) отработка цикла загрузки/ выгрузки паллеты (время $t_{см}$); перемещение робокары к i -му станку (путь доставки S_δ), отработка цикла загрузки/ выгрузки со временем $t_{см}$.

Перемещение включает стадии разгона (путь S_p), перемещение с маршевой (постоянной) скоростью V и торможения (путь S_m). Математически время t_{mp} можно описать как

$$t_{mp} = \frac{(S_n - S_p - S_m + S_\delta - S_p - S_m)}{V} + 2 \times (t_p + t_m) + 2 \times t_{см}, \quad (5.5)$$

где t_p – время разгона робокары;

t_m – время торможения робокары;

V – маршевая скорость перемещения робокары.

Известно, что при равнопеременном движении без начальной скорости

$$V = a \times t \text{ и } S = \frac{V \times t}{2}, \text{ откуда } t = \frac{V}{a} \text{ и } t = \frac{2 \times S}{V}. \quad (5.6)$$

С учетом этого последовательно преобразуем выражение 1

$$t_{mp} = \frac{(S_n + S_\delta)}{V} - 2 \times \frac{(S_p + S_m)}{V} + 2 \times (t_p + t_m) + 2 \times t_{см}, \quad (5.7)$$

$$t_{mp} = \frac{(S_n + S_\delta)}{V} - (t_p + t_m) + 2 \times (t_p + t_m) + 2 \times t_{см}, \quad (5.8)$$

$$t_{mp} = \frac{(S_n + S_\delta)}{V} + (t_p + t_m) + 2 \times t_{см}, \quad (5.9)$$

$$t_{mp} = \frac{(S_n + S_\delta)}{V} + \left(\frac{V}{a_p} + \frac{V}{a_m} \right) + 2 \times t_{см}. \quad (5.10)$$

Примем, что ускорение при разгоне a_p и замедление при торможении a_m равны: $a_p = a_m = a$. Тогда

$$t_{mp} = \frac{(S_n + S_\delta)}{V} + 2 \times \frac{V}{a} + 2 \times t_{см} \quad (5.11)$$

Из расчетной схемы на рисунке (5.15) следует, что проходимые робокарой в процессе выполнения заявки расстояния можно выразить через координаты точек останова робокары на планировке ГПЯ

$$S_n = |D_j - L_x - L \times (x - 1)|, S_o = |D_i - L_x - L \times (x - 1)|, \quad (5.12)$$

где $i, j \in \{1, 2, \dots, C\}$ - номера станков;
 C - число станков ГПЯ;
 $x \in \{1, 2, \dots, N\}$ - номер позиции склада с нужной паллетой;
 N - число позиций на складе.

Окончательно выражение, описывающее связь между планировочными параметрами ГПЯ, техническими характеристиками робокары и временем выполнения очередной транспортной операции примет вид

$$t_{mp} = \frac{|D_j - L_x - L \times (x - 1)| + |D_i - L_x - L \times (x - 1)|}{v} + 2 \times \frac{v}{a} + 2 \times t_{cm} \quad (5.13)$$

В зависимости от текущих значений D_i, D_j и x время выполнения операции может рассеиваться в широких пределах. Границы интервала возможных значений t_{mp} можно определить как $t_{mp}^{\min} = 2 \times t_{cm}$,

$$\left| \begin{aligned} t_{mp}^{\max} &= 2 \times \frac{(D_c - L_x)}{v} + 2 \times \frac{v}{a} + 2 \times t_{cm}, \text{ если } (D_c - L_x) \geq (L_x + L \times (N - 1) - D_1), \\ t_{mp}^{\max} &= 2 \times \frac{|D_1 - L_x - L \times (N - 1)|}{v} + 2 \times \frac{v}{a} + 2 \times t_{cm}, \text{ если } (D_c - L_x) < (L_x + L \times (N - 1) - D_1). \end{aligned} \right. \quad (5.14)$$

Поле рассеяния возможных значений t_{mp} составит

$$\omega_t = \frac{2 \times (D_c - L_x)}{v} + 2 \times \frac{v}{a}. \quad (5.15)$$

Выражение (5.15) позволяет уже на стадии проектирования прогнозировать возможное рассеяние времени выполнения транспортной операции, в то время как выражения 3 позволяют рассчитать границы ее возможных значений. Полученные выражения могут быть использованы для аналитического расчета, например, интервала допустимых значений скорости робокары. Так, если при проектировании АТСС наибольшее время транспортной операции ограничено каким-то допустимым значением $t_{mp}^{\text{дон}}$, то можно записать

$$2 \times \frac{(D_c - L_x)}{v} + 2 \times \frac{v}{a} + 2 \times t_{cm} \leq t_{mp}^{don}. \quad (5.16)$$

Перепишав выражение относительно неизвестной скорости v , получим

$$\frac{v^2}{a} + t_{cm} * v - \left[\frac{t_{mp}^{don}}{2} - (D_c - L_x) \right] \leq 0 \quad (5.17)$$

Неравенство выполняется при следующих граничных значениях v

$$v_{\min} = \frac{-t_{cm} - \sqrt{t_{cm}^2 + (2 * t_{mp}^{don} - 4 * D_c - L_x) / a}}{2} \times a$$

$$v_{\max} = \frac{-t_{cm} + \sqrt{t_{cm}^2 + (2 * t_{mp}^{don} - 4 * D_c - L_x) / a}}{2} \times a \quad (5.18)$$

Выражения (5.18) позволяют аналитически рассчитать граничные значения интервала скорости робокары. Однако уточнить и проанализировать качество работы ГПЯ при различных значениях скорости из выявленного интервала можно с помощью имитационной модели, использующей выражение (5.12).

Выражения (5.18) раскрывают физический смысл утверждения о том, что есть критические скорости перемещения транспортного средства, выше и ниже которых эффективность использования ГПЯ падает (рисунок 5.16).

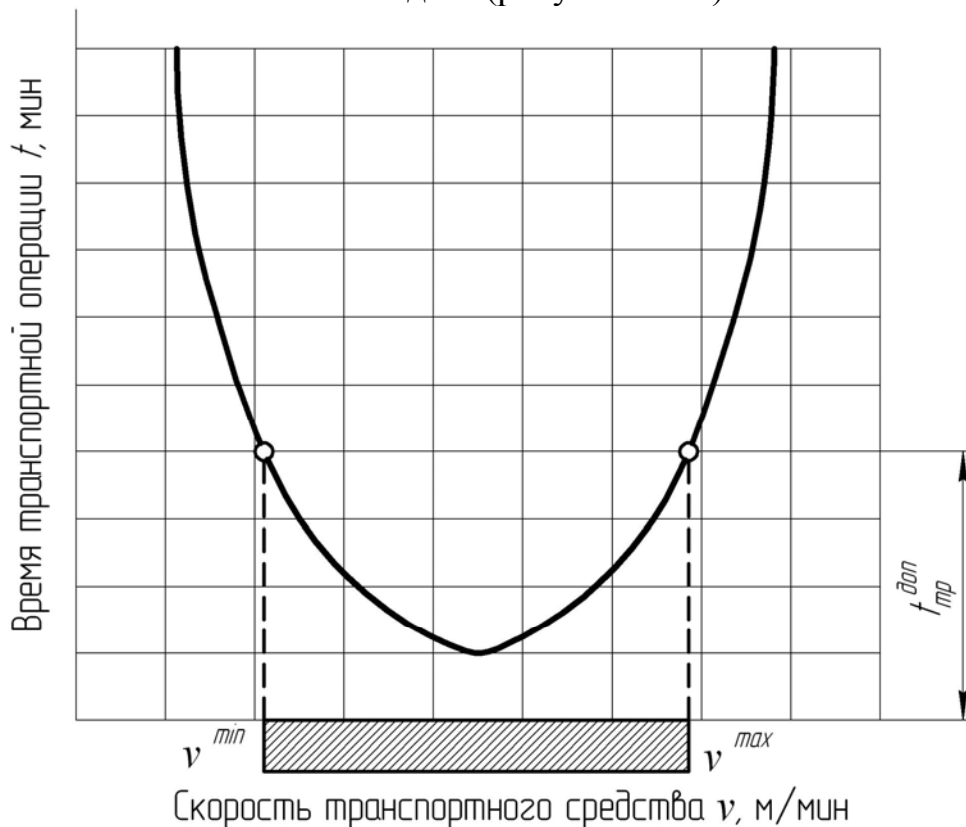
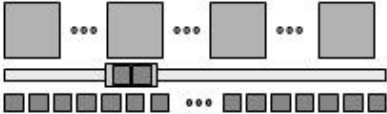
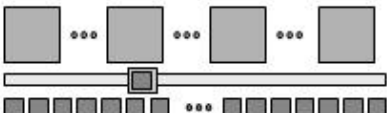
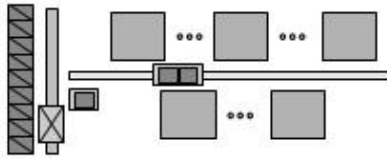
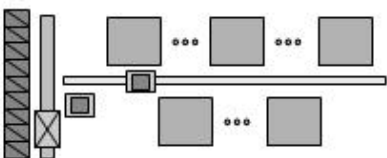
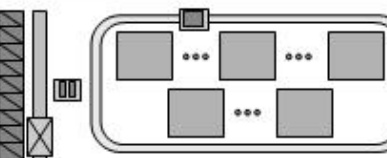


Рисунок 5.16 – Схема, иллюстрирующая выбор интервала допустимых значений скорости транспортного средства ГПЯ

Выражение (5.12) служит для описания длительности транспортной операции, выполняемой двухместной робокарой при линейном исполнении автоматизированного склада ГПЯ. Однако приведенная в разделе последовательность вывода формулы применима к различным вариантам размещения оборудования и используемым типам транспортных средств. В таблице 5.4 представлены конечные формулы для расчета длительности транспортных операций при различных вариантах построения транспортных систем ГПЯ.

Таблица 5.4 – Формулы для расчета времени транспортной операции

Расчетная схема ГПС	Расчетная формула
	$t_{\text{тп}} = \frac{ D_i - L \cdot x + D_j - L \cdot x }{v} + 2 \cdot \frac{v}{a} + 2 \cdot t_{\text{см}}$
	$t_{\text{тп}} = \frac{L \cdot x - y + D_i - L \cdot x + D_j - L \cdot x }{v} + 3 \cdot \frac{v}{a} + 4 \cdot t_{\text{см}}$
	$t_{\text{тп}} = \frac{D_i + D_j}{v} + 2 \cdot \frac{v}{a} + 2 \cdot t_{\text{см}}$
	$t_{\text{тп}} = \frac{D_i + D_j}{v} + 2 \cdot \frac{v}{a} + 4 \cdot t_{\text{см}}$
	$t_{\text{тп}} = \frac{S_4 - D_i + D_j}{v} + 2 \cdot \frac{v}{a} + 4 \cdot t_{\text{см}}$
 - однопозиционная тележка  - двухпозиционная тележка  - кран-штабелер  - одноярусный накопитель  - многоярусный накопитель  - траектория движения ТС	

5.2 Модели ГПЯ на уровне технологического перехода

5.2.1 Схема обобщенного алгоритма моделирования

Укрупненная схема алгоритма моделирования представлена в соответствии с рисунком 5.17. При моделировании на уровне технологического перехода ГПЯ с автоматическим складом режущих инструментов возмущениями, с частотой которых фиксируются состояния модулей, являются отрезки времени выполнения транспортных операций по доставке к станкам РИ. Если на участке станков имеется лишь транспортное устройство для передачи ДУ, а весь запас РИ хранится в ИМ станков, то приращение модельного времени производится по времени транспортирования ДУ. Таким образом, продолжительность моделируемого процесса (или модельное время) увеличивается скачкообразно из-за переменной частоты появления возмущений (приращений модельного времени). Принципиальным отличием модели ГПЯ на уровне технологического перехода служит учет работы многоцелевого станка на протяжении цикла выполнения технологической операции, состоящей из множества технологических переходов.

В качестве исходных данных при моделировании используется машинное основное и вспомогательное время технологических переходов по всей номенклатуре техпроцессов из состава текущего сменного задания, а также код и ресурс применяемых на переходах режущих инструментов.

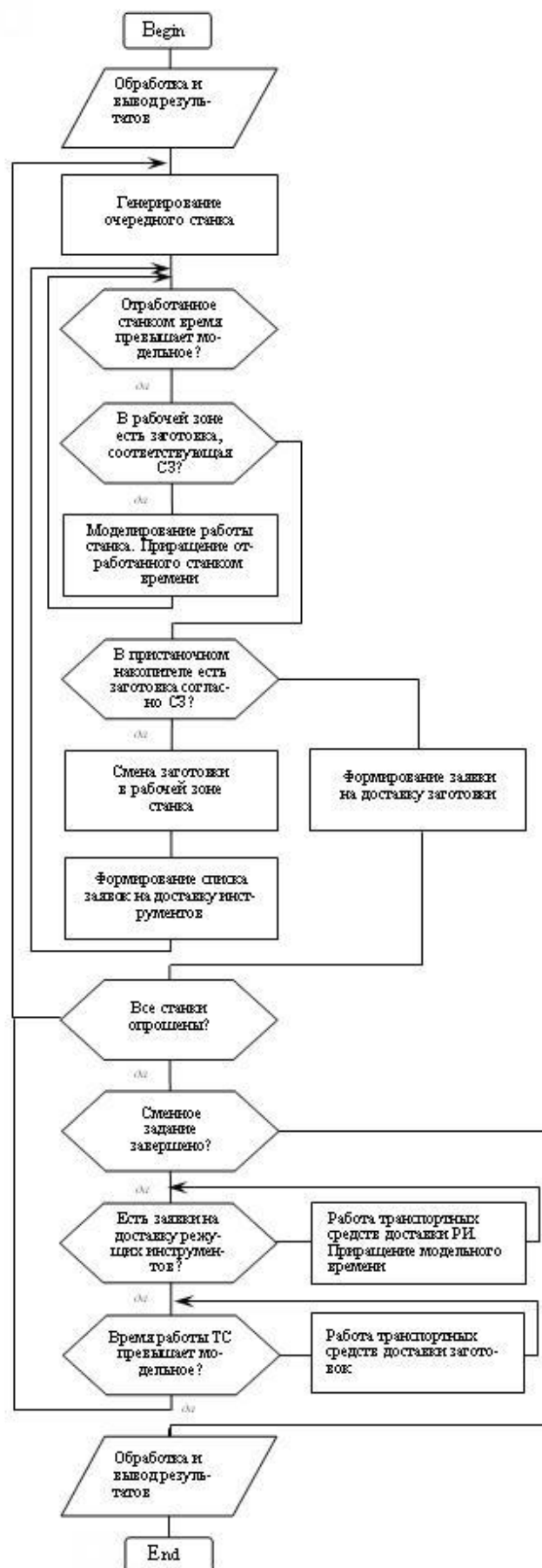


Рисунок 5.17 – Схема процедуры моделирования ГПЯ на уровне технологического перехода

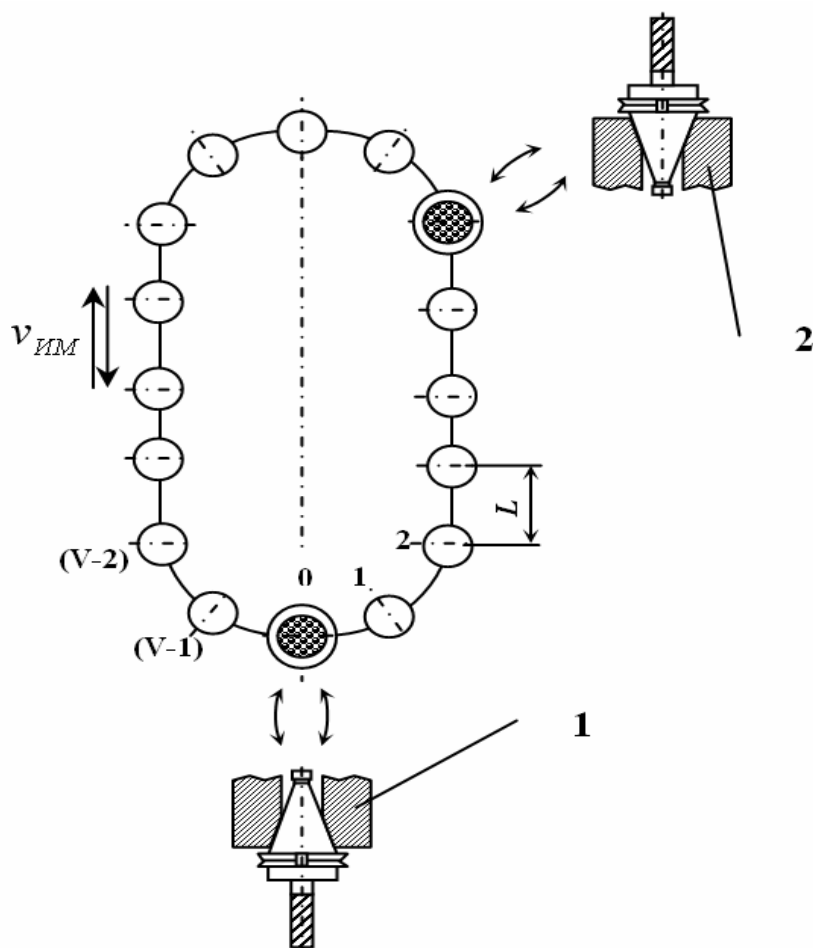
5.2.2 Схема алгоритма моделирования многоцелевого станка ГПЯ

В соответствии с рисунком 5.17, моделирование работы станка начинается при условии, что на его столе имеется заготовка из состава сменного задания. Моделирование включает генерирование очередных технологических переходов (т.е. выбор их из техпроцесса обрабатываемой заготовки), расчет затрат времени на подготовку режущих инструментов (РИ) в позиции смены, сравнение указанных затрат с длительностью выполняемых в данный момент технологических переходов и выявление таким образом неплановых внутриоперационных простоев станка. Данный подход позволил отразить в модели не только паспортные данные станка (время смены инструмента, скорость поворота магазина, время смены паллет и другие технические параметры), но и увязать их в едином алгоритме с технологическими параметрами: машинным временем выполнения переходов и соответствующей стойкостью применяемых режущих инструментов.

Каждый РИ в системе, описывается двумя параметрами - десятичным кодом и стойкостью. Десятичный код РИ – допускает одновременное использование на участке до 10 инструментов-дублеров и позволяет вести учет остаточного ресурса стойкости каждого из них. Кроме того, десятичный код РИ задается в технологической базе данных элементом двумерного массива $INSTR[i, j]$ (i - номер перехода, j - номер техпроцесса).

При нахождении РМ в центральном накопителе его код задается элементом одномерного массива $INSTR1[i]$. Наконец, РМ в гнездах инструментального магазина станка кодируется элементом двумерного массива $KODINS[gnezdo, stanok]$ ($gnezdo$ - текущий номер гнезда ИМ, $stanok$ - текущий номер станка участка), где $stanok = (1, 2, \dots, Stanki)$ ($Stanki$ - суммарное число станков на участке), $gnezdo = (1, 2, \dots, mkMag[stanok] + 2)$ ($EmkMag[stanok]$ - емкость ИМ станка).

Кроме того, каждому РИ специальной процедурой присваивается признак «не нужен в магазине данного станка» (массив логических переменных $UDALEN[gnezdo, stanok]$). Нумерация гнезд ИМ ведется, начиная со шпинделя станка и заканчивая перегрузочной позицией магазина (рисунок 5.18).



1 – шпиндель станка; 2 – позиция обмена инструментами с внешним транспортом
Рисунок 5.18 – Схема к разработке модели многоцелевого станка

5.2.3 Логика работы устройств АСИ и АЗИ станка

Если в ГПЯ реализована лишь автоматическая смена РИ в шпинделях станков, а замена их в магазинах производится вручную в периоды плановых остановов участка, то работа устройств АСИ (автооператора смены инструмента АО₁ и ИМ) осуществляется по достаточно простому алгоритму. Если же в ГПЯ реализуется автоматическая замена РИ в магазинах в процессе работы станков, то могут иметь место различные варианты управления очередностью отработки циклов смены и замены РИ (рисунок 5.19). Варианты появляются уже при решении вопроса о том, в каком случае РИ следует удалять из магазина: с одной стороны, отработавший, в шпинделе и удаленный в центральный склад РИ может быть использован на других станках участка, тем самым, сокращая общий расход РИ.

С другой, этот РИ вновь может потребоваться на данном станке, и снова потребуется доставлять его в магазин, дополнительно загружая транспортное средство и рискуя получить дополнительный простой станка из-за отсутствия РИ.

Рациональное решение может быть получено лишь в результате моделирования. В работе выделены и реализованы в виде меню алгоритмов системы моделирования три возможных условия удаления РИ из ИМ:

1) Удаление производится, если данный РИ больше не нужен на обрабатываемой ДУ. Условие имеет смысл, если распределение ДУ между станками ведется согласно общей очереди в сменном задании. При распределении ДУ по имеющемуся на станке комплекту РИ данное условие игнорируется.

2) РИ удаляется из ИМ только в том случае, когда он выработал положенный ресурс или поломан. Реализация данного условия может привести к увеличению количества РИ в ИМ при работе станка.

3) РИ удаляется только в том случае, если в ИМ нет пустого гнезда для установки доставленного в позицию смены инструмента. В данном случае вопрос о том, какой именно РИ удалять решается на основании первого условия и его разновидностей (РИ не нужен на всех последующих переходах данной ДУ, на 10 последующих переходах, на 3 и т.д.).

Другая группа вариантов появляется при решении вопроса об очередности отработки циклов смены и замены РИ. В случае если на станке используются устройства АСИ с отдельным циклом смены РИ, двухзахватный АО₁ с позицией ожидания, то цикл смены РИ в захвате АО₁ всегда предшествует отработке цикла замены. При совмещенном цикле смены замена РМ в магазине предшествует смене в шпинделе. Однако и в том, и в другом случаях остается вопрос: в какие моменты времени выполнять циклы замены в магазине РИ, доставленных транспортным средством АСИО в перегрузочную позицию станка? Возможные следующие варианты:

1) Цикл замены обрабатывается, если в перегрузочной позиции есть доставленный инструмента.

2) Замена осуществляется, если данный РИ потребовался в шпинделе. При данном варианте может блокироваться работа транспортного средства, так как бывает некуда выгружать доставленный инструмент.

3) Замена производится адаптивно, во время выполнения «длинных» технологических переходов, либо если РИ нужен в шпинделе. Алгоритмическую и программную реализацию данных вариантов управления циклами смены и замены рассмотрим в следующих разделах.

5.2.4 Формальное описание процесса моделирования

Процедура моделирования работы станка (рисунок 5.19) активизируется головной программой при выполнении условий:

а) на столе рассматриваемого станка имеется деталиустановка, соответствующая текущему составу сменного задания;

б) обработка ДУ не завершена;

в) отработанное станком время не превышает модельное время системы.

Моделирование начинается с идентификации ДУ на столе станка

$$DETAL=KDU [OBRZAG [stanok]],$$

где $stanok$ - порядковый номер опрашиваемого станка;
 $OBRZAG[stanok]$ - код ДУ;
 $DETAL$ - порядковый номер техпроцесса обработки ДУ с данным кодом
(из активной технологической базы данных).

Номер текущего перехода (перед которым завершилось моделирование на прошлом шаге итерации) хранится в переменной $PERH[stanok]$, количество переходов в активном техпроцессе - в переменной $FEREN[DETAL]$. Затем рассчитывается машинное время выполнения предшествующего технологического перехода, определяемое как сумма машинного основного и машинного вспомогательного (кроме связанного со сменой РИ) времени

$$PREPRH = T_{mash}[PERHOD-i, DETAL] + T_{wsp}[PERHOD-1, DETAL].$$

Далее начинается моделирование работы устройств АСИ и АЗИ станка. Рассмотрим вариант, когда на участке имеется автоматический склад РИ и реализована замена инструментов в ИМ станков (рисунок 5.18). Очередность отработки циклов смены и замены РИ зависит от того, отдельно или совместно обрабатывается на станке цикл смены РИ в шпинделе. При отдельной отработке ($АО_1$ с позицией ожидания) безусловный приоритет имеет подготовка и смена РИ. Если же $АО_1$ обрабатывает цикл удаления отработавшего и установки в шпиндель нового РИ одновременно, то решается вопрос об очередности выполнения циклов АСИ и АЗИ. Для этого выясняется, есть ли физическая возможность установки данного РИ в перегрузочную позицию ИМ. Такая возможность имеется, если в перегрузочном гнезде находится РИ, доставленный по заявке из внешнего склада, (его можно поставить в гнездо удаляемого РИ), либо перегрузочное гнездо свободно (при доставке РИ группами проверяется наличие пустого гнезда в перегрузочной позиции).

Если перегрузочная позиция ИМ занята удаляемым РИ, то начинается моделирование подготовки и смены РИ в шпинделе.

Если перегрузочная позиция свободна, то выясняется условие удаления отработавшего в шпинделе РИ из магазина станка. Возможно одно из трех условий, выбираемых пользователем из меню:

1) *«Исчерпан ресурс РИ или он сломан»*. Состояние поломки РИ может быть сгенерировано специальной процедурой, построенной на основе вероятностных законов распределения стойкости РИ. Поскольку же разрабатываемая система моделирования строится на прямом и точном соответствии производственному процессу, примем, что отсутствие поломки РИ гарантировано технологией обработки за счет выбора режимов резания и превентивной смены РИ по достижению ими щадящего времени наработки. Остаточный ресурс РИ хранится в переменной $RESURS(KODINS[gnezdo, zstanok1])$, где $KODINS[gnezdo, stanok]$ - десятичный код РИ, установленного в гнездо магазина с номером $gnezdo$ станка с номером $zstanok$.

В том случае, если остаточный ресурс РИ, выведенного из шпинделя, исчерпан, то моделируется процесс поворота ИМ для доставки гнезда с РИ из позиции смены в позицию замены и отработка цикла замены РИ в магазине. Моделирование

производится расчетом времени поворота ИМ (включая фиксацию и расфиксацию гнезда) и отработки цикла замены автооператором АО₂ с последующим перекодированием состояния ИМ.

2) «РИ больше не нужен на обрабатываемой ДУ». Условие проверяется, будет ли использован данный РИ на оставшихся переходах обработки данной ДУ. Если нет, то удаление РИ моделируется аналогично 1-му варианту. В случае если данное условие удаления РИ задано совместно с правилом распределения ДУ между станками «в технологической последовательности», то оно игнорируется, а удаление ведется по 3-му условию.

3) «Нужно освободить гнездо в ИМ для следующего РИ». Данное условие предполагает удаление РИ лишь в том случае, когда все гнезда ИМ заполнены, и некуда установить требуемый РИ, доставленный из внешнего склада РИ. Выявление РИ, подлежащих удалению, производится процедурами *POISK* и *UDALIN* в моменты замены РИ в ИМ.

Моделирование процесса подготовки и смены РИ в шпинделе проводится в следующей последовательности. По известному коду технологического процесса DETAL и известному номеру перехода PERH в технологической базе данных выявляется шифр нужного РИ и требуемый его ресурс. Поскольку в системе допускается наличие до 10 одноименных инструментов-дублеров, поиск ведется любого из них, имеющего достаточный ресурс.

Поиск РИ начинается с проверки содержимого гнезд магазина, т.е. значений элементов массива KODINS[gnezdo, stanok]. Результаты поиска могут быть следующие.

1) РИ с требуемым кодом и достаточным ресурсом обнаружен в 1-м гнезде магазина. Если при этом $i = 1$, т.е. на следующем переходе нужен установленный в шпинделе РИ, то: 1) отработанное станком время WREOTR[stanok] увеличивается на длительность следующего перехода; 2) остаточный ресурс РИ уменьшается на величину машинного основного времени; 3) счетчик переходов PERH увеличивается на единицу, и цикл моделирования повторяется уже для нового перехода.

Если $i > 1$, то производится расчет времени TPOISK, затрачиваемого на поворот i -го гнезда в позицию смены магазина и отработку цикла смены РИ в захвате автооператора смены РИ (при раздельном цикле смены РИ). Время поворота рассчитывается следующим образом. Если ИМ без реверса (REWERS[stanok]=false) или $i \leq (\text{EmkMag}[\text{stanok}] / 2)$

$$t_{\text{поиск}} = \frac{L \times (i - 2)}{v_{\text{ИМ}}}, \quad (5.19)$$

Если REWERS[stanok]=true и $i > (\text{EmkMag}[\text{stanok}] / 2)$

$$t_{\text{поиск}} = \frac{L \times (V - i + 2)}{v_{\text{ИМ}}}, \quad (5.20)$$

где L - расстояние между осями соседних позиций магазина (рисунок 5.18);
 $v_{ИМ}$ - линейная скорость поворота ИМ;
 V - число гнезд в магазине.

Время стандартного цикла смены РИ в захвате АО при раздельном цикле смены задается пользователем в исходных данных и суммируется с переменной $TPOISK$. После расчета время $TPOISK$ сравнивается со временем перехода $PREPRH$, в процессе которого производится подготовка. Если $TPOISK > PREPRH$, то рассчитывается простой станка из-за несвоевременной подготовки РИ

$$PROSTO = TPOISK - PREPRH + TSMEIN[stanok],$$

где $TSMEIN[stanok]$ - неперекрываемая часть цикла смены РИ в шпинделе рассматриваемого станка.

Затем с учетом возможного простоя $PROSTO$ рассчитывается отработанное станком время как

$$WREOTR[stanok] = WREOTR[stanok] + PREPRH + PROSTO.$$

2) Нужного РИ в ИМ нет, склад РИ на участке не предусмотрен.

Замена РМ в ИМ производится в перерывах между циклами безлюдной работы ГПЯ. В данном случае ведется поиск ближайшего к шпинделю свободного гнезда ИМ и нужный РМ «материализуется» в данном гнезде. При этом код РИ выдается с учетом числа уже задействованных на участке его дублеров. Пользователь имеет возможность задать режим размещения РМ в ИМ «по умолчанию», когда все гнёзда ИМ обнуляются, и размещение РИ ведётся по указанной схеме. В данном случае на печать может быть выведена карта оптимального размещения РИ (строго говоря, оптимальным будет размещение РИ лишь для первого перехода, однако данный вопрос требует еще статистического обоснования).

3) *Нужный РИ с достаточным ресурсом обнаружен в позиции замены магазина* (предполагается при этом, что в ГПЯ имеется склад РИ). В этом случае подготовка РИ начинается с поиска гнезда в ИМ для установки данного РИ. После обнаружения гнезда для установки РИ из перегрузочной позиции ИМ считается время замены РИ. Затем считается время на смену РИ, с которым суммируется время выполнений замены: $TPOISK = TPOISK + TPOWOR + TZAM[stanok]$. Далее расчет простоев станка ведется так же, как и по первому варианту.

4) Нужного РИ на станке нет, на участке имеется, склад РИ. В этом случае формируется аварийная заявка на доставку нужного РИ из инструментального склада, проверяется готовность перегрузочного гнезда ИМ к приему нового РИ, рассчитывается текущий простой станка из-за отсутствия РИ.

В ходе проверки готовности перегрузочного гнезда выясняется, пустое оно или в нем имеется РИ. Если имеется РИ, то выясняется, какое значение имеет признак ($UDALEN[2, stanok]$). Если РИ удаляемый ($UDALEN[2, stanok]=1$), то станок готов к приему нового РИ. Если $UDLEN[2, stanok]=0$, то последующие дей-

ствия зависят от используемого условия передачи РИ из перегрузочной позиции в гнездо ИМ. Как следует из рисунка 5.19, таких условия три.

1) «*РИ потребовался в шпинделе*». Выполнение данного условия обеспечивается в процессе поиска РИ для предстоящего перехода при условии, что нужный РИ находится в перегрузочной позиции.

2) «*Адаптивно*» (по *min* простоев станка). При выполнении данного условия отрабатываются процедуры поиска удаляемого РИ, и рассчитывается время замены. Если окажется, что расчетное время замены РИ $TZAM[stanok]$ не превышает оставшуюся часть времени перехода ($TZAM[stanok] \leq PREPRH - TPOISK - TSMEIN[stanok]$), то замена РИ выполняется, т.е. время замены суммируется с отработанным станком временем и производится перекодировка расположения РИ и гнезд магазина. В противном случае расчетное значение $TZAM[stanok]$ игнорируется.

3) «*В перегрузочной позиции имеется доставленный РИ*». В этом случае производится безусловная замена РИ.

После выполнения циклов смены и (или) замены РИ, связанных с изменением относительного положения гнезд ИМ или расположения РИ, отрабатываются процедуры перекодировки состояния и накопления и обработки статистической информации по длительности циклов АСИ и АЗИ. При этом моделирование работы данного станка может прекратиться, если:

а) смоделирована подготовка и смена РИ для последнего перехода ($PERH[stanok] = PEREN[DETAL]$) обработки данной детали;

б) отработанное станком время превышает модельное время ($WREOTR[stanok] > SIOSUM$).

Если ни одно из двух данных условий не выполняется, то значение переменной $PERH[stanok]$ увеличивается на единицу и процесс моделирования повторяется уже для следующего технологического перехода.

5.2.5 Функциональные процедуры

Подпрограмма, моделирующая работу станка, включает ряд функциональных процедур, последовательность выполнения которых устанавливается в зависимости от заданных пользователем алгоритмов работы устройств АСИ и АЗМ, их характеристик и складывающейся ситуации в производственном процессе. Каждая из функциональных процедур моделирует определенную последовательность действий исполнительных органов станка и его агрегатов. Рассмотрим состав и содержание указанных процедур.

Процедура `AWARIQ` генерирует появление аварийной заявки на доставку режущего инструмента из автоматического склада РИ к текущему станку. Процедура вызывается, если в инструментальном магазине нет РИ для следующего перехода либо РИ нужного типоразмера есть, но имеет недостаточный ресурс стойкости. Формирование аварийной заявки на РИ начинается с проверки уже имеющегося списка заявок от данного станка.

Если заявка на данный РИ уже имеется, то ей присваивается ранг аварийной

(RANG[i]=2). Если нет, то формируется заявка на требуемый РИ с необходимым ресурсом, текущим моментом времени начала работы РИ и с рангом = 2. Сообщение о появлении аварийной заявки записывается в протоколе работы станка.

Процедура POISK служит для поиска в ИМ гнезда для установки в него РИ из перегрузочной позиции магазина. Чтобы сократить время поворота ИМ поиск ведется, начиная с гнезда, ближайшего к позиции замены ИМ дифференцированно для ИМ с возможностью реверса и без. Гнездо пригодно для установки РИ из перегрузочной позиции, если оно пустое либо в нем находится РИ, имеющий признак «не нужен в ИМ данного станка» (UDALEN[гнездо, станок]=i (гнездо - номер текущего гнезда ИМ, станок - номер текущего станка). В случае если пустых гнезд и удаляемых РИ в ИМ не обнаружено, вызывается процедура UDALIN, просматривающая список РИ в ИМ и выявляющая ненужные РИ. Выявление ненужных РИ производится так, чтобы сохранить в ИМ набор РИ для выполнения всех оставшихся переходов обработки ДУ. Если в это случае удаляемых РИ не обнаруживается, то последний переход отбрасывается, и процедура повторяется. Процесс длится до тех пор, пока не будет обнаружен удаляемый РИ или число рассматриваемых переходов не станет равным 1. В последнем случае выдается сообщении об ошибке.

Процедура TZAMRI служит для расчета времени выполнения цикла замены РИ в ИМ. Вызывается непосредственно за процедурой POISK.

После обнаружения гнезда для установки РИ из перегрузочной позиции ИМ считается время замены РИ, включающее время на поворот гнезда в позицию замены и замену РИ автооператором А0₂. Расчет времени поворота ведется дифференцированно для ИМ с реверсом и без согласно выражениям (5.19) или (5.20).

Процедура KODIR выполняет перекодировку гнезд ИМ при замене в нем режущего инструмента. В процессе моделирования работы станка имитируются повороты ИМ, а также изменяется состав и размещение РИ в гнездах. Кроме того, изменяется местоположение гнезд относительно шпинделя станка. С целью установить новые координаты положения гнезд ИМ и состояние гнезд, шпинделя и перегрузочного гнезда магазина выполняется их перекодировка.

Процедуры STATSI и STATZI служат для накопления статистической информации о затратах времени на смену и замену режущих инструментов, используемой в дальнейшем для построения практических кривых рассеяния времени работы устройств АСИ и АЗИ.

Выводы

Выполнена проработка алгоритмов моделирования работы ГПЯ на иерархических уровнях технологической операции и технологического перехода.

Детально проработано содержание процедур и функций, реализующих различные алгоритмы взаимодействия технологических и транспортных модулей, а также устройств автоматической смены и замены режущих инструментов.

Получены аналитические зависимости, описывающие длительность выполнения транспортных операций, циклов смены заготовок и режущих

инструментов для различных вариантов планировки производственной системы, компоновки оборудования, типов используемых транспортных средств, способов идентификации режущих инструментов, алгоритмов взаимодействия оборудования, узлов и агрегатов.

Проработана инфология компьютерных моделей, включая логику протекания производственного процесса и ее отображение в программной реализации алгоритмов.

Описано содержание процедур и функций, реализующих алгоритмы моделирования в виде программного кода.

6 Программные продукты, основанные на методе циклограмм

6.1 Интегрированная система расчета и моделирования ГПЯ механообработки «Каскад»

Система предназначена для решения широкого спектра технологических, проектных и организационных задач на этапах создания и эксплуатации ГПЯ. Функционирование производственной системы рассматривается на уровне технологического перехода. Моделирование основано на аналитическом описании длительности циклов работы и логическом описании многоуровневого взаимодействия модулей и их агрегатов. Длительность цикла работы каждого агрегата включает постоянные составляющие (например, цикл смены паллеты, цикл смены РИ) и переменные, продолжительность которых описана специально полученными аналитическими зависимостями. Используемый подход позволил учесть в совокупности практически все основные факторы реального производственного процесса, влияющие на загрузку и производительность оборудования - от планировочных размеров ГПЯ до алгоритмов работы отдельных агрегатов модулей, ресурса каждого отдельного инструмента и его местонахождения в магазине или накопителе.

6.1.1 Уровни задач, решаемых с помощью системы «Каскад»

Предусмотрено использование системы «Каскад» для решения задач трех уровней (рисунок 6.1).

1 уровень задач – объемные календарные расчеты производственной системы, связанные с выбором вместимости накопителей заготовок и режущих инструментов (склада паллет, инструментальных магазинов станков и автоматического инструментального склада ГПЯ). Вместимость накопителей рассчитывается так, чтобы разместить в них заготовки и режущие инструменты, необходимые для предстоящего цикла автономной работы ГПЯ. Расчеты основаны на формировании состава возможных вариантов сменных заданий, расчете количества изготавливаемых в смену деталей и определении необходимой потребности в режущих инструментах. Варианты сменного задания формируются на основе производственной программы ГПЯ, содержащей номенклатуру и годовой объем выпуска деталей.

По результатам расчета строятся кривые вероятностного распределения числа обрабатываемых в смену заготовок и количества требующихся режущих инструментов. На основе анализа кривых вероятностного распределения заготовок и инструментов рассчитывается требуемая вместимость накопителей. Задачи данного уровня решаются с помощью процедуры экспресс-анализа производственного процесса (раздел «Каскад/ Аналитический расчет/ Экспресс-анализ»). Пример экрана в одном из разделов результатов данного уровня расчетов представлен на рисунке 6.2.

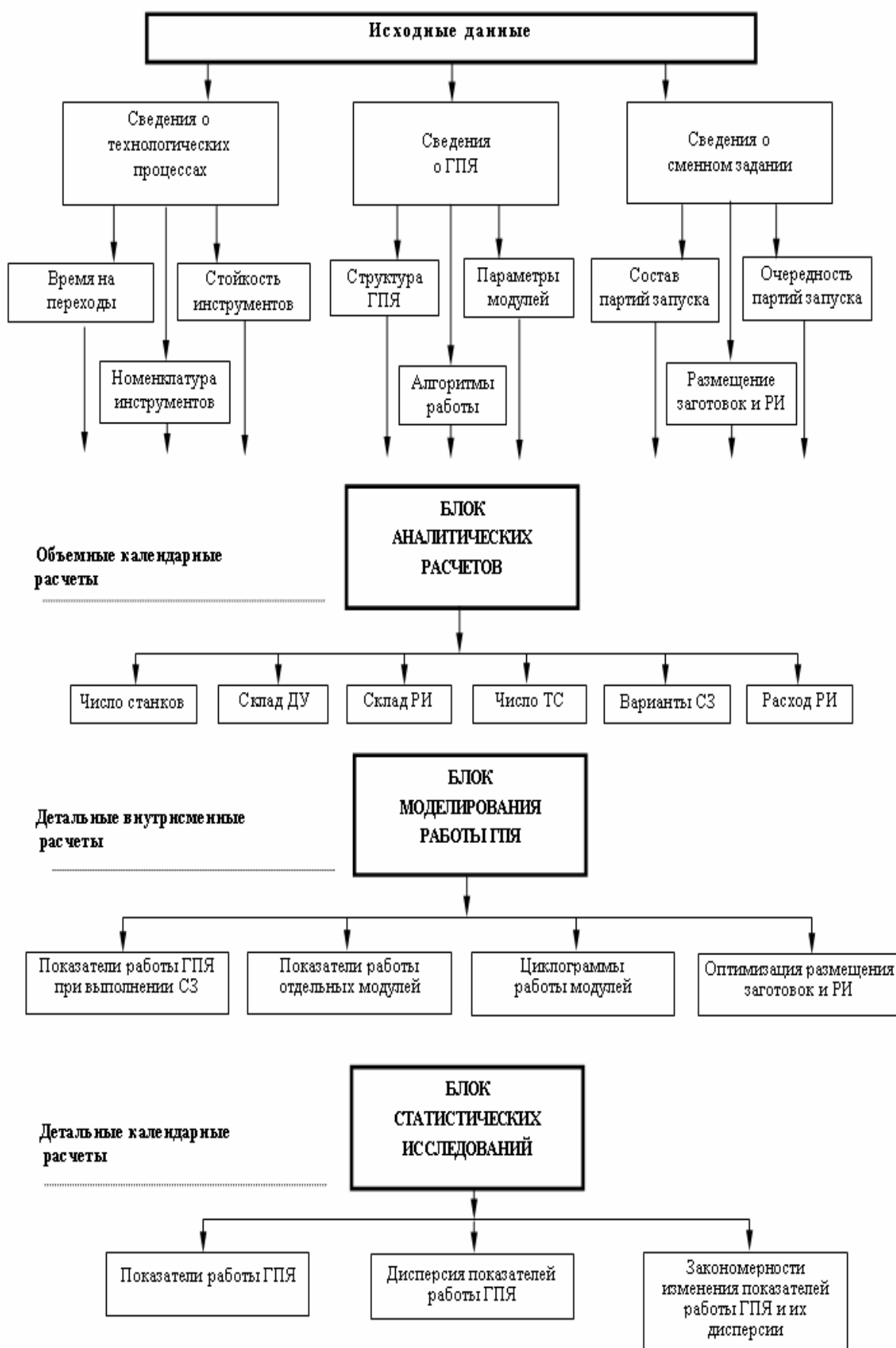


Рисунок 6.1 - Структурная схема и уровни использования интегрированной системы моделирования участков станков с ЧПУ и ГПЯ «Каскад»

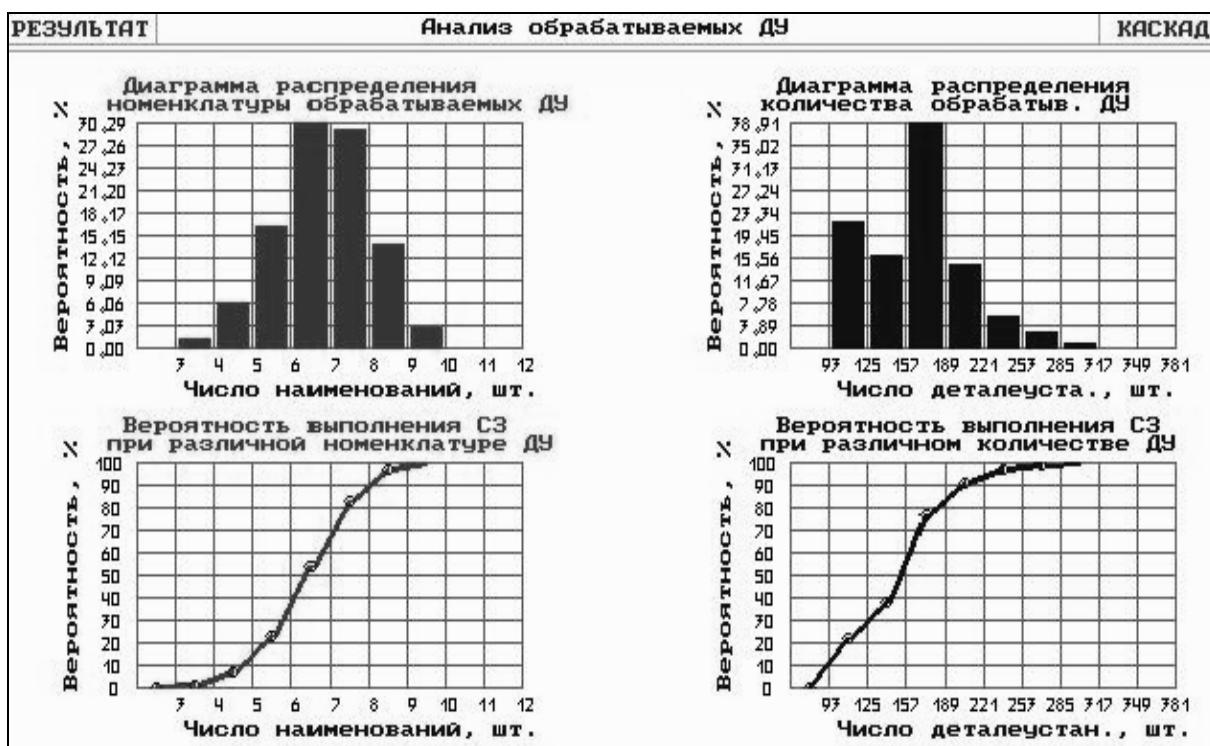


Рисунок 6.2 – Пример результатов расчета распределения числа деталей-установок в сменном задании на цикл безлюдной работы ГПЯ

2 уровень задач связан с оптимизацией работы ГПЯ при выполнении сменного задания. Цель – свести к минимуму внутрицикловые (внутрисменные) простои основного оборудования из-за несвоевременной доставки в рабочие зоны заготовок и режущих инструментов. Критерием оптимизации является коэффициент загрузки ГПЯ $k_{ГПЯ}$. Оптимизация состоит в целенаправленном изменении технических параметров оборудования, моделировании работы ГПЯ с выбранными параметрами и анализе полученных результатов по $k_{ГПЯ}$. Задачи данного уровня решаются с помощью блока моделирования производственного процесса в разделе «Каскад/ Моделирование». Примеры экранов с результатами моделирования в данном разделе представлены в соответствии с рисунками 6.3 и 6.4.

3 уровень задач – статистические исследования, состоящие в проверке качества функционирования ГПЯ при различных вариантах СЗ. На данном уровне оценивается чувствительность ГПЯ, параметры которой определены на 2-м уровне, к изменению состава сменного задания. Если на предыдущем уровне расчетов качество работы ГПЯ характеризуется конкретным значением параметра, например, коэффициента загрузки $k_{ГПЯ}$, то здесь критериями качества служат предельные и среднее значения коэффициента загрузки ГПЯ, а так же закономерности его вероятностного распределения (рисунок 6.5 а). Тем самым обеспечивается более объективная оценка качества работы системы. Статистический анализ позволяет и более точно подобрать параметры ГПЯ (рисунок 6.5 б).

Помимо $k_{ГПЯ}$ могут использоваться и другие критерии (всего 12), среди которых: продолжительность выполнения сменного задания, расход режущего инструмента и т.д.

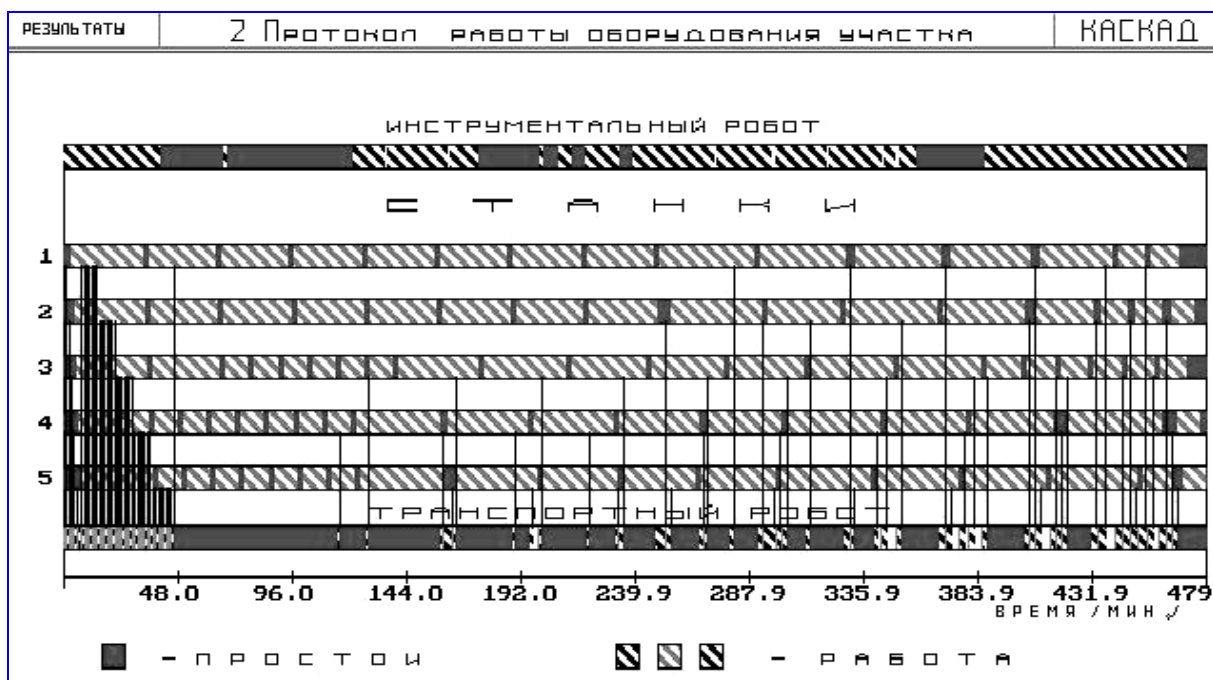


Рисунок 6.3 – Пример циклограммы работы и простоев оборудования за цикл моделирования работы ГПЯ

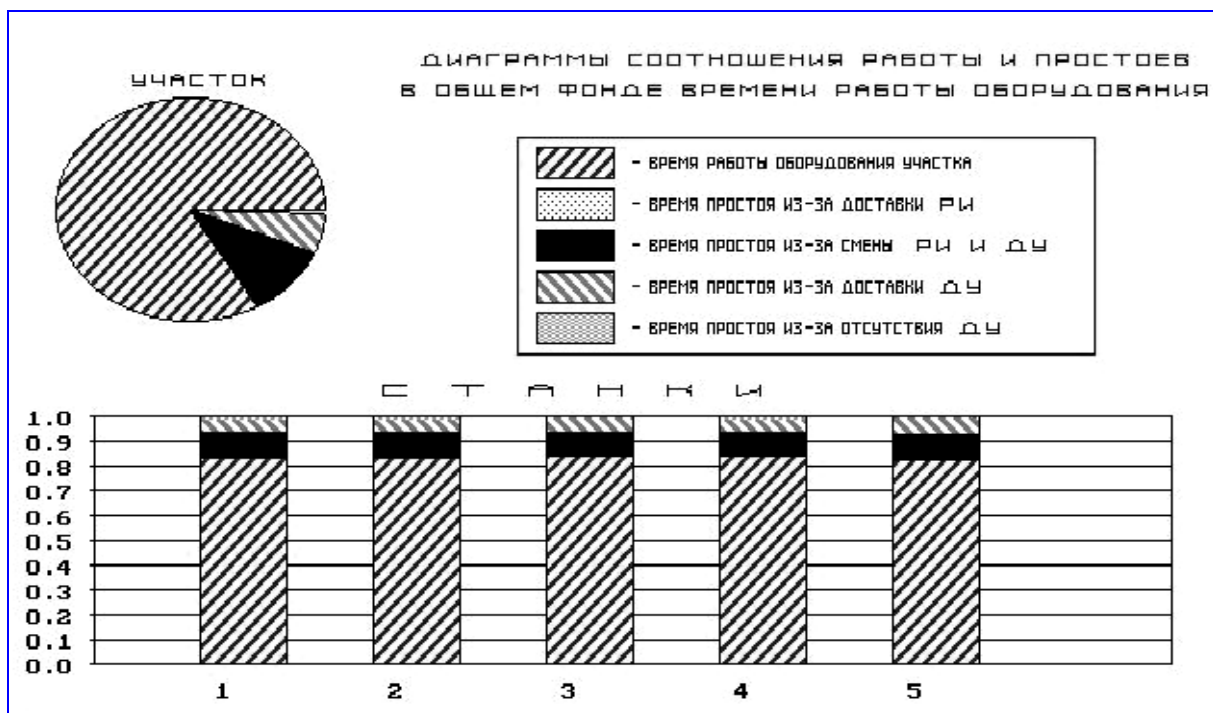
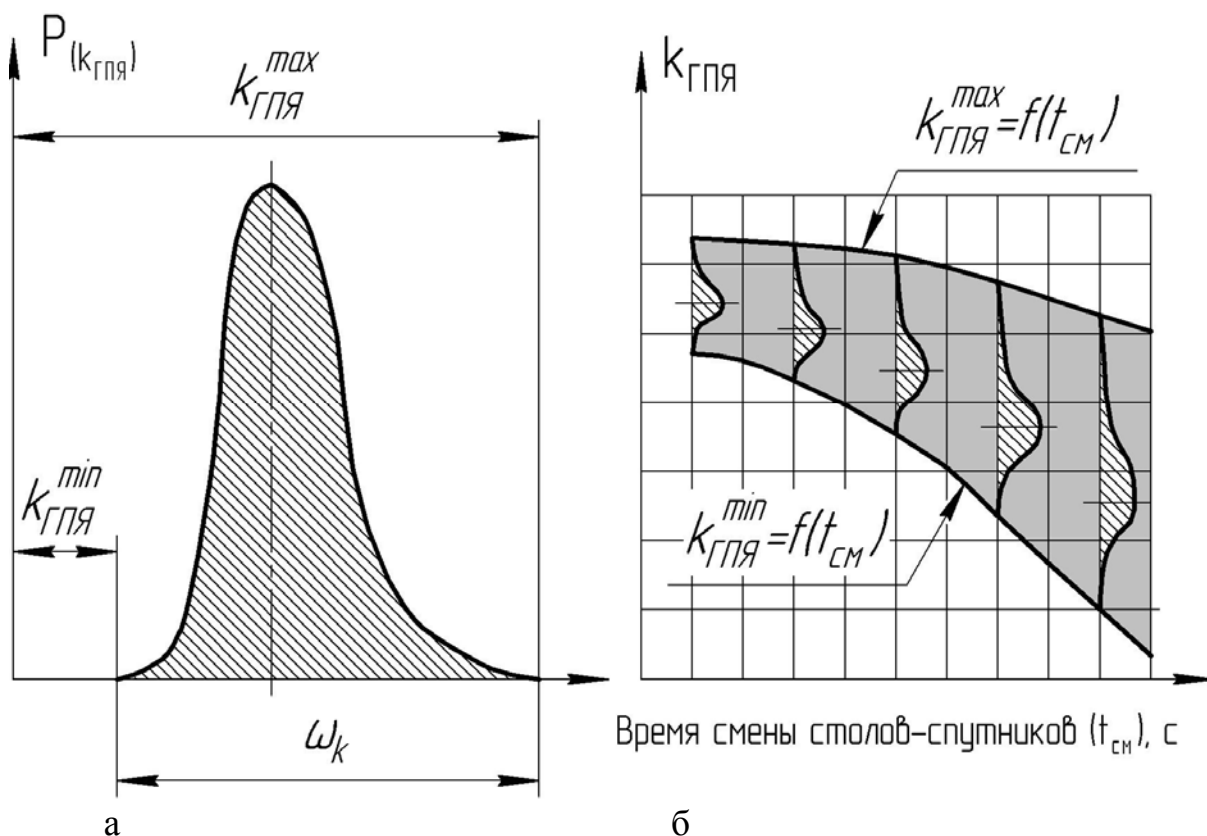


Рисунок 6.4 – Пример диаграмм распределения работы и простоев оборудования за цикл моделирования работы ГПЯ



а - вероятностное распределение коэффициента загрузки ГПЯ при выполнении нескольких сменных заданий;

б - закономерности изменения $k_{ГПЯ}$ при различной длительности цикла смены столов-спутников

Рисунок 6.5 – Статистические исследования ГПЯ в системе «Каскад»

6.1.2 Области использования системы «Каскад»

В общем случае интегрированная система расчета и моделирования ГПЯ механообработки «Каскад» как проблемно-ориентированный инструмент позволяет:

а) технолог:

- 1) сформировать рациональную номенклатуру деталей для изготовления в данной ГПЯ;
- 2) оценить целесообразность внедрения в данной ГПЯ с устоявшейся номенклатурой технологии изготовления новой детали (набора деталей);
- 3) правильно распределить технологические переходы между технологическими операциями;
- 4) осуществить выбор номенклатуры режущих инструментов и режимов резания, исходя из условия наиболее эффективной работы ГПЯ;
- 5) оценить качество работы ГПЯ при заданном наборе технологических процессов;

б) проектировщику, конструктору:

- 1) сравнить по целому ряду показателей и выбрать наиболее рациональный вариант планировки проектируемого или модернизируемого участка;
- 2) при заданной планировке выявить требуемые технические характеристики

модулей участка (станков, транспортных средств, накопителей) и их агрегатов (напр., устройств автоматической смены инструментов (АСИ));

3) выбрать наиболее рациональные варианты функционирования модулей и их агрегатов;

4) оценить рядом количественных показателей качество работы спроектированного участка;

в) диспетчеру (мастеру, начальнику участка):

1) оперативно определить по задаваемым ограничениям состав рекомендуемых вариантов сменно-суточного задания (например, исходя из номенклатуры и количества имеющейся в наличии инструментальной оснастки);

2) выявить наиболее рациональный состав сменного задания, размеры партий запуска и очередность их обработки;

3) выбрать оптимальный вариант размещения заготовок и режущих инструментов в ячейках накопителей перед началом цикла работы участка;

4) определить требуемое количество паллет с различными ложементами, динамику (моменты времени) их потребления, а также номенклатуру и количество режущих инструментов и инструментальной оснастки, необходимой для выполнения предписанного сменного задания;

5) оценить качество работы участка при выполнении предписанного сменного задания;

г) ученому-исследователю:

1) проанализировать связи между различными характеристиками производственных процессов;

2) вывести и исследовать эмпирические зависимости, обуславливающие качество проектных и эксплуатационных решений;

3) провести апробирование оптимизационных и логико-вероятностных процедур организации и управления производством, и т.д.

Таким образом, моделирование производственного процесса с использованием системы «Каскад» позволяет оценить качество работы производственной системы при различных проектных, технологических и организационных решениях, выявить «узкие места», которые могут стать причиной снижения эффективности участка либо требуют неоправданно высоких капитальных вложений.

К особенностям системы «Каскад» относятся:

а) высокая гибкость, обуславливающая возможность моделирования как ГПЯ со складом паллет и с автоматическим складом режущих инструментов при поштучной замене их в магазинах сверлильно-фрезерно-расточных станков, так и участков из токарных станков с ЧПУ, не связанных автоматическим транспортом;

б) учет практически всех основных факторов реального производственного процесса - от планировочных размеров участка до алгоритмов работы отдельных агрегатов модулей, ресурса каждого отдельного инструмента и его местонахождения в магазине или накопителе (таблица 6.1);

Таблица 6.1 - Сводная ведомость регулируемых параметров интегрированной системы моделирования «Каскад»

<i>Регулируемый параметр производственного процесса</i>	<i>Предельное значение</i>
Технологические параметры	
Число технологических переходов в техпроцессе	50
Типоразмер РИ на каждом переходе	500
Стойкость РИ каждого типоразмера	*
Основное технологическое время перехода	*
Вспомогательное время перехода	*
Организационные параметры	
Размеры партий запуска	*
Число партий запуска в СЗ	10
Очередность партий запуска	*
Варианты исходного размещения заготовок	*
Варианты исходного размещения РИ	*
Проектные и конструкторские параметры	
Схемы расстановки оборудования	16
Планировочные размеры	*
Формы транспортных маршрутов	3
Система основного оборудования	
Число станков, типа ОЦ, шт.	10
Число позиций в ПН паллет, шт.	10
Время смены паллеты на столе, с	*
Время смены РИ в шпинделе, с	*
Число гнезд в ИМ станка, шт.	*
Скорость поворота ИМ, м /мин	*
Время замены РИ в ИМ, с	*
Положение гнезда замены, номер	*
Транспортно - накопительная система	
Местоположение склада паллет, м.	*
Число позиций склада паллет, шт.	400
Число ярусов склада паллет, шт.	*
Расстояние между позициями склада, м	*
Тип транспортного средства (ТС)	4
Отработка перемещений ТС	2
Число позиций на ТТ, шт.	*
Скорость перемещения ТС, м/с	*
Ускорение ТТ, м/с ²	*
Время смены паллет на ТТ, с	*
Система инструментаобеспечения	
Местоположение склада РИ, м	*
Число ярусов склада РИ, шт.	*
Число позиций склада РИ, шт.	1000
Расстояние между позициями склада, м	*
Отработка перемещений ИРа	2
Скорость перемещения ИРа, м/с	*
Ускорение ИРа, м/с ²	*
Время цикла смены РИ, с	*
Алгоритмы системы управления	
Варианты распределения заготовок между станками	2
Способ идентификации РИ	2
Приоритеты обслуживания ТС	5
Приоритеты смены и замены инструментов на станке	3
* - параметр задается пользователем	

в) наличие блока аналитических расчетов, позволяющего выявить интервалы допустимых значений ряда параметров участка и выполняемых технологических процессов.

Ограничения по использованию системы «Каскад» накладываются особенностями реализации моделирующего алгоритма в среде операционной системы ПЭВМ: с целью обеспечить приемлемое соотношение между временем одного прогона блока моделирования и размерами моделируемого участка, в системе приняты предельные значения параметров моделируемых участков, представленные в соответствии с таблицей 6.1. Логика многоуровневого взаимодействия элементов системы настраивается посредством выбора правил приоритетов, полный перечень которых представлен в таблице 6.2.

Таблица 6.2 - Список правил, регламентирующих работу устройств моделируемой ГПЯ

Правило	Варианты значений
Распределение заготовок между станками	а) по составу комплекта РИ на станке; б) произвольно согласно очереди ССЗ.
Приоритеты доставки заготовок к станкам	а) по времени поступления заявок; б) по приоритетам станков; в) по <i>min</i> времени транспортной операции; г) по <i>min</i> очереди ДУ в пристан. накопителе; д) по <i>min</i> станкоемкости ДУ в накопителе.
Передача режущего инструмента из перегрузочной позиции в магазин станка	а) как только робот доставил туда РИ; б) если только данный РИ нужен в шпинделе; в) адаптивно (по <i>min</i> простоя станков).
Приоритеты доставки к станкам режущих инструментов	а) по времени поступления заявки; б) по приоритетам станков; в) по приоритетам деталиустановок; г) по <i>max</i> простаивающему станку; д) по <i>min</i> времени трансп. операции.
Причина удаления режущего инструмента из магазина	а) исчерпана стойкость или поломанн; б) больше не нужен на обрабатываемой ДУ; в) чтобы освободить гнездо для след. РИ.

6.2 Программа многокритериальной оценки проектных решений в гибких производственных ячейках механообработки «Fania»

Программное средство для реализации вычислительных экспериментов – программа Fania – реализовано в среде Delphi 7 на языке программирования Object Pascal.

Алгоритм программы рассчитан на моделирование работы ГПЯ на уровне технологической операции.

6.2.1 Входные данные

Для работы программы необходимо ввести:

- а) параметры оборудования;
- б) алгоритмы взаимодействия оборудования;
- в) производственную программу;
- г) сменное задание;
- д) исходное размещение заготовок в позициях накопителей.

Визуально вводимые данные представлены в виде двух структурных модулей – данных об оборудовании и данных об изделиях (рисунок 6.6 а, б).

Данные об оборудовании содержат необходимые параметры станков, накопителей, транспортного средства и исследуемые правила обслуживания. Схема на рисунке 6.6 а) иллюстрирует логику формирования состава входных данных. Если состав входных данных по станкам и правилам обслуживания остается неизменным, то состав запрашиваемых параметров транспортного средства зависит от типа используемого склада заготовок.

Ввод данных об изделиях начинается с описания производственной программы, выполняемой на исследуемом производственном участке ГПЯ (рисунок 6.6 б). В программе в соответствии с методикой исследования, описанной во 2-ой главе, предусмотрено два режима моделирования:

- а) для конкретного состава сменного задания с расчетом показателей загрузки, производительности и срока окупаемости системы (режим «Сменное задание»);
- б) для множества вариантов сменных заданий, что предусматривает получение статистической информации о вероятностном распределении показателей загрузки, производительности и срока окупаемости (режим «Варианты СЗ»).

Согласно выбранному режиму работы запрашивается и необходимый набор исходных данных.

Исходное размещение заготовок в позициях склада формируется по одному из 7 способов, формальное описание которых представлено в разделе 5.1.3.

Экранные формы разработанной программы, реализующие ввод данных, представлены в соответствии с рисунком 6.7 а, б.

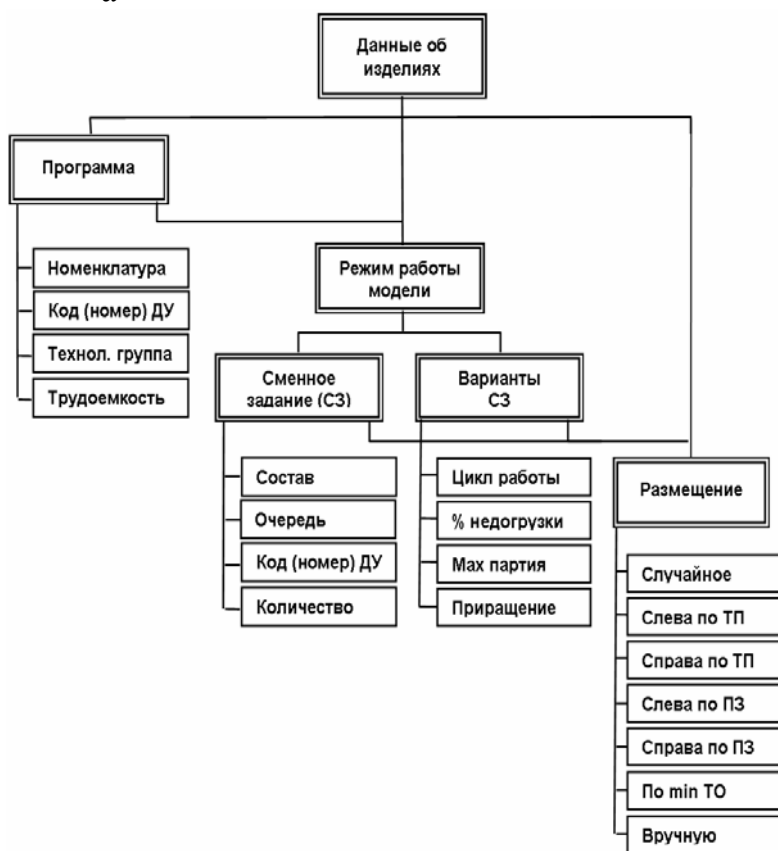
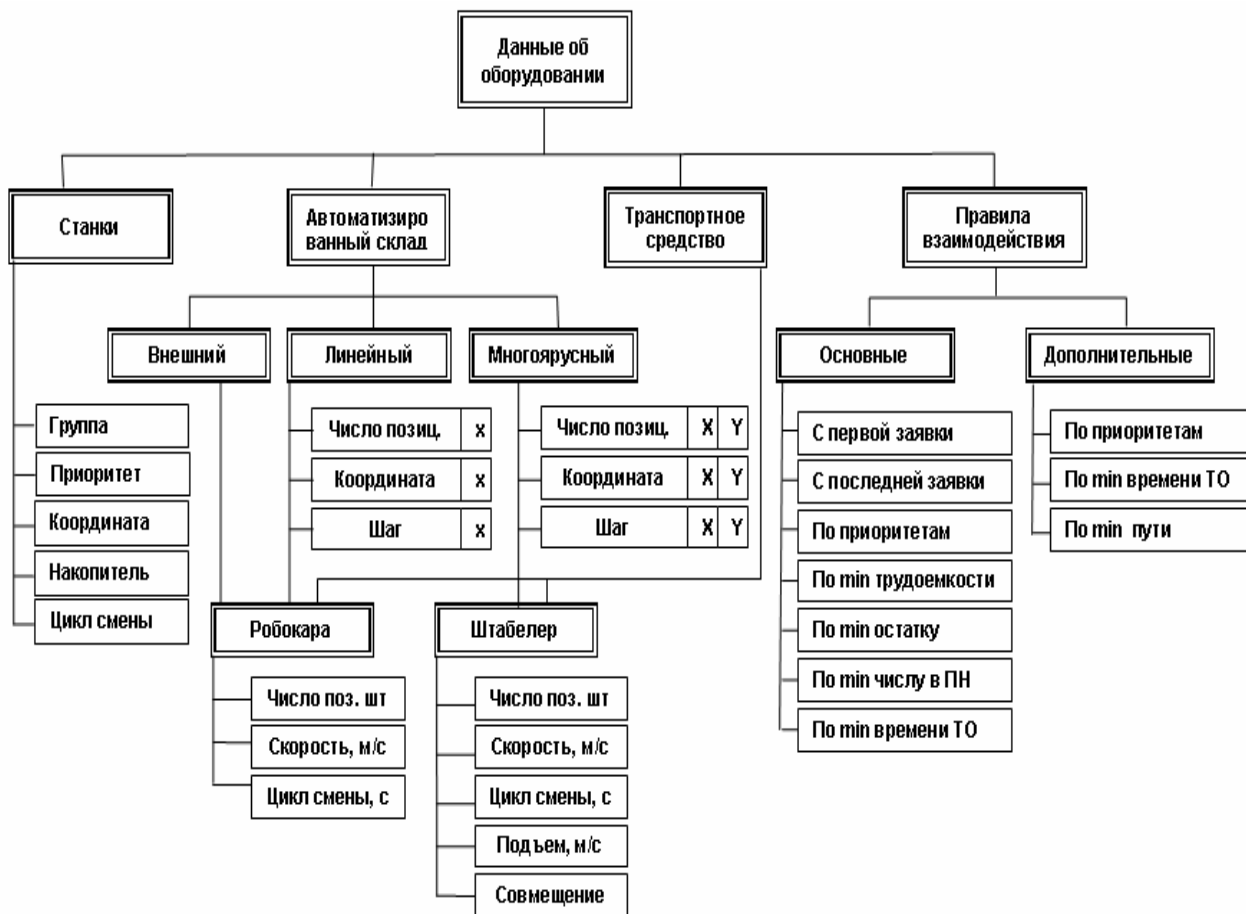


Рисунок 6.6 - Схемы входных данных программы моделирования «Fania» об оборудовании (а) и об изделиях (б)

Fania. Ввод данных для моделирования работы гибкой производственной ячейки
 Инструкции Разработчики Выход

Данные об оборудовании

Станки
 Число станков, штук **10**

Станок № =>	1	2	3	4	5
Технологическая группа	1	1	1	2	2
Приоритет станка	1	1	2	1	1
Координата перегрузки, м	1	5	9	13	17
Число позиций в ПН, шт	6	6	6	8	8
Цикл смены палеты, сек	50	50	50	45	45

Автоматизированный склад
 Тип склада **Многоярусный**

	X	Y
Число позиций на складе, шт	100	5
Координата первой позиции, м	2	1
Шаг позиций склада, м	1	1

Транспортное средство Кран-штабелер

Число позиций для палет, шт	2
Скорость перемещения, м/с	2
Время цикла смены палет, с	20
Скорость подъема платформы, м/с	1

☒ одновременные перемещения по осям

Правило выбора заявок на обслуживание:
 5. По min остаточному времени в рабочей зоне

Дополнительное правило:
 3. По min пути ТС к ячейке склада

Данные об оборудовании Загрузить Сохранить

Данные об изделиях

Производственная программа
 Номенклатура деталей, шт. **6**

Порядковый номер деталей	1	2	3	4	5
Технологическая группа	1	1	1	1	2
Трудоемкость, мин	5	19	23	11	10

Сменное задание
 Номенклатура партий запуска, шт. **7**

Номер партии запуска =>	1	2	3	4	5
Порядковый номер деталей	6	1	3	2	4
Число деталей, шт.	35	12	17	29	36

Исходное размещение заготовок на складе
 Расположить заготовки
 1. В случайной последовательности

Старт Данные об изделиях Загрузить Сохранить

☒ сменное задание (СЗ) ☐ варианты СЗ

а

Fania. Ввод данных для моделирования работы гибкой производственной ячейки
 Инструкции Разработчики Выход

Данные об оборудовании

Станки
 Число станков, штук **10**

Станок № =>	1	2	3	4	5
Технологическая группа	1	1	1	2	2
Приоритет станка	1	1	2	1	1
Координата перегрузки, м	1	5	9	13	17
Число позиций в ПН, шт	6	6	6	8	8
Цикл смены палеты, сек	50	50	50	45	45

Автоматизированный склад
 Тип склада **Многоярусный**

	X	Y
Число позиций на складе, шт	100	5
Координата первой позиции, м	2	1
Шаг позиций склада, м	1	1

Транспортное средство Кран-штабелер

Число позиций для палет, шт	2
Скорость перемещения, м/с	2
Время цикла смены палет, с	20
Скорость подъема платформы, м/с	1

☒ одновременные перемещения по осям

Правило выбора заявок на обслуживание:
 4. По min трудоемкости заготовок в ПН

Дополнительное правило:
 2. По min времени транспортной операции

Данные об оборудовании Загрузить Сохранить

Данные об изделиях

Производственная программа
 Номенклатура деталей, шт. **6**

Порядковый номер деталей	1	2	3	4	5
Технологическая группа	1	1	1	1	2
Трудоемкость, мин	5	19	23	11	10

Варианты сменного задания

Цикл безлюдной работы ГПЯ, мин.	480
Допустимая недогрузка (1...3) %	1
Max размер партии запуска, шт.	100
Приращение партий запуска, шт.	10

Пуск

NN	1	2	3	4	5	6	Деталей
Min	0	0	0	0	0	0	260
Max-426	100	100	100	100	100	100	460
1	100	80	100	40	0	0	320
2	60	80	100	60	0	0	300
3	80	100	60	100	0	0	340
4	60	80	80	100	0	0	320
5	80	100	80	40	20	0	320
6	60	80	100	40	20	0	300
7	20	80	100	60	20	0	280
8	80	100	60	80	20	0	340
9	40	100	60	100	20	0	320

Старт Данные об изделиях Загрузить Сохранить

☐ сменное задание (СЗ) ☒ варианты СЗ

б

Рисунок 6.7 – Экранные формы ввода данных в режимах «Сменное задание» (а) и «Варианты СЗ» (б)

6.2.2 Результаты моделирования

Выбор режима моделирования производится щелчком мыши на соответствующей радиокнопке (в правой нижней части формы ввода данных), тем самым изменяется содержание правой части экранной формы, а последующий запуск программы нажатием кнопки «Старт» приводит к запуску соответствующих программных процедур и визуализации разных экранных форм с результатами работы программы.

Запуск программы после ввода данных в режиме «Сменное задание» активизирует форму результатов моделирования, представленную на рисунке 6.8. Форма содержит:

а) циклограмму работы станков и транспортного средства на протяжении цикла выполнения предписанного сменного задания;

б) круговую диаграмму, иллюстрирующую соотношение работы и простоев по различным причинам в общем фонде времени цикла работы ГПЯ;

в) сводные результаты и показатели эффективности работы системы, представленные в табличной форме. Переход между панелью ввода данных рисунка 6.7 а и панелью результатов моделирования рисунка 6.8 происходит по нажатию левой клавиши мыши. Тем самым реализована возможность изменить входные данные и тут же отследить изменение получаемых результатов.

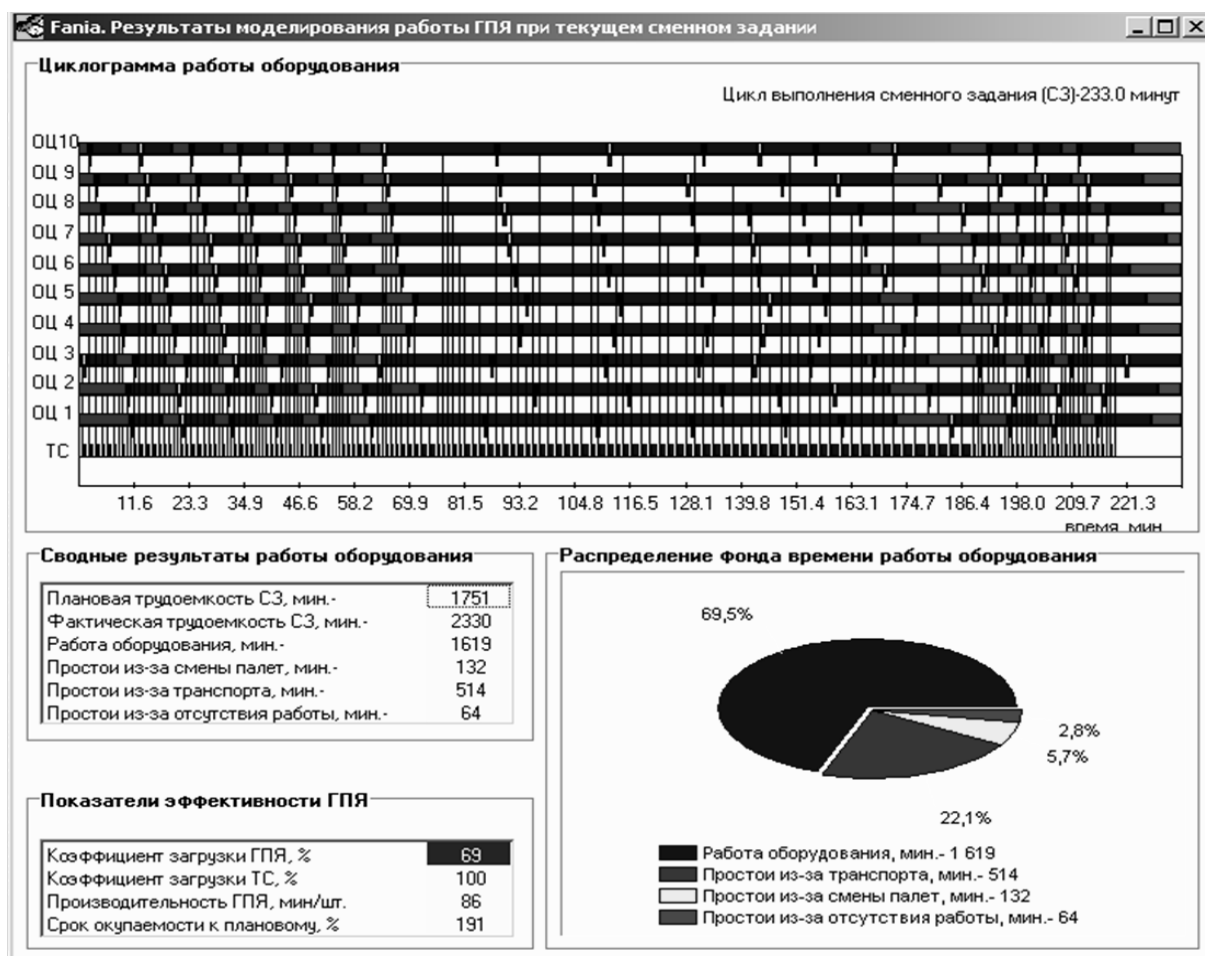


Рисунок 6.8 – Экранная форма с результатами моделирования в режиме «Сменное задание»

Запуск программы из режима «Варианты СЗ» активизирует одну из форм результатов, представленных в соответствии с рисунком 6.9.

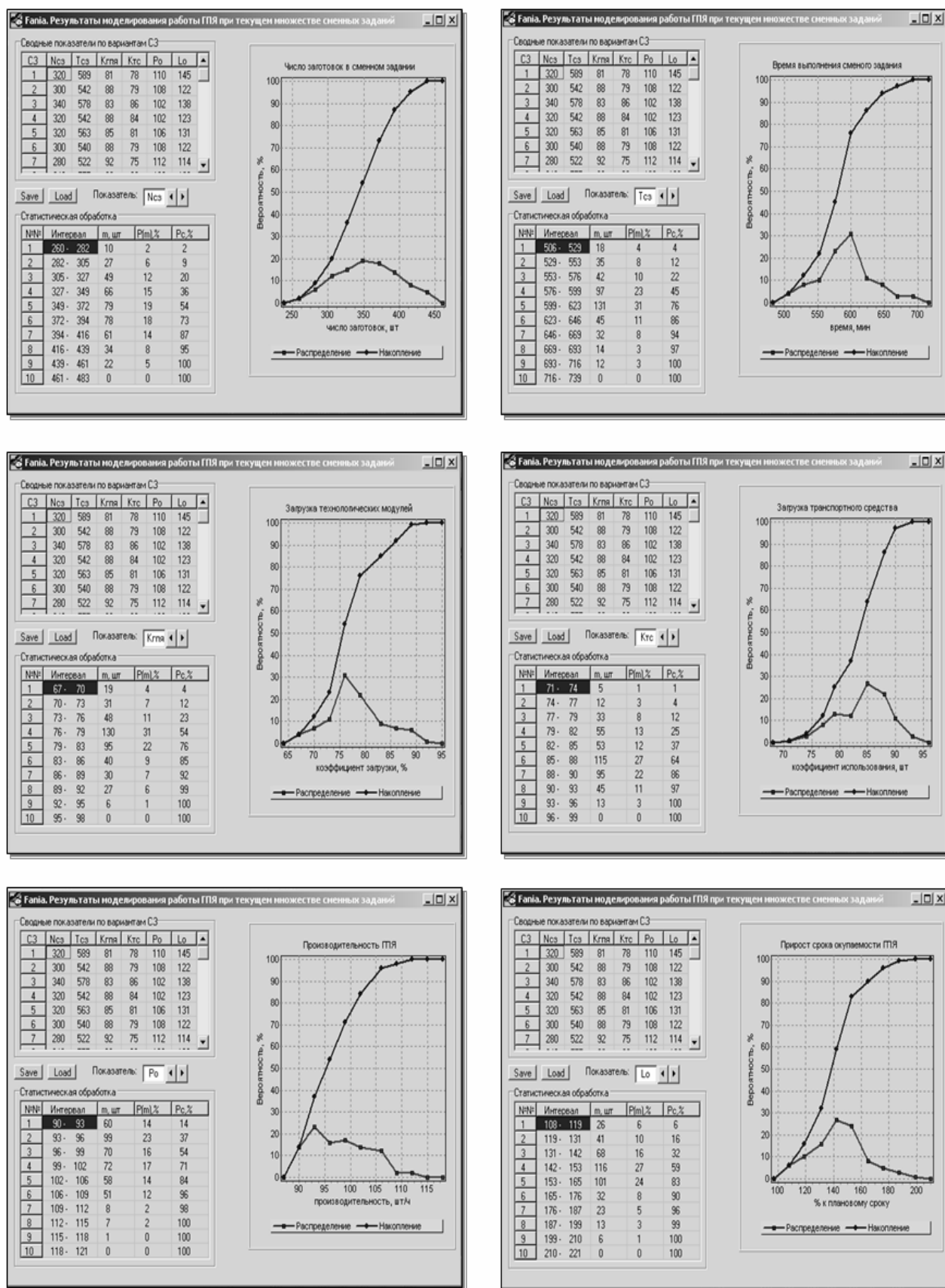


Рисунок 6.9 – Экранные формы показателей эффективности ГПЯ в режиме работы «Варианты СЗ»

Каждая из форм выполнена по одному сценарию и содержит а) список вариантов сменного задания, подвергнутых моделированию, и 6 показателей эффективности выполнения СЗ; б) статистическое распределение значений выбранного показателя; в) вероятностные кривые распределения и накопления значений выбранного показателя. Показатели эффективности работы ГПЯ при каждом варианте сменного задания включают (рисунок 6.9): число заготовок в составе СЗ, $N_{СЗ}$; фактическое (с учетом простоев) время выполнения СЗ, $T_{СЗ}$; коэффициент загрузки ГПЯ, $k_{ГПЯ}$; коэффициент использования транспортного средства, $k_{ТС}$; производительность, P_o ; процент прироста срока окупаемости ГПЯ к номинальному, L_o .

Таким образом, реализована возможность статистической оценки эффективности ГПЯ посредством показателей загрузки оборудования, производительности и срока окупаемости при различных значениях входных данных.

Для повышения уровня доверия к результатам моделирования, получаемым с помощью программы «Fania», выполнены испытания, предусмотренные стандартами на разработку автоматизированных систем. Испытания включали:

а) оценку чувствительности модели, выполненную визуальным сравнением циклограмм, построенных для различных комбинаций правил обслуживания;

б) формальные процедуры верификации, состоящие в проверке соответствия логики программы «Fania» известным закономерностям [2] между входными и выходными параметрами ГПЯ;

в) проверку соответствия входных данных и получаемых результатов, выполненную с помощью расчетов вручную на тестовых примерах;

г) сравнение полученных результатов моделирования с результатами работы программы-аналога (в качестве такой программы использовалась интегрированная система «Каскад»).

6.2.3 Пример применения программы

Разработанная программа позволяет производить оценку эффективности выбора правил обслуживания в АСУ при различных параметрах оборудования и изготавливаемых изделий. Кроме того, программа «Fania» предоставляет возможность исследовать эффективность работы комплекса технологического оборудования (прессов, станков, координатно-измерительных машин), обслуживание которых производится промышленным роботом, транспортной тележкой или краном-штабелером. При этом учитываются различные компоновки склада заготовок (внешний, линейный, многоярусный) и схемы расстановки оборудования. Возможно использование программы как на этапе выбора проектных параметров ГПЯ, так и на стадии эксплуатации.

В качестве примера на рисунке 6.10 представлены результаты вычислительных экспериментов по выявлению оптимальной длительности цикла безлюдной работы ГПЯ.

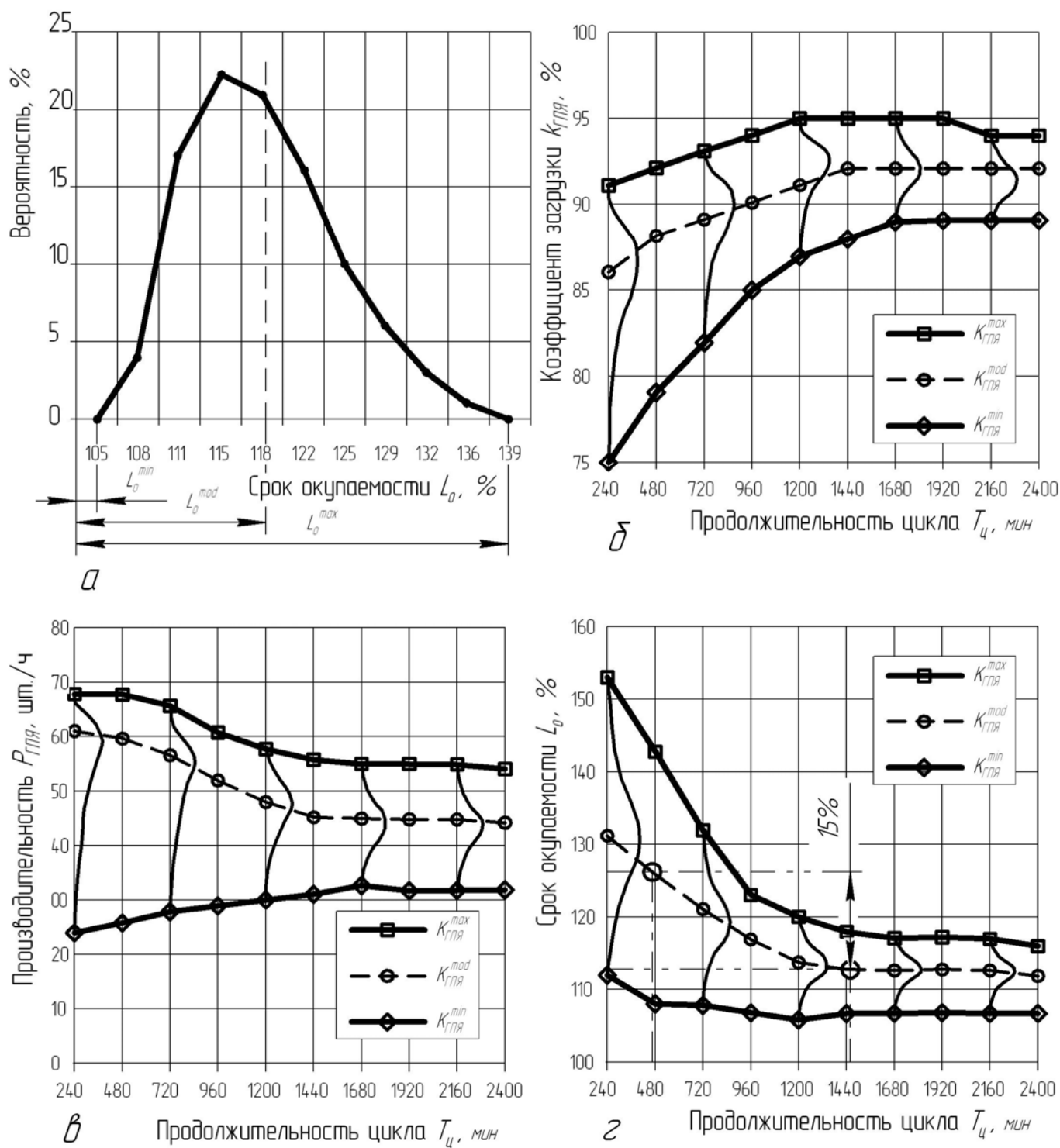


Рисунок 6.10 - Графики, иллюстрирующие вероятностное распределение прироста срока окупаемости L_O для множества вариантов сменных заданий (а); изменение вероятностного распределения $k_{ГПЯ}$ (б), $P_{ГПЯ}$ (в) и L_O (г) при изменении продолжительности цикла $T_{ц}$ безлюдной работы ГПЯ

Под безлюдным режимом работы понимается функционирование производственной ячейки в автоматическом режиме не менее 8 ч [13]. В литературе имеются данные о ГПЯ, работающих в безлюдном режиме 140 ч в неделю и 20 ч в сутки [75, 82]. Теоретически можно, выполняя своевременное пополнение запасов заготовок и инструментов, обеспечить сколь угодно долгое автоматическое

функционирование ГПЯ. Однако закономерен вопрос: зависит ли эффективность ГПЯ от длительности цикла $T_{ц}$ ее безлюдной работы?

Для поиска ответа изучена работа ГПЯ, состоящей из 8 станков типа «обрабатывающий центр», функционирующих по схеме «склад-станок-склад». Автоматизированный склад заготовок имеет приемопередающие позиции для обмена паллетами со станками и с внешним транспортом, обеспечивающим своевременное пополнение склада. Рассматривалось функционирование ГПЯ при организации длительности цикла безлюдной работы $T_{ц}=240...2400$ мин. Эффективность работы ГПЯ оценивалась средними значениями и параметрами вероятностного распределения трех показателей: коэффициента загрузки $k_{ГПЯ}$, производительности $P_{ГПЯ}$ и прироста срока окупаемости L_o (рисунок 6.10 б - г). Полученные результаты показывают, что наиболее высокая средняя производительность ГПЯ обнаруживается при $T_{ц}=240...480$ мин. Однако в этом диапазоне значения показателей $k_{ГПЯ}$, $P_{ГПЯ}$ и L_o наиболее нестабильны, что характеризуется большими интервалами рассеяния значений. Так, значения $k_{ГПЯ}$ при $T_{ц}=480$ мин рассеиваются в интервале от 75 до 92 %, L_o колеблется от 109 до 143 %. Дальнейшее увеличение $T_{ц}$ ведет к улучшению стабильности показателей, повышению среднего значения $k_{ГПЯ}$ и сокращению среднего значения L_o . Начиная с $T_{ц}=1440$ мин, работа ГПЯ характеризуется наиболее высокой стабильностью при максимальной загрузке оборудования и минимальном приросте срока окупаемости. Таким образом, для использованных исходных данных увеличение длительности цикла безлюдной работы ГПЯ с 8 до 24 ч позволяет сократить срок окупаемости ГПЯ в среднем на 15 % и резко повысить стабильность контролируемых показателей эффективности. Дальнейшее увеличение $T_{ц}$ не приводит к увеличению эффективности функционирования производственной ячейки. Можно сделать вывод, что в рассмотренном случае наиболее целесообразной является организация работы ГПЯ в безлюдном режиме в течение 24 ч.

Пример оценки длительности цикла безлюдной работы ГПЯ продемонстрировал возможность использования программы «Fania» при оптимизации организационных параметров ГПЯ. В общем случае программа может быть использована для многокритериальной оценки любых параметров, применяемых в качестве исходных данных (таблицы 6.3 и 6.4).

Полученные результаты показывают, что разработанное программное средство «Fania»:

а) обеспечивает поддержку технических, технологических и организационных решений, принимаемых при проектировании и эксплуатации гибких производственных ячеек;

б) может использоваться для многокритериальной оценки параметров системы технологического оборудования, транспортно-накопительной системы и автоматизированной системы управления гибкими производственными ячейками;

Таблица 6.3 - Сводная ведомость входных данных программы «Fania»

Входные данные	Диапазон регулирования или варианты значений
Система основного технологического оборудования	
Число станков в ГПЯ, шт.	1-50
Технологическая группа станка	*
Приоритет в обслуживании транспортным средством (ТС)	*
Координата точки останова ТС при перегрузке заготовок	*
Число позиций в накопителе заготовок, шт.	1-50
Время цикла смены заготовки в рабочей зоне, с	*
Автоматизированная транспортно-складская система	
Тип склада заготовок	Линейный, многоярусный, внешний
Число позиций на складе, шт.: по оси X (горизонталь) по оси Y (вертикаль)	1 – 500 1 - 20
Координаты склада по осям X , Y , м	*
Шаг ячеек склада по осям X , Y , м	*
Тип транспортного средства	Промышленный робот
	Робокара
	Кран-штабелер
Число позиций (манипуляторов), шт.	1-2
Синхронность перемещений по осям X , Y	Последовательная
	Одновременная
Скорость перемещения ТС по осям X , Y , м/с	*
Время цикла смены заготовки на ТС, с	*
Автоматизированная система управления	
Правила обслуживания технологического оборудования	21 комбинация из основного и дополнительного правил обслуживания
Производственная программа	
Число технологических процессов в программе, шт.	1-200
Станкоемкость технологического процесса, мин	О
Технологическая группа оборудования, номер	О
Сменное задание на цикл безлюдной работы	
Число партий запуска (ПЗ), шт.	1-100
Номер техпроцесса партии запуска	О
Размер партии запуска, шт.	О
Исходное размещение заготовок на складе	
Варианты упорядоченного заполнения позиций склада	6 вариантов
Параметры процесса моделирования	
Число изделий в сменном задании, шт., не более	5000
Число вариантов сменного задания, шт., не более	5000

Таблица 6.4 - Сводная ведомость и выходных результатов программы «Fania»

Показатели эффективности ГПЯ*	В режиме «Сменное задание»	В режиме «Варианты СЗ»
Простои оборудования, мин: из-за смены заготовок в рабочей зоне; из-за доставки заготовок; из-за отсутствия работы.	+	-
Простои ТС, мин.	+	-
Число изделий в сменном задании, шт.	+	++
Станкочасовое время сменного задания, мин	+	++
Трудоемкость сменного задания (по циклограмме), мин	+	++
Коэффициент загрузки ГПЯ, %	+	++
Коэффициент использования ТС, %	+	++
Производительность ГПЯ, шт./ч.	+	++
Коэффициент увеличения срока окупаемости ГПЯ	+	++
Производные показатели эффективности ГПЯ**		
Себестоимость станко-часа, руб./ч.	+	++
Себестоимость обработки, руб./шт.	+	++
Срок окупаемости, лет	+	++
* — значение вводится пользователем. Показатели, непосредственно получаемые в программе: — не рассчитывается; + единичное значение показателя; ++ параметры вероятностного распределения показателя; ** — Показатели, расчет которых возможен с использованием полученных показателей и дополнительных данных		

6.3 Моделирование работы ГПЯ с несколькими транспортными средствами. Программа «PolyTrans»

Дальнейшим развитием алгоритмов моделирования на уровне технологической операции служит программа «PolyTrans».

Программа «Fania», использованная в предыдущей главе для проведения вычислительных экспериментов, предоставляет возможности для компьютерного моделирования ГПЯ при трех различных компоновках автоматического склада с использованием в качестве транспортных средств робокары или крана-штабелера, взаимодействующих с технологическим оборудованием по выбираемым из меню правилам обслуживания. Недостатком программы служит учет функционирования лишь одной транспортной единицы.

Вместе с тем логика программы «Fania» известна, что позволило усовершенствовать программу с тем, чтобы учитывать при моделировании ГПЯ возможность использования нескольких транспортных средств.

6.3.1 Ограничения и допущения на моделирование работы нескольких робокар

Вначале необходимо сформулировать ряд ограничений, которым будет удовлетворять разрабатываемый алгоритм моделирования.

1) Технически использование в ГПЯ нескольких ТС возможно по-разному. При использовании рельсового транспорта – кранов-штабелеров и рельсовых тележек [2, 6, 19] необходимо разделить склад и технологическое оборудование на зоны обслуживания. В противном случае возможны ситуации, когда тележки блокируют друг другу доступ к обслуживаемым станкам или ячейкам склада с нужными заготовками.

При разделении на зоны обслуживания внутри ГПЯ образуются несколько автономно работающих систем, каждая из которых обладает собственным запасом заготовок. Каждый фрагмент системы в данном случае может управляться отдельной АСУ и рассматриваться как отдельная ГПЯ.

При использовании безрельсовых тележек – робокар, снабженных средствами дистанционной навигации и обхода препятствий, становится возможным доступ любого ТС к любому станку и любой ячейке склада. В этом случае ГПЯ функционирует как единая система.

При разработке алгоритма моделирования будем рассматривать случай оснащения системы безрельсовыми тележками – робокарами.

2) Все используемые робокары имеют одинаковые технические параметры по длительности цикла смены паллет $T_{см}$ и скорости перемещения $V_{ТС}$. Ограничение представляется рациональным по причине того, что при создании ГПЯ дешевле изготовить или заказать несколько одинаковых робокар.

3) Функционирование робокар регламентируется общим для всех алгоритмом обслуживания технологического оборудования, который настраивается выбором из предлагаемого программой списка комбинаций правил обслуживания. Данное ограничение установлено с целью упрощения, а, значит, удешевления и повышения надежности АСУ ГПЯ.

4) Поступившая очередная заявка на обслуживание станка выполняется робокарой, имеющей на текущий момент времени наибольшие простои после предыдущей выполненной транспортной операции.

5) Время транспортной операции t_{mp} , выполняемой робокарой, описывается известными выражениями [56, 57, 61]. Случайное приращение времени, связанное с взаимными объездами робокар при пересечениях их маршрутов, будем учитывать как случайное нормальное распределение с заданным математическим ожиданием и среднеквадратичным отклонением (рисунок 6.11).

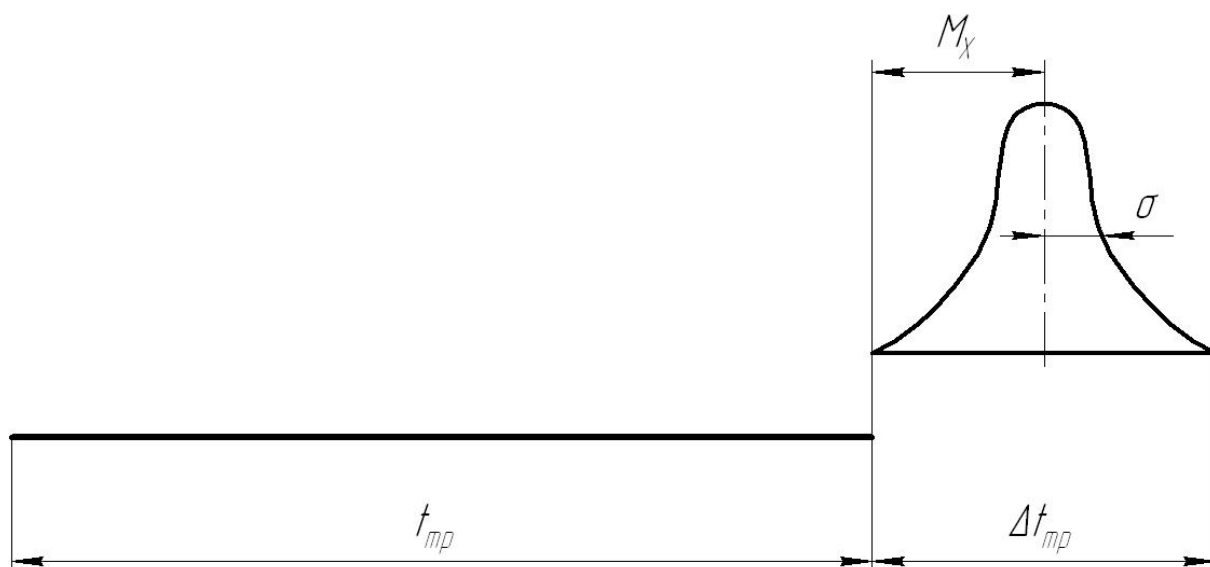


Рисунок 6.11 - Схема, иллюстрирующая случайный прирост времени транспортной операции из-за взаимных объездов роботов

6.3.2 Подход к моделированию ГПЯ с несколькими транспортными средствами

В работах [61, 62] описан алгоритм автоматизированного построения циклограмм автоматической работы технологического и сервисного оборудования, отражающий один из подходов к имитационному моделированию ГПЯ. На его основе разработан подробный алгоритм, реализованный в программном продукте Fania и представленный на рисунке 5.1.

Блок Dostawka символизирует выполнение транспортной операции по доставке очередной заготовки из позиции склада к станку, очередность обслуживания которого установлена в результате работы процедур блока Wybor. По умолчанию в блоке Dostawka подразумевается наличие в ГПЯ одного транспортного средства, которое и используется для обслуживания станков. Блок Dostawka представлен программными процедурами Wremq_TR и Perekodir программного модуля Model.pas программы Fania. В нем работа транспортного средства описана несколькими переменными:

KodRobot – код заготовки, установленный на платформе ТС;

Otpraw – порядковый номер станка, возле которого ТС находится после завершения очередной транспортной операции (позиция отправления ТС для следующей транспортной операции);

Pribyt – порядковый номер станка, обслуживаемого в текущей транспортной операции;

WreOtr – текущее отработанное робота-время (календарь событий транспортного средства);

ProstT – текущие простои ТС перед получением текущей заявки на обслуживание;

SumT – суммарные, или накопленные простои транспортного средства за цикл моделирования ГПЯ.

Для внедрения в программный код моделирования нескольких транспортных средств (число ТС обозначено идентификатором Kol_TS), потребовалось переопределить вышеперечисленные переменные в массивы переменных. Так, переменная Wreotr типа extended переопределена как массив WreOtr: array[1..Kol_TS] of extended. Аналогичные изменения коснулись и других переменных. Тем самым в состав моделируемой ГПЯ стало возможным ввести Kol_TS однотипных автономных транспортных средств.

Автоматический выбор из заданного множества транспортного средства для выполнения очередной транспортной операции реализован на основе выбора из календарей событий WreOtr того ТС, у которого отработанное время, а, следовательно, и момент завершения последней транспортной операции, минимально. Программно данное условие реализуется простым поиском минимального значения min из нескольких заданных значений массива WreOtr[1..Kol_TS]:

```

Procedure Wybor_TS;
Var
  min: double;
  i: integer;
begin
  Min:=WreOtr[1]; TS:=1;
  for i:=Kol_TS downto 1 do
    if Min>WreOtr[i] then
      begin Min:= WreOtr[i]; TS:= i; end;
    end;
  end.

```

Результатом выполнения данной процедуры служит порядковый номер транспортного средства TS, которому и назначается выполнение текущей транспортной операции.

Случайное приращение времени транспортной операции, вызванное возможными взаимными объездами робокар, генерируется специальной программной функцией Gauss, текст которой заимствован в Internet и принадлежит неизвестному автору:

```

function Gauss(Mx, Sigma: Extended): Extended;
  {распределение Гаусса}
  var
    a, b, r, Sq: Extended;
  begin
    repeat
      a := 2 * Random - 1;
      b := 2 * Random - 1;
      r := Sqr(a) + Sqr(b);
    until r < 1;
    Sq := Sqrt(-2 * Ln(r) / r);
    Result := Mx + Sigma * a * Sq;
  end;
end.

```

Для проверки работоспособности функции написан простой программный код, позволяющий визуально оценить ее работоспособность: для вводимых в режиме диалога исходных данных для функции Gauss строятся статистически обработанные распределения случайных значений. Пример одного из вариантов визуали-

зации результатов использования функции Gauss представлен в соответствии с рисунком 6.12. Представленные графические результаты не дают оснований усомниться в работоспособности процедуры и возможности ее использования.

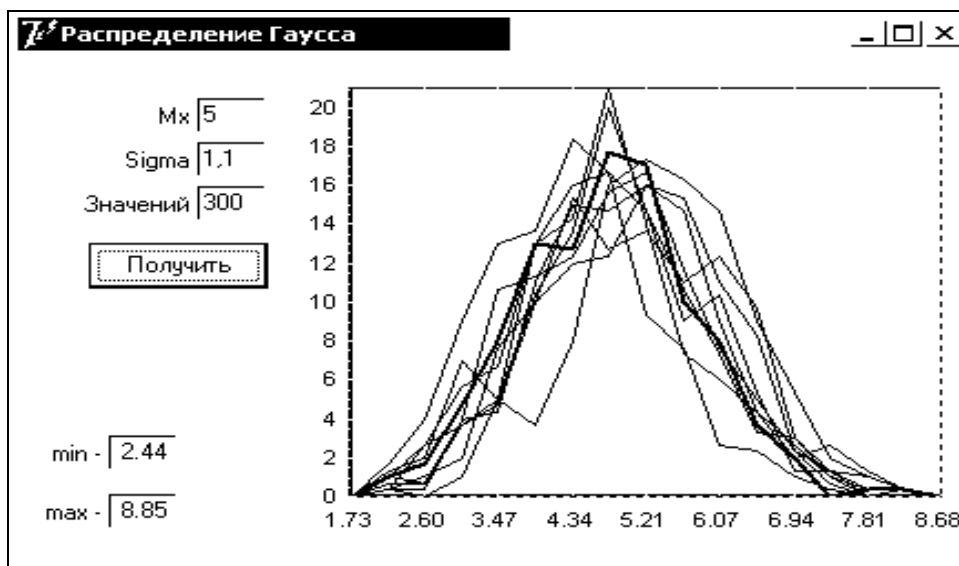
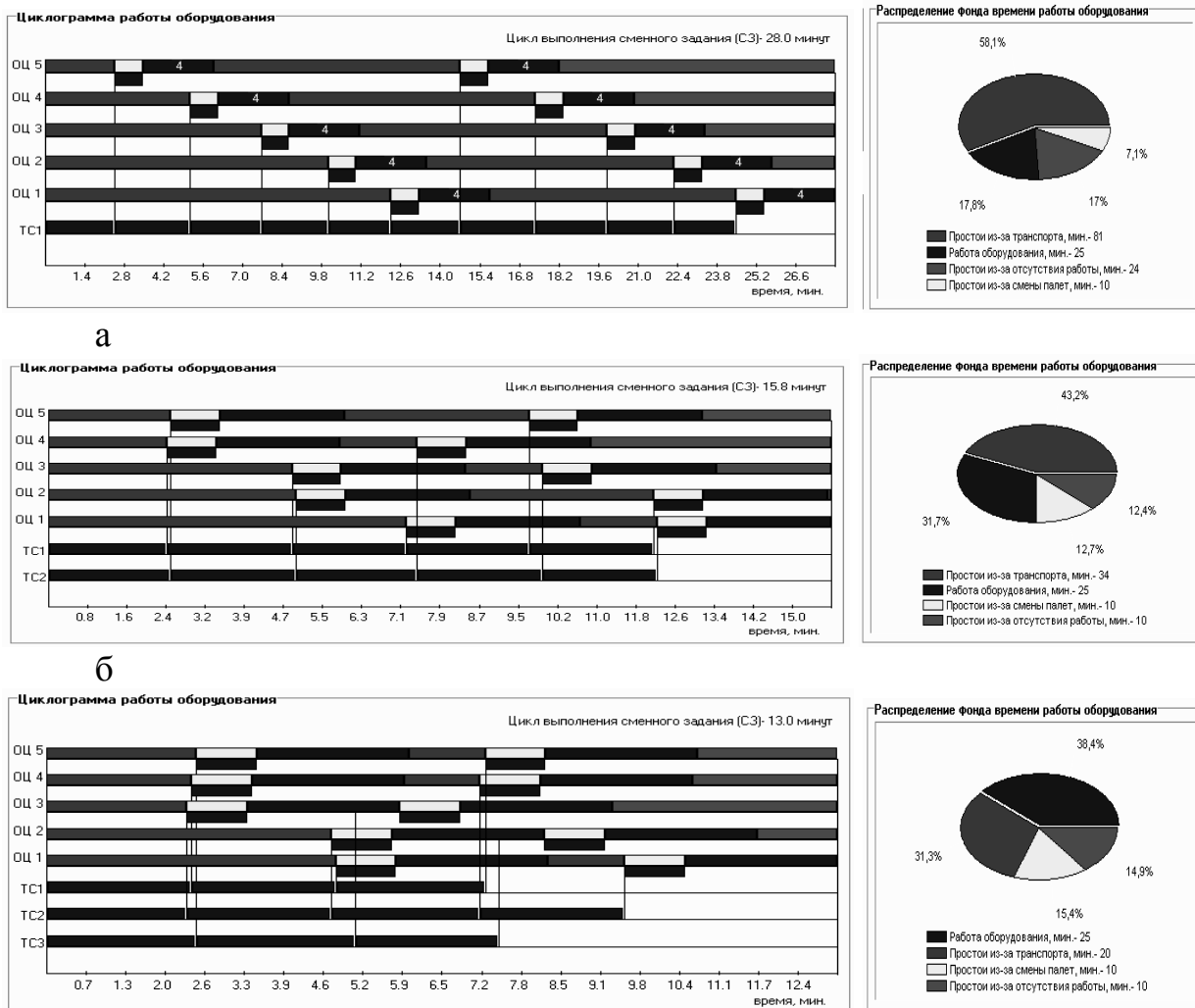


Рисунок 6.12 – Экранная форма процедуры проверки функции Gauss

Следующим шагом послужило изменение процедур графического вывода циклограмм работы оборудования ГПЯ для интерактивной настройки элементов свойства Canvas компонента VCL Image. В результате проделанной работы по динамическому масштабированию данных графическое отображение циклограмм стало визуализировать работу заданного количества транспортных средств. Для тестирования результатов работы программы PolyTrans подготовлен тестовый пример, результаты программного выполнения которого можно выполнить в ручном режиме. Исходные данные для тестового примера приведены в таблице 6.4 и на рисунке 6.13 а, б, в.

Таблица 6.4 – Исходные данные для тестирования программного кода

Число станков	5 шт. модели ИР800ПМФ4					Вариант а) – с одной робока- рой	Вариант б) – с двумя робока- рами	Вариант в) – с тремя робока- рами
Номер станка	1	2	3	4	5			
Технологическая группа	1	1	1	1	1			
Приоритет	1	1	1	1	1			
Координата, м	4,8	8,8	13,8	17,8	22,8			
Позиций в ПН, шт.	6	6	6	6	6			
Цикл смены паллет, с	60	60	60	60	60			
Тип склада	внешний							
Транспорт	двухместная робокара, скорость 1,0 м/с, цикл смены паллет – 60 с.							
Комбинация правил	FIFO2							
Изготавливаемые детали	10 шт.							
Номера деталей	4							
Трудоемкость (t _{шт.}), мин	2,5							



В

Рисунок 6.13 – Циклограммы работы (слева) и диаграммы распределения фонда времени (справа) работы оборудования при использовании в ГПЯ разного количества транспортных средств: а) $N_{TC} = 1$; б) $N_{TC} = 2$; в) $N_{TC} = 3$ робокары

6.3.3 Результаты тестирования программы «PolyTrans»

Для проверки корректности работы программных процедур, что должно найти отражение в правильном построении циклограмм и круговых диаграмм рисунка 6.13, выполним ручной расчет тестового примера для каждого варианта оснащения ГПЯ робокарами.

Для варианта «а» в таблице 6.4 и на рисунке 6.13 а, согласно комбинации правил FIFO2, с началом работы ГПЯ робокара выполняет транспортную операцию по доставке заготовки из перегрузочной позиции внешнего склада к 5-му станку. Время операции t_{TP} рассчитываем по формуле

$$t_{TP} = \frac{D_i + D_j}{V_{TC}} + 2 \times T_{\Pi}, \quad (6.1)$$

где $D_i = 4,8$ – координата перегрузочной позиции 1-го станка (по умолчанию

для первой транспортной операции), м;

$D_j=22,8$ - координата перегрузочной позиции обслуживаемого 5-го станка, м;

$V_{TC} = 1,0$ – скорость робокары, м/с;

$T_{ц} = 60$ – цикл смены паллеты на робокаре, с (таблица 6.4).

С учетом принятых числовых значений $T_{TP}(1)$ составит

$$T_{TP}(1) = \frac{4,8 + 22,8}{1,0} + 2 \times 60 = 147,6 \text{ с}$$

После перегрузки заготовки с робокары в накопитель 5-го станка отрабатывается цикл загрузки паллеты в рабочую зону со временем $T_C = 60$ с (1,0 мин). В момент времени $147,6 + 60,0 = 207,6$ с 5-й станок начинает обработку заготовки, трудоемкость которой задана как $t_{ум} = 2,5$ мин (таблица 6.4), или 150,0 с. Обработка данной заготовки заканчивается в момент времени t

$$t = T_{TP}(1) + T_C + t_{ум} = 147,6 + 60,0 + 150,0 = 357,6 \text{ с}$$

По аналогии для 2-ой транспортной операции согласно выражению (6.1) рассчитываем время ее выполнения. Из-за перемещения робокары теперь $D_i=22,8$ м – координата 5-го станка, где находится робокара, $D_j = 17,8$ м – координата обслуживаемого 4-го станка

$$T_{TP}(2) = \frac{22,8 + 17,8}{1,0} + 2 \times 60 = 160,6 \text{ с}$$

Обработка второй заготовки на 4-м станке закончится в момент времени

$$t = T_{TP}(1) + T_{TP}(2) + T_C + t_{ум} = 147,6 + 160,6 + 60,0 + 150,0 = 518,2 \text{ с}$$

Полученные результаты для первых двух транспортных операций, если их перевести в минуты, полностью совпадают с началом циклограммы на рисунке 6.13 а. Однако, поскольку циклограмма служит лишь средством визуализации, а исходными данными для ее построения служит файл протокола работы ГПМ (файл `ciklgr.mod`), то сравнение удобнее делать с данными этого файла (рисунок 6.14). Цветом выделены строки, соответствующие 1-ой и 2-ой транспортным операциям.

Выделенные на рисунке 6.14 строки полностью тождественны расчетам, выполненным вручную, поэтому для случая «а» на рисунке 6.13 нет оснований не доверять результатам моделирования.

Для варианта «б», отличающегося использованием 2-х робокар, начальное обслуживание станков ведется в той же последовательности, однако первые две транспортные операции начинаются одновременно. Вторая транспортная операция выполняется 2-ой робокарой, которая перемещается от позиции ожидания возле 1-го станка (по умолчанию) к 4-му станку. Время $T_{TP}(2)$ составит

$$T_{TP}(1) = \frac{4,8 + 17,8}{1,0} + 2 \times 60 = 142,6 \text{ с},$$

а обработка второй заготовки на 4-м станке закончится в момент времени

$$t = T_{TP}(2) + T_C + t_{um} = 142,6 + 60,0 + 150,0 = 352,6 \text{ с}$$

Полученные результаты соответствуют циклограмме на рисунке 6.13 б и данным файла Cijklgr.mod для варианта «б» таблицы 6.4.

Число станков в ГПЯ		Число транспортных средств		Время цикла загрузки/разг рузки станка					
5	1	60	60	60	60	60	60	60	60
транс опер.	обсл. станко	код ДУ	прост ТТ	время транс	момент достав	начало обраб.	конец обраб.	NN ТС	
1	5	4	0.0	47.6	147.6	207.6	357.6	1	
2	4	4	0.0	60.6	308.2	368.2	518.2	1	
3	3	4	0.0	51.6	459.8	519.8	669.8	1	
4	2	4	0.0	42.6	602.4	662.4	812.4	1	
5	1	4	0.0	33.6	736.0	796.0	946.0	1	
6	5	4	0.0	47.6	883.6	943.6	1093.6	1	
7	4	4	0.0	60.6	1044.2	1104.2	1254.2	1	
8	3	4	0.0	51.6	1195.8	1255.8	1405.8	1	
9	2	4	0.0	42.6	1338.4	1398.4	1548.4	1	
10	1	4	0.0	33.6	1472.0	1532.0	1682.0	1	

Рисунок 6.14 – Формат файла – протокола моделирования Cijklgr.mod:

Аналогичным способом проведено тестовое сравнение и для случая «в», отличающегося использованием 3-х робокар: распечатка файла протокола позволяет произвести при необходимости полную проверку каждого элементарного шага в процессе моделирования. При расчете длительности транспортных операций для точного учета составляющих времени был выключен режим случайного прираще-ния их длительности из-за взаимных объездов. Данный режим отдельно был иссле-дован ранее (рисунок 6.12).

Другим способом тестирования программных продуктов служит визуальное сравнение получаемых и логически ожидаемых результатов [15, 16]. В рас-

смаатриваемом случае следует ожидать, что увеличение количества робокар в системе должно привести к сокращению простоев станков из-за транспорта. Проверка данной закономерности может быть выполнена по данным на круговых диаграммах баланса времени (рисунок 6.13 а, б, в). Сводные данные по простоям станков из-за транспорта на круговых диаграммах представлены на рисунке 6.15. Судя по рисунку 6.15, графическая иллюстрация результатов моделирования совпадает с логически ожидаемой закономерностью.

Таким образом, результаты проведенного тестирования разработанного программного обеспечения, учитывающего работу нескольких транспортных средств в составе ГПЯ, подтверждают адекватность компьютерной модели как передаточной функции между входными данными и получаемыми результатами. Модернизированная программа приобрела возможности инструментального средства для оценки эффективности функционирования ГПЯ с несколькими единицами транспортных средств.

Количество робокар в ГПЯ, шт.	Простои станков из-за транспорта (рисунок 6.13 а, б, в)	
	мин	%
1	81	58,1
2	34	43,2
3	20	31,3

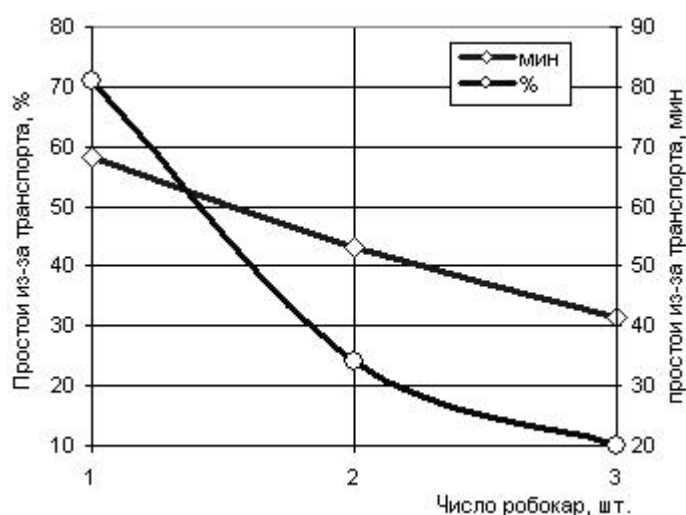


Рисунок 6.15 – Иллюстрация уменьшения простоев станков из-за транспорта при увеличении количества транспортных средств в ГПЯ

6.4 Моделирование работы участков традиционного производства. Программа «Расписание»

По данным одного из машиностроительных предприятий Оренбуржья, хранение производственных запасов составляет порядка 70 суток, а время добавления ценности на единицу продукции составляет лишь 3 часа. Подобную ситуацию нельзя считать приемлемой. Жесткая конкуренция заставляет предприятия изыскивать способы сокращения издержек производства, на практике решать многоуровневые и многокритериальные задачи оптимизации материальных и информационных потоков. Зачастую полученные решения служат основанием для реконструкции производства и выбора современных технологий формообразования изделий.

Решение задач оптимизации производства требует разработки компьютерных систем поддержки принятия решений (СППР), выступающих своего рода передаточными функциями между входными параметрами оборудования, технологических

процессов и диспетчерских решений, с одной стороны, и показателями эффективности производства, с другой (рисунок 6.16).

Практика показывает, что для создания СППР может быть использован формализованный аппарат, разработанный для моделирования гибких производственных ячеек в программных продуктах «Каскад», «Fania» и «PolyTrans».

Принципиальными отличиями производственных процессов в традиционном производстве и в гибких производственных ячейках являются:

- широкая номенклатура технологического оборудования, предполагающая наличие рабочих мест для выполнения как взаимозаменяемых, так и взаимодополняемых технологических операций;
- наличие технологических операций разделения (пруток разделяется на несколько заготовок) и слияния (сборочные операции);
- передача заготовок между технологическими операциями осуществляется передаточными партиями разных размеров;

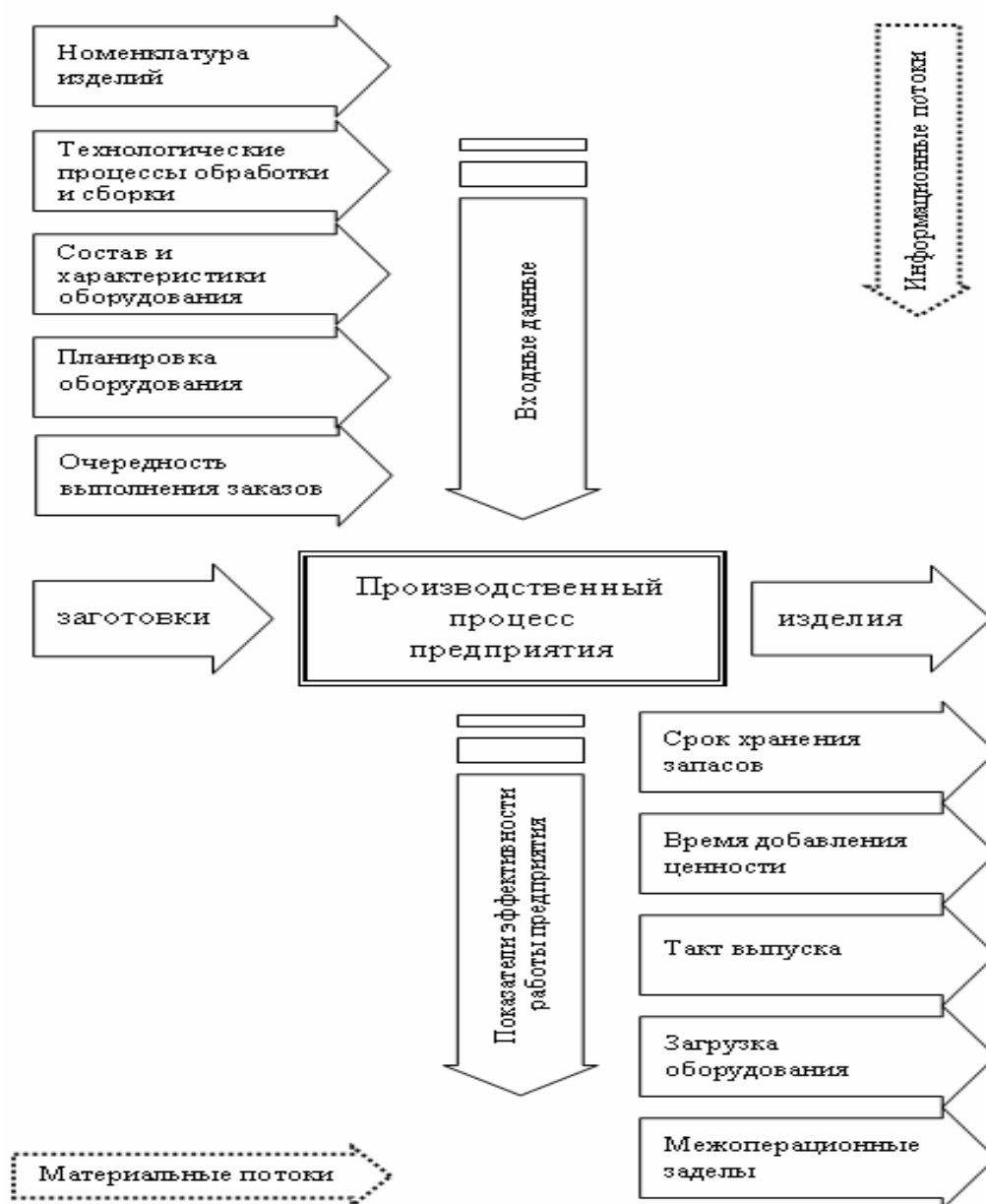


Рисунок 6.16 – Схема преобразования материальных и информационных потоков предприятия

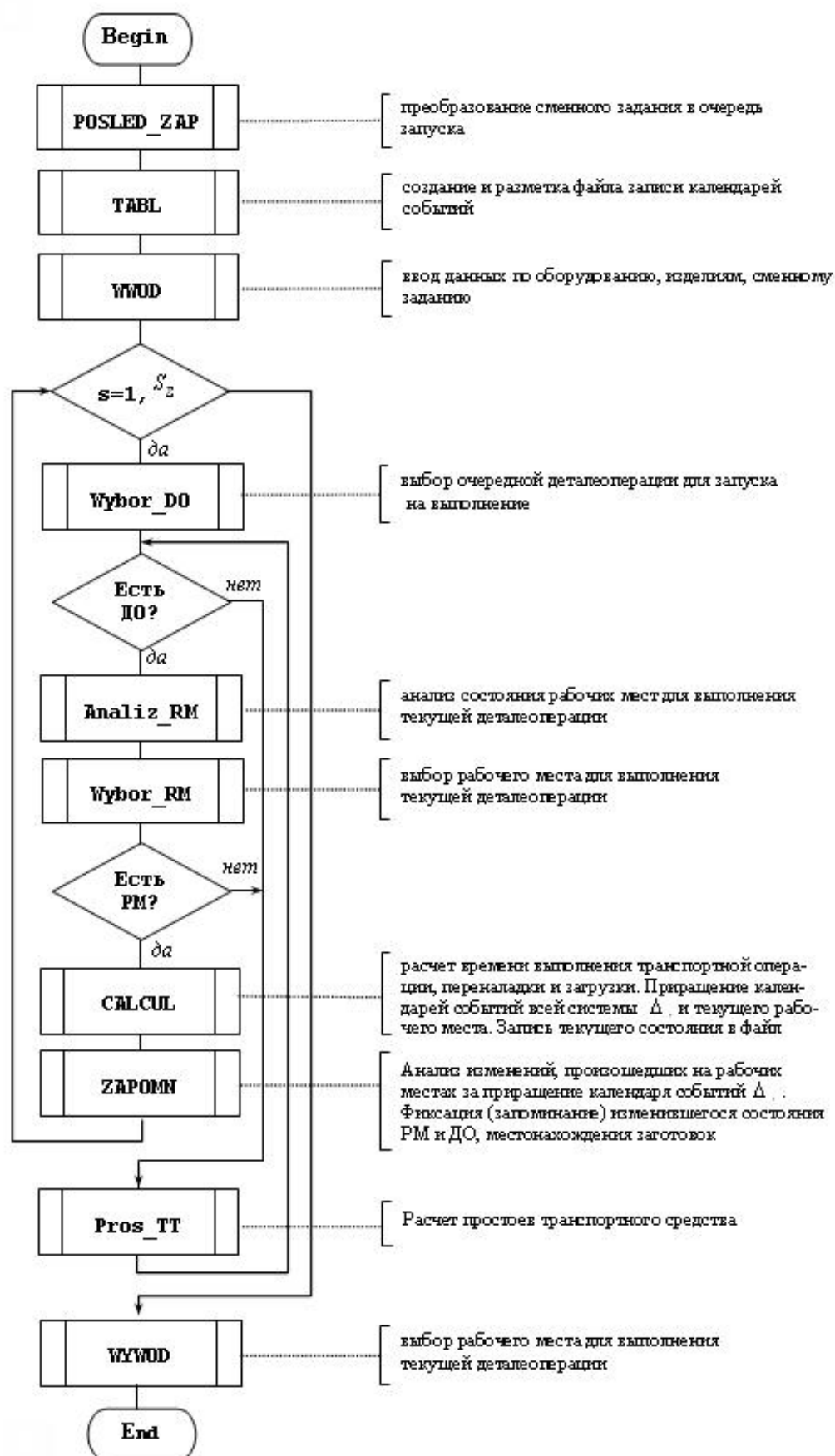


Рисунок 6.18 – Схема программы моделирования работы производственного участка традиционного производства

Программная реализация представлена в виде программы «Расписание», экранная форма ввода данных представлена в соответствии рисунком 6.19.

Описание рабочих мест (PM)

Число PM, штук: 10

Номера станков =>	1	2	3	4	5
Группа станков	1	1	2	3	2
Позиций в ПН, шт	4	4	4	4	4
Цикл загрузки, сек	50	50	50	50	50
Координата станка, м	5	10	15	20	25

Описание изделий

Групп изделий, всего: 3

Описание деталей 2 группы:

Группа станков =>	1	2	3
Маршрут (приоритеты)	1	1	1
Время обработки, мин	5	4	7

Группа изделий: 2

Описание трансп. средств (ТС)

Число ТС, штук: 2

Скорость, м/сек	2
Цикл смены, сек	60

Описание партий запуска (ПЗ)

Число ПЗ, шт: 7

№№ партий запуска=>	1	2	3	4	5	6
Группа деталей	1	3	2	1	2	3
Число деталей, шт.	100	50	50	10	10	10

Buttons: Выход, Загрузить, Сохранить, Расчет

Рисунок 6.19 - Форма ввода данных в программе «Расписание»

В листинге программы предусмотрена возможность настройки предельных значений входных данных моделируемого подразделения (рисунок 6.20).

```

unit Varr;

interface

Const
    { *** НАСТРОЙКА ПРЕДЕЛЬНЫХ ЗНАЧЕНИЙ ВХОДНЫХ ДАННЫХ *** }

    Stan_D = 100; // допустимое число станков
    Grup_Stan = 20; // допустимое число технологических групп станков
    Nakop_D = 100; // допустимое число позиций в пристаночном накопителе
    Nomkl_D = 20; // допустимая номенклатура изделий
    Zagot_D = 2000; // допустимое число изделий в СЗ
    Part_Zap = 20; // допустимое число партий запуска в СЗ
    Trans_D = 5000; // допустимое число транспортных операций
    Kol_TS_D = 30; // допустимое число транспортных средств

Var
    Koord: array[0..Stan_D] of real; // координата станка
    Gruppa_Stan: array[1..Stan_D] of integer; // техн. группа станка
    Kod_DU: array[1..Nomkl_D, 1..Grup_Stan] of integer;
    Wremq_DU: array[1..Nomkl_D, 1..Grup_Stan] of real;
    KodPZ, RazmerPZ: array[1..Part_Zap] of integer;
    Nomenkl, PZ, Grupp, Stanki, Gruppa_D, Kol_TS, TS,
    PN: integer;
    Smena_DU, Skor_TT, Kzag,
    T_sm: real;
    Var_Time: extended; // текущее модельное время

implementation
  
```

Рисунок 6.20 - Программный модуль настройки предельных значений входных данных программы «Расписание»

Использованные идентификаторы приведены в таблице 6.5.

Таблица 6.5 - Идентификаторы программы «Расписание»

Идентификатор	Тип идентификатора	Смысловое значение идентификатора
1	2	3
Глобальные переменные (модуль Varr)		
Nomenkl	integer	Номенклатура (число групп) деталей на участке
PZ	integer	Число партий запуска деталей в сменном задании
Grupp	integer	Число групп станков
Stanki	integer	Число станков
Gruppa_D	integer	Текущая группа деталей
PN	integer	Число позиций в пристаночном накопителе
Smena_DU	real	Время смены заготовки на станке
Skor_TT	real	Скорость тележки
T_sm	real	Время смены заготовки на тележке
Koord	array[0..10] of real	Координата станка
Gruppa_Stan	array[1..10] of integer	Технологическая группа станка
Kod_DU	array[1..20,1..10] of integer	наличие и приоритет технол. операции i-ой группы деталей на j-ой группе оборудования i – идентификатор группы деталей j – идентиф. группы станков
Wremq_DU	array[1..20,1..10] of real	время выполнения технол. операции на детали i-ой группы на j-ой группе оборудования i – идентификатор группы деталей j – идентиф. группы станков
KodPZ	array[1..10] of integer	код партии запуска (код группы деталей)
RazmerPZ	array[1..10] of integer	размер партии запуска (число заготовок в группе)
Глобальные переменные процедуры моделирования Model_ras (модуль Calcul)		
Zapret_1	array [1..500] of integer	запрет (=1) или разрешение (=0) на запуск очередной заготовки в обработку
Posled		код заготовки в общей последовательности в СЗ запуска в обработку
NomT		порядковый номер заготовки (1,2,3,4,...) в общей последовательности запуска в обработку
KodSklad		код детали в ячейке склада
KodT		код заготовки в общей последовательности в СЗ запуска в обработку (вначале то же, что и Posled)
Trud	array [1..500] of real	трудоемкость изготовления детали как сумма трудоемкости всех операций
Transp		время выполнения транспортной операции
MomentZap		момент запуска новой заготовки в обработку
MomentSklad		момент установки готовой детали в ячейку склада
PutT	array [1..500,1..10] of integer	наличие приоритет технологической операции i – порядковый номер детали в СЗ j – идентификатор группы станков
WreT	array [1..500,1..10] of real	время выполнения технологической операции i – порядковый номер детали в СЗ j – идентификатор группы станков
KodStol	array [1..20] of integer	код заготовки в рабочей зоне станка i – номер станка

Продолжение таблицы 6.5

1	2	3
SumPR_S	array [1..20] of real	суммарные простои i-го станка к текущему моменту времени
ProStan		текущие простои i-го станка в текущий момент времени
Konec		текущее отработанное i-м станком время
MomentStol		момент времени подачи очередной заготовки в рабочую зону i-го станка
Zapros	array [1..20] of boolean	наличие запроса (true, false) от i-го станка на доставку заготовки
Zapret		??
Priznak		признак того, что заготовка на i-м станке в j-й позиции накопителя обработана (1) или нет (0)
KodPN		код заготовки на i-м станке в j-й позиции накопителя
MomentPN	array [1..20, 1..20] of integer	момент времени установки в накопитель заготовки, находящейся на i-м станке в j-й позиции накопителя
obslug		номер обслуживаемого станка
KodRobot		код заготовки на платформе ТТ
zaqwka		порядковый номер заготовки из СЗ, запускаемой в обработку
Otpraw	integer	станок отправления
Pribyt		станок прибытия
sos		номер текущей транспортной операции
sklad		порядковый номер очередной заготовки, поступившей на склад
Razmer_SZ		число заготовок в сменном задании
poz1		текущий номер позиции накопителя станка
zz		признак (0 или 1) необходимости перевода строки в файле результатов
poz		номер позиции накопителя станка
DU		код заготовки, удаляемой из накопителя станка после ее обработки
SumT		общее число транспортных операций с учетом неоднократной перевозки каждой заготовки от станка к станку
ii		??
ProstTT		текущие простои транспортного средства
SumTT		суммарные простои транспортного средства
Nomin		плановая (номинальная) трудоемкость сменного задания
WreOtr		текущее время в системе
Sum		суммарное модельное время
Wremq		время выполнения текущей транспортной операции
FF1	textFile	имя файла для записи протокола событий

Исходными данными для работы программы «Расписание» служат данные по оборудованию (рисунок 6.19), по технологическим процессам и по сменному заданию.

Данные по оборудованию включают технические характеристики станков (рабочих мест), их принадлежность к определенной технологической группе и привязку к планировке производственного подразделения (рисунок 6.21).



Рисунок 6.21 – Возможная планировка механосборочного цеха

Также задается количество обслуживающих транспортных средств и их характеристики. Под транспортными средствами могут пониматься рабочие, осуществляющие на ручных или механизированных тележках передачу партий полуфабрикатов между рабочими местами подразделения.

Исходная информация по технологическим процессам задаются по каждому наименованию изделия, включенному в производственную программу.

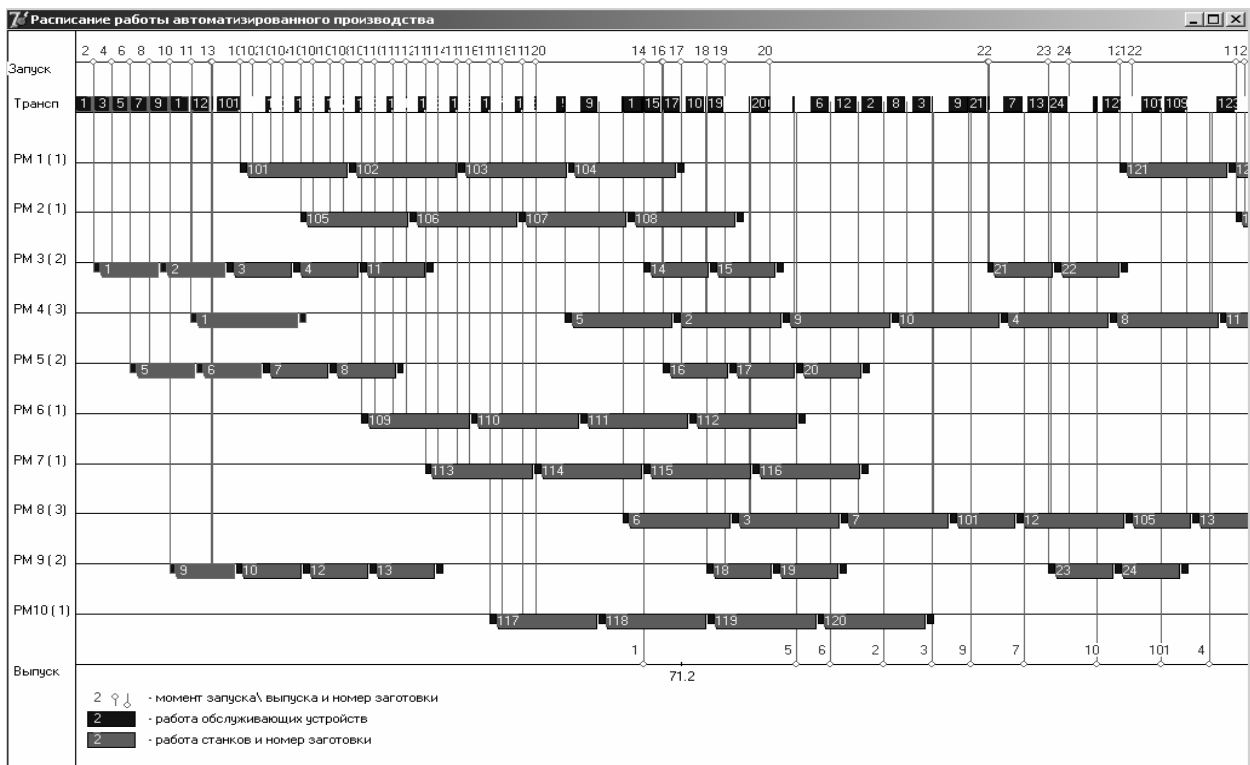


Рисунок 6.22 – Циклограмма работы участка

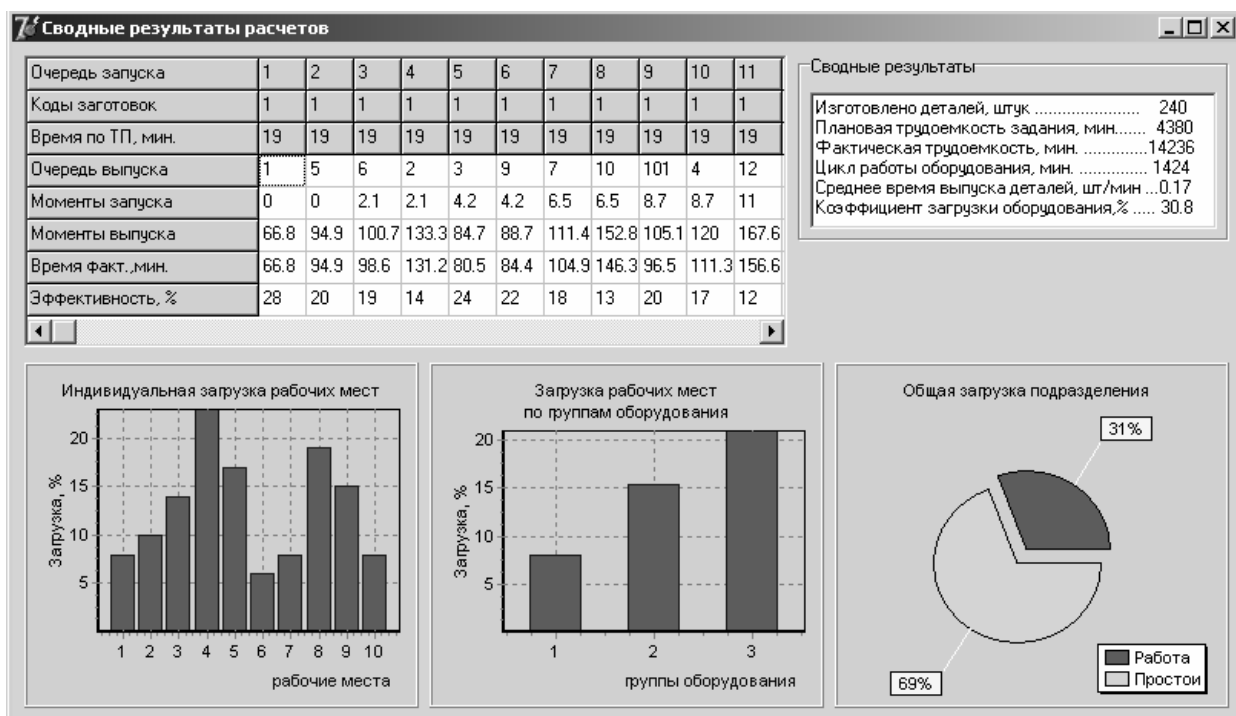


Рисунок 6.23 – Сводные результаты расчета эффективности работы участка

Информация включает список технологических операций, очередность их выполнения в виде приоритетов и время выполнения (трудоемкость).

Сменное задание вводится в виде матричной таблицы, в которой указываются коды и размеры последовательных партий запуска заготовок в обработку. Результаты моделирования отображают автоматически сформированное расписание работы подразделения при выполнении текущего сменного задания (рисунок 6.22), а так же сводные результаты об эффективности использования отдельных рабочих мест, групп оборудования и подразделения в целом (рисунок 6.23).

В дальнейшем предполагается наращивание возможностей программы в рамках общей концепции применения метода циклограмм для оптимизации проектных параметров и эксплуатационных режимов производственных систем с разным уровнем автоматизации (рисунок 6.24).

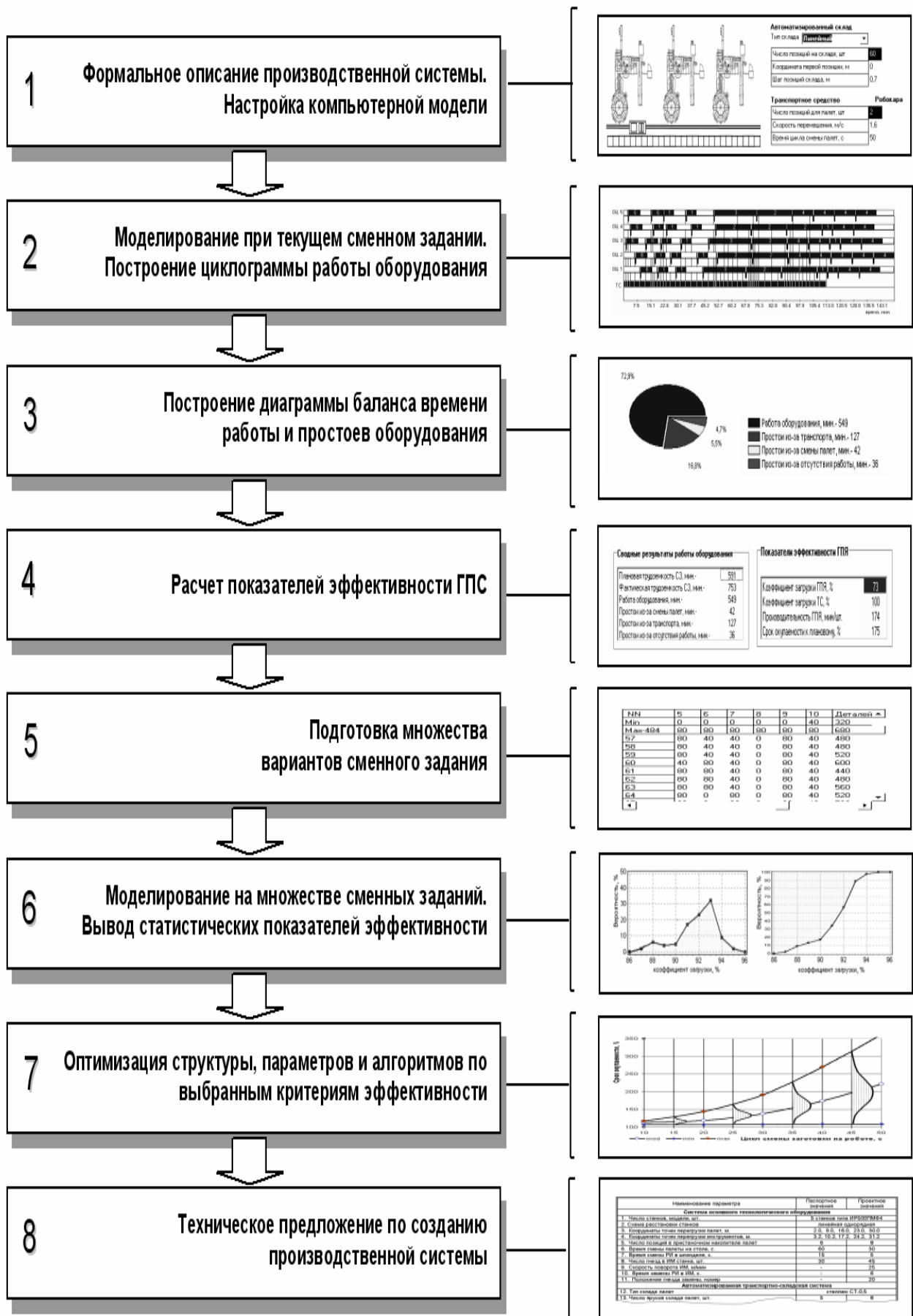


Рисунок 6.24 – Последовательность синтеза проектных параметров и эксплуатационных режимов производственных систем

Заключение

Одним из направлений технического перевооружения машиностроительного производства служит создание и внедрение компьютерно управляемых комплексов оборудования, согласно ГОСТ 26228-90 именуемых гибкими производственными ячейками и системами.

Отсутствие этапа создания опытного образца при проектировании подобных систем предъявляет повышенные требования к методам их расчета.

В представленном проекте разрабатывается новый метод предпроектных расчетов ГПЯ, основанный на компьютерном моделировании и анализе эффективности функционирования системы при различных вариантах проектных параметров и эксплуатационных режимов.

Разработано формализованное представление компьютерных моделей, основанное на аналитическом описании работы и логическом описании многоуровневого взаимодействия элементов производственной системы.

Для аналитического описания работы гибкой производственной ячейки разработан комплекс математических моделей, учитывающих технические характеристики оборудования и его агрегатов, типы применяемых транспортно-складских систем и компоновочные решения системы.

Логическое описание многоуровневого взаимодействия элементов системы представлено в виде алгоритмов их взаимодействия на иерархических уровнях технологической операции и технологического перехода при использовании различных правил обслуживания, используемых в АСУ.

Для количественной оценки эффективности работы ГПЯ по результатам моделирования обоснован выбор и предложено математическое представление основных показателей эффективности – коэффициента загрузки технологического оборудования и производительности системы. Получено аналитическое выражение для расчета комплексного показателя – срока окупаемости затрат на создание системы.

Для повышения степени объективности и достоверности результатов вычислительного эксперимента разработан метод статистического моделирования, основанный на итерационном формировании вариантов сменного задания и статистической обработке получаемых значений показателей эффективности. Формализованное представление метода обеспечивают разработанные математические модели и схемы алгоритмов.

Совокупность полученных результатов представляет собой иерархический комплекс компьютерных моделей, преобразующих параметры оборудования, технологии формообразования, алгоритмы управления и режимы эксплуатации в комплекс показателей эффективности производственной системы.

Примеры программной реализации разработанного метода в виде программ «Каскад», «Fania», «PolyTrans» и «Расписание» доказывают возможность его применения как при создании и эксплуатации высокоавтоматизированных систем нового поколения, так и при минимизации производственных потерь машиностроительных подразделений с традиционным универсальным оборудованием.

Список использованных источников

- 1 **Абчук, В.А.** Управление в гибком производстве / В.А. Абчук, Ю.С. Карпенко. - М.: Радио и связь, 1990. – 128 с.
- 2 **Азбель, В.О.** Организационно-технологическое проектирование ГПЯ / В.О.Азбель, А.Ю. Звоницкий, В.Н. Каминский. – Л.: Машиностроение, 1986. – 294 с.
- 3 **Баховский, Л.Ф.** Сокращение срока окупаемости затрат на ГПЯ / Л.Ф. Баховский // АСТ : Автоматизация и современные технологии. - 1998. - №6. - С. 35-38.
- 4 **Бондаренко, В.А.** Основы создания ГПЯ механообработки: учебное пособие для вузов / В.А. Бондаренко, А.И. Сердюк. - Оренбург: ОГУ, 2001. – 206 с.
- 5 **Васильев, В.Н.** Организация, управление и экономика гибкого интегрированного производства в машиностроении / В.Н. Васильев. - М.: Изд. Стандартов, 1987. - 288 с.
- 6 **Вильчевский, Н.О.** Автоматизированные транспортно-складские системы ГПЯ механообработки: учебное пособие / Н.О. Вильчевский, О.Б. Маликов, С.Ф. Пилипчук. - Л.: Изд. ЛПИ, 1988.- 92 с.
- 7 Гибкие производственные системы (ГПЯ) и интегрированные компьютеризированные производства (КИП) [Электронный ресурс]. - Режим доступа : WWW.URL : http://www.cals.ru/annotation/concept_R/history/index.html.
- 8 **Белянин, П.Н.** Гибкие производственные комплексы / под ред. П.Н. Белянина, В.А. Лещенко - М.: Машиностроение, 1984. - 384 с.
- 9 Гибкое автоматическое производство / В.О. Азбель [и др.]. - Л.: Машиностроение, 1985. - 454 с.
- 10 Гибкие производственные системы Японии. - М.: Машиностроение, 1987. - 230 с.
- 11 **Гнеденко, Б.В** Приоритетные системы обслуживания / Б.В. Гнеденко [и др.]. – М.: МГУ, 1973. - 446 с.
- 12 **Горнев, В.Ф.** Оперативное управление в ГПЯ / В.Ф. Горнев, В.В. Емельянов, М.В. Овсянников. - М.: Машиностроение, 1990. - 256 с.
- 13 **ГОСТ 26228-90.** Системы производственные гибкие. Термины и определения. Номенклатура показателей. – Введ. 01.01.1991. - М.: Изд-во стандартов, 1990. - 12 с.
- 14 **ГОСТ 34.601-90.** Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы стадии создания. – Введ. 01.01.1992. - М.: Издательство стандартов, 2002. – 6 с.
- 15 **ГОСТ 34.603-92.** Информационная технология. Виды испытаний автоматизированных систем. – Введ. 01.01.1993. - М.: Издательство стандартов, 2002. – 6 с.
- 16 **ГОСТ 19.301-79.** ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению. – Введ. 01.01.1981. - М.: Издательство стандартов, 2005. – 2 с.
- 17 ГПЯ на малом предприятии [Электронный ресурс] // Tramelal. - 2002. - Nr. 67. - С. 35 – 37. - Режим доступа : WWW.URL : <http://www.stankoinform.ru /15.CNC-2.htm>.
- 18 **Грачев, Л.Н.** Автоматизированные участки для точной размерной обработки деталей / Л.Н. Грачев, Д.С. Гиндин. - М.: Машиностроение, 1981. - 240с., ил.

19 **Хартли, Д.** ГПЯ в действии / Джон Хартли : пер. с англ. - М.: Машиностроение, 1987. - 328 с.

20 **Дорнан, С.Б.** Экономическая эффективность ГПЯ / С.Б. Дорнан // Технологии, оборудование, организация и экономика машиностроительного производства. Серия 1. Автоматизация производства, гибкие производственные системы и робототехника. Зарубежный опыт. Экспресс-информация. - М.: ВНИИТЭМР, 1988. - Вып. 24. - С. 37-41.

21 **Дудорин, В.И.** Информатика в управлении производством: учебник / В.И. Дудорин. - М.: Издательство «Менеджер», 2001. - 360 с.

22 **Егоров, В.А.** Транспортно-накопительные системы для ГПЯ / В.А. Егоров, В.Д. Лузанов, С.М. Щербаков. - Л.: Машиностроение, 1989. - 293 с.

23 **Емельянов, В.В.** Введение в интеллектуальное имитационное моделирование сложных дискретных систем и процессов / В.В. Емельянов, С.И. Ясиновский. - М.: АНВИК, 1998. - 427 с.

24 Принятие оптимальных решений в интеллектуальных имитационных системах: учебное пособие / В.В. Емельянов [и др.]. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. - 60 с.

25 **Жданович, В.Ф.** Комплексная механизация и автоматизация в механических цехах / В.Ф. Жданович, Л.Б. Гай. - М.: Машиностроение, 1976. - 288 с.

26 **Иванов, А.А.** Интегрированные производственные системы: Автоматизация информ. потоков : учеб. пособие / А.А. Иванов, А.А. Москвичев, А.Р. Кварталов. - Н. Новгород: НГТУ, 1997. - 76 с.

27 **Ивченко, Г.И.** Теория массового обслуживания: учеб. пособие для вузов / Г.И. Ивченко, В.А. Каштанов, И.Н. Коваленко. - М.: Высшая школа, 1982. - 256 с.

28 **Колосов, В. Г.** Инструментальная сеть наукоемкого инжиниринга (о республиканской программе НТП «Создание центров инженерного обеспечения машиностроительных производств» / В. Г. Колосов, И. Л. Туккель. - СПб.: ДНТП, общество "Знание", 1992. - 22 с.

29 Институт Автоматизации и Системных Исследований (ASRI) [Электронный ресурс]. - Режим доступа : WWW.URL : <http://www.polarcom.ru/~vvtsv/ref2000/r023.htm>.

30 Информационно-вычислительные системы в машиностроении CALS-технологии / Ю.М. Соломенцев, В.Г. Митрофанов, В.В. Павлов, А.В. Рыбаков. - М.: Наука, 2003. - 292 с.

31 Итоги заседания совета по CALS РОСАВИАКОСМОСА [Электронный ресурс]. - Режим доступа : WWW.URL : <http://www.mati.ru/magazine/archive/102003/main.php>.

32 **Кетов, А.В.** Особенности маркетинга станкостроительной и машиностроительной продукции и пути решения задачи : Материалы Интернет-конференции «Стратегия развития Дальнего Востока России : возможности и перспективы» [Электронный ресурс] / А.В. Кетов, В.М. Давыдов. - Режим доступа : WWW.URL : <http://www.feststrategy.ru/materials.php?action=show&id=212> .

33 **Клейнрок, Л.** Теория массового обслуживания / Л. Клейнрок. - М.: Машиностроение, 1979. - 432 с.

34 **Конвей, Р.В.** Теория расписаний / Р.В. Конвей, В.Л. Максвелл, Л.В. Миллер. - М.: Наука, 1975. - 360 с.

35 **Колосов, В. Г.** Концепция развития в России распределенной инфраструктуры комплексной автоматизации / В. Г. Колосов. - СПб.: "Политехника", 1991. - 19 с.

36 **Кривошеев, И.А.** Внедрение компонентов CALS-технологии в авиадвигателестроении: проблемы и перспективы [Электронный ресурс] / И.А. Кривошеев // Инженерное образование: электронный журнал. – 2005, №4. - Режим доступа : [WWW.URL : http://www.techno.edu.ru:16000/db/msg/24972.html](http://www.techno.edu.ru:16000/db/msg/24972.html).

37 Критические технологии федерального уровня [Электронный ресурс]. - Режим доступа : [WWW.URL : http://www.extech.ru/s_e/min_s_niokr/krittech/annot/2-05.htm](http://www.extech.ru/s_e/min_s_niokr/krittech/annot/2-05.htm).

38 **Лескин, А.А.** Гибкие автоматические производства. Спецификация внешних требований при проектировании систем управления / А.А. Лескин, В.М. Пономарев, А.М. Спиридонов. - Л.: ЛИИАН, 1986. - 24 с.

39 **Лищинский, Л.Ю.** Структурный и параметрический синтез гибких производственных систем / Л.Ю. Лищинский. - М.: Машиностроении, 1990. - 312 с.

40 **Малышко, И.А.** Влияние гибких автоматизированных систем на эффективность машиностроительного производства [Электронный ресурс] / И.А. Малышко. - Режим доступа : [WWW.URL : http://www.uran.donetsk.ua/~masters/2002/mech/poloka/lib/stat5.htm](http://www.uran.donetsk.ua/~masters/2002/mech/poloka/lib/stat5.htm).

41 **Земсков, Г.Г.** Металлорежущие системы машиностроительных производств / под ред. Г.Г. Земскова, О.В. Таратынова. - М.: Высшая школа, 1988. - 464 с.

42 Гибкие производственные системы / Н.П. Меткин, М.С. Лапин, С.А. Клейменов, В.М. Критский. - М.: Изд-во стандартов, 1989. - 312 с.

43 **Назаретов, В.М.** Моделирование гибких производственных систем // Сб. Робот. Компьютер. Гибкое производство / под ред. И.М. Макарова. – М.: Наука, 1990. – 176 с. (Серия «Кибернетика – неограниченные возможности и возможные ограничения»).

44 ОАО НИТИ «ПРОГРЕСС» [Электронный ресурс]. - Режим доступа : [WWW.URL : http://www.niti-rogress.ru/development.htm](http://www.niti-rogress.ru/development.htm).

45 Основы политики Российской Федерации в области развития науки и технологий на период до 2010 года и дальнейшую перспективу: документ правительства РФ Пр-576 от 30 марта 2002 г. [Электронный ресурс]. - Режим доступа : [WWW.URL : http://www.cals.ru](http://www.cals.ru).

46 **ОСТ 107-460641.002-87.** Система управления автоматизированной гибкой производственной системы. Общие требования к интерфейсу. – Введ. 01.07.1988. – 21с.

47 **Подберезкин, А.И.** Белая Книга российских спецслужб / А.И. Подберезкин. - 2-е изд., перераб. - М. : Информ.-изд. агентство "Обозреватель", 1996. - 268 с.

48 **Пуш, В.Э.** Автоматические станочные системы / В.Э. Пуш, Р. Пигерт, В.Л. Сосонкин. - М.: Машиностроение, 1982. – 319 с.

49 **Раимов, Ф.Ф.** Разработка метода и алгоритмов решения задач составления расписаний в подсистемах АСУП : автореферат дис. канд. тех. наук / Ф.Ф. Раимов. - Оренбург: ИПК ОГУ, 2005. – 18 с.

50 ГОСТ Р 50.1.028-2001. Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. – Введ. 01.07.2002. - М.: Издательство стандартов, 2001. – 53 с.

51 РД 50-34.698-90. Автоматизированные системы. Требования к содержанию документов. – Введ. 01.01.1992. - М.: Издательство стандартов, 2007. – 29 с.

52 Соломенцев, Ю.М. Роботизированные технологические комплексы и гибкие производственные системы в машиностроении: Альбом схем и чертежей: учебное пособие для втузов / под ред. Ю.М. Соломенцева. – М.: Маш., 1989. - 192 с.

53 Козырева, Ю.Г. Роботизированные комплексы «оборудование-робот» стран-членов СЭВ: Каталог / под общ. ред. Ю.Г. Козырева. - М.: НИИМаш, 1984. - 173 с.

54 Родионов, С.В. Проектирование оптимальных расписаний: учебное пособие / С.В. Родионов, М.В. Москина. – М.: МГТУ им. Н.Э. Баумана, 2002. – 19 с.

55 Сергеев, А.И. Использование систем моделирования “ANYLOGIC” и «КАС-КАД» в курсе автоматизации производства [Электронный ресурс] / А.И. Сергеев. - Режим доступа : WWW.URL : http://agpi.armavir.ru/institut/kaf/ped_kaf/SRS_konf/Pages/Members/Sergeeva.htm.

56 Сердюк, А.И. Интегрированная система моделирования ГАУ механообработки/ А.И. Сердюк // Станки и инструмент. - 1994. - № 3. - С. 2 – 4.

57 Сердюк, А.И. Моделирование производственного процесса ГПЯ / А.И. Сердюк // СТИН. – 1994. - № 11. - С. 11-13.

58 Сердюк, А.И. Интегрированная система моделирования и расчета производственных участков станков с ЧПУ и ГПЯ : учебное пособие для вузов / А.И. Сердюк, В.А. Гречишников. - Оренбург: ОГУ, 1994. – 255 с.

59 Сердюк, А.И. Оценка влияния решений на качество функционирования ГПЯ/ А.И. Сердюк, Л.Ф. Баховский // Автоматизация и современные технологии. - 1998. - № 7. - С. 29 – 32.

60 Сердюк, А.И. Проектирование гибких производственных систем с заданным сроком окупаемости / А.И. Сердюк, А.И. Сергеев // СТИН. – 2005. - № 11. – С. 20-25.

61 Сердюк, А.И. Метод циклограмм в построении компьютерных моделей ГПЯ / А.И. Сердюк, А.И. Сергеев // Автоматизация и современные технологии. – 2005. - № 11. – С. 17-23.

62 Сердюк, А.И. Электронный учебный курс «Основы создания ГПЯ механообработки» / А.И. Сердюк, А.И. Сергеев, М.А. Корнипаев // Отраслевой фонд алгоритмов и программ. - М. - Код 02069024.00033-01 от 11.04.2005.

63 Сердюк, А.И. Переход от технического задания к техническому предложению на создание ГПЯ: учебное пособие для вузов / А.И. Сердюк, Л.В. Карагулова. - Оренбург: ГОУ ОГУ, 2005. - 130 с.

64 Сердюк, А.И. К проблеме подготовки инженеров в области гибких производственных систем / А.И. Сердюк // Машиностроение и инженерное образование. – 2005. - № 4. – С. 52-61.

65 Сердюк, А.И. Проектирование гибких производственных систем с заданным сроком окупаемости / А.И. Сердюк, А.И. Сергеев // Станки и инструмент. – 2005. - №11. – С. 20-25.

66 **Сморodinский, С.С.** Оптимизация решений на основе методов и моделей мат. программирования: учеб. пособие / С.С. Смородинский, Н.В. Батин. – М.: БГУИР, 2003. – 136 с.

67 **Соломенцев, Ю.М.** Управление гибкими производственными системами / Ю.М. Соломенцев, В.Л. Сосонкин. - М.: Машиностроение, 1988. - 351 с.

68 Станки, современные технологии и инструмент для металлообработки [Электронный ресурс]. - Режим доступа : WWW.URL : <http://www.stankoinform.ru/Stanki6.htm>.

69 **Рощина, Н.** Станкоинструментальная отрасль России [Электронный ресурс] / Нина Рощина; ПромМаш-инструмент. - Режим доступа : WWW.URL : http://www.instrument.spb.ru/zurnals/23/zurnal_23Theme.shtml.

70 **Танаев, В.С.** Теория расписаний. Многостадийные системы / В.С. Танаев, Ю.Н. Сотсков, В.А. Струевич. – М.: Наука, 1989. – 328 с.

71 **Танаев, В.С.** Введение в теорию расписаний / В.С. Танаев, В.В. Шкурба. - М.: Наука, 1975. – 256 с.

72 Теория расписаний (Shedule theory) [Электронный ресурс]. - Режим доступа : WWW.URL : <http://www.math.spbu.ru/en/chairs/dep09.html>.

73 **Фроман, Б.** ГПЯ в механической обработке / Б. Фроман, Ж.Ж. Лезаж ; пер. с франц. Н.А.Шнуровой; под ред. В.А.Лещенко. - М.: Машиностроение, 1988. - 120 с.

74 Учебно-методическое обеспечение внедрения CALS-технологии на предприятиях МИНАТОМа России и создание отраслевого учебно-научного центра МИФИ по CALS-технологиям / В.В. Харитонов [и др.] // Сборник научных трудов. Том 6. – М.: МИФИ, 2002. – С. 150-151.

75 **Черпаков, Б.И.** Тенденции развития мирового станкостроения в начале века [Электронный ресурс] / Б.И. Черпаков). - Режим доступа : WWW.URL : <http://www.stankoimport.com/presscentre/forum/cherpakov.html>.

76 **Чудаков, А.Д.** Системы управления гибкими комплексами механообработки / А.Д. Чудаков. - М. : Машиностроение, 1990. - 236 с.

77 **Чудаков, А.Д.** Автоматизированное оперативно-календарное планирование в гибких комплексах механообработки / А.Д. Чудаков, Б.Я. Фалевич. - М.: Машиностроение 1986. – 222 с.

78 **Широков, А.Г.** Склады в ГПЯ / А.Г. Широков. - М.: Машиностроение, 1988. - 216 с.

79 Язык UML) [Электронный ресурс]. - Режим доступа : WWW.URL : <http://uml.shl.com>.

80 AnyLogic) [Электронный ресурс]. - Режим доступа : WWW.URL : <http://www.xjtek.ru/anylogic>.

81 Arena Enterprise Suite) [Электронный ресурс]. - Режим доступа : WWW.URL : <http://www.itshop.ru/Level4.asp?ItemId=4670>.

82 **Destefani, J.** Особенности экономного производства на фирме Pratt & Whitney [Электронный ресурс] / J. Destefani). - Режим доступа : WWW.URL : http://www.stankoinform.ru/journal/manufacturing_engineering2005.htm.

83 GPSS World) [Электронный ресурс]. - Режим доступа : WWW.URL : <http://www.gpss.ru/systems/gpssw.html>.

84 **Chung, C.A.** Simulation Modeling Handbook: A practical Approach / Chris Chung. - CRC Press, 2003. – 608 p.

85. **Zeigler, B.B.** Theory of Modeling and Simulation / B.B. Zeigler, Herbert Praehofer, T.G. Kim. - Academic Press, 2000. – 510 p. (2-nd edition).

86. **Waurzyniak, P.** Экономная автоматизация на фирме Renishaw [Электронный ресурс] / P. Waurzyniak. - С. 145, 146, 148 – 150, 152, 154, ил. Manufacturing Engineering. 2005. V.134. Nr. 5 (май)). - Режим доступа : WWW.URL : http://www.stankoinform.ru/journal/manufacturing_engineering2005.htm.

Приложение А

(справочное)

Структурные и функциональные схемы системы «Каскад»

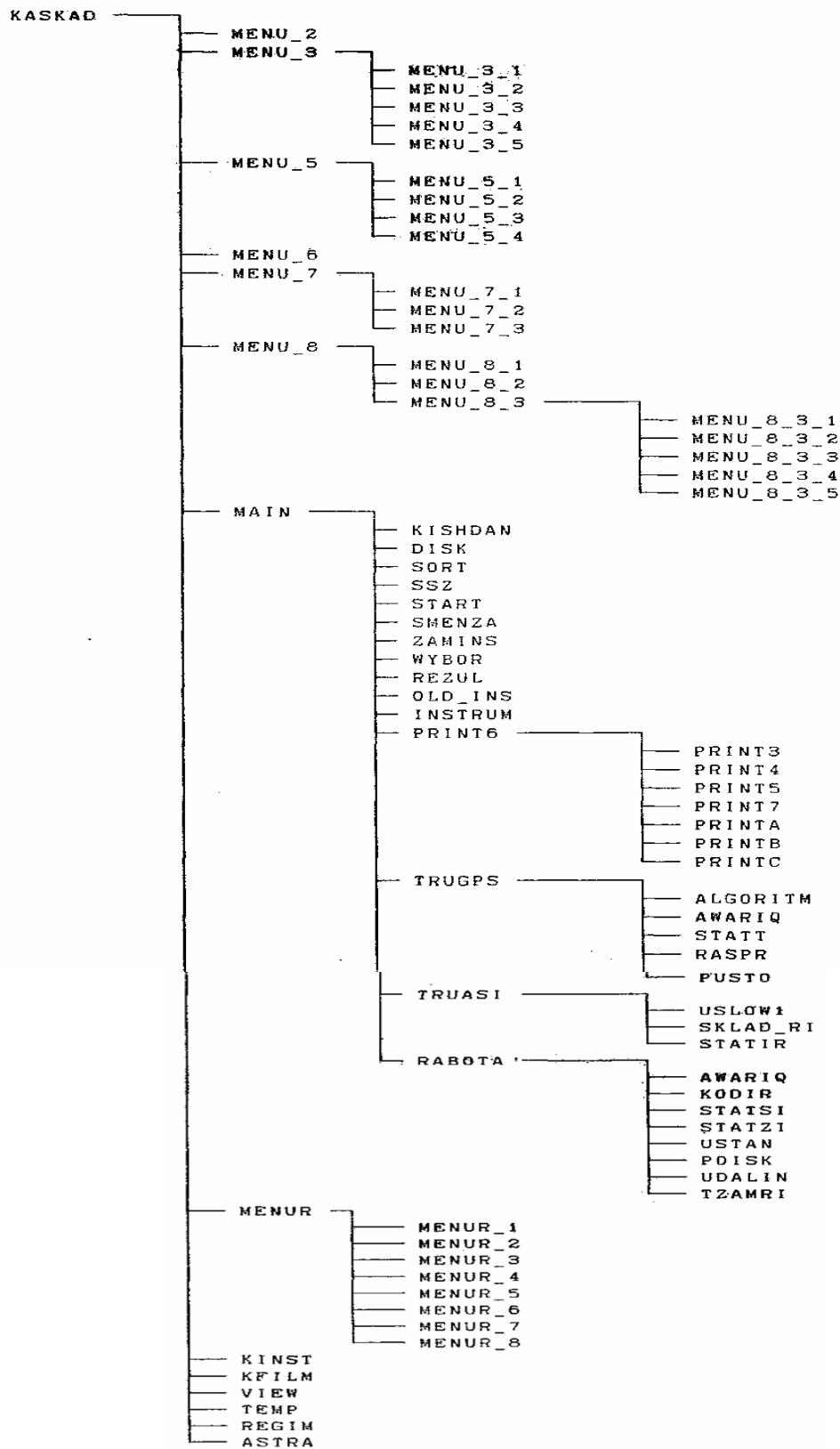


Рисунок А.1 - Структурная схема программных модулей системы «Каскад»

Таблица А.1 – Функциональное назначение программных модулей системы «Каскад»

Уровень вложенности	Наименование программного модуля	Функциональное назначение модуля
1	2	3
1.1	MENU_2	Регистрация, нового варианта' ГПЯ, выбор имеющегося из базы, задание структуры ГПЯ
1.2	MENU_3	Ввод данных о параметрах модулей (станков, накопителей, транспортный средств) ГПЯ
1.2.1	MENU_3_1	ввод и редактирование параметров станков
1.2.2	MENU_3_2	ввод параметров склада заготовок
1.2.3	MENU_3_3	ввод параметров, транспортного средства THC
1.2.4	MENU_3_4	ввод параметров склада инструментов
1.2.5	MENU_3_5	ввод параметров - инструментального робота
1.3	MENU_5	ввод исходного размещения заготовок и режущих инструментов
1.3.1	MENU_5_1	задание размещения заготовок на столах и в пристаночных накопителя
1.3.2	MENU_5_2	размещение заготовок на складе
1.3.3	MENU_5_3	размещение РИ в гнездах ИМ станков
1.3.4	MENU_5_4	размещение РИ в гнездах инструмент, склада
1.4	MENU_6	ввод сменного задания на цикл моделирования
1.5	MENU_7	ввод сведений по технологическим процессам изготовления деталей для данной ГПЯ
1.5.1	MENU_7_1	формирование ведомости режущих инструментов
1.5.2	MENU_7_2	формирование ведомости деталиустановок
1.5.3	MENU_7_3	сведения по технологическим переходам
1.6	MENU_8	модуль аналитических расчетов участка
1.6.1	MENU_8_1	расчет емкости автоматизированного склада заготовок
1.6.2	MENU_8_2	расчет параметров транспортных средств АСИ
1.6.3	MENU_8_3	экспресс-анализ производственного процесса
1.6.3.1	MENU_8_3_1	выполнение расчетов и обработка статистики
1.6.3.2	MENU_8_3_2	ввод и редактирование исходных данных
1.6.3.3	MENU_8_3_3	выдача базы данных по статистике использования режущих инструментов
1.6.3.4	MENU_8_3_4	выдача базы данных по вариантам СЗ
1.6.3.5	MENU_8_3_5	статистические диаграммы распределения числа заготовок, режущих инструментов, времени выполнения технологических переходов
1.7	MAIN	головной модуль блока моделирования производственного процесса
1.7.1	KISHDAN	считывание исходных данных по участку
1.7.2	DISK	считывание данных по технологическим процессам
1.7.3	SORT	сортировка и упорядочивание кодов инструментов-дублеров
1.7.4	SSZ	упорядочивание состава сменного задания
1.7.5	START	расчет граничных значений интервалов накопления статистики
1.7.6	SMENZA	моделирование смены заготовок на столе станка
1.7.7	ZAMINS	Анализ, состава инструментов в ИМ станка и выдача заявок на доставку недостающих
1.7.8	WYBOR	Выбор очередного обслуживаемого станка
1.7.9		Обработка и запись результатов моделирования .
1.7.10	OLD_INS	формирование файла данных об исходном размещении инструментов «по умолчанию»
1.7.11	INSTRUM	формирование ведомости расхода режущих инструментов
1.7.12	PRINT6	модуль, включающий процедуры создания текстовых-файлов с исходными данными и промежуточной (отладочной) информацией
1.7.12.1	PRINT3	
1.7.12.2	PRINT4	
1.7.12.3	PRINT5	
1.7.12.4	PRINT7	
1.7.12.5	PRINTA	
1.7.12.6	PRINTB	

Продолжение таблицы А.1

1	2	3
1.7.12.7	PRINTC	
1.7.13	TRUGPS	моделирование работы робокары
1.7.13.1	ALGORITM	выбор очередного обслуживаемого станка согласно заданному алгоритму
1.7.13.2	AWARTT	моделирование выполнения аварийной заявки
1.7.13.3	STATT	накопление статистики по длительности транспортных операций
1.7.13.4	RASPR	выбор очередной заготовки для доставки к обслуживаемому станку
1.7.13.5	PUSTO	поиск позиции склада - пустой или с нужной заготовкой
1.7.14	TRUASI	моделирование работы INSTR. робота
1.7.14.1	USLOWI	выбор очередной заявки на обслуживание
1.7.14.1	SKLAD_RI	поиск гнезда с требуемым инструментом
1.7.14.1	STATIR	накопление статистики по работе ИРа
1.7.15	RABOTA	моделирование работы станка с автоматической сменой и заменой инструментов
1.7.15.1	AWARIO	выдача аварийной заявки на доставку РИ
1.7.15.2	KODIR	перекодировка гнезд магазина после поворота и смены или замены РИ
1.7.15.3	STATSI	накопление статистики по времени смены инструментов в шпинделе
1.7.15.4	STATZI	накопление статистики по времени замены инструментов в шпинделе
1.7.15.5	USTAN	установка РИ в свободное гнездо магазина (при отсутствии внеш. склада)
1.7.15.6	POISK	поиск пустого гнезда в ИМ станка
1.7.15.7	UDALIN	поиск удаляемого РИ в ИМ станка
1.7.15.8	TZAMRI	расчет времени поворота ИМ при смене и замене инструмента
1.8	MENUR	модуль вывода результатов моделирования
1.8.1	MENUR_1	общие сведения по работе ГПЯ
1.8.2	MENUR_2	протокол (циклограмма) работы ГПЯ
1.8.3	MENUR_3	использование накопителей ГПЯ
1.8.4	MENUR_4	статистика по простоям станка из-за смены и замены инструментов
1.8.5	MENUR_5	статистика по работе инструментального робота
1.8.6	MENUR_6	вывод на печать результатов моделирования
1.8.7	MENUR_7	статистика по работе робокары
1.8.8	MENUR_8	статистика по количеству изготовленных деталей и числу использованных РИ
1.9	K INST	инсталляция системы; управление созданием файлов отладочной информации
1.10	KFILM	компьютерный видеофильм о работе ГПЯ
1.11	VIEW	просмотр и вывод на печать результатов моделирования
1.12	TEMP	связь с программно-методическим комплексом «ТЕМП», используемым для разработки технологических процессов
1.13	ASTRA	размерный анализ технологического процесса с помощью автоматизированной системы технологического размерного анализа «АСТРА» (файл astra.bas)
1.14	REG IM	связь с программой «РЕЖИМ», используемой для нормирования (расчета режимов резания и норм времени) техпроцесса
1.15	DEKODER	модуль перекодировки данных по технологическим процессам из формата *.dbf в формат данных системы «КАСКАД»

Таблица А.2 - Список идентификаторов программного модуля моделирования системы «Каскад»

Наименование идентификатора	Размерность массива	Тип идентификатора	Смысловое значение идентификатора
1	2	3	4
Описание станков ГПЯ			
R	-	INTEGER*4	число станков на участке
STANOK	-	INTEGER*4	текущий номер станка
RANGST	(10)	INTEGER*4	ранг, (приоритет) станка
GRUPPA	(10)	INTEGER*4	.технологическая группа станка
STANAK	(10)	INTEGER*4	число позиций в пристаночном накопителе
TSMZAG	(10)	REAL*4	время смены заготовки на столе станка
KORDST	(10)	REAL*4	координата позиции перегрузки паллет на станке
TSMEIN	(10)	REAL*4	время смены РИ в шпинделе станка
EMKMAG	(10)	INTEGER*4	число гнезд в ИМ станка
GNEZAM	(10)	INTEGER*4	номер гнезда- позиции замены РИ в магазине станка
L	(10)	REAL*4	расстояние между осями соседних гнезд инструментального магазина
TZAM	(10)	REAL*4	время цикла замены РИ в ИМ из перегруз, гнезда
SKOR	(10)	REAL*4	линейная скорость поворота ИМ
REWERS	(10)	INTEGER*4	признак наличие реверса в приводе ИМ
RASSTO	(10)	REAL*4	расстояние в выбранной системе координат до перегрузочного гнезда ИМ станка
OBRZAG	(10)	INTEGER*4	код заготовки на столе станка
NETZAG	(10)	INTEGER*4	признак отсутствия заготовки на столе станка
ZAGOT	(10,10)	INTEGER*4	код заготовки в позиции пристаночного накопителя станка
RANGZA	(10,10)	INTEGER*4	ранг заготовки в позиции пристаночного накопителя
NOMERD	(10,10)	INTEGER*4	шифр техпроцесса обработки заготовки, установленной в позицию пристаночного накопителя
KODINS	(102,10)	INTEGER*4	коды РИ в гнездах ИМ станков
UDALEN	(102,10)	INTEGER*4	признаки удаления РИ, установленных в гнездах ИМ
Описание автоматизированного склада паллет			
SKLDU	-	INTEGER*4	признак типа используемого склада паллет.
SKLAD	(2)	INTEGER*4	число позиций на складе паллет
LNAKOP	(2)	REAL*4	расстояние между осями позиций склада паллет
KRDSKL	(2)	REAL*4	координата склада паллет
ZAGOTO.	(400)	INTEGER*4	коды заготовок в позициях склада (исходное состояние)
ZAGTOI	(400)	INTEGER*4	коды заготовок в позициях склада («размещение по умолчанию»)
Описание транспортного средства ТС			
USKTR	(2)	REAL*4	ускорение (замедление) ТС
SKORTR	(2)	REAL*4	маршевая скорость ТС (по двум осям)
SINHRO		INTEGER*4	Признак синхронной отработки перемещений по коорд. осям транспортным средством
POZTR	-	REAL*4	число позиций у трансп. тележки
TSMTR	-	REAL*4	время цикла смены заготовки на ТС
OTPRAW	-	INTEGER*4	текущий номер станка, от которого отправляется робота
PRIBYT	-	INTEGER*4	текущий номер станка, к которому направляется робота
Описание автоматического склада режущих инструментов			
SKLRI	-	INTEGER*4	признак типа склада режущих инструментов
EMKNAK	(2)	INTEGER*4	число гнезд центр, склада РИ
LGNEZD	(2)	REAL*4	расстояние между осями гнезд склада инструментов
KRDNRI	(2)	REAL*4	координата склада РИ
INSTR1	(1000)	INTEGER*4	коды РИ гнездах склада РИ
INSTR2	(1000)	INTEGER*4	коды РИ гнездах склада РИ (заполнение в режиме «по умолчанию»)

Продолжение таблицы А.2

Описание транспортного средства доставки режущих инструментов			
1	2	3	4
SINHR	-	INTEGER*4	признак синхронной отработки перемещений по коорд. осям инструментальным роботом (ИР)
EMKTEL	-	REAL*4	число позиций на ИР
SKORAO	(2)	REAL*4	маршевая скорость ИРа (по двум осям)
USKOR	(2)	REAL*4	ускорение (замедление) ИРа
TSMEN	-	REALA4	время цикла смены РИ ИРом
KODRIR		INTEGER*4	текущий код режущего инструмента в захвате ИР
Описание сменного задания			
ALGOR	(5)	INTEGER*4	порядковые номера алгоритмов управления работой модулей
KOLPZ		INTEGER*4	число партий запуска в сменном задании
KOL	(10)	INTEGER*4	число заготовок в партии запуска
KPZ	(10)	INTEGER*4	код партии запуска
Описание процесса моделирования			
GPSSUM	-	REAL*8	текущее время (календарь) ТС
SIOUSUM	-	REAL*8	текущее время (календарь) ИРа
KOLZAG	-	INTEGER*4	общее число заготовок в смен. задании
KOLZAG	(10)	INTEGER*4	текущее число заготовок из состава СЗ, запущенных в обработку .
POSOBR.	(200)	INTEGER-	код заготовки согласно очереди в. сменном задании
PRAWIL	(10)	INTEGER*4	признак соответствия заготовки на столе станка составу сменного задания
USTAN	(100,10)	REAL*8	момент установки заготовки в пристаночный накопитель
MIGUST	(10,10)	REAL*8	момент установки заготовки на стол станка
DU STAN	(50)	INTEGER*4	шифр техпроцесса обработки заготовки
KDU	(50)	INTEGER*4	резерв, шифр техпроцесса обработки
PERH	(10)	INTEGER*4	номер текущего перехода обработки заготовки на столе станка
TMASH	(50,10)	REAL*4	основное время техн. перехода
TWSPOM	(50,10)	REAL*4	вспомогательное время техн. перехода
PEREH	(10)	INTEGER*4	число переходов в техпроцессе обработки заготовки на столе станка
TTPMSH	(10)	REAL*4	суммарное машинное время техпроцесса в сменном задании
TTPWSP	(10)	REAL*4	суммарное вспомогательное время техпроцесса
INSTR	(50,10)	INTEGER*4	код РИ на данном технологическом переходе
ZAPROS	(50,10)	INTEGER*4	код РИ, запрашиваемого в заявке станка
RANG	(50,10)	INTEGER*4	ранг заявки на данный РИ
ZAP INS	(10)	INTEGER*4	признак выдачи заявок на требуемые РИ
ZAPSUM	(10)	INTEGER*4	число заявок от станка на доставку РИ при смене заготовки на столе
ZAQWKA	-	INTEGER*4	порядковый номер заготовки в очереди сменного задания на доставку к станкам
ZAGTEL	-	INTEGER*4	текущий код заготовки на платформе робокары
WRTRAN	-	REAL*4	время выполнения текущей транспортной операции робокарой
WREMQ	(50,10)	REAL*8	момент времени, в который потребуется в шпинделе станка запрошенный РИ
RESURS	(5000)	REAL*4	текущий ресурс режущего инструмента
RESINS	(500)	REAL*4	исходный ресурс инструментов-дублеров
WYP	(10)	INTEGER*4	признак завершения обработки в рабочей зоне
SNQTIE	(100,10)	REAL*8	момент снятия заготовки со стола
DU	(100,10)	INTEGERA4	код заготовки, обработанной на станке

Продолжение таблицы А.2

1	2	3	4
PROST	-	INTEGER*4	признак наличия удаляемых заготовок в пристаночных накопителях
SMENAZ	-	INTEGER*4	текущее число заготовок, запущенных в обработку
KONEC	(10)	REAL*8	момент завершения станком обработки последней заготовки в данном цикле
WREMZA	(10)	REAL*8	текущее отработанное станком время
WREOTR	(10)	REAL*8	резерв, текущее отработанное станком время
BEGINT'	(200)	REAL*8	момент начала выполнения транспортной операции
STANT	(200)	INTEGER*4	номер текущего обслуживаемого станка
ENDT	(200)	REAL*8 /	момент завершения выполнения транспортной операции
ZAGSUM	-	INTEGER*4	число транспортных операций робокары
SUMPTT	-	REAL*4	суммарные простои транспортной тележки
BEGNIR	(1000)	REAL*8	момент начала выполнения ИРом транспортной операции
STANIR	(1000)	INTEGER*4	номер станка, обслуженного ИРом в данной транспортной операции
ENDIR	(1000)	REAL*8	момент окончания выполнения ИРом транспортной операции
WREOBS	-	REAL*4	время выполнения текущей транспортной операции ИРом
PRSTIR	-	REAL*4	текущее время простоев ИРа
ZAQSUM	-	INTEGER*4	текущее число заявок на замену РИ у ИРа
Описание результатов моделирования			
PODGRI	(10)	REAL*4	простои станка из-за подготовки РИ
PROUSI	(10)	REAL*4	простои станка из-за подготовки и смены РИ
PROASI	(10)	REAL*4	простой станка из-за доставки реж. инструментов
SMENRI	(10)	REAL*4	простой станка из-за смены реж. инструментов
PROSTT	(10)	REAL*4	время простоев робокары
SUMPIR	-	REAL*4	суммарное время простоев инструментального робота
NOMER	(10)	INTEGER*4	текущее число заготовок, обработанных на данном станке
WRENOM	(10)	REAL*4	номинальное время (по техпроцессам), отработанное станком за цикл моделирования
ISPRIS	(10)	INTEGER*4	количество РИ, использованных на станке за цикл моделирования
KOLINS	(500)	INTEGER*4	количество РИ каждого наименования, использованных за цикл работы участка
KOLSRI	(10)	INTEGER*4	число смен режущих инструментов
KOLZRI	(10)	INTEGER*4	число замен режущих инструментов
KZAGMI		REAL*4	к-т загрузки магазина инструментов
KZAGPN		REAL*4	к-т загрузки пристаноч. накопителя
KZGSKL		REAL*4	к-т использ. склада паллет
KIR		REAL*4	к-т использования ИР
KTT		REAL*4	к-т использования робокары
KZGCNI		REAL*4	к-т использ. центр, склада РИ
OTL3		INTEGER*4	ключи создания файлов с отладочной информацией
OTL4		INTEGER*4	
OTL5		INTEGER*4	
OTL6		INTEGER*4	
IOST		INTEGER*4	признак аварийного завершения выполнения программной процедуры

Фрагмент программного кода системы «Каскад». Обслуживание станков транспортом

```

subroutine trugps(obrzag,stanak,zagot,rangza,posobr,prawil,wyp,
* ttpmsh,ttpwsp,kordst,zagoto,lnakop,gpssum,sklad,skortr,
* usktr,tsmtr,wrtran,zaqwka,otpraw,zagtel,r,perh,pereh,tmash,
* twspom,netzag,prostt,siosum,sumptt,migust,SINHRO,KRDSKL,
* tTTmin,shagTT,warTT,tmaxTT,tminTT,awariq,wremza,prost,
* beginT,endT,stanT,KzagPN,SkldU,poztr,otl3,otl6,iost,KDU,
* zagtol,algor,zaqwk1,zagsum,KPZ,KOLPZ,RangSt,DU_STAN,Gruppa,
* DU,Nomer,SINTEZ)
C
C      выбора очередного обслуживаемого станка
C      и расчета времени транспортной операции
C
      real ttpmsh,ttpwsp,kordst,lnakop,skortr,usktr,tsmtr,
* wrtran,tmash,twspom,prostt,sumptt,tminTT,tmaxTT,
* shagTT,tTTmin,wredop,put,predel,skora,POZTR,KRDSKL,Wrem_X,
* Wrem_Y,yyy,KzagPN
      real*8 SioSum,Gpssum,beginT,endT,migust,minwr,sumwr,wremza
      integer sumst,stanok,гнездо,pribyt,zagtel,kolzag,zagtol,TekKod,
* prawil,r,obrzag,stanak,zagot,perhod,perh,pereh,warTT,wyp,
* rangza,posobr,zagoto,sklad,zaqwka,otpraw,mogu,netzag,prost,A,
* obraz,awariq,udalDU,net,stanT,SkldU,otl3,otl6,KDU,Gruppa,
* zaqwk1,zagsum,okon,net1,algor,RangSt,spisok,stoit,DU_STAN,SINHRO,
* X,Y,XX,YY,netdu,DU,Nomer, SINTEZ
      dimension prawil(10),kolzag(10),netzag(10),prostt(10),wyp(10),
* obrzag(10),stanak(10),zagot(10,10),rangza(10,10),posobr(500),
* migust(10,10),wartt(12),udalDU(10),ttpmsh(10),ttpwsp(10),
* kordst(10),zagoto(400),tmash(50,10),twspom(50,10),algor(5),
* perh(10),pereh(10),wremza(10),awariq(10),beginT(500),endT(500),
* stanT(500),sumwr(10),KzagPN(10),zagtol(400),kdu(50),okon(10),
* spisok(10),RangSt(10),DU_STAN(50),Gruppa(10),LNAKOP(2),SKLAD(2),
* SKORTR(2),USKTR(2),KRDSKL(2),netdu(10),DU(100,10),Nomer(10)
C      Write(*,*) 'TRUGPS вход'
      do 1 i=1,r
      spisok(i)=0
      sumwr(i)=0
      netdu(i)=0
1      okon(i)=1
2      prost=0
      stoit=0
      do i=1,r
         If (NETDU(i).LT.1) goto 6
      end do
      goto 999
C
C      ФОРМИРОВАНИЕ АЛЬТЕРНАТИВНОГО СПИСКА СТАНКОВ НА ОБСЛУЖИВАНИЕ
C      *****
C      поиск станков, на к. нет обрабатываемой ДУ
C
6      minwr=9999999
      mogu=0
      do 3 stanok=1,r
      If (okon(stanok).le.0.or.netdu(stanok).gt.0) goto 3
      IF (netzag(stanok).lt.1) then
         mogu=mogu+1
         spisok(mogu)=stanok
         if (wremza(stanok).lt.minwr) minwr=wremza(stanok)
      END IF
3      continue
      if (mogu.gt.0) then
         stoit=1
         CALL ALGORITM(algor,spisok,wremza,pribyt,mogu,minwr,
* sumwr,RangSt,kolzag,udaldu,stoit,KordSt,otpraw,SkldU)
C      write(*,*) '1 обслужив. станок N ',pribyt
         goto 680
      end if
      If (Algor(2).eq.1) goto 320
C
C      поиск станков, которые в принципе могут быть обслужены

```

```

c
7 minwr=9999999.
  sumst=0
  mogu=0
  do 50 stanok=1,r
    spisok(stanok)=0
    sumwr(stanok)=0
    udalDU(stanok)=0
    kolzag(stanok)=0
c    Write(*,*) stanok
      if (okon(stanok).lt.1) then
        goto 50
      else
        if (wyp(stanok).ge.10) goto 312
      end if
      A=KDU(ObrZag(stanok))

c
c      расчет остаточного времени на столе
c
c    Write(*,17) a,perh(stanok),pereh(A)
c17 format('+', ' код ДУ-',i3,'1 пер.-',i3,
c * ' посл.пер-',i3)
    DO perhod=perh(stanok),pereh(A)
      sumwr(stanok)=sumwr(stanok)+tmash(perhod,A)+twspom(perhod,A)
    END DO
312 DO 40 il=1,stanak(stanok)
    if (zagot(il,stanok).le.0) goto 40
c    or.migust(il,stanok).gt.gpssum)
15 kolzag(stanok)=kolzag(stanok)+1
    IF (rangza(il,stanok).eq.0) THEN
      udalDU(stanok)=udalDU(stanok)+1
    ELSE
c      расчет сум. станкоемкости в пристан. накопителе
      IA=KDU(zagot(il,stanok))
      sumwr(stanok)=sumwr(stanok)+ttpmsh(IA)+ttpwsp(IA)
    END IF
40 continue
    yyy=(kolzag(stanok)*100)/(stanak(stanok))
    if (KzagPN(stanok).lt.yyy) KzagPN(stanok)=yyy
    if (kolzag(stanok).ge.(stanak(stanok)-1)) goto 50
41 mogu=mogu+1
    spisok(mogu)=stanok
    if (sumwr(stanok).lt.minwr) minwr=sumwr(stanok)
50 continue
c write(*,*) 'по 2 условию mogu==',mogu
c *****
    if (mogu.gt.0) then
      CALL ALGORITM(algor,spisok,wremza,pribyt,mogu,minwr,
c * sumwr,RangSt,kolzag,udaldu,stoit,KordSt,otpraw,SkldU)
c write(*,*) '2 обслужив. станок N ',pribyt
      goto 680
    end if
c может, есть удаляемые ДУ : ищем станок, на котором их больше..
51 GOTO 320
c51 imax=0
    DO stanok=1,r
      IF (okon(stanok).GT.0.AND.IMAX.LT.udalDU(stanok)) THEN
        imax=udalDU(stanok)
        istan=stanok
      END IF
    END DO
    if (imax.eq.0) goto 320
c есть однако удаляемые ДУ на istan станке :
c
c315 write(*,*) ' есть удаляемые '
315 pribyt=istan
    prost=10
26 CALL AWARTT(skortr,usktr,tsmtr,SINHRO,lnakop,kordst,zagot,
c * rangza,stanak,pribyt,otpraw,zagtel,zagoto,wrTRAN,sklad,
c * gnezdo,zagtol,SkldU,X,Y,KRDSKL,POZTR)
    If (SkldU.gt.0) then
      gpssum=gpssum+wrtran

```

```

        prostTT(pribyt)=ProsTT(pribyt)+gpssum-wremza(pribyt)
        if (tmaxTT.lt.wrtran) tmaxTT=wrTRAN
        if (tminTT.gt.wrtran) tminTT=wrTRAN
    end if
c27    if (OTL6.ne.1) goto 888
        IF (SINTEZ.NE.0) GOTO 777
87     write(7,28) otpraw,zagtel,gnezdo,pribyt,wrtran,gpssum
28     format(1x,'I Авар. заявкаI',4(i4,2x,'I'),6x,'I',6x,'I',
* f5.0,1x,'I',f7.0,'I')
888    if(OTL3.ne.1) goto 777
887    write(3,32)gnezdo,pribyt,wrtran,gpssum,zagot(i,pribyt)
32     FORMAT(1x,'I TT-Авар.заяв',4x,' гнездо=',i3,' стан=',i2,
* ' BP.=',f5.1,' СУМ.=',f6.0,' УД=',i4,'Авария',5x,'I')
777    CALL STATtt(tTTmin,shagTT,wrTRAN,warTT,iost)
        goto 999
c ..   нет и удаляемых ДУ , простой TT
320    sumptt=sumptt+abs(siosum-gpssum)
c      write(*,1887) sumptt,(siosum-gpssum)
c1887  FORMAT(1x,'I TT- Простой суммарный -',f8.2,' текущий- ',f8.2)
c      write(*,*) 'TRUGPS- нет удаляемых ДУ'
        prost=1
        if (gpssum.lt.siosum) gpssum=siosum+1
c      gpssum=gpssum+1
        goto 999
c *****
c
c      Определение очередной транспортируемой к станку ДУ
c      -----
680    if (ALGOR(1).EQ.1) goto 2030
c      2... при распределении ДУ произвольно согласно общей очереди
c
2025    do i=1,zagsum
        if (posobr(i).gt.0) then
            ii=posobr(i)
            TekKod=int(DU_STAN(ii)/10)
            TekKod=DU_STAN(ii)-TekKod*10
            If (TekKod.eq.Gruppa(pribyt)) then
                zaqwka=i
                goto 2035
            end if
        end if
    end do
    okon(pribyt)=0
    goto 2
c
c      1... при распределении ДУ согласно имеющемуся на станке
c      комплекту инструментов
c
2030    CALL RASPR(posobr,obrzag,pribyt,zaqwka,zagsum,r,
* kpz,kolpz,ttpmsh,ttpwsp,stanak,zagot,net1,DU_STAN,Gruppa,DU,
* Nomer)
        if (net1.le.0) goto 2035
2073    okon(pribyt)=0
        goto 2
2035    zaqwk1=zaqwkl+1
        beginT(zaqwkl)=gpssum
        obraz=posobr(zaqwka)
        wredop=0
        if (Skldu.ne.1.or.poztr.ne.1.or.zagtel.eq.0) goto 2005
c ***** тележка однопозиционная и занята *****
2010    obraz=0
        CALL PUSTO(kordst,lnakop,gnezdo,zagoto,sklad,otpraw,obraz,net,
* zagtol,SkldU,X,Y,KRDSKL)
        iyyy=gnezdo
        XX=X
        YY=Y
        zagoto(gnezdo)=zagtel
        zagtel=0
        obraz=posobr(zaqwka)
        CALL PUSTO(kordst,lnakop,gnezdo,zagoto,sklad,otpraw,obraz,net,
* zagtol,SkldU,X,Y,KRDSKL)
        If (SkldU.eq.1) then

```

```

        wredop=lnakop(1)*(abs(iyyy-gnezdo))
        wredop=wredop/skortr(1)+skortr(1)/usktr(1)+tsmtr
    end if
    IF (SkldU.eq.2) then
        Wrem_X=lnakop(1)*(abs(XX-X))
        Wrem_X=Wrem_X/skortr(1)+skortr(1)/usktr(1)
        Wrem_Y=lnakop(2)*(abs(YY-Y))
        Wrem_Y=Wrem_Y/skortr(2)+skortr(2)/usktr(2)
    If (SINHRO.EQ.2) THEN
        wredop=Wrem_X+Wrem_Y+tsmtr
    ELSE
        If (Wrem_X.lt.Wrem_Y) then
            wredop=Wrem_Y+tsmtr
        else
            wredop=Wrem_X+tsmtr
        end if
    END IF
END IF
END IF

C
C ***** тележка двухпозиционная *****
2005      net=1
          CALL PUSTO(kordst,lnakop,gnezdo,zagoto,sklad,otpraw,obraz,net,
* zagtol,SkldU,X,Y,KRDSKL)
          if (net.eq.1) goto 750

C
c1290      write(*,*) '   на складе нет нужной ДУ N',obraz
C          проверим, начальный ли код у разыскиваемой ДУ (id=1 ?)
C
c1290      mk=Int(DU_STAN(posobr(zagwka))/10)
C          id=(DU_STAN(posobr(zagwka))-mk*10)
C          Write(*,1291) DU_STAN(posobr(zagwka)),id
c1291      format(1x,'запрос на ДУ-',i4,'id-',i3)
C          IF (Id.eq.1) then
C              goto 1701
C          else
C              NetDU(pribyt)=1 ! признак того, что нет на складе нужной ДУ
C              Call GDEDU      ! поищем ее в пристаночных накопителях
C              goto 2
C          end if
C
C          pause
C          «установим» в пустое гнездо нужную ДУ
1701      obraz=0
          net=5
          CALL PUSTO(kordst,lnakop,gnezdo,zagoto,sklad,otpraw,obraz,net,
* zagtol,SkldU,X,Y,KRDSKL)
          if (net.gt.0) goto 1710
1702      Write(*,*) ' в складе ДУ нет пустых гнезд !!'
          Write(*,78) (zagoto(i),i=1,200)
78          format(1x,10(20(i3)/))
          pause
          Write(*,78) (zagtol(i),i=1,200)
          pause
          Iost=10
          goto 999
1710      zagoto(gnezdo)=posobr(zagwka)
C
C      РАСЧЕТ ВРЕМЕНИ ТРАНСПОРТНОЙ ОПЕРАЦИИ :
C      ОТ СТАНКА отпраw ДОЕХАТЬ ДО ГНЕЗДА gnezdo,ВЗЯТЬ ДУ С
C      КОДОМ posobr(zagwka), ПРИВЕЗТИ ЕЕ К СТАНКУ pribyt И ВЫГРУЗИТЬ
C      В ГНЕЗДО С УДАЛЯЕМОЙ ДУ, ЗАХВАТИВ ЕЕ
750      zagtol(gnezdo)=posobr(zagwka)
          if (SkldU.ne.3) goto 755
C
C ***** если склад ДУ внешний *****
C
753      poisk=abs(kordst(otpraw))
          dostaw=abs(kordst(pribyt))
          A=1
          goto 757
C
C ***** если склад ДУ линейный *****

```



```

C
755      A=1
          If (Sk1DU.eq.1) X=gnezdo
poisk =abs(kordst(otpraw)-lnakop(A)*X-KRDSKL(1))
dostaw=abs(kordst(pribyt)-lnakop(A)*X-KRDSKL(1))
757      put=(poisk+dostaw)/2
          predel=skortr(A)*skortr(A)/usktr(A)
          skora=skortr(A)
          if (predel.lt.put) goto 846
844      skora=put/usktr(A)
          put=0
846      wrtran=2*put/skortr(A)+2*skora/usktr(A)+2*tsmtr
          if (pozTR.eq.1) wrtran=wrtran+wredop
          If (Sk1DU.ne.2) goto 880

C
C *****  если склад многоярусный *****
C
1500     If (A.lt.2) then
          A=2
          wrem_X=wrtran
          poisk=abs(PozTR-lnakop(2)*Y-KRDSKL(2))
          dostaw=abs(PozTR-lnakop(2)*Y-KRDSKL(2))
          goto 757
        end if
          If (SINHRO.NE.1) THEN
              wrtran=wrtran+wrem_X-2*tsmtr
              ELSE
                  IF (wrtran.LT.wrem_X) wrtran=wrem_X
              END IF
          wrtran=wrtran+wredop
880     If (Sk1DU.eq.0) wrtran=1
          gpssum=gpssum+wrtran
          endT(zaqwk1)=gpssum
          stanT(zaqwk1)=pribyt
          If (Sk1DU.eq.0) goto 199
881     if(tmaxTT.lt.wrtran) tmaxTT=wrTRAN
          if(tminTT.gt.wrtran) tminTT=wrTRAN
          CALL STATtt(tTTmin,shagTT,wrTRAN,warTT,iost)
          if (iost.Eq.10) then
            Write(*,231) otpraw,X,Y,pribyt,poisk,dostaw,wrtran
231     * ' Format(1x,' otpraw- ',i2,' X- ',i3,' Y- ',i3,' pribyt- ',i3,
          * ' poisk- ',f6.2,' dostaw- ',f6.1,' wrtran- ',f6.1)
          Write(*,*) 'Sklad(1)- ',Sklad(1),'Sklad(2)- ',Sklad(2)
          goto 999
        end if

C      ПОИСК В ПРИСТАНОЧНОМ НАКОПИТЕЛЕ pribyt СТАНКА
C      УДАЛЯЕМОЙ ЗАГОТОВКИ
199     If (Sk1DU.eq.0) then
          i=1
          goto 250
        end if
          DO i=1,stanak(pribyt)
            if(zagot(i,pribyt).gt.0.and.rangza(i,pribyt).le.0) GOTO 250
C      *      and.migust(i,pribyt).le.GpsSum) goto 250
          END DO
C      . . . НЕТ УДАЛЯЕМОЙ ЗАГОТОВКИ; ТОГДА ИЩЕМ ПУСТУЮ ЯЧЕЙКУ :
          DO i=1,stanak(pribyt)
            if(zagot(i,pribyt).eq.0) goto 250
          END DO
          write(*,212) pribyt,obrzag(pribyt),
          * (zagot(i,pribyt),i=1,stanak(pribyt))
212     * format(20x,' в TRUGPS ошибка !!!'/
          * 10x,'станок N ',i2,' удаляемых ДУ и пустых гнезд нет'/
          * 1x,' стол :',10(i3,1x))
          write(*,213) prawil(pribyt),
          * (rangza(i,pribyt),i=1,stanak(pribyt))
213     * format(1x,7x,10(i3,1x))
          Iost=10
          goto 999
250     if (otl6.ne.1.OR.SINTEZ.NE.0) goto 1188
7811    write(7,220) zaqwk1,posobr(zaqwka),otpraw,zagot(i,pribyt),gnezdo,
          * pribyt,prostt(pribyt),wrtran,gpssum

```

```

220  format(1x,'I',6(i4,2x,'I'),f6.1,'I',f5.0,1x,'I',
* f8.0,'I')
1188  if (otl3.ne.1.OR.SINTEZ.NE.0) goto 2199
2198  write(3,300)zaqwkl,posobr(zaqwka),гнездо,pribyt,wrtran,
* gpssum,zagot(i,pribyt),minwr
300  FORMAT(1x,'I TT-',i3,' ДУ=',i2,' гнездо=',i3,' стан=',i2,
* ' ВР.',f4.0,'GpsS.',f8.0,' УД=',i3,' MINw=',f9.0,'I')
c  перекодировка размещения заготовок
2199  iiii=zagot(i,pribyt)
      zagot(i,pribyt)=posobr(zaqwka)
      migust(i,pribyt)=gpssum
      rangza(i,pribyt)=1
      zagoto(гнездо)=zagtcl
      zagtcl=iiii
      otpraw=pribyt
      awariq(pribyt)=0
      netzag(otpraw)=1
      posobr(zaqwka)=0
cc  if (otpraw.ne.6) goto 999
cc999  write(*,900) zaqwka, (i,i=1,10)
c900  format(/5x,'размещ загот на станке после ',i3,'трансп. опер.'/
c  *5x,'! станок ! стол !',10(i2,1x,'!'),' ( сек )!')
c  write(*,905)pribyt,obrzag(pribyt), (zagot(i1,pribyt),i1=1,10),
c  *wremza(pribyt)
c905  format(5x,'!',i4,5x,'!',i3,4x,'!',10(i2,1x,'!'),f8.0,'!')
c  Write(*,*) 'netzag=',Netzag(otpraw)
c  pause
999  return
end

SUBROUTINE ALGORITM(algor,spisok,wremza,pribyt,mogu,minwr,
* sumwr,RangSt,kolzag,udaldu,stoit,KordSt,otpraw,SkldU)
real*8 wremza,minwr,sumwr,minwrl
real rassto,KordSt
integer spisok,pribyt,mogu,algor,stanok,max,rangSt,Kol,TekAl,
* spisl,kolzag,udaldu,stoit,otpraw,SkldU
dimension wremza(10),spisok(10),algor(5),RangSt(10),sumwr(10),
* spisl(10),kolzag(10),udaldu(10),KordSt(10)
      pribyt=0
c  Write(*,*) 'ALGOR'
c  WRITE(*,20) mogu, (spisok(i),i=1,mogu)
c123  format( 10(i5:))
      If (ALGOR(2).gt.5.or.ALGOR(2).lt.1) ALGOR(2)=1
      TekAl=ALGOR(2)

c
c  выбор очередного станка из альтернативного списка
c
      IF (mogu.eq.1) then
        pribyt=spisok(1)
        goto 999
      end if

c
c  0. Станки альтерн. списка простаивают, выявляем max ранг станка
c
      if (stoit.eq.1) then
        Max=0
        Do i=1,mogu
          stanok=spisok(i)
          if (RangST(stanok).gt.Max) Max=RangST(stanok)
        end Do
c  выявляем простаивающие станки с max рангом
        minwrl=9999999
        kol=0
        Do i=1,mogu
          stanok=spisok(i)
          if (RangST(stanok).eq.Max) then
            kol=kol+1
            spisl(kol)=stanok
          if (wremza(stanok).lt.minwrl) minwrl=wremza(stanok)
          end if
        end Do
        if (Kol.eq.1) then

```

```

        pribyt=spisl(kol)
        goto 999
    end if
    If (kol.le.0) then
    Write(*,*) 'ОШИБКА АЛГОРИТМ-0'
        pause
        goto 999
    end if
    do i=1,kol
        spisok(i)=spisl(i)
    end do
    mogu=kol
C    ВЫЯВЛЯЕМ ДОЛЬШЕ ПРОСТАИВАЮЩИЕ СТАНКИ С МАХ РАНГОМ
        kol=0
        do i=1,mogu
            stanok=spisok(i)
            if (wremza(stanok).eq.minwrl) then
                kol=kol+1
                spisl(kol)=stanok
            end if
        end do
        if (Kol.eq.1) then
            pribyt=spisl(kol)
            goto 999
        end if
        If (kol.le.0) then
        Write(*,*) 'ОШИБКА АЛГОРИТМ-0'
            pause
            goto 999
        end if
        do i=1,kol
            spisok(i)=spisl(i)
        end do
        mogu=kol
C    ВЫЯВЛЯЕМ БЛИЖАЙШИЙ К ТТ ДОЛГО ПРОСТАИВАЮЩИЙ СТАНОК
C    С МАХ РАНГОМ
        minwrl=9999999
        do i=1,mogu
            stanok=spisok(i)
            Rassto=ABS(Kordst(stanok)-KordSt(otpraw))
            If (Rassto.lt.minwrl) then
                pribyt=stanok
                minwrl=Rassto
            end if
        end do
        goto 999
    end if
C *****
C    1. ПО ВРЕМЕНИ ПОСТУПЛЕНИЯ ЗАЯВКИ
5    if (TekAl.eq.1) then
        Kol=0
        minwrl=9999999
        Do i=1,mogu
            stanok=spisok(i)
        if (wremza(stanok).lt.minwrl) minwrl=wremza(stanok)
        end do
        Do i=1,mogu
            stanok=spisok(i)
            if (wremza(stanok).eq.minwrl) then
                Kol=Kol+1
                Spisl(Kol)=stanok
            end if
        end Do
        If (Kol.le.0) then
            write(*,*) 'ОШИБКА АЛГОРИТМ-1'
            PAUSE
        end if
        pribyt=spisl(kol)
        goto 999
    end if
C *****
C    2. ПО ПРИОРИТЕТАМ СТАНКОВ

```

```

        Max=0
        if (TEKAL.eq.2) then
C      определяем max ранг станка из списка
            Do i=1,mogu
                stanok=spisok(i)
                if (RangST(stanok).gt.Max) then
                    Max=RangSt(stanok)
                end if
            end Do
C      определяем, сколько станков с таким рангом
            Kol=0
            Do i=1,mogu
                stanok=spisok(i)
                if (RangST(stanok).eq.Max) then
                    Kol=Kol+1
                    spis1(kol)=stanok
                end if
            end Do
            If (Kol.le.0) then
                write(*,*) 'ОШИБКА АЛГОРИТМ-2'
                PAUSE
            end if
            If (kol.eq.1) then
                pribyt=spis1(kol)
                goto 999
            end if
C      .... станков несколько; ищем отработавший меньшее время
            Mogu=Kol
            Do i=1,Mogu
                spisok(i)=spis1(i)
            end do
            TEKAL=1
            goto 5
        end if !
C      *****
C      3. ПО МИН ВРЕМЕНИ ТРАНСПОРТНОЙ ОПЕРАЦИИ
        IF (TEKAL.EQ.3) THEN
            If (SkldU.eq.1.or.SkldU.eq.2) then
                minwrl=999999
                do i=1,mogu
                    stanok=spisok(i)
                    Rassto=ABS(Kordst(stanok)-KordSt(otpraw))
                    If (Rassto.lt.minwrl) then
                        pribyt=stanok
                        minwrl=Rassto
                    end if
                end do
                goto 999
            End if
            If (SkldU.eq.3) then
                minwrl=999999
                do i=1,mogu
                    stanok=spisok(i)
                    Rassto=Kordst(stanok)
                    If (Rassto.lt.minwrl) then
                        pribyt=stanok
                        minwrl=Rassto
                    end if
                end do
                goto 999
            End if
            WRITE(*,*) 'ОШИБКА АЛГОРИТМ-3'
            Pause
        END IF
C      *****
C      4. ПО МИН ОЧЕРЕДИ ДУ
        if (TEKAL.eq.4) then
            Max=100
            Do i=1,mogu
                stanok=spisok(i)
                If (KolZag(stanok).gt.0) then
                    Dlina=KolZag(stanok)-UdalDU(stanok)

```

```

        If (Dlina.lt.Max) Max=Dlina
        else
        Max=0
        end if
    end do
    Kol=0
    Do i=1,mogu
        stanok=spisok(i)
        Dlina=KolZag(stanok)-UdalDU(stanok)
        if (Dlina.eq.Max.or.KolZag(stanok).eq.0) then
            Kol=Kol+1
            Spis1(Kol)=stanok
        end if
    end Do
    If (Kol.le.0) then
        write(*,*) 'ОШИБКА АЛГОРИТМ-4'
        Do i=1,mogu
            stanok=spisok(i)
            Write(*,*)i,Kolzag(stanok),UdalDu(stanok),Max
        end do
        PAUSE
    end if
    If (kol.eq.1) then
        pribyt=spis1(kol)
        goto 999
    else
        Mogu=Kol
        Do i=1,Mogu
            spisok(i)=spis1(i)
        end do
        TEKAL=1
        goto 5
    end if
end if
! ALGOR(2)=4
*****
C 5. ПО MIN СУММАРНОЙ СТАНКООЕМКОСТИ ДУ
C   if (TEKAL.eq.5) then
        Kol=0
        Do i=1,mogu
            stanok=spisok(i)
            if (sumwr(stanok).eq.minwr) then
                Kol=Kol+1
                Spis1(Kol)=stanok
            end if
        end Do
        If (Kol.le.0) then
            write(*,*) 'ОШИБКА АЛГОРИТМ-5'
            PAUSE
        end if
        If (kol.eq.1) then
            pribyt=spis1(kol)
            goto 999
        else
            Mogu=Kol
            Do i=1,Mogu
                spisok(i)=spis1(i)
            end do
            TEKAL=1
            goto 5
        end if
    end if
    ! ALGOR(2)=5
*****
C999 write(*,20) pribyt,(spisok(i),i=1,mogu)
C20  format(' ',i3,' ',10(i5,))
C    pause
C    write(*,25)
C25  format(//////////' ')
999  If (Pribyt.eq.0) then
        Write(*,*) ' Ошибка в пп ALGORИТМ'
        pause
    end if
return

```

end

```
SUBROUTINE RASPR (posobr, obrzag, pribyt, zaqwka, zagsum, r,  
* kpz, kolpz, ttpmsh, ttpwsp, stanak, zagot, net1, DU_STAN, Gruppa, DU,  
* Nomer)
```

```
C  
C ... выбор ДУ при доставке на станок по комплекту РИ .....  
C  
      integer i, y, posobr, obrzag, pribyt, zaqwka, zagsum, r, kolpz, kpz,  
*      stanak, zagot, net1, TekKod, DU_STAN, Gruppa, DU, Nomer  
      real ttpmsh, ttpwsp, sum, max  
      dimension posobr(500), obrzag(10), ttpmsh(10), ttpwsp(10),  
* sum(10), stanak(10), zagot(10,10), kpz(10), DU_STAN(50), Gruppa(10),  
* DU(100,10), Nomer(10)  
      zaqwka=0  
C      write(*,*) 'RASPR вход, zagsum-', zagsum  
      net1=0  
1      do 37 i=1, zagsum  
        if (posobr(i).eq.0) goto 37  
2        if (posobr(i).eq.obrzag(pribyt)) goto 40  
        if (posobr(i).eq.DU(Nomer(pribyt), pribyt)) goto 40  
C        Write(*,89) pribyt, ObrZag(pribyt)  
c89      format(' ', 'На станок' , i3, ' не найдено. На столе ДУ', i4)  
C  
C      проверим, а есть ли подобная заготовка где-нибудь на станке  
C  
3      y=0  
      do 30 j=1, R  
C      if (j.eq.pribyt) goto 30  
28      if (posobr(i).eq.obrzag(j)) y=y+1  
      if (Nomer(j).gt.0.and.posobr(i).eq.DU(Nomer(j), j)) y=y+1  
      if (Nomer(j).le.1) then  
        DO 29 il=1, stanak(j)  
29      if (posobr(i).eq.zagot(il, j)) y=y+1  
      end if  
30      continue  
C ... такой заготовки нет, поэтому запускаем ее на станок pribyt ....  
      if (y.lt.1) goto 40  
37      continue  
C ... нечем догрузить данный станок ; заявки от него не выполняем  
      if (kolPZ.lt.r) goto 97  
96      net1=1  
      goto 999  
C      write(*,*)  
C      WRITE(*,*) 'ДОГРУЖАЕМ'  
97      max=0  
      do 70 j=1, KolPZ  
        Sum(j)=0  
        do 50 i=1, zagsum  
          if (posobr(i).gt.0) then  
            ii=posobr(i)  
            TekKod=int(DU_STAN(ii)/10)  
            TekKod=DU_STAN(ii)-TekKod*10  
          else  
            TekKod=0  
          end if  
          If (TekKod.ne.Gruppa(pribyt)) goto 50  
49      if (posobr(i).eq.KPZ(j)) sum(j)=sum(j)+ttpmsh(j)+ttpwsp(j)  
50      continue  
          if (max.ge.sum(j)) goto 70  
67      max=sum(j)  
          y=j  
70      continue  
          do 80 i=1, zagsum  
            if (posobr(i).gt.0.and.posobr(i).eq.KPZ(y)) goto 40  
80      continue  
      write(*,*) 'ошибка выбора очередной ДУ для трансп. тележки!'  
      pause  
c39      write(*,*) 'догружаем ', KPZ(y)  
C      pause  
40      zaqwka=i
```

```

c      write(*,11) pribyt,Obrzag(pribyt),(zagot(i,pribyt),
c      * i=1,Stanak(pribyt)),zaqwka,posobr(zaqwka)
c11    format(' ',20(/),' ','стан-',i3,':',7(i4,' '),
c      * ' заяв-',i3,' код-',i4)
c      pause
999    return
      end

      SUBROUTINE STATtt(tTTmin,shagTT,wrTRAN,warTT,iost)
c
c      накопление статистики по работе TT
c
c      real zna,tTTmin,shagTT,wrTRAN
c      integer i,warTT
c      dimension warTT(12)
c      do 10 i=1,12
c      zna=tTTmin+shagTT*i
10     IF (zna.ge.wrTRAN) goto 20
c      continue
c      pause
c      write(*,*) 'в STATtt бардак'
c      Write(*,*) 'wrtran=',wrtran,' ,a max=',zna
c      iost=10
20     warTT(i)=warTT(i)+1
c      return
c      end

      SUBROUTINE AWARTT(skortr,usktr,tsmtr,SINHRO,lnakop,kordst,
c      * zagot,rangza,stanak,pribyt,otpraw,zagtel,zagoto,wrTRAN,sklad,
c      * gnezdo,zagtol,SkldU,X,Y,KRDSKL,POZTR)
c      real kordst,udalen,wozwr,wrtran,skortr,usktr,tsmtr,lnakop,
c      * KRDSKL(2),wrem_X,POZTR
c      integer zagot,rangza,stanak,pribyt,otpraw,zagtel,gnezdo,
c      * zagoto,iiii,obraz,sklad,zagtol,X,Y,SkldU,A,SINHRO
c      Dimension Kordst(10),zagot(10,10),rangza(10,10),stanak(10),
c      * zagoto(400),zagtol(400),SKORTR(2),USKTR(2),LNAKOP(2),
c      * SKLAD(2)
c      Write(*,*) 'AWARTT'
c      выполнение аварийной заявки от pribyt станка
c      есть ли на TT ДУ :
c      if (zagtel.ne.0) goto 20
c      ... нет, считаем время перемещения к обсл станку:
10     wozwr=abs(kordst(pribyt)-kordst(otpraw))
c      wrtran=wozwr/skortr(1)+2*skortr(1)/usktr(1)+2*tsmtr
c      goto 30
c      .. есть, ищем гнездо в складе для ее установки
20     obraz=0
c      CALL PUSTO(kordst,lnakop,gnezdo,zagoto,sklad,otpraw,obraz,net,
c      * zagtol,SkldU,X,Y,KRDSKL)
c      .. нашли, везем туда заготовку, выгружаем , возвращаемся
c      к pribyt станку
c      ***** СКЛАД ВНЕШНИЙ *****
c      IF (SkldU.eq.3) THEN
c      udalen=abs(kordst(otpraw))
c      wozwr=abs(kordst(pribyt))
c      wrtran=(udalen+wozwr)/skortr(A)+2*skortr(A)/usktr(A)+2*tsmtr
c      goto 30
c      END IF
c      ***** СКЛАД - ЛИНЕЙНЫЙ *****
c      A=1
60     IF (SkldU.eq.1) X=gnezdo
c      udalen=abs(kordst(otpraw)-lnakop(A)*X-KRDSKL(A))
c      wozwr=abs(kordst(pribyt)-lnakop(A)*X-KRDSKL(A))
c      wrtran=(udalen+wozwr)/skortr(A)+2*skortr(A)/usktr(A)+2*tsmtr
c      IF (SkldU.eq.1) goto 30
c      ***** СКЛАД МНОГОЯРУСНЫЙ *****
c      If (A.lt.2) then
c      A=2
c      wrem X=wrtran
c      udalen=abs(PozTR-lnakop(2)*Y-KRDSKL(2))
c      wozwr=abs(PozTR-lnakop(2)*Y-KRDSKL(2))
c      goto 60

```

```

        end if
        If (SINHRO.NE.1) THEN
            wrtran=wrtran+wrem_X-2*tsmtr
        ELSE
            IF (wrtran.LT.wrem_X) wrtran=wrem_X
        END IF
C
C        ПОИСК В ПРИСТАНОЧНОМ НАКОПИТЕЛЕ pribyt СТАНКА
C        УДАЛЯЕМОЙ ЗАГОТОВКИ
C
30      do 40 i=1,stanak(pribyt)
        if(zagot(i,pribyt).gt.0.and.rangza(i,pribyt).eq.0) goto 50
40      continue
        Write(*,*) pribyt,' стан: нет удаляемых по аварийной заявке ДУ : '
        write(*,43) (zagot(i,pribyt),i=1,stanak(pribyt))
        write(*,43) (rangza(i,pribyt),i=1,stanak(pribyt))
43      format(1x,10(i3,2x))
        pause
C      перекодировка размещения заготовок
C
50      iiii=zagot(i,pribyt)
        zagot(i,pribyt)=0
        rangza(i,pribyt)=0
        zagoto(gnezdo)=zagtel
        zagtel=iiii
        otpraw=pribyt
999     return
        end
        SUBROUTINE PUSTO(kordst,lnakop,gnezdo,zagoto,sklad,otpraw,
* obraz,net,zagtol,SkldU,X,Y,KRDSKL)
        real kordst,lnakop,KRDSKL
        integer j1,i,i1,gnezdo,obraz,zagoto,sklad,otpraw,net,X,Y,
* zagtol,X1,SkldU,A
        dimension kordst(10),zagoto(400),zagtol(400),LNAKOP(2),
* SKLAD(2),KRDSKL(2)
        if (net.ne.5) net=1
C        write(*,*) 'ищем ДУ N',OBRAZ,' в ЦНДУ'
C ***** СКЛАД ДУ - ВНЕШНИЙ ИЛИ ОТСУТСТВУЕТ *****
        If (SkldU.eq.3.or.SkldU.eq.0) then
            gnezdo=1
            goto 999
        end if

        If (SkldU.eq.2) goto 1800
C
C ***** СКЛАД ДУ - ЛИНЕЙНЫЙ *****
C
C      ищем гнездо с требуемой заготовкой OBRAZ или пустое
C      в центральном складе
C      ( ИЩЕТСЯ ГНЕЗДО, БЛИЖАЙШЕЕ К ОБСЛУЖИВАЕМОМУ СТАНКУ )
C
C      if (net.ne.5) goto 8
C3     if (net.eq.5) write(*,*) 'ищем ДУ N',OBRAZ,' в ЦНДУ'
C      write(*,6) (i,i=1,25)
C6     format(1x,25(i3))
C      write(*,7) (zagtol(i),i=1,25)
C7     format(1x,25(i3))
32     Y=1
        j2=0
        j1=1
        i=int((kordst(otpraw)-KRDSKL(1))/lnakop(1))
        if (i.lt.1) i=1
        i1=i
        gnezdo=i
C700    write(*,11) gnezdo,zagoto(gnezdo)
C11     format(1x,'гнездо- ',i3,' ДУ N ',i2)
9       if (net.ne.5) goto 700
10      if (zagtol(gnezdo).le.0) goto 999
699     goto 715
700     if(zagoto(gnezdo).eq.obraz) goto 999
715     if (j1.gt.0) goto 737
720     j1=1

```



```

        if (i.ge.sklad(1)) goto 730
725    i=i+1
        gnezdo=i
        goto 9
730    if (j2.ge.10) goto 747
735    goto 740
737    j1=0
740    if (i1.gt.1) goto 745
742    j2=10
        goto 720
745    i1=i1-1
        gnezdo=i1
        goto 9
c747    write(*,*) 'ДУ N ',OBRAZ,' на складе нет'
747    net=0
        goto 999
        pause

c
c ***** СКЛАД ДУ - МНОГОЯРУСНЫЙ *****
c
c ищем гнездо с требуемой заготовкой OBRAZ или пустое
c                в центральном складе
c    ( ИЩЕТСЯ ГНЕЗДО, БЛИЖАЙШЕЕ К ОБСЛУЖИВАЕМОМУ СТАНКУ )
1800    Y=1
        j2=0
        j1=1
        X=int((kordst(otpraw)-KRDSKL(1))/lnakop(1))
        If (X.lt.1) X=1
        X1=X
        A=X
1710    gnezdo=Sklad(1)*(Y-1)+A
        if (net.ne.5) then
            if (zagoto(gnezdo).eq.obraz) goto 999
            else
                if (zagtol(gnezdo).le.0) goto 999
            end if
        if (Y.lt.Sklad(2)) then
            Y=Y+1
            goto 1710
        else
            Y=1
        end if
1715    if (j1.gt.0) then
        j1=0
        if (X1.le.1) then
            j2=10
            goto 1720
        else
            X1=X1-1
            A=X1
            goto 1710
        end if
    end if
1720    j1=1
    if (X.ge.sklad(1)) then
        if (j2.ge.10) then
            net=0
            goto 999
        else
            goto 1715
        end if
    else
        X=X+1
        A=X
        goto 1710
    end if
c999    Write(*,*)A,Y,gnezdo,Zagtol(gnezdo),Zagoto(gnezdo)
999    return
        END
c    SUBROUTINE GDEDU(obrzag(10),stanak(10),zagot(10,10),
c    * rangza(10,10),posobr(500),migust(10,10),gpssum,r,stan)
c    stan=0

```

```

c      Do i=1,r
c      Do j=1,stanak(i)
c      If (zagot(j,i).EQ.posobr(zawka).AND.migust(j,i).LE.gpssum)
c      then
c      stan=i
c      zagot(j,i)=0
c      migust(j,i)=0
c      goto 9999
c      end if
c      end do
c  end do
c      return
c      end

```

Приложение Б

(справочное)

Фрагмент исходного кода программы «Fania». Правила обслуживания

```
unit Algoritm;

interface
Uses    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, Grids, Menus, Buttons, Varr;

Type
    MM=array[1..100] of integer;

Var
    KodStol: array [1..100] of integer;
    KodRobot, Otpraw, Pribyt, poz: integer;
    ProstT, SumTT,
    Wremq, Nomin, WreOtr: double;
    Moment_Zaprosa, SumPR_S, ProStan, Konec,
    OtrStan, WreStan: array[1..100] of double;
    Transp: array[1..5000] of real;
    KodPN1, Priznak, Priznak1, KodPN: array[1..100,1..50] of integer;
    MomentPN1, MomentPN: array[1..100,1..50] of double;
    Zapro: array[1..100] of boolean;
    spisok: MM;

Procedure Prawilo_1(Var pribyt: integer);
Procedure Prawilo_2(Var pribyt: integer);
Procedure Prawilo_3(Var pribyt: integer);
Procedure Prawilo_4(Var pribyt: integer);
Procedure Prawilo_5(Var pribyt: integer);
Procedure Prawilo_6(Var pribyt: integer);
Procedure Prawilo_7(Var pribyt: integer);

implementation

Procedure Dopoln_1;
{дополнительно - по приоритету}
Var max, stanok: integer;
begin
{по приоритету}
max:=0; pribyt:=999;
for stanok:=1 to Stanki do
If spisok[stanok]>0 then
If Prior_Stanka[stanok]>max then max:= Prior_Stanka[stanok];
for stanok:=1 to stanki do
If (Spisok[stanok]>0) AND (Prior_Stanka[stanok]=max) then
begin
pribyt:=stanok;
end;
If pribyt=999 then
begin ShowMessage('Ошибка выбора станка по дополнительному правилу-1'); Halt; end;
end;

Procedure Dopoln_2;
{дополнительно - по времени транспортной операции}
Var stanok: integer;
min, Poisk, Dostawka, Transp: real;
begin
min:=999999.9; pribyt:=999;
```

```

for stanok:=1 to Stanki do
If Spisok[stanok]>0 then
begin
  If Tip<2 then
    begin
      Poisk:=ABS(Koord_Stanka[Otpraw]-KorSkl_X-Lskl_X*(Gnezdo-1));
      Dostawka:=ABS(Koord_Stanka[Pribyt]-KorSkl_X-Lskl_X*(Gnezdo-1));
    end
    else
    begin
      Poisk:= Koord_Stanka[Otpraw];
      Dostawka:=Koord_Stanka[Pribyt];
    end;
    Transp:=(Poisk+Dostawka)/Skor_X+2*T_sm);
    If Transp<min then begin Min:= Transp; pribyt:=stanok; end;
  end;
If pribyt=999 then
begin ShowMessage('Ошибка выбора станка по дополнительному правилу-2'); Halt; end;
end; {Dopoln_2}

  Procedure Dopoln_3;
{дополнительно - по min пути к ячейке склада}
Var stanok: integer;
    min,Poisk: real;
begin
  min:=999999.9; pribyt:=999;
  for stanok:=1 to Stanki do
    If Spisok[stanok]>0 then
      begin
        If Tip <2 then
          Poisk:=ABS(Koord_Stanka[Otpraw]-KorSkl_X-Lskl_X*(Gnezdo-1))
          else Poisk:=Koord_Stanka[Otpraw];
          If Poisk<min then begin Min:= Poisk; pribyt:=stanok; end;
        end;
      end;
If pribyt=999 then
begin ShowMessage('Ошибка выбора станка по дополнительному правилу-3'); Halt; end;
end; {Dopoln_3}

Procedure Wybor_Dop;
begin
  Case Prawilo_Dop of
    1: Dopoln_1;
    2: Dopoln_2;
    3: Dopoln_3;
    else ShowMessage('Нет такого доп. правила!');
  end; {of case}
end;

Procedure Prawilo_1(Var pribyt: integer);
{по очереди поступления заявок}
Var
    min: real;
    stanok, stankov: integer;
    ss: string;
    FF: textFile;
begin
  min:=Moment_Zaprosa[1]*5;
  for stanok:=1 to Stanki do
    If (Zapros[stanok]) AND
      (Moment_Zaprosa[stanok]<min) then min:=Moment_Zaprosa[stanok];
  stankov:=0;
  for stanok:=1 to stanki do
    If (Zapros[stanok]) AND (Moment_Zaprosa[stanok]=min) then
      begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end

```

```

        else spisok[stanok]:=0;
    If stankov=1 then Exit;
    If stankov>1 then Wybor_Dop
        else
        begin
            ss:=IntToStr(Stankov);
            ss:=Concat('Ошибка в правиле_1: станков ',ss);
            AssignFile(FF,'Prawilo_1.txt'); Rewrite(FF);
            for stanok:=1 to stanki do
                Writeln(FF, stanok:2, Zaproso[stanok]:8,Moment_Zaprosa[stanok]:8:1);
            CloseFile(FF);
            ShowMessage(ss); Halt;
        end;
end; {Prawilo_1}

```

```

Procedure Prawilo_2(Var pribyt: integer);
{с последней заявки}
Label 5,15;

```

```

Var
    max: real;
    mogno, zagwok,i,stanok,stankov: integer;
    Zapret: array[1..50] of boolean;
    ss: string;

```

```

Procedure Stoit;
Var stanok: integer;
begin
    { 1. есть ли простаивающие станки с учетом времени на их обслуживание}
    stankov:=0; max:=0;
    for stanok:=1 to Stanki do
        If (Zaproso[stanok]) AND (WreOtr{+stanki*Wremq}-Konec[stanok]>=0) then
            begin stankov:=stankov+1; Spisok[stanok]:=1; end
            else Spisok[stanok]:=0;

    If Stankov<1 then Exit;
    {есть простаивающие. Ищем среди них}
    for stanok:=1 to Stanki do
        begin
            If (Zaproso[stanok]) AND (Spisok[stanok]>0) AND
                (Moment_Zaprosa[stanok]>max) then max:=Moment_Zaprosa[stanok];
        end;
    stankov:=0;
    for stanok:=1 to stanki do
        If Zaproso[stanok] AND (Spisok[stanok]>0) AND (Moment_Zaprosa[stanok]=max)
            then begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end
            else spisok[stanok]:=0;
    end;{STOIT}

```

```

Procedure WSTANET;
Var stanok: integer;
    min: real;
begin
    {нет простаивающих. Ищем тот, который может остановиться в ближайшее время}
    stankov:=0; min:=9999.9;
    for stanok:=1 to Stanki do
        If (Zaproso[stanok]) AND (WreOtr+stanki*Wremq-Konec[stanok]>0) then
            begin stankov:=stankov+1; Spisok[stanok]:=1; end
            else Spisok[stanok]:=0;

    If Stankov<1 then Exit;
    for stanok:=1 to Stanki do
        begin
            If (Zaproso[stanok]) AND (Spisok[stanok]>0) AND
                (Konec[stanok]<min) then min:=Konec[stanok];
        end;
    end;

```

```

    end;
    stankov:=0;
for stanok:=1 to stanki do
    If Zapros[stanok] AND (Spisok[stanok]>0) AND (Konec[stanok]=min)
        then begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end
        else spisok[stanok]:=0;
end; {Wstanet}

Procedure ZapretNO;
Var stanok,j,k: integer;
begin
{for i:=1 to zaqwok do
    begin Str(Obslug[i]:2, ss); aa:=Concat(aa,ss); end;  ShowMessage(aa);}
    for stanok:=1 to Stanki do
        begin
            Zapret[stanok]:=false; k:=0;
            for j:=zaqwok downto zaqwok-mogno do
                If obslug[j]=stanok then k:=k+1;
            if k=mogno then Zapret[stanok]:=true;
            k:=0;
            for j:=1 to Nakop_Stanka[stanok] do
                If (KodPN1[stanok,j]>0) AND (MomentPN1[stanok,j]>WreOtr) then k:=k+1;
                If k=Nakop_Stanka[stanok] then Zapret[stanok]:=true;
            end;
end; {Zapret}

begin {Prawilo_2}
mogno:=1;{можно подряд операций}
ss:='';
{ считаем, сколько всего заявок выполнено}
i:=0; Repeat i:=i+1 Until Obslug[i]<1; zaqwok:=i;
STOIT; If stankov>0 then goto 15;
WSTANET; If stankov>0 then goto 15;
5: ZAPRETNO;
{ нет простаивающих. Ищем запрещенный по числу последовательных операций}
{ Ищем последний запрос с учетом запрета на обслуживание}
for stanok:=1 to Stanki do
    begin
        spisok[stanok]:=0;
        If (Not Zapret[stanok]) AND (Zapros[stanok]) AND
        { (WreOtr+stanki*Wremq-Konec[stanok]>0) AND}
        (Moment_Zaprosa[stanok]>max) then max:=Moment_Zaprosa[stanok];
    end;
stankov:=0;
for stanok:=1 to stanki do
    if (Not Zapret[stanok]) AND (Zapros[stanok]) AND (Moment_Zaprosa[stanok]=max)
then
    begin
        stankov:=stankov+1;
        spisok[stanok]:=1;
        pribyt:=stanok;
    end;
15: If stankov=0 then
    begin
        If mogno<3 then begin mogno:=mogno+1; goto 5; end
        else begin Prawilo_1(pribyt); Exit; end;
    end;
    If stankov=1 then begin {ShowMessage('правило 2');} Exit; end
    else begin {ShowMessage('не 2');} Wybor_Dop; end
end; {Prawilo_2}

Procedure Prawilo_3(Var pribyt: integer);
{по приоритетам}

```

```

Label 5, 15;
Var
  max, mogno, zaqwok,i,stanok,stankov: integer;
  Zapret: array[1..50] of boolean;
  ss: string;

Procedure Stoit;
Var stanok: integer;
begin
  { 1. есть ли простаивающие станки }
  stankov:=0; max:=0;
  for stanok:=1 to Stanki do
    If (Zapros[stanok]) AND (WreOtr-Konec[stanok]>=0) then
      begin stankov:=stankov+1; Spisok[stanok]:=1; end
      else Spisok[stanok]:=0;

  If Stankov<1 then Exit;
  {есть простаивающие. Ищем среди них}
  for stanok:=1 to Stanki do
    begin
      If (Zapros[stanok]) AND (Spisok[stanok]>0) AND
        (Prior_Stanka[stanok]>max) then max:=Prior_Stanka[stanok];
    end;
  stankov:=0;
  for stanok:=1 to stanki do
    If Zapros[stanok] AND (Spisok[stanok]>0) AND (Prior_Stanka[stanok]=max)
      then begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end
      else spisok[stanok]:=0;
  end;{STOIT}

Procedure WSTANET;
Var stanok: integer;
  min: real;
begin
  {нет простаивающих. Ищем тот, который может остановиться в ближайшее время}
  stankov:=0; min:=9999.9;
  for stanok:=1 to Stanki do
    If (Zapros[stanok]) AND (WreOtr+stanki*Wremq-Moment_Zaprosa[stanok]>0) then
      begin stankov:=stankov+1; Spisok[stanok]:=1; end
      else Spisok[stanok]:=0;

  If Stankov<1 then Exit;
  for stanok:=1 to Stanki do
    begin
      If (Zapros[stanok]) AND (Spisok[stanok]>0) AND
        (Moment_Zaprosa[stanok]<min) then min:=Moment_Zaprosa[stanok];
    end;
  stankov:=0;
  for stanok:=1 to stanki do
    If Zapros[stanok] AND (Spisok[stanok]>0) AND (Moment_Zaprosa[stanok]=min)
      then begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end
      else spisok[stanok]:=0;
  end; {Wstanet}

Procedure ZapretNO;
Var stanok,j,k: integer;
begin
  {for i:=1 to zaqwok do
    begin Str(Obslug[i]:2, ss); aa:=Concat(aa,ss); end; ShowMessage(aa);}
  for stanok:=1 to Stanki do
    begin
      Zapret[stanok]:=false; k:=0;
      for j:=zaqwok downto zaqwok-mogno do
        If obslug[j]=stanok then k:=k+1;
        if k=mogno then Zapret[stanok]:=true;
        k:=0;

```

```

        for j:=1 to Nakop_Stanka[stanok] do
            If (KodPN1[stanok,j]>0) AND (MomentPN1[stanok,j]>WreOtr) then k:=k+1;
            If k=Nakop_Stanka[stanok] then Zapret[stanok]:=true;
        end;
end; {Zapret}

begin {Prawilo_3}
max:=0;
mogno:=1;{можно подряд операций}
ss:='';
{ считаем, сколько всего заявок выполнено}
i:=0; Repeat i:=i+1 Until Obslug[i]<1; zaqwok:=i;
STOIT; If stankov>0 then begin {ShowMessage('проход 1');} goto 15; end;
WSTANET; If stankov>0 then begin {ShowMessage('проход 2');} goto 15; end;
5: ZAPRETNO;
{ нет простаивающих. }
{ Ищем запрос с учетом запрета на обслуживание}
max:=0;
for stanok:=1 to Stanki do
begin
    spisok[stanok]:=0;
    If (Not Zapret[stanok]) AND (Zapros[stanok]) AND
        (Prior_Stanka[stanok]>max) then max:=Prior_Stanka[stanok];
end;
stankov:=0;
for stanok:=1 to stanki do
    if (Not Zapret[stanok]) AND (Zapros[stanok]) AND (Prior_Stanka[stanok]=max)
then
begin
    stankov:=stankov+1;
    spisok[stanok]:=1;
    pribyt:=stanok;
end else spisok[stanok]:=0;
{ShowMessage('проход 3');}
15: If stankov=0 then
begin
    If mogno<3 then begin mogno:=mogno+1; goto 5; end
    else begin Prawilo_1(pribyt); Exit; end;
end;
If stankov=1 then begin {ShowMessage('правило 3');} Exit; end
else begin {ShowMessage('не 3');} Wybor_Dop; end
end; {Prawilo_3}

Procedure Prawilo_4(Var pribyt: integer);
{по мин трудоемкости заготовок в ПН}
Var
    sum: array[1..100] of real;
    min: real;
    stanok, gnezdo, stankov: integer;
begin
    Min:=999999.9;
    for stanok:=1 to stanki do
        if zapros[stanok] then
            begin
                spisok[stanok]:=0;
                sum[stanok]:=0;
                for gnezdo:=1 to Nakop_Stanka[stanok] do
                    begin
                        if (KodPN1[stanok,gnezdo]>0) then
                            if ((Priznak1[stanok,gnezdo]>0) AND (MomentPN1[stanok,gnezdo]<=WreOtr)) OR
                                ((Priznak1[stanok,gnezdo]=0) AND (MomentPN1[stanok,gnezdo]>WreOtr))
                            then
                                sum[stanok]:=sum[stanok]+WremqDU[KodPN1[stanok,gnezdo]]*60;
                            end;
                    end;
                end;
            end;
        end;
    end;
end;

```



```

        if sum[stanok]<min then min:=sum[stanok];
    end;
    stankov:=0;
    for stanok:=1 to stanki do
        if sum[stanok]=min then
            begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end;
        if stankov=1 then begin {ShowMessage('правило 4');} Exit; end;
        If (stankov>1) AND (min=0) then begin {ShowMessage('не 4');} Prawilo_1(pribyt);
    end;
    If stankov>1 then Wybor_Dop
        else begin ShowMessage('Ошибка в правиле_4'); Halt; end;
    end; {of Prawilo_4}

Procedure Prawilo_5(Var pribyt: integer);
{по мин. остаточному времени в рабочей зоне}
Var
    Ostatok: array[1..100] of real;
    maxN,min: real;
    X, gnezdo, stanok, stankov: integer;
begin
    min:=99999.9;
    for stanok:=1 to stanki do
        if zapros[stanok] then
            begin
                spisok[stanok]:=0;
                MaxN:=0; X:=999;
                for gnezdo:=1 to Nakop_Stanka[stanok] do
                    begin
                        if (KodPN1[stanok,gnezdo]>0) AND (KodPN1[stanok,gnezdo]<=Nomenk1)
                            AND(MomentPN1[stanok,gnezdo]>WreOtr) then
                            if MomentPN1[stanok,gnezdo]>maxN then
                                begin maxN:=MomentPN1[stanok,gnezdo]; X:= gnezdo; end;
                        end;
                        If X<999 then Ostatok[stanok]:=maxN-WremqDU[KodPN1[stanok,X]]*60
                            else Ostatok[stanok]:=0;
                        If Ostatok[stanok]< 0 then Ostatok[stanok]:=0;
                        If Ostatok[stanok]< Min then Min:=Ostatok[stanok];
                    end;
                end;
            end;
        stankov:=0;
        for stanok:=1 to stanki do
            if Ostatok[stanok]=min then
                begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end;
        if stankov=1 then begin {ShowMessage('правило 5');} Exit; end;
        If (stankov>1) AND (min=0) then begin {ShowMessage('не 5');} Prawilo_1(pribyt);
    end;
    If stankov>1 then Wybor_Dop
        else begin ShowMessage('Ошибка в правиле_5'); Halt; end;
    end; {of Prawilo_5}

Procedure Prawilo_6(Var pribyt: integer);
{по минимальному числу заготовок в пристаночном накопителе}
Var
    sum: array[1..100] of integer;
    min, stanok, gnezdo, stankov: integer;
begin
    Min:= Nakop_Stanka[1]*stanki;
    for stanok:=1 to stanki do
        if zapros[stanok] then
            begin
                spisok[stanok]:=0;
                sum[stanok]:=0;
                for gnezdo:=1 to Nakop_Stanka[stanok] do
                    begin

```

```

        if (KodPN1[stanok,гнездо]>0) then
    if ((Priznak1[stanok,гнездо]>0) AND (MomentPN1[stanok,гнездо]<=WreOtr)) OR
        ((Priznak1[stanok,гнездо]=0) AND (MomentPN1[stanok,гнездо]>WreOtr))
        then sum[stanok]:=sum[stanok]+1;
    end;
    if sum[stanok]<min then min:=sum[stanok];
    end;
    stankov:=0;
    for stanok:=1 to stanki do
        if sum[stanok]=min then
            begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end;
    If stankov=1 then Exit;
    If (stankov>1) AND (min=0) then Prawilo_1(pribyt);
    If stankov>1 then Wybor_Dop
        else begin ShowMessage('Ошибка в правиле_6'); Halt; end;
end; {of Prawilo_6}

Procedure Prawilo_7(Var pribyt: integer);
{по приоритетам}
Label 5, 15;
Var
mogno, zaqwok,i,stanok,stankov: integer;
Zapret: array[1..50] of boolean;
ss: string;
min: real;

Function Transp(Stanok: integer): real;
Var poisk, dostawka: real;
begin
If (Tip=0) OR (Tip=1) then
    begin
        Poisk:=ABS(Koord_Stanka[Otpraw]-KorSk1_X-Lsk1_X*(Gnezdo-1));
        Dostawka:=ABS(Koord_Stanka[Stanok]-KorSk1_X-Lsk1_X*(Gnezdo-1));
    end
else
    begin
        Poisk:=Koord_Stanka[Otpraw];
        Dostawka:=Koord_Stanka[Stanok];
    end;
    Transp:=(Poisk+Dostawka)/Skor_X+2*T_sm;
end; {Transp}

Procedure Stoit;
Var stanok: integer;
begin
{ 1. есть ли простаивающие станки }
    stankov:=0; min:=9999.9;
    for stanok:=1 to Stanki do
        If (Zapros[stanok]) AND (WreOtr-Konec[stanok]>0) then
            begin stankov:=stankov+1; Spisok[stanok]:=1; end
            else Spisok[stanok]:=0;

    If Stankov<1 then Exit;
    {есть простаивающие. Ищем среди них}
    for stanok:=1 to Stanki do
        begin
            If (Zapros[stanok]) AND (Spisok[stanok]>0) AND
                (TRANSP(stanok)<min) then min:=TRANSP(stanok);
        end;
    stankov:=0;
    for stanok:=1 to stanki do
        If Zapros[stanok] AND (Spisok[stanok]>0) AND (TRANSP(stanok)=min)
            then begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end
            else spisok[stanok]:=0;
    end;{STOIT}

```

```

Procedure WSTANET;
Var stanok: integer;
    min: real;
begin
{нет простаивающих. Ищем тот, который может остановиться в ближайшее время}
stankov:=0; min:=9999.9;
for stanok:=1 to Stanki do
    If (Zapros[stanok]) AND (WreOtr+stanki*Wremq-Moment_Zaprosa[stanok]>0) then
        begin stankov:=stankov+1; Spisok[stanok]:=1; end
        else Spisok[stanok]:=0;

    If Stankov<1 then Exit;
for stanok:=1 to Stanki do
    begin
        If (Zapros[stanok]) AND (Spisok[stanok]>0) AND
            (Moment_Zaprosa[stanok]<min) then min:=Moment_Zaprosa[stanok];
    end;
stankov:=0;
for stanok:=1 to stanki do
    If Zapros[stanok] AND (Spisok[stanok]>0) AND (Moment_Zaprosa[stanok]=min)
        then begin stankov:=stankov+1; spisok[stanok]:=1; pribyt:=stanok; end
        else spisok[stanok]:=0;
end; {Wstanet}

Procedure ZapretNO;
Var stanok,j,k: integer;
begin
{for i:=1 to zaqwok do
    begin Str(Obslug[i]:2, ss); aa:=Concat(aa,ss); end; ShowMessage(aa);}
    for stanok:=1 to Stanki do
        begin
            Zapret[stanok]:=false; k:=0;
            for j:=zaqwok downto zaqwok-mogno do
                If obslug[j]=stanok then k:=k+1;
            if k=mogno then Zapret[stanok]:=true;
            k:=0;
            for j:=1 to Nakop_Stanka[stanok] do
                If (KodPN1[stanok,j]>0) AND (MomentPN1[stanok,j]>WreOtr) then k:=k+1;
                If k=Nakop_Stanka[stanok] then Zapret[stanok]:=true;
            end;
        end;
end; {Zapret}

begin {Prawilo_7}
mogno:=1;{можно подряд операций}
ss:='';
{ считаем, сколько всего заявок выполнено}
i:=0; Repeat i:=i+1 Until Obslug[i]<1; zaqwok:=i;
STOIT; If stankov>0 then begin {ShowMessage('проход 1');} goto 15; end;
WSTANET; If stankov>0 then begin {ShowMessage('проход 2');} goto 15; end;
5: ZAPRETNO;
{ нет простаивающих. }
{ Ищем запрос с учетом запрета на обслуживание}
min:=9999.9;
for stanok:=1 to Stanki do
    begin
        spisok[stanok]:=0;
        If (Not Zapret[stanok]) AND (Zapros[stanok]) AND
            (TRANSP(stanok)<min) then min:=TRANSP(stanok);
    end;
stankov:=0;
for stanok:=1 to stanki do
    if (Not Zapret[stanok]) AND (Zapros[stanok]) AND (TRANSP(stanok)=min) then
        begin
            stankov:=stankov+1;

```

```

        spisok[stanok]:=1;
        pribyt:=stanok;
    end else spisok[stanok]:=0;
    { ShowMessage('проход 3');}
15: If stankov=0 then
    begin
        If mogno<3 then begin mogno:=mogno+1; goto 5; end
        else begin Prawilo_1(pribyt); Exit; end;
    end;
    If stankov=1 then begin {ShowMessage('правило 3');} Exit; end
    else begin {ShowMessage('не 3');} Wybor_Dop; end
end; {Prawilo_7}

begin

end.

```

Приложение В

(справочное)

Фрагмент исходного кода программы «Расписание». Описание работы оборудования

```
Procedure Model_ras;

Label 10;
Var
  Zapret_1, Posled, NomT, KodSklad, KodT: array [1..Zagot_D] of integer;
  Trud, MomentZap, MomentSklad: array [1..Zagot_D] of Extended;
  PutT: array [1..Zagot_D, 1..Grup_Stan] of integer;
  WreT: array [1..Zagot_D, 1..Grup_Stan] of real;
  KodStol: array [1..Stan_D] of integer;
  SumPR_S, ProStan, Konec, MomentStol: array [1..Stan_D] of Extended;
  ZaproS: array [1..Stan_D] of boolean;
  Zapret, Priznak, KodPN: array [1..Stan_D, 1..Nakop_D] of integer;
  MomentPN: array [1..Stan_D, 1..Nakop_D] of Extended;
  Transp: array [1..Trans_D] of real;
  WreOtr: array [1..Kol_TS_D] of extended;
  ProsTT, SumTT: array [1..Kol_TS_D] of real;
  Pribyt, KodRobot, Otpraw: array [1..Kol_TS_D] of integer;
  obslug, zaqwka, sos, sklad,
  Razmer_SZ, pozl, zz, poz, DU, SumT, ii: Integer;
  Nomin, Sum, Wremq: Extended;
  FF1: TextFile;

Procedure POSLED_ZAP;
Var i, j: integer;
begin
  Razmer_SZ:=0;
  For i:=1 to PZ do
    for j:=1 to RazmerPZ[i] do
      begin
        Razmer_SZ:= Razmer_SZ+1;
        Posled[razmer_SZ]:=KodPZ[i];
      end;
    end;
  end; {Posled_Zap}

Procedure WWOD;
Var i, j: integer;
begin
  sklad:=0;
  For i:=1 to Kol_TS do
    begin
      KodRobot[i]:=0; Otpraw[i]:=0; Pribyt[i]:=0;
      WreOtr[i]:=0; ProsTT[i]:=0; SumTT[i]:=0; Var_Time:=0;
    end;
  For i:=1 to Razmer_SZ do
    begin
      Trud[i]:=0;
      NomT[i]:=i; KodT[NomT[i]]:=Posled[i];
      For j:=1 to Grupp do begin PutT[NomT[i],j]:=Kod_DU[Posled[i],j];
        WreT[NomT[i],j]:=Wremq_DU[Posled[i],j];
        Trud[i]:=Trud[i]+Wremq_DU[Posled[i],j];
      end;
    end;
  end;
  For i:=1 to Stanki do
    begin
      KodStol[i]:=0; ProStan[i]:=0; SumPR_S[i]:=0; ZaproS[i]:=False; Konec[i]:=0;
      For j:=1 to PN do
```

```

begin
    KodPN[i,j]:=0; MomentPN[i,j]:=0; Priznak[i,j]:=0; Zapret[i,j]:=0;
end;
end;
Nomin:=0; SumT:=0;
For i:=1 to Razmer_SZ do
begin
    Zapret_1[i]:=0;
    for j:=1 to Grupp do
        begin
            If Kod_DU[Posled[i],j]>0 then Nomin:=Nomin+Wremq_DU[Posled[i],j];
            If Kod_DU[Posled[i],j]>0 then SumT:=SumT+1;
        end;
    end;
    SumT:=SumT+Razmer_SZ;
    zaqwka:=0;
    For i:=1 to PZ do
        For j:=1 to RazmerPZ[i] do
            begin zaqwka:=zaqwka+1; Posled[zaqwka]:=KodPZ[i]; end;
    end; {wwod}

    Procedure TABL;
    begin
        AssignFile(FF1,'Ciklgr.mod');
        Rewrite(FF1);
        Writeln(FF1,' №№ №№ ном код ст. ст. ст. прост время момент стан ном момент
оконч ');
        Writeln(FF1,' TC опер ДУ ДУ отпр обсл дост TC транс прибыт смДУ ДУ смены об-
раб ');
    end;

    Procedure Pros_TT;
    begin
        ProsTT[TS]:=ProsTT[TS]+1; WreOtr[TS]:=WreOtr[TS]+1;
    end;

    Procedure Wybor_DO(var obslug,poz1,DU: integer);
    Var i,j: integer;
    mig: real;
    begin
        {ищем ДУ, обработанную на каком-либо станке}
        mig:=Var_Time{WreOtr[TS]}; obslug:=999; poz1:=999;
        For i:=1 to Stanki do
            begin
                for j:=1 to PN do
                    If Zapret[i,j]=0 Then
                        If (KodPN[i,j]>0) AND (Priznak[i,j]=0) AND (MomentPN[i,j]<=mig) THEN
                            begin mig:=MomentPN[i,j]; poz1:=j; obslug:=i; DU:= KodPN[i,j]; end;
                end;
            end;
        If obslug>PN then
            {нет удаляемой ДУ на станках. Запускаем новую}
            begin
                for i:=1 to Razmer_SZ do
                    If (Posled[i]>0) AND (Zapret_1[i]<1) then
                        begin
                            zaqwka:=i; DU:=NomT[zaqwka]; obslug:=0;
                            MomentZap[zaqwka]:=WreOtr[TS];
                            Exit;
                        end;
                end;
            end;
        end; {Wybor_DO}

    Procedure Wybor_RM(DU: integer; Var pribyt,poz: integer);
    Var i,j,x,Gruppa,max: integer;

```

```

begin
{определяем очередную max группу станков Gruppa. Если max=0, то }
{заготовка полностью обработана, везем на склад (pribyt=0)}
max:=0; gruppa:=0;
For j:=1 to Grupp do
  If PutT[DU,j]>max then begin max:=PutT[DU,j]; Gruppa:=j; end;
  If Max=0 then begin pribyt:=0; Exit; end;
  {..max>0. Ищем свободный станок из группы}
pribyt:=999;
For i:=1 to Stanki do
begin
  If Gruppa_Stan[i]=Gruppa then
  begin
    x:=0;
    for j:=1 to PN do
      If KodPN[i,j]=0 then begin pribyt:=i; poz:=j; x:=x+1; end;
    If x>1 then Exit;
  end;
end;
{нет свободного станка для продолжения обработки данной ДУ}
{ Ставим запрет на ее выбор процедурой WyborDU}
zapret[obslug,poz]:=1;
If obslug=0 then Zapret_1[DU]:=1;
pribyt:=999;
end; {Wybor_RM}

Procedure Wybor_TS;
// выбор транспортного средства
Var Min: extended;
    i: integer;
begin
  Min:=WreOtr[1]; TS:=1;
  For i:=1 to Kol_TS do
    If WreOtr[i]<Min then begin Min:=WreOtr[i]; TS:=i; end;
  Var_Time:=min;
end; {Wybor_TS}

Procedure DOSTAWKA;
begin
{расчет длительности транспортной операции. Приращение времени}
Wremq:=(ABS(Koord[Obslug]-Koord[otpraw[TS]])/Skor_TT+T_sm);
Wremq:=wremq+(ABS(Koord[Otpraw[TS]]-Koord[Pribyt[TS]])/Skor_TT+T_sm);
Transp[sos]:=Wremq;
WreOTR[TS]:=WreOtr[TS]+Wremq;
Write(FF1,TS:2,sos:5,DU:5,KodT[DU]:4,otpraw[TS]:5,obslug:5,
pribyt[TS]:5,prostT[TS]:5:0,wremq:7:0,WreOtr[TS]:7:0{,TS:5});
zz:=0;
end; {Dostawka}

Procedure ZAPOMN(DU,pribyt,poz: integer);
Var i,j: integer;
begin
  { перекодируем состояние}
  KodRobot[TS]:=DU;
  If pribyt>0 then
  begin
    KodPN[pribyt,poz]:=KodRobot[TS];
    MomentPN[pribyt,poz]:=WreOtr[TS];
    Priznak[pribyt,poz]:=1;
    If DU=NomT[zaqwka] then Posled[zaqwka]:=0;
  end
  Else
  begin
    sklad:=sklad+1; KodSklad[sklad]:=KodRobot[TS];

```

```

    MomentSklad[KodRobot[TS]]:=WreOtr[TS];
end;
If (obslug>0) then
begin KodPN[obslug,poz1]:=0; MomentPN[obslug,poz1]:=0; end;
For i:=1 to Stanki do
    For j:=1 to PN do zapret[i,j]:=0;
KodRobot[TS]:= 0; otpraw[TS]:=pribyt; ProstT[TS]:=0;
for i:=1 to Razmer_SZ do Zapret_1[i]:=0;
end; {PEREKODIR}

Procedure WYGRUZ(stanok,гнездо: integer);
Var j: integer;
begin
    { выгрузка заготовки }
    j:=KodStol[stanok];
    KodPN[stanok,гнездо]:=j;
    KodStol[stanok]:=0;
    PutT[j,Gruppa_Stan[stanok]]:=0;
    Priznak[stanok,гнездо]:=0;
    MomentPN[stanok,гнездо]:=Konec[stanok]+Smena_DU;
    Konec[stanok]:=MomentPN[stanok,гнездо];
end;

Function Sost_RM(stanok:integer; Var гнездо: integer): boolean;
    Var    min: real;
           j: integer;
begin
    {Ищем в ПН самую раннюю доставленную заготовки}
    Sost_RM:=false;
    Min:=WreOtr[TS]{Var_Time}; гнездо:=999;
    {Ищем ДУ в накопителе для установки в раб. зону}
    for j:=1 to PN do
        begin
            If (KodPN[stanok,j]>0) AND (MomentPN[stanok,j]<=MIN) AND
                (Priznak[stanok,j]>0) THEN
                begin MIN:=MomentPN[stanok,j]; гнездо:=j; end;
        end;
    {не нашли. Если в раб.зоне есть обработанная ДУ, выгружаем в накопитель }
    If (гнездо>PN) AND (KodStol[stanok]>0) then
        begin
            for j:=1 to PN do If KodPN[stanok,j]=0 then гнездо:=j;
                WYGRUZ(stanok,гнездо); гнездо:=999;
            end;
        If Gнездо<999 then Sost_RM:=true;
    end;

Procedure SmenaDU(stanok,гнездо:integer);
Var j: integer;
    AA: real;
begin
    If Konec[stanok]=0 then Konec[stanok]:=WreOtr[TS];
    j:=KodStol[stanok];          KodStol[stanok]:=KodPN[stanok,гнездо];
    KodPN[stanok,гнездо]:=j;    PutT[j,Gruppa_Stan[stanok]]:=0;
    Priznak[stanok,гнездо]:=0;
    AA:=0; If (j>0) AND (Konec[stanok]<=MomentPN[stanok,гнездо]) then
        AA:=Konec[stanok]+Smena_DU;
    If (Konec[stanok]>=MomentPN[stanok,гнездо]) then
        begin
            ProStan[stanok]:=0;
            Konec[stanok]:=Konec[stanok]+Smena_DU;
        end
        ELSE
        begin
            ProStan[stanok]:=MomentPN[stanok,гнездо]-Konec[stanok];

```



```

        Konec[stanok]:=MomentPN[stanok,gnezdo]+Smena_DU;
    end;
    MomentStol[stanok]:=Konec[stanok];
    MomentPN[stanok,gnezdo]:=MomentStol[stanok];
    If AA>0 then MomentPN[stanok,gnezdo]:=AA;
If KodStol[stanok]>0 then
    Konec[stanok]:=Konec[stanok]+WreT[KodStol[stanok],Gruppa_Stan[stanok]]*60;
SumPR_S[stanok]:=SumPR_S[stanok]+ProStan[stanok];
If KodPN[stanok,gnezdo]=0 then MomentPN[stanok,gnezdo]:=0;
end; {SmenaDU}

Procedure Analiz_RM;
Var stanok: integer;
begin
    ii:=0;
    For stanok:=1 to Stanki do
        If (Konec[stanok]<=WreOtr[TS]) AND Sost_RM(stanok,poz) then
            begin
                SmenaDU(stanok,poz); zz:=1;
                Writeln(FF1,stanok:4,KodStol[stanok]:4, MomentStol[stanok]:8:0,Konec[stanok]:8:0);
                end;
            If zz=0 then begin Writeln(FF1); zz:=1; end;
    end; {Analiz_RM}

Procedure Control;
Var i,j: integer;
begin
    If (sos>35) then
        begin
            Writeln('wreOtr-',WreOtr[TS]:6:0);
            For i:=1 to Stanki do
                begin
                    Writeln('станок-',i:2,' на столе-',KodStol[i]:2,' конец-',Konec[i]:7:0);
                    for j:=1 to PN do Write(j:6); Writeln;
                    for j:=1 to PN do Write(KodPN[i,j]:6); Writeln;
                    for j:=1 to PN do Write(Priznak[i,j]:6); Writeln;
                    for j:=1 to PN do Write(MomentPN[i,j]:6:0); Writeln;
                    for j:=1 to PN do Write(Zapret[i,j]:6); Writeln;
                end;
            Writeln('sos-',sos:2,'DU-',DU:2,' obslug-',obslug:2,' pribyt-',pribyt[TS]:3,' poz-
',poz:3);
            Readln;
        end;
    end;

Procedure Grafik;
Const Interw=10;
Type
    MM= array [1..Zagot_D] of real;
    NN= array [0..Interw+2] of double;
Var i,j: integer;
Zna: MM; Kol: NN;
Delta, min, max: real;
begin
    {Статистика по фактической трудоемкости}
{ Form2.Chart1.Series[0].Clear;
Form2.Chart2.Series[0].Clear;}
Min:=999; Max:=0; for j:=0 to Interw do Kol[j]:=0;
For i:=1 to Razmer_SZ do
    begin
        Zna[i]:=(MomentSklad[i]-MomentZap[i])/60;
        If Min>Zna[i] then Min:=Zna[i];
        If Max<Zna[i] then Max:=Zna[i];
    end;
end;

```

```

end;
Delta:=(Max-Min)/Interw;
for i:=1 to Razmer_SZ do
for j:=0 to Interw do
    If (Zna[i]>=Min+Delta*j) AND (Zna[i]<Min+Delta*(j+1))then Kol[j]:=Kol[j]+1;
end;

end;

Procedure Wywod;
Var      i: integer;
        Konec: real;
begin
    Konec:=WreOtr[1];
    For i:=1 to TS do  If Konec<WreOtr[i] then Konec:= WreOtr[i]; Konec:=Konec/60;
    Kzag:= (Nomin/(Konec*Stanki))*100;
    Form2.memol.Lines[0]:=Concat('Изготовлено деталей, штук .....
',IntToStr(Razmer_SZ));
    Form2.memol.Lines[1]:=Concat('Плановая трудоемкость задания, мин.....
',FloatToStr(Nomin));
    Form2.memol.Lines[2]:=Concat('Фактическая трудоемкость, мин.
.....',FloatToStr(RoundTo(Konec*Stanki,-0)));
    Form2.memol.Lines[3]:=Concat('Цикл работы оборудования, мин. ....
',FloatToStr(RoundTo(Konec,-0)));
    Form2.memol.Lines[4]:=Concat('Среднее время выпуска деталей, шт/мин
... ',FloatToStr(RoundTo(Razmer_SZ/Konec,-2)));
    Form2.memol.Lines[5]:=Concat('Коэффициент загрузки оборудования,% .....
',FloatToStr(RoundTo(Kzag,-1)));

    With Form2.StringGrid1 do
    begin
    ColWidths[0]:=150;
    Cells[0,0]:='Очередь запуска';
    Cells[0,1]:='Коды заготовок';
    Cells[0,2]:='Время по ТП, мин.';
    Cells[0,3]:='Очередь выпуска';
    Cells[0,4]:='Моменты запуска';
    Cells[0,5]:='Моменты выпуска';
    Cells[0,6]:='Время факт.,мин.';
    Cells[0,7]:='Эффективность, %';
    ColCount:=Razmer_SZ+1;
    { Width:=ColWidths[0]+(Razmer_SZ-1)*(DefaultColWidth);
    If Width>Form2.Width then Width:=Form2.Width-30;}
    end;
    With Form2.StringGrid1 do
    begin
        For i:=1 to Razmer_SZ do
        begin
            Cells[i,0]:=IntToStr(i);
            Cells[i,1]:=IntToStr(KodT[i]);
            Cells[i,2]:=FloatToStr(RoundTo(Trud[i],-1));
            Cells[i,3]:=IntToStr(KodSklad[i]);
            Cells[i,4]:=FloatToStr(RoundTo(MomentZap[i]/60,-1));
            Cells[i,5]:=FloatToStr(RoundTo(MomentSklad[i]/60,-1));
            Cells[i,6]:=FloatToStr(RoundTo((MomentSklad[i]-MomentZap[i])/60,-1));
            Cells[i,7]:=FloatToStr(RoundTo((Trud[i]*100*60/(MomentSklad[i]-Moment-
Zap[i])), -0));
        end;
    end;
    { Grafik;}
end;

begin {main}

```

```

    POSLED_ZAP;
    TABL;
    WWOD;
    sos:=0;
    TS:=1;
    Repeat
        begin
10:    WYBOR_TS;
        Wybor_DO(obslug,poz1,DU);
        If obslug>=999 then begin {Writeln('*'); Control;} Pros_TT; Analiz_RM; goto
10; end;
        Wybor_RM(DU,pribyt[TS],poz);
        If pribyt[TS]>=999 then begin {Writeln('*');} Pros_TT; Analiz_RM; goto 10;
    end;
        sos:=sos+1;
    {    Control;}
        DOSTAWKA;
        ZAPOMN(DU,pribyt[TS],poz);
        ANALIZ_RM;
        end;
    Until sos=SumT;
    Close(FF1);
    WYWOD;
    end;
end.

```