

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа природных ресурсов
 Направление подготовки 21.04.01 Нефтегазовое дело
 Отделение школы (НОЦ) Отделение нефтегазового дела

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
«Использование генетического алгоритма для оптимизации системы разработки месторождения углеводородов на примере синтетического аналога»

УДК 622.276-048.34

Студент

Группа	ФИО	Подпись	Дата
2ТМ01	Новак Максим Викторович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Матвеев И.В.	к.ф-м.н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Инженер-исследователь	Мельников М.О.	-		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Рукавишников В.С.	Ph.D.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Сечин А.А.	к.т.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Чернова О.С.	д.г-м.н.		

**Результаты освоения основной образовательной программы
Petroleum Engineering /Нефтегазовый инжиниринг**

Категория компетенций	Код и наименование компетенции	Индикаторы достижения компетенции
Системное и критическое мышление	УК-1. Способен осуществлять критический анализ проблемных ситуаций на основе системного анализа, вырабатывать стратегию действий	И.УК(У)-1.1. Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
		И.УК(У)-1.2. Определяет пробелы в информации, необходимой для решения проблемной ситуации, и проектирует процессы по их устранению
		И.УК(У)-1.3. Разрабатывает стратегию решения проблемной ситуации на основе системного и других современных междисциплинарных подходов; обосновывает выбор темы исследований на основе анализа явлений и процессов в конкретной области научного знания
		И.УК(У)-1.4. Использует логико-методологический инструментарий для критической оценки современных концепций в своей предметной области
Разработка и реализация проектов	УК(У)-2. Способен управлять проектом на всех этапах его жизненного цикла	И.УК(У)-2.1. Определяет проблему и способ ее решения через реализацию проектного управления
		И.УК(У)-2.2. Разрабатывает концепцию проекта в рамках обозначенной проблемы: формулирует цель, задачи, обосновывает актуальность, значимость, ожидаемые результаты и возможные сферы их применения
		И.УК(У)-2.3. Осуществляет мониторинг за ходом реализации проекта, корректирует отклонения, вносит дополнительные изменения в план реализации проекта
Командная работа и лидерство	УК(У)-3. Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	И.УК(У)-3.1. Планирует и корректирует свою социальную и профессиональную деятельность с учетом интересов, особенностей поведения и мнений людей, с которыми работает и взаимодействует

Категория компетенций	Код и наименование компетенции	Индикаторы достижения компетенции
		И.УК(У)-3.2. Организует дискуссии по заданной теме и обсуждение результатов работы команды
		И.УК(У)-3.3. Планирует командную работу, распределяет поручения и делегирует полномочия членам команды
Коммуникация	УК(У)-4. Способен применять современные коммуникативные технологии, в том числе на иностранном(ых) языке(ах), для академического и профессионального взаимодействия	И.УК(У)-4.1. Решает конкретные задачи профессиональной деятельности на основе академического и профессионального взаимодействия с учетом анализа мнений, предложений, идей отечественных и зарубежных коллег
		И.УК(У)-4.2. Составляет, переводит и редактирует различные академические тексты (рефераты, эссе, обзоры, статьи и т.д.)
		И.УК(У)-4.3. Представляет результаты академической и профессиональной деятельности на различных научных мероприятиях, включая международные
		И.УК(У)-4.4. Планирует и организывает совещания, деловые беседы, дискуссии по заданной теме; аргументированно и конструктивно отстаивает свою точку зрения, позицию, идею в академических и профессиональных дискуссиях на государственном и иностранном языках
Межкультурное взаимодействие	УК(У)-5. Способен анализировать и учитывать разнообразие культур в процессе межкультурного взаимодействия	И.УК(У)-5.1. Осуществляет профессиональную и социальную деятельность с учетом особенностей поведения и мотивации людей различного социального и культурного происхождения, в том числе особенностей деловой и общей культуры представителей других этносов и конфессий
		И.УК(У)-5.2. Выстраивает социальное и профессиональное взаимодействие с учётом особенностей деловой и общей культуры представителей разных этносов и конфессий, других социальных групп
		И.УК(У)-5.3. Обеспечивает создание недискриминационной среды для участников межкультурного взаимодействия при личном общении и при выполнении профессиональных задач

Категория компетенций	Код и наименование компетенции	Индикаторы достижения компетенции
Самоорганизация и саморазвитие (в том числе здоровьесбережение)	УК(У)-6. Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	И.УК(У)-6.1. Анализирует использование рабочего времени в широком спектре деятельности: планирование, распределение, постановка целей, делегирование полномочий, анализ временных затрат, мониторинг, организация, составление списков и расстановка приоритетов
		И.УК(У)-6.2. Сочетает выполнение текущих производственных задач с повышением квалификации; корректирует планы в соответствии с имеющимися ресурсами
		И.УК(У)-6.3. Планирует профессиональную траекторию с учетом особенностей как профессиональной, так и других видов деятельности и требований рынка труда

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа природных ресурсов
 Направление подготовки (специальность) 21.04.01 Нефтегазовое дело
 Отделение школы (НОЦ) Отделение нефтегазового дела

УТВЕРЖДАЮ:
 Руководитель ООП

 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

Магистерской диссертации (бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
2ТМ01	Новак Максиму Викторовичу

Тема работы:

«Использование генетического алгоритма для оптимизации системы разработки месторождения углеводородов на примере синтетического аналога»	
Утверждена приказом директора (дата, номер)	№70-34/с от 11.03.2022

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Возможные значения параметров систем разработки, возможные значения параметров месторождения, возможные значения дебитов скважин, возможные значения экономических параметров.</p> <p>Тексты и графические материалы отчетов и исследовательских работ, фондовая и периодическая литература, нормативные документы.</p>
---	--

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>Генетические алгоритмы; Применение генетического алгоритма для оптимизации системы разработки синтетической модели месторождения нефти; Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; Социальная ответственность.</p>
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Рисунок 1.1 – Разница между глобальными и локальными экстремумами функции. Рисунок 1.2 – Общая схема работы генетических алгоритмов. Рисунок 2.1 – Схема оптимизационной системы. Рисунок 2.2 – Пример зависимости дебита скважины от времени. Рисунок 2.3 – Пример профиля добычи нефти. Рисунок 2.4 – Пример графика изменения значения целевой функции в поколении. Рисунок 2.5 – Пример столбчатой диаграммы количества поколений в итерации. Рисунок 2.6 – Пример столбчатой диаграммы количества расчетов в итерации. Рисунок 2.7 – Пример столбчатой диаграммы доли возможных вариантов в итерации. Рисунок 2.8 – Пример столбчатой диаграммы максимального значения целевой функции в итерации. Рисунок 2.9 – График тестовой функции чистой приведенной стоимости. Рисунок 2.10 – Зависимость среднего количества поколений в итерации от гиперпараметров для тестовой функции. Рисунок 2.11 – Зависимость среднего общего количества расчетов в итерации и количества расчетов до получения максимального результата от гиперпараметров для тестовой функции. Рисунок 2.12 – Зависимость средней доли от возможных вариантов от гиперпараметров для тестовой функции. Рисунок 2.13 – Зависимость медианных максимумов метода случайных чисел и генетического алгоритма от гиперпараметров для тестовой функции. Рисунок 2.14 – Зависимость доли достижения генетическим алгоритмом абсолютного максимума и доли случаев, когда генетический алгоритм работал лучше метода случайных параметров от гиперпараметров для тестовой функции. Рисунок 2.15 – Зависимость среднего количества поколений в итерации от гиперпараметров.</p>

	<p>Рисунок 2.16 – Зависимость среднего общего количества расчетов в итерации и количества расчетов до получения максимального результата от гиперпараметров.</p> <p>Рисунок 2.17 – Зависимость средней доли от возможных вариантов от гиперпараметров.</p> <p>Рисунок 2.18 – Зависимость медианных максимумов метода случайных чисел и генетического алгоритма от гиперпараметров.</p> <p>Рисунок 2.19 – Зависимость доли достижения генетическим алгоритмом абсолютного максимума и доли случаев, когда генетический алгоритм работал лучше метода случайных параметров от гиперпараметров.</p> <p>Рисунок 3.1 – Пример профиля дохода с продажи нефти.</p> <p>Рисунок 3.2 – Пример графика накопленного дисконтированного денежного потока.</p> <p>Рисунок 4.1 – Схема вертикальных расстояний для светильников.</p> <p>Рисунок 4.2 – Схема размещения светильников на плане.</p>
--	---

Консультанты по разделам выпускной квалификационной работы

(с указанием разделов)

Раздел	Консультант
Генетические алгоритмы	Мельников М.О., инженер-исследователь
Применение генетического алгоритма для оптимизации системы разработки синтетической модели месторождения нефти	Мельников М.О., инженер-исследователь
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Рукавишников В.С., доцент, Ph.D.
Социальная ответственность	Сечин А.А., доцент, к.т.н.
Названия разделов, которые должны быть написаны на русском и иностранном языках:	
Генетические алгоритмы / Genetic algorithms	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Матвеев И.В.	к.ф-м.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
2ТМ01	Новак Максим Викторович		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа природных ресурсов
 Направление подготовки (специальность) 21.04.01 Нефтегазовое дело
 Уровень образования Магистратура
 Отделение школы (НОЦ) Отделение нефтегазового дела
 Период выполнения Весенний семестр 2021/2022 учебного года

Форма представления работы:

Магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
01.04.2022	Изучение литературы по выбранной теме, написание теоретического раздела	25
10.04.2022	Написание раздела на английском языке	15
05.05.2022	Написание программы на языке программирования Python 3 и практического раздела по ней	30
15.05.2022	Написание раздела по финансовому менеджменту, ресурсоэффективности и ресурсосбережение	15
24.05.2022	Написание раздела по социальной ответственности	15

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Матвеев И.В.	к.ф-м.н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Инженер-исследователь	Мельников М.О.	-		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор	Чернова О.С.	д.г-м.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
2ТМ01	Новак Максиму Викторовичу

Школа	ИШПР	Отделение школы (НОЦ)	Отделение нефтегазового дела
Уровень образования	Магистратура	Направление/специальность	21.04.01 Нефтегазовое дело

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

Возможные значения экономических параметров.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

Расчет чистой приведенной стоимости, рассмотрение других экономических параметров, используемых для оценки инвестиционной привлекательности проектов и рисков, связанных с ними.

Перечень графического материала (с точным указанием обязательных чертежей):

Рисунок 3.1 – Пример профиля дохода с продажи нефти.

Рисунок 3.2 – Пример графика накопленного дисконтированного денежного потока.

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Рукавишников В.С.	Ph.D.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
2ТМ01	Новак Максим Викторович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
2ТМ01		Новак Максиму Викторовичу	
Школа	Инженерная школа природных ресурсов	Отделение (НОЦ)	Отделение нефтегазового дела
Уровень образования	Магистратура	Направление/специальность	21.04.01 «Нефтегазовое дело»

Тема ВКР:

Использование генетического алгоритма для оптимизации системы разработки месторождения углеводородов на примере синтетического аналога	
Исходные данные к разделу «Социальная ответственность»:	
<p>Введение</p> <ul style="list-style-type: none"> – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения 	<p>Объект исследования: генетические алгоритмы для оптимизации системы разработки месторождений углеводородов.</p> <p>Область применения: разработка нефтяных, газовых и газоконденсатных месторождений.</p> <p>Рабочая зона: офис.</p> <p>Размеры помещения: 6 на 6 метров.</p> <p>Количество и наименование оборудования рабочей зоны: один стол, одно офисное кресло, один персональный компьютер.</p> <p>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне: разработка генетического алгоритма для оптимизации системы разработки месторождений углеводородов, используется компьютерная техника.</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<p>СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы;</p> <p>ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре;</p> <p>ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора;</p> <p>ГОСТ Р ИСО 9241-7-2007. Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов (ВДТ). Часть 7. Требования к дисплеям при наличии отражений.</p>
<p>2. Производственная безопасность при разработке проектного решения:</p> <ul style="list-style-type: none"> – Анализ выявленных вредных 	<p>Вредные производственные факторы:</p> <ul style="list-style-type: none"> - перенапряжение зрительного анализатора;

<p>и опасных производственных факторов</p> <p>– Расчет уровня опасного или вредного производственного фактора</p>	<p>- умственное перенапряжение;</p> <p>- повышенный уровень шума на рабочем месте;</p> <p>- повышенный уровень электромагнитных излучений;</p> <p>- повышенная или пониженная температура воздуха рабочей зоны;</p> <p>- недостаточная освещенность рабочей зоны.</p> <p>Опасные производственные факторы:</p> <p>- электрический ток, воздействующий на персонал.</p> <p>Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: осветительные приборы, оградительные устройства, звукоизолирующие устройства, устройства защитного заземления и зануления.</p> <p>Расчет: расчет системы искусственного освещения.</p>
<p>3. Экологическая безопасность <u>при разработке проектного решения:</u></p>	<p>Воздействие на селитебную зону: возникновение стихийных свалок компьютерной техники.</p> <p>Воздействие на литосферу: утилизация твердых отходов компьютерной техники.</p> <p>Воздействие на гидросферу: утилизация жидких отходов компьютерной техники.</p> <p>Воздействие на атмосферу: тепловое воздействие от работы систем охлаждения компьютерной техники.</p>
<p>4. Безопасность в чрезвычайных ситуациях <u>при разработке проектного решения:</u></p>	<p>Возможные ЧС:</p> <p>- геологические воздействия (землетрясения, оползни, обвалы, провалы территории и т.д.);</p> <p>- пожары, взрывы;</p> <p>- природные катастрофы (наводнения, цунами, ураган и т.д.).</p> <p>Наиболее типичная ЧС: пожар по причине возгорания компьютерной техники.</p>
<p>Дата выдачи задания для раздела по линейному графику</p>	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Сечин А.А.	к.т.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
2ТМ01	Новак Максим Викторович		

Реферат

Магистерская диссертация содержит 119 страниц, 25 рисунков, 8 таблиц, 60 источников и 2 приложения.

Ключевые слова: генетический алгоритм, система разработки, месторождение углеводородов, оптимизация, машинное обучение, моделирование, чистая приведенная стоимость.

Объект исследования: генетический алгоритм для оптимизации системы разработки месторождений углеводородов.

Цель работы: написание генетического алгоритма для оптимизации системы разработки месторождения углеводородов на примере синтетического аналога.

Для достижения данной цели ставились следующие научные задачи:

- 1) изучить генетические алгоритмы;
- 2) написать генетический алгоритм;
- 3) применить генетический алгоритм на тестовой функции и проанализировать результаты;
- 4) применить генетический алгоритм на синтетической модели месторождения нефти и проанализировать результаты.

Актуальность темы исследования. Верный подбор системы разработки во многом определяет экономическую эффективность проекта разработки месторождения. Генетические алгоритмы могут быть использованы для оптимизации системы разработки, что позволяет повысить рентабельность проектов.

Область применения: оптимизация системы разработки месторождений углеводородов.

Экономическая эффективность/значимость работы связана с улучшением системы разработки месторождений нефти и газа и экономией

рабочего времени инженеров-разработчиков месторождений нефти и газа и специалистов по геологическому и гидродинамическому моделированию.

Прогнозные предположения о развитии объекта исследования: улучшение алгоритма, тестирование на гидродинамических моделях месторождений нефти и газа с дальнейшим применением алгоритма на практике разработки месторождений.

Благодарности. Выражаю особую благодарность сотрудникам ЦППС НД ТПУ: И.В. Матвееву, М.О. Мельникову и О.С. Черновой за предоставленную помощь при написании данной магистерской диссертации.

Обозначения и сокращения

- ВНР (IRR) – внутренняя норма рентабельности (internal rate of return).
- ГА – генетический алгоритм.
- ГДМ – гидродинамическая модель.
- ГРП – гидравлический разрыв пласта.
- ГС – горизонтальный ствол.
- ДДП (DCF) – дисконтированный денежный поток (discounted cash flow).
- ДУ – допустимый уровень.
- ИТ – информационные технологии.
- КЕО – коэффициент естественной освещенности.
- МВК – максимальное вложение капитала.
- МО – машинное обучение.
- МПК – максимальная потребность в капитале.
- ПЗП – призабойная зона пласта
- ПК – персональный компьютер.
- ПЭВМ – персональная электронно-вычислительная машина
- СР – система разработки.
- УВ – углеводороды.
- ФП – функция приспособленности.
- ФЭМ – финансово-экономическая модель.
- ЦФ – целевая функция.
- ЧДП (NCF) – чистый денежный поток (net cash flow).
- ЧПС (NPV) – чистая приведенная стоимость (net present value).
- ЧС – чрезвычайная ситуация.
- ЭМП – электромагнитное поле.

Оглавление

Введение.....	17
1 Генетические алгоритмы	19
1.1 Введение в генетические алгоритмы	19
1.2 История разработки и использования генетических алгоритмов.....	21
1.3 Применение генетических алгоритмов.....	23
1.4 Отличия от традиционных алгоритмов	25
1.5 Достоинства и ограничения генетических алгоритмов	26
1.6 Общая схема работы генетических алгоритмов	29
1.7 Создание исходной популяции.....	30
1.8 Оценка целевой функции	31
1.9 Отбор особей для размножения.....	32
1.10 Скрещивание	34
1.11 Мутация.....	36
1.12 Расширение и редукция популяции	37
1.13 Проверка критериев остановки и завершение работы.....	38
1.14 Выводы по разделу	39
2 Применение генетического алгоритма для оптимизации системы разработки синтетической модели месторождения нефти	40
2.1 Описание системы	40
2.2 Параметры месторождения и систем разработки.....	41
2.3 Описание генетического алгоритма.....	48
2.4 Тестовая функция чистой приведенной стоимости	53
2.5 Функция чистой приведенной стоимости	59
2.6 Выводы по разделу	63
3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	65
4 Социальная ответственность.....	72
4.1 Правовые и организационные вопросы обеспечения безопасности ...	72

4.2 Производственная безопасность	74
4.3 Экологическая безопасность	83
4.4 Безопасность в чрезвычайных ситуациях	84
4.5 Выводы по разделу	85
Заключение	86
Список использованных источников	88
Приложение А	94
Приложение Б	113

Введение

Структура запасов месторождений нефти и газа за последние десятилетия претерпела значительные изменения. Возросла доля месторождений, имеющих относительно небольшие размеры, сложное строение, низкопроницаемые и нетрадиционные породы-коллекторы, высоковязкие флюиды [26]. В такой ситуации все более остро встает вопрос рентабельности разработки месторождений. Во многом экономическую эффективность разработки месторождений нефти и газа определяет применяемая система разработки [16].

В настоящее время информационные технологии широко используются инженерами-разработчиками месторождений углеводородов (УВ) в качестве помощи в принятии решений. В данной работе было выдвинуто предположение, что генетические алгоритмы (как один из оптимизационных методов машинного обучения) могут быть использованы в качестве такой помощи для оптимизации системы разработки. Для проверки данного предположения было решено создать синтетическую модель месторождения нефти и применить на ней генетический алгоритм для получения оптимальной системы разработки (с точки зрения экономики).

Объектом исследования данной работы являются генетические алгоритмы. Предметом исследования является генетический алгоритм, используемый для оптимизации системы разработки месторождений углеводородов.

Целью данной работы является написание генетического алгоритма для оптимизации системы разработки месторождения углеводородов на примере синтетического аналога.

Для достижения данной цели ставились следующие научные задачи:

- 1) изучить генетические алгоритмы;
- 2) написать генетический алгоритм;

3) применить генетический алгоритм на тестовой функции и проанализировать результаты;

4) применить генетический алгоритм на синтетической модели месторождения нефти и проанализировать результаты.

Актуальность исследования связана с тем, что верный подбор системы разработки во многом определяет экономическую эффективность проекта разработки месторождения. Генетические алгоритмы могут быть использованы для оптимизации системы разработки, что позволяет повысить рентабельность проектов.

Личный вклад автора заключается в написании генетического алгоритма и анализе его применения на синтетической модели месторождения.

Практическая значимость работы заключается в возможном применении написанного генетического алгоритма для оптимизации разработки месторождений нефти и газа инженерами-разработчиками месторождений углеводородов и специалистов по геологическому и гидродинамическому моделированию.

1 Генетические алгоритмы

Генетические алгоритмы являются концепцией, относящейся к информационным технологиям. Так как основная часть данной диссертации связана с ними, для понимания работы абсолютно необходимо иметь представление о генетических алгоритмах.

1.1 Введение в генетические алгоритмы

Генетические алгоритмы представляют собой семейство поисковых алгоритмов, которые основаны на эволюции живых существ, происходящей в природе [58]. Генетические алгоритмы используют упрощенную версию так называемой Дарвиновской эволюции, основные принципы которой могут быть сформулированы следующим образом.

Принцип изменчивости. Гены особей, принадлежащих к какой-либо популяции, могут отличаться от особи к особи, соответственно, отличаются сами особи [48].

Принцип наследования. Некоторые признаки постоянно передаются из поколения в поколение. Соответственно, потомки походят на своих предков больше, чем на другие особи в популяции [48].

Принцип отбора. Более приспособленные особи имеют более высокие шансы на выживание и рождение потомства [48].

Таким образом, эволюция способствует тому, что появляются более и более приспособленные особи [43].

Приведем аналоги между эволюцией живой природы и генетическими алгоритмами [60]:

- особь – возможное решение решаемой задачи;
- популяция (совокупность особей) – совокупность возможных решений решаемой задачи;

- ген (определенный признак особи) – какой-либо входной параметр, присущий возможному решению решаемой задачи;

- генотип (совокупность генов особи) – все входные параметры, присущие возможному решению решаемой задачи;

- приспособленность (иначе фитнес-функция, способность особи к выживанию и размножению, зависит от генотипа) – значение функции, которая является целевой и подлежит максимизации или минимизации, определяемое входными параметрами;

- отбор (иначе селекция, выбор особей, участвующих в размножении и передающих свой генотип потомкам) – выбор возможных решений, параметры которых будут использованы для формирования новых возможных решений;

- размножение (иначе кроссовер, кроссинговер или скрещивание, появление новых особей от уже существующих особей) – использование существующих возможных решений для формирования новых возможных решений;

- потомок (особь, продукт размножения, произведенный предшествующими особями – предками) – возможное решение, полученное на основе предшествующих возможных решений;

- предок (иначе родитель, особь, участвующая в размножении и производящая новую особь – потомка) – возможное решение, на основании которого формируются новые возможные решения;

- мутация (случайное изменение генотипа особи) – случайное изменение параметров возможного решения;

- редукция (снижение количества особей в популяции) – снижение количества возможных решений в совокупности данных возможных решений.

Таким образом, существуют особи, являющиеся частью популяции, которые имеют определенные признаки (генотип, состоящий из генов). Особи стремятся к выживанию и размножению, причем шанс на выживание и размножение тем выше, чем выше приспособленность особи, что определяется в процессе отбора. Размножаясь, особи производят новые особи (потомков),

которые наследуют определенные признаки родительских особей (предков). Также признаки потомков могут случайным образом меняться в процессе мутации. Так как шанс на выживание и размножение наиболее высок у самых приспособленных особей, потомкам передаются «хорошие» гены и приспособленность новых поколений растет. На этом и основана эволюция живых организмов и генетические алгоритмы [33].

В дальнейшем при упоминании терминов естественнонаучной эволюции будем понимать значение этих терминов в рамках генетических алгоритмов.

1.2 История разработки и использования генетических алгоритмов

Изучение эволюционных систем началось в 1950-х и 1960-х годах несколькими учеными независимо друг от друга. Предполагалось, что данные эволюционные системы могут быть использованы для нахождения решений различных инженерных задач. Как уже было сказано, концепция заключалась в эволюции определенной популяции решений-кандидатов для какой-либо задачи, используя операторы, вдохновленные эволюцией живой природы [45, 48].

В 1964 году немецкий ученый Инго Рехенберг представил концепцию эволюционных стратегий, метод был использован им для оптимизации параметров различных устройств (например, формы крыльев летательных аппаратов). Данная концепция была развита немецким ученым Хансом-Полом Швифелом. Отметим, что сфера эволюционных стратегий остается сферой, в которой происходят активные исследования, причем параллельно генетическим алгоритмам (хотя в последнее время эти две области начали активно взаимодействовать) [45, 48].

Британские ученые Фогель, Оуэнс и Уолш примерно в то же самое время разработали «эволюционное программирование» – метод, в котором кандидаты на решение поставленных задач представлялись в виде механизмов,

решающих определенные задачи, которые эволюционировали путем случайного изменения их параметров и отбора наиболее приспособленных. Эволюционное программирование также остается областью активных исследований. Вместе эволюционные стратегии, эволюционное программирование и генетические алгоритмы составляют основу области эволюционных вычислений [45, 48].

Несколько других ученых, работавших в 1950-х и 1960-х годах, разработали алгоритмы оптимизации и машинного обучения, также вдохновленные эволюцией. Кроме того, ряд эволюционных биологов использовали ЭВМ для моделирования эволюции с целью проведения экспериментов в сфере эволюционной биологии [45, 48].

Непосредственно генетические алгоритмы были изобретены Джоном Холландом в 1960-х годах и разрабатывались им и его студентами и коллегами в Мичиганском университете в 1960-х и 1970-х годах. В отличие от эволюционных стратегий и эволюционного программирования, Холланд не преследовал цель создания алгоритмов для решения конкретных задач, а изучал сам феномен адаптации, в виде, в котором он происходит в природе. Позже Холланд перешел к разработке способов переноса механизмов естественной адаптации в компьютерные системы. В итоге Холланд смог представить генетический алгоритм как модель биологической эволюции в компьютерных системах и выдвинул теоретические основы адаптации в рамках генетических алгоритмов. Генетические алгоритмы Холланда – это метод перехода от одной популяции «хромосом» (например, строк единиц и нулей, или «битов») к новой популяции с помощью «естественного отбора» вместе с операторами кроссинговера, мутации и инверсии, вдохновленными генетикой [45, 48].

Создание Холландом подобного алгоритма было важным нововведением. Более того, Холланд был первым, кто попытался поставить «вычислительную эволюцию» на прочную теоретическую основу. Этот теоретический фундамент был и остается основой почти всех последующих теоретических работ по генетическим алгоритмам [45, 48].

За последние несколько лет началось широкое взаимодействие между исследователями, изучающими различные методы эволюционных вычислений, и границы между генетическими алгоритмами, эволюционными стратегиями, эволюционным программированием и другими эволюционными подходами в определенной степени разрушились. Сегодня исследователи часто используют термин «генетический алгоритм» для описания всех этих концепций [45, 48].

1.3 Применение генетических алгоритмов

Генетические алгоритмы способны давать решения для различных задач оптимизации, обучения и поиска. Кроме того, генетические алгоритмы могут справляться с задачами, которые не могут быть решены с использованием классических поисковых и оптимизационных алгоритмов, особенно с задачами со сложными математическими представлениями и большим количеством параметров.

Генетические алгоритмы могут быть полезны для двух в значительной степени различных целей. Первая цель – это подбор параметров для оптимизации работы системы. Эти системы могут быть различны: системы газораспределительных трубопроводов, светофоры, коммивояжеры, распределение средств на проекты, составление расписания, обработка и смешивание материалов и так далее. Такие системы обычно зависят от входных параметров, на которых основаны принятия решений, выбираемые разработчиком или оператором системы. Правильный или неправильный выбор параметров решения приводит к тому, что система работает лучше или хуже, что измеряется достижением некоторой соответствующей цели [42, 47].

В реальных крупных системах взаимодействие между параметрами, как правило, не поддается аналитической обработке, и исследователю приходится прибегать к соответствующим методам поиска. Большинство опубликованных работ по генетическим алгоритмам посвящено использованию их для

оптимизации систем или, по крайней мере, для их улучшения путем приближения к оптимуму. Отметим, что в данной работе генетические алгоритмы используются именно в этом направлении [42, 47].

Второе потенциальное направление применения генетических алгоритмов обсуждается меньше, но также является очень важным и перспективным направлением – оно лежит в области тестирования и подгонки количественных моделей. Научное исследование проблемной области можно описать как итерационный процесс. Строится описательная модель, собираются данные и используются для проверки модели. Если обнаруживаются расхождения, модели изменяются. Процесс повторяется, пока проблема не будет решена (если генетический алгоритм работает правильно). В отличие от ситуации, когда генетические алгоритмы используются для максимизации производительности системы, в данном случае происходит поиск параметров, которые минимизируют несоответствие между моделью и реальными данными. Оптимизация включает в себя поиск значений параметров для модели, которые минимизируют невязку [42, 47].

Итак, было выделено две основные области применения генетических алгоритмов: оптимизация системы и подгонка количественной модели. Это можно сравнить с разницей между максимизацией показателей производительности системы и минимизацией расхождения между моделью и набором наблюдаемых данных. Это различие фактически несущественно, поскольку максимизация и минимизация всегда могут быть взаимозаменяемы. Максимизация производительности операционной системы эквивалентна минимизации ее отставания от какого-то оптимума. И наоборот, минимизация функции невязки эквивалентна максимизации отрицательного значения функции соответствия [42, 47].

Стоит отметить, что генетические алгоритмы уже используются и в программных продуктах, используемых в нефтегазовом деле, прежде всего геологическом и гидродинамическом моделировании. Например, программный комплекс «tNavigator» от компании «Rock Flow Dynamics» использует

генетические алгоритмы для автоадаптации гидродинамических моделей – уменьшении невязки исторических и расчетных параметров. Стоит отметить, что в данном программном продукте наряду с генетическими алгоритмами могут быть использованы и другие алгоритмы машинного обучения, также и используемые генетические алгоритмы могут быть различными, их параметры могут изменяться в зависимости от преследуемых целей [57].

1.4 Отличия от традиционных алгоритмов

Генетические алгоритмы довольно сильно отличаются от традиционных алгоритмов поиска и оптимизации (таких как, например, градиентные методы). Приведем основные отличительные особенности.

Представление параметров в виде генетических особенностей. Параметры особей представляются в виде хромосом (которые являются определенными кодированными значениями). Такое представление позволяет использовать эволюционные операции. При этом стоит понимать, что из-за подобного представления происходит некое отдаление от области решаемой проблемы – хромосомы должны быть точно настроены, чтобы быть интерпретируемыми в рамках определенной задачи [19, 36].

Поведение, основанное на вероятностях. Традиционные алгоритмы часто являются детерминистическими, в то время как генетические алгоритмы основаны на вероятностном подходе. Эволюционные операции, проводимые над особями, определяются вероятностями. При этом стоит заметить, что генетические алгоритмы не являются чисто случайным поиском, наоборот, вероятностный подход служит для направления особей к увеличению приспособленности, то есть к оптимизации функции [19, 36].

Использование функции приспособленности. Генетические алгоритмы направлены на получение максимальных приспособленностей особей. В Генетических алгоритмах не используются дифференцирование или

интегрирование (как во множестве других традиционных алгоритмов), что позволяет использовать генетические алгоритмы в случаях, когда слишком сложно или невозможно провести дифференцирование или интегрирование [19, 36].

Генетические алгоритмы являются популяционными, имеют дело не просто с особями, но с их совокупностями. В каждый момент времени в памяти хранится целая популяция (как минимум одна) особей. Чаще всего традиционные алгоритмы имеют дело только с одним возможным решением, которое итеративно изменяется [19, 36].

1.5 Достоинства и ограничения генетических алгоритмов

Генетические алгоритмы, как и любые другие подходы, имеют определенные сильные и слабые стороны. Рассмотрим достоинства данных алгоритмов.

Решение задач со сложным математическим представлением. Как уже упоминалось ранее, генетическим алгоритмам для работы нужны только функции приспособленности для каждой особи, нет необходимости в производных или первообразных функций. Соответственно, как говорилось ранее, генетические алгоритмы могут быть использованы для функций со сложным математическим представлением. Также генетические алгоритмы могут быть использованы для функций с большим количеством переменных и различными их типами (например, дискретные и непрерывные параметры) [25, 29, 59].

Решение задач с затрудненным математическим представлением. Некоторые решаемые задачи не могут быть представлены в математическом виде (как минимум напрямую) – например, когда функция приспособленности представляет собой мнение людей. Также генетические алгоритмы могут быть использованы даже тогда, когда непосредственно значение функции не может

быть получено, но может быть выявлено относительное положение приспособленностей особей [25, 29, 59].

Возможность использования параллельных вычислений. Чаще всего вычисления в генетических алгоритмах могут быть легко параллелизованы – приспособленность для каждой особи может быть посчитана отдельно, также и эволюционные операторы могут быть применены параллельно для особей и пар особей [25, 29, 59].

Возможность находить глобальный максимум (минимум). Часто функции имеют несколько локальных максимумов (минимумов) – см. Рисунок 1.1 Большинство традиционных алгоритмов не может «выйти» из найденного локального максимума (минимума). Генетические алгоритмы используют популяции особей и операторы, основанные на вероятностях, из-за чего локальные максимумы (минимумы) могут быть преодолены [25, 29, 59].

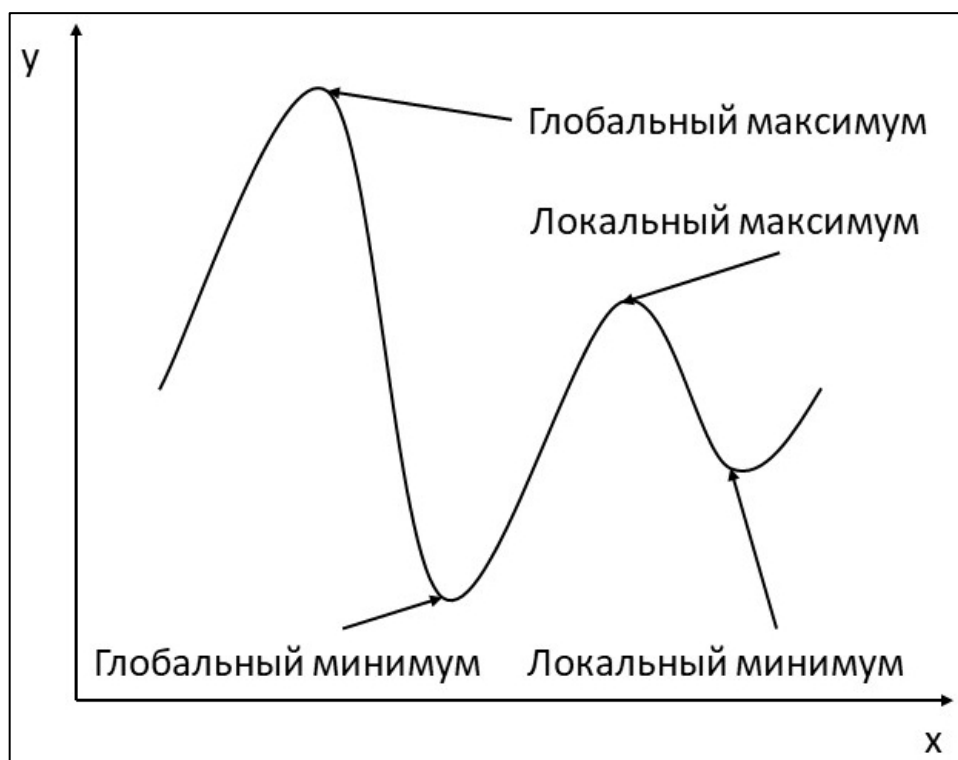


Рисунок 1.1 – Разница между глобальными и локальными экстремумами функции

Устойчивость к информационному шуму. Иногда функция приспособленности может давать разные выходные значения для одних и тех же входных значений (например, из-за погрешности измерений). При неизменном общем тренде генетические алгоритмы способны решать задачи с подобным шумом [25, 29, 59].

Непрерывное обучение. Генетические алгоритмы способны подстраиваться под изменяющиеся входные и выходные условия [25, 29, 59].

Также рассмотрим недостатки и ограничения генетических алгоритмов.

Генетические алгоритмы может нуждаться в большом количестве вычислений. Так как популяции могут быть достаточно большими, а функции приспособленности достаточно сложными (и должны быть посчитаны для каждой особи), работа генетических алгоритмов может быть затратной по количеству вычислений и времени до достижения результата [25, 29, 59].

Гиперпараметры алгоритма должны быть настроены. Гиперпараметры, такие как количество особей в популяции, вероятности мутации и так далее, нуждаются в настройке под конкретную задачу и сложно масштабируются [25, 29, 59].

Данные для решений должны быть представлены в особом виде. Для каждой решаемой задачи вид представления входных параметров, функции приспособленности, а также эволюционные операторы будут разными, должны быть подобраны для каждой конкретной задачи. Стоит отметить, что на нынешнем уровне развития генетических алгоритмов практически всегда возможно найти похожую задачу, уже решенную с помощью генетических алгоритмов [25, 29, 59].

Отсутствует гарантированно верное решение, есть риск нахождения только локального экстремума. При неправильной настройке гиперпараметров или при неудачном совпадении генетические алгоритмы могут быть подвержены преждевременной сходимости – поиску локальных, а не глобальных максимумов. Поэтому невозможно быть уверенным в том, что

полученный результат является самым лучшим решением. Стоит отметить, что данные проблемы присущи и всем другим подобным алгоритмам [25, 29, 59].

1.6 Общая схема работы генетических алгоритмов

Все генетические алгоритмы имеют уникальные схемы, тем не менее, все они опираются на одну исходную общую схему [23]:

- 1) создается исходная популяция из определенного количества особей;
- 2) для каждой особи рассчитывается и присваивается ей значение целевой функции;
- 3) отбираются особи, которые будут участвовать в размножении;
- 4) отобранные особи скрещиваются, создаются потомки;
- 5) потомки могут быть подвержены мутации;
- 6) полученные особи добавляются в старую популяцию или образуют совершенно новую;
- 7) если новые особи были добавлены в старую популяцию, популяция редуцирует – количество особей в популяции сокращается до исходного количества – данный шаг выполняется не всегда;
- 8) происходит проверка критерия останова алгоритма. Если он не выполняется, цикл начинается снова с оценки целевой функции для каждой особи. Если критерий выполняется, находится лучшая особь, которая считается решением задачи.

Общая схема работы генетических алгоритмов представлена на Рисунке 1.2.

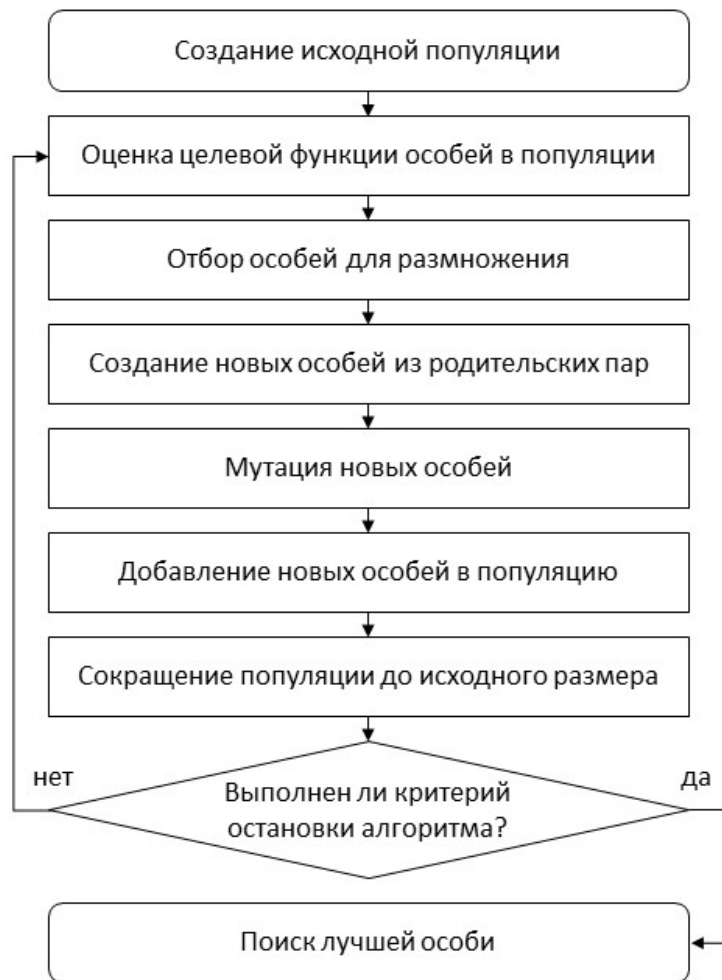


Рисунок 1.2 – Общая схема работы генетических алгоритмов

1.7 Создание исходной популяции

Исходная популяция представляет собой группу особей, параметры которых генерируются случайным образом. При этом типы параметров могут быть разными (числа из непрерывного распределения, дискретные целые числа, десятичные дроби, бинарные числа и так далее). На данном этапе закладывается такой гиперпараметр, как размер популяции [46].

Стоит отметить, что для получения качественного решения исходная популяция должна содержать достаточное количество особей, причем распределенных по области поиска как можно более равномерно. При слишком малой выборке или при создании исходной популяции из слишком сильно похожих особей генетический алгоритм может столкнуться с проблемой

преждевременной сходимости (хотя оператор мутации может помочь справиться с этой проблемой). Именно поэтому исходная популяция должна проверяться перед каждым запуском генетического алгоритма, либо на исходную популяцию должны накладываться ограничения по «похожести» особей. Так как создание исходной популяции обычно является процессом, не требующим сложных и долгих расчетов, можно повторять его до получения приемлемой выборки [46].

В некоторых случаях исходная популяция может создаваться в две стадии: на первом этапе создается исходная популяция, равномерно покрывающая всю область поиска, на втором этапе выделяются подобласти на основании высоких значений функции приспособленности, именно в этих областях происходит создание новых исходных популяций и запуск генетического алгоритма. Данный подход часто позволяет ускорить нахождение решения, но подходит не для всех задач (в некоторых случаях данный подход может дать высокие значения функции приспособленности только в локальных максимумах, соответственно, выбираемые подобласти будут исключать глобальный максимум, что может привести к преждевременной сходимости). Обычно такой подход работает хорошо для задач, в которых достаточно хорошо известно предполагаемое положение искомого решения [46].

1.8 Оценка целевой функции

Целевая функция представляет собой функцию нескольких переменных, подлежащую оптимизации, то есть максимизации или минимизации. Целевая функция может задаваться в явном или неявном виде [44].

Фактически, целевая функция может быть любым выражением или способом, позволяющим сравнивать особи друг с другом. Единственное

условие – каждая особь (в исходном и всех последующих поколениях) должна обладать присвоенным значением целевой функции [44].

В некоторых случаях расчет целевой функции может занимать значительное время. Значение целевой функции одной особи не зависит от значений целевой функции всех других особей. По этой причине расчет целевой функции может проводиться параллельно для каждой особи, что может позволить значительно сократить время расчета целевой функции для поколения и, соответственно, ускорить работу алгоритма и получение результата [44].

1.9 Отбор особей для размножения

Функция отбора представляет собой выбор из популяции особей, которые затем будут участвовать в размножении. Снова заметим, что отбор имеет вероятностный характер. По этой причине каждая особь в поколении имеет возможность быть отобранной, но вероятность отбора зависит от приспособленности особи – чем выше приспособленность, тем выше вероятность отбора. Стоит заметить, что шанс на отбор особей с относительно низким значением целевой функции является важным элементом генетических алгоритмов, так как такие особи могут нести в себе «полезные» гены, которые могут быть использованы для решения задачи, иначе эти гены могут быть потеряны [29].

Существуют различные методы отбора (в зависимости от решаемой задачи), рассмотрим основные из них.

Отбор методом рулетки. Является самым простым и чаще всего используемым методом отбора – вероятность отбора особи напрямую пропорциональна целевой функции. Отбор с использованием данного метода походит на применение рулетки, используемой в казино. При этом угол сектора рулетки пропорционален доле значения целевой функции в сумме значений

целевой функции для всех особей в поколении. При этом условно фиксируется точка выбора, а рулетка вращается и останавливается в случайный момент. Такие вращения рулетки продолжаются, пока не будет отобрано необходимое количество особей для размножения. Очевидно, что в таком случае одна и та же особь может быть отобрана несколько раз. Заметим, что при явно более высоком значении целевой функции у одной из особей она может быть выбрана большое количество раз, что может приводить к преждевременной сходимости. Использование данного метода затруднено в случае возможности получения отрицательного значения целевой функции [29].

Случайная равномерная выборка. Данный метод походит на метод рулетки. Угол сектора рулетки также пропорционален доле значения целевой функции в сумме значений целевой функции для всех особей в поколении, но существуют несколько точек отбора, равномерно рассредоточенных по всей рулетке. Рулетка может вращаться несколько раз или количество точек отбора может соответствовать количеству особей, которое необходимо отобрать. При использовании данного метода совокупность отобранных особей является более неоднородной. Шанс на отбор у особей с относительно низким значением целевой функции значительно повышается, что может помочь решить проблему преждевременной сходимости. Использование данного метода затруднено в случае возможности получения отрицательного значения целевой функции [29].

Метод отбора ранжированием. Метод очень похож на метод отбора рулеткой, но в данном случае угол сектора рулетки пропорционален не самому значению целевой функции у особи, а пропорционален рангу. То есть, значения целевой функции необходимы только для того, чтобы отсортировать все особи. Каждой особи присваивается ранг (от количества особей до единицы). Соответственно, угол сектора рулетки также пропорционален доле ранга в сумме рангов для всех особей в поколении. В таком случае отбор может происходить по схеме метода рулетки или по схеме случайной универсальной

выборки. Заметим, что в данном случае отрицательные значения целевой функции не являются проблемой [29].

Нормирование целевой функции. Для снижения перевеса особей с высокими значениями целевой функции может быть использовано нормирование целевой функции и, соответственно, нормирование угла сектора на рулетке. Для проведения нормирования необходимо выбрать пределы нормирования, тогда худшая особь будет соответствовать нижнему пределу нормирования, а лучшая – верхнему. Это обычно позволяет увеличить шансы на отбор особей с низким значением целевой функции. В таком случае отбор может происходить по схеме метода рулетки или по схеме случайной универсальной выборки. Заметим, что пределы масштабирования будут отличаться от задачи к задаче [29].

Отбор по турнирной системе. В данном методе предусматривается проведение сравнения нескольких особей. Определенное количество особей случайным образом выбирается из популяции, затем лучшая из них отбирается для размножения. Чем больше количество отбираемых особей, тем выше вероятность на выбор лучших особей. Данный метод интересен тем, что при его использовании нет необходимости в самом значении функции приспособленности, единственное необходимое условие – это возможность сравнить особи между собой [29].

1.10 Скрещивание

Скрещивание представляет собой процесс, в ходе которого определенное количество особей (обычно две) создают особь (или особи) со смешанным генотипом особей-родителей. В некоторых случаях применяют вероятность скрещивания – в некоторых (редких) случаях скрещивание не происходит, а предки переходят в новое поколение неизменными [49].

Существуют различные методы скрещивания (в зависимости от решаемой задачи), рассмотрим основные из них.

Одноточечное скрещивание. Данный метод применяется (прежде всего) для двоичных чисел. В данном методе случайным образом выбирается какой-либо промежуток между двумя знаками числа, левая и правая части меняются местами. В данном случае появляются два потомка, которые обладают параметрами двух особей-родителей [52].

Скрещивание с несколькими точками. В данном методе случайным образом выбираются несколько промежутков между знаками числа, а затем определенным образом обмениваются между родителями. В классическом варианте используются два промежутка, гены между которыми подлежат обмену. При большем количестве точек происходит обмен знаками между ними. Данный метод также применим (прежде всего) для бинарных чисел [52].

Равномерное скрещивание. В данном методе решение об обмене генами принимается для каждого гена по отдельности. Каждый ген имеет вероятность (обычно 50%) на то, чтобы остаться неизменным и вероятность на то, чтобы обменяться. Вероятность может быть также больше или меньше стандартных 50%. Потомок может формироваться как выбором генов родителей (тогда формируется один потомок) или обменом генов родителей (тогда формируется два потомка). Данный метод подходит для любых типов данных. Данный метод позволяет добиваться большего разнообразия в результирующем поколении, так как обмен происходит похромосомно [52].

Скрещивание упорядоченных списков. В некоторых случаях необходимо, чтобы значение генов присутствовало в каждой особи ровно один раз, для таких случаев используется подобный метод. В нем у каждого родителя случайным образом выбирается определенный промежуток, который переходит потомку. Далее оставшиеся промежутки последовательно заполняются оставшимися значениями из генов родителя [52].

Скрещивание смешиванием. Метод используется, прежде всего, для вещественных чисел из непрерывного распределения. В данном случае

значение параметра потомка выбирается случайным образом из интервала между значениями параметра родительских особей. Иногда также используется расширение (сужение) интервала выбора. Расширение интервала способствует большей «случайности» выбора значения, при этом может позволить увеличить разнообразие генов потомков [52].

Скрещивание с симуляцией бинарных чисел. Метод используется, прежде всего, для вещественных чисел из непрерывного распределения. Симулируется одноточечное скрещивание. Значения параметра потомка меняется таким образом, что потомки представляют собой значения родителей, отдалившиеся (приблизившиеся) друг от друга (среднего значения параметра для родителей) на определенную случайным образом долю [52].

1.11 Мутация

Мутация является очень важным составляющим элементом генетических алгоритмов, так как позволяет получать новые гены в процессе работы алгоритма, что препятствует преждевременной сходимости. Мутация производится над созданными особями с определенной (обычно низкой) вероятностью, что является гиперпараметром алгоритма. Слишком высокая вероятность мутации может ухудшить популяцию и даже превратить генетический алгоритм в случайный поиск решения. В некоторых случаях вероятность мутации возрастает от поколения к поколению, так как сходимость обычно наблюдается на более поздних поколениях [13].

Существуют различные методы мутации (в зависимости от решаемой задачи), рассмотрим основные из них.

Мутация сменой бита. В случае, когда мутация совершается над бинарными числами, она может быть произведена сменой значений одного или нескольких битов с единицы на ноль или наоборот [55].

Мутация перемешиванием. Для бинарных и целых чисел мутация может быть произведена перемешиванием параметров на определенных позициях. Заметим, что такой метод подходит для упорядоченных списков [55].

Мутация инверсией. В данном методе определенная случайным образом часть параметров приобретает обратный порядок. Заметим, что такой метод подходит для упорядоченных списков [55].

Мутация смешиванием. В данном случае определенная случайным образом часть параметров приобретает случайный порядок. Заметим, что такой метод подходит для упорядоченных списков [55].

Мутация вещественных чисел. Для вещественных чисел может быть использована мутация заменой каких-либо параметров на случайные. Так как данный подход несет в себе опасность того, что получившаяся особь может не быть похожей на предшествующую, мутация может быть произведена с помощью распределения (например, нормального) от значения параметра предка с определенным стандартным отклонением. В таком случае вероятность выбора значения тем выше, чем ближе оно к значению параметра предка [55].

1.12 Расширение и редукция популяции

Существует четыре принципиальных подхода к расширению и редукции, появляющихся в процессе размножения популяции [53].

Полученные в ходе размножения особи добавляются в предшествующую популяцию. Затем работает механизм редукции – особи с самыми низкими значениями функции приспособленности удаляются из популяции до того момента, пока количество особей в популяции не станет равным изначально заданному [53].

Все полученные в результате скрещивания особи попадают в популяцию, причем редукция не работает. Число особей в популяции растет с каждым поколением. Данный метод удобен тем, что снижается количество

расчетов функции приспособленности, так, как ее значение известно, для всех предшествующих особей [53].

Сохраняется только полученное новое поколение. Поколение предков удаляется. Преимущество данного метода заключается в экономии памяти вычислительного устройства [53].

Последний метод основан на элитизме – определенное количество или определенная доля лучших особей из прошлого поколения всегда переходит в новое поколение. Это позволяет гарантированно не терять достижения прошлых поколений [53].

Стоит также заметить, что возможно хранение в памяти значений функции приспособленности для определенных значений параметров. Это позволяет не рассчитывать функцию приспособленности снова в том случае, если подобная особь уже появлялась [31].

1.13 Проверка критериев останова и завершение работы

Завершение работы алгоритма наступает тогда, когда выполняется критерий его останова. Существуют разные критерии останова, рассмотрим основные из них.

Достижение максимального количества расчетов функции приспособленности или максимального количества поколений. Данный метод используется для сокращения времени работы алгоритма или сокращения потребляемых вычислительных ресурсов, но может относительно часто приводить к тому, что алгоритм «не успевает» найти глобальное решение исследуемой задачи [27].

Отсутствие значительных изменений максимального значения функции приспособленности в поколении на протяжении нескольких поколений. Для использования данного метода необходимо хранить максимальные значения функции приспособленности в прошедших поколениях. При данном подходе

вероятность получения глобального решения максимальна, при этом затраты времени и вычислительных мощностей могут быть относительно велики [27].

Истечение предопределенного количества времени. В некоторых случаях (когда необходимо оптимизировать что-либо как можно быстрее) ограничение по времени является более важным критерием, чем нахождение глобального решения [27].

Достижение определенного количества занятой памяти устройства. При хранении информации о работе алгоритма память устройства может быть переполнена [27].

Лучшая особь занимает определенную долю из всех особей в популяции. При данном подходе считается, что из-за потери разнообразия особей более хорошее решение найти не представляется возможным [27].

Особь, имеющая при завершении работы алгоритма самое высокое значение функции приспособленности принимается решением изучаемой задачи [27].

1.14 Выводы по разделу

В данном разделе были представлены основные принципы генетических алгоритмов, используемых для оптимизации. Были обсуждены базовые понятия, используемые в генетических алгоритмах, история их разработки и применения, основные направления применения и отличия от традиционных алгоритмов оптимизации, преимущества и недостатки, общая схема работы и основные шаги (включая методы) генетических алгоритмов.

Стоит отметить, что данный раздел не ставит цель покрыть всю теоретическую базу, используемую в генетических алгоритмах, но направлен на понимание структуры диссертации.

2 Применение генетического алгоритма для оптимизации системы разработки синтетической модели месторождения нефти

Для оптимизации системы разработки синтетической модели нефтяной залежи была написана программа на языке программирования «Python 3», на основе генетического алгоритма. Полный код программы представлен в Приложении Б. Код программы тщательно закомментирован для упрощения понимания.

2.1 Описание системы

Генетический алгоритм является лишь частью системы, служащей для оптимизации системы разработки месторождения (см. Рисунок 2.1). Сначала генетический алгоритм создает исходную популяцию систем разработок определенного объема. Системы разработки представляют собой совокупность параметров расположения и конструкции скважин. Для каждой системы разработки формируется профиль добычи на основании синтетической модели месторождения нефти. Профиль добычи и экономические данные переходят в финансово-экономическую модель, которая считает чистую приведенную стоимость. Чистая приведенная стоимость принимается целевой функцией. То есть, считаем, что система разработки тем лучше, чем выше чистая приведенная стоимость (ЧПС), получаемый с ее использованием. Соответственно, генетический алгоритм стремится максимизировать чистую приведенную стоимость. После присвоения всем особями популяции значения целевой функции работает генетический алгоритм, и цикл продолжается, пока не выполняется критерий останова генетического алгоритма.



Рисунок 2.1 – Схема оптимизационной системы

2.2 Параметры месторождения и систем разработки

Система разработки месторождений нефти и газа подразумевает под собой организацию движения нефти и газа в пласте для их последующего извлечения. В частности, в данной работе под системой разработки месторождения нефти понимается размещение скважин на эксплуатационном объекте, их конструкцию, порядок ввода в работу и применяемые на них методы стимуляции [15].

Во многом именно система разработки, ее продуманность и эффективность определяет экономический успех разработки месторождения (с технической точки зрения) [38].

В качестве объекта исследования использована синтетическая математическая модель нефтяного месторождения, с априори однородными петрофизическими параметрами, отражающими реальные данные. Условные параметры месторождения представлены в Таблице 2.1.

Таблица 2.1 – Условные параметры синтетического месторождения.

Параметр	Обозначение	Единицы измерения	Значение
Площадь	S	м ²	10 ⁸
Мощность коллектора	h	м	20
Пористость	φ	д.ед.	0,2
Нефтенасыщенность	S _o	д.ед.	0,8
Объемный коэффициент нефти	B _o	м ³ /м ³	1,16
Проницаемость	k	м ²	10 ⁻¹³
Вязкость нефти	μ	Па·с	0,003

Зная эти параметры, можно рассчитать условные начальные геологические запасы (СТОИР) [51]:

$$СТОИР = \frac{Sh\varphi S_o}{B_o} = \frac{10^8 \times 20 \times 0,2 \times 0,8}{1,16} = 275862069 \text{ м}^3. \quad (2.1)$$

Также рассчитаем условный ориентировочный дебит скважины (Q), приняв депрессию (ΔP) равной 10⁶ Па и натуральный логарифм отношения радиусов контура питания и скважины (ln(r_к/r_с)) равным 7, используя формулу Дюпюи [54]:

$$Q = \frac{2\pi kh\Delta P}{\mu \ln \frac{r_k}{r_c}} = \frac{2 \times 3,14 \times 10^{-13} \times 20 \times 10^6}{0,003 \times 7} = 0,000598 \frac{\text{м}^3}{\text{с}} = 51,7 \frac{\text{м}^3}{\text{сут}}. \quad (2.2)$$

Следует заметить, что данные параметры являются ориентировочными и показаны для того, чтобы будущие расчеты были близки к реально возможным параметрам. При этом для упрощения модели было решено использовать добычу нефти на истощение (без нагнетания). Для описания системы разработки месторождения использовано шесть параметров, которые могут изменяться и влиять на ЧПС проекта:

- 1) наличие и конфигурация горизонтального ствола (ГС). Изменяемыми параметрами могут быть длина и направление горизонтального ствола;
- 2) расстояние между скважинами, которое также влияет на количество скважин;
- 3) радиус скважин;
- 4) наличие и конфигурация трещин гидравлического разрыва пласта (ГРП), изменяемый параметр – длина трещины;
- 5) наличие и конфигурация кислотной обработки призабойной зоны пласта; изменяемый параметр – радиус обработки.

Было решено моделировать работу скважины (то есть добычу нефти и ее динамику) экспоненциальной зависимостью, используя подход Дейка [34]. В таком случае зависимость дебита нефти скважины от времени (t) определяется начальным дебитом (Q_0) и скоростью падения дебита (v):

$$Q = Q_0 e^{tv}. \quad (2.3)$$

Соответственно, зная начальные дебиты и скорость падения дебита (далее – константы дебита), можно определить дебит нефти в любой момент времени. Зная дебиты всех скважин, можно определить общий дебит по месторождению. Таким образом, формируется профиль добычи.

Параметры системы разработки, о которых говорилось ранее, будут определенным образом изменять зависимость дебита от времени. Заметим, что в работе приняты одинаковые параметры для всех скважин.

Для того, чтобы была возможность проверить работоспособность генетического алгоритма, было решено применить дискретизацию параметров системы разработки. Минимальные и максимальные значения параметров системы разработки и шаги их дискретизации представлены в Таблице 2.2.

Таблица 2.2 – Возможные значения параметров системы разработки

Параметр	Минимальное значение	Максимальное значение	Шаг дискретизации	Количество вариантов
Расстояние между скважинами, м	500	1000	50	11
Радиус скважины, см	10	30	5	5
Длина ГС, м	0	1000	100	11
Направление ГС, град	0	180	10	19
Длина трещины ГРП, м	0	100	10	11
Радиус обработки кислотой, дм	0	50	10	6

Рассмотрим то, как различные параметры системы разработки будут влиять на константы дебита. При этом базовыми значениями начального дебита и скорости падения добычи принимаем $10000 \text{ м}^3/\text{мес.}$ (без учета расстояние между скважинами и радиуса скважин) и $0,01$, соответственно. Различные параметры системы разработки будут модифицировать эти константы путем добавления определенных коэффициентов.

2.2.1 Расстояние между скважинами и радиус скважин

Сразу стоит отметить, что расстояние между скважинами (l) будет определять количество скважин на месторождении (n). Конечное количество скважин на месторождении можно найти по формуле [32]:

$$n = \left(\frac{\sqrt{S}}{l} \right)^2. \quad (2.4)$$

Знак дроби в данном случае означает целочисленное деление.

Также примем время бурения одной скважины равным одному месяцу, одновременно бурится только одна скважина [30].

Примем радиус контура питания скважины равным половине расстояния между скважинами. Из формулы Дюпюи (2.2) дебит обратно пропорционален

натуральному логарифму отношения радиусов контура питания и скважины. Примем данный коэффициент для модификации начального дебита. Соответственно, начальный дебит нефти тем больше, чем меньше расстояния между скважинами и чем больше радиус скважин.

При большом расстоянии между скважинами каждой скважине соответствует большая площадь дренирования, поэтому принимаем, что коэффициент действия расстояния между скважинами на скорость падения добычи линейно меняется от 1 до 2,5 [56].

Большой радиус скважины будет незначительно повышать скорость падения дебита за счет того, что отбор нефти идет быстрее, поэтому принимаем, что коэффициент действия радиуса скважины на скорость падения добычи линейно меняется от 1 до 1,1 [56].

2.2.2 Длина и направление горизонтального ствола

Горизонтальный ствол значительно повышает площадь дренирования скважины и, соответственно, повышает дебит, при этом скорость падения добычи нефти также повышается, поэтому принимаем, что коэффициент действия длины горизонтального ствола на начальный дебит линейно меняется от 1 до 1,2, а коэффициент действия длины горизонтального ствола на скорость падения добычи также линейно меняется от 1 до 1,2 [28].

Предположим, что архитектура резервуара такова, что в некотором направлении проницаемость породы выше, поэтому в этом направлении дебит выше, поэтому принимаем, что коэффициент действия направления горизонтального ствола на начальный дебит меняется от 1 до 1,05 по синусоидальному закону. Скорость падения добычи при этом не меняется [28].

2.2.3 Гидравлический разрыв пласта

Трещина гидравлического разрыва пласта способствует увеличению дебита, при этом и скорость падения добычи растет, поэтому принимаем, что коэффициент действия длины трещины ГРП на начальный дебит линейно меняется от 1 до 1,1, а коэффициент действия длины трещины ГРП на скорость падения добычи также линейно меняется от 1 до 1,1 [41].

2.2.4 Кислотная обработка призабойной зоны пласта

Кислотная обработка призабойной зоны пласта способствует относительно небольшому росту дебита и небольшому увеличению скорости падения добычи, поэтому принимаем, что коэффициент действия радиуса кислотной обработки на начальный дебит линейно меняется от 1 до 1,01, а коэффициент действия длины радиуса кислотной обработки на скорость падения добычи линейно меняется от 1 до 1,05 [35].

2.2.5 Профиль добычи нефти

Зная параметры системы разработки, можно построить график зависимости дебита одной скважины от времени. На Рисунке 2.2 представлен такой график для системы разработки, в которой горизонтальный ствол имеет длину 600 метров и направление 60 градусов, расстояние между скважинами 600 метров, радиус скважины 20 сантиметров, длину трещины ГРП 60 метров и радиус обработки кислотой 30 дециметров.

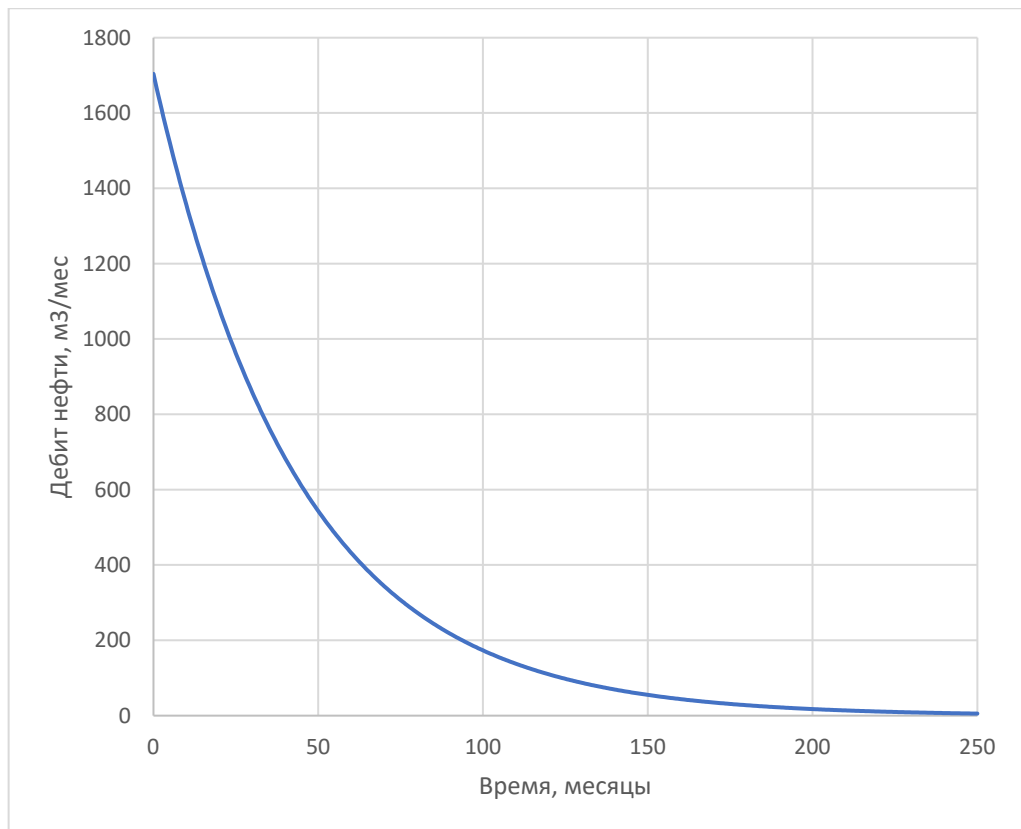


Рисунок 2.2 – Пример зависимости дебита скважины от времени

Учитывая изменение дебита каждой скважины по времени и время бурения, можно построить профиль добычи нефти по всему месторождению для системы разработки с теми же самыми параметрами (см. Рисунок 2.3).

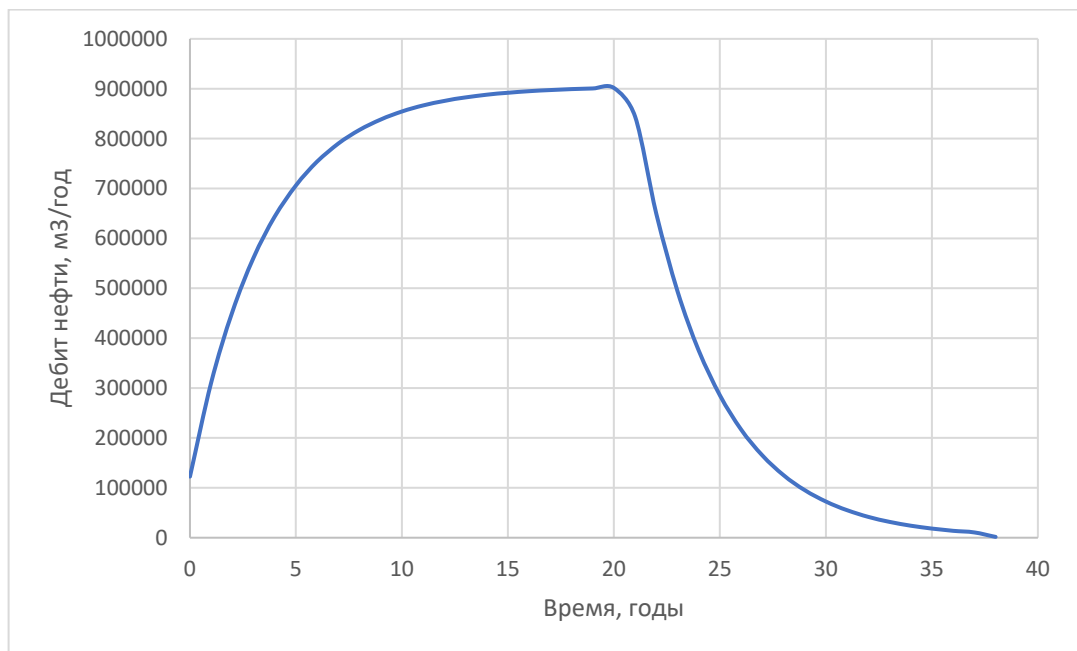


Рисунок 2.3 – Пример профиля добычи нефти

2.3 Описание генетического алгоритма

В данном подразделе кратко опишем написанная программа, код которого можно увидеть в Приложении Б.

Генетический алгоритм начинается с создания исходной популяции. На данном этапе фиксируется определенный размер популяции, что является гиперпараметром. Создание происходит таким образом, что в популяцию добавляются особи со случайным набором параметров, пока их количество не будет равно заданному гиперпараметру. После создания исходной популяции для каждой особи считается целевая функция.

Далее происходит отбор особей для размножения. Был выбран метод отбора ранжированием для того, чтобы вероятность отбора особей была более равномерной. Для того, чтобы произвести отбор, для каждой особи рассчитывался ранг, через ранг рассчитывалась вероятность отбора для каждой особи.

Далее следует создание родительских пар с учетом вероятности их отбора. Пары формируются до тех пор, пока их количество не будет совпадать с количеством особей в популяции. При этом контролируется, чтобы размножающиеся пары не совпадали друг с другом, то есть, чтобы каждая размножающаяся пара была уникальна. Также использован метод элитизма – пара из двух одинаковых лучших особей всегда участвует в размножении, что создает условия для того, чтобы лучшие особи прошлых поколений точно учитывались.

Сформированные пары участвуют в скрещивании, каждая пара дает одного потомка. Значения параметров каждого потомка определяются случайно из интервала значений этого параметра у родительских особей, плюс этот интервал расширяется за счет гиперпараметра альфа, который контролирует ширину добавляемого интервала. Альфа используется потому, что лучший параметр может находиться не между значениями параметра родителей, а в их окрестностях вне этого интервала.

Полученный потомок может мутировать с определенной вероятностью, которая является гиперпараметром. Если потомок мутирует, все его параметры меняются на случайно сгенерированные, что служит для добавления новых генов в процесс.

Так как каждая пара дает одного потомка и создается новая популяция без добавления особей из предыдущей, редукция популяции до исходных размеров не требуется.

Для каждой полученной особи рассчитывается и присваивается ей значение целевой функции.

Алгоритм повторяется вплоть до выполнения условия остановки. Условием остановки работы алгоритма было решено выбрать определенное количество поколений (это количество поколений выбирается заранее и является гиперпараметром), в течение которых не появляются особи с более высоким значением целевой функции.

Также было рассчитано количество всех возможных вариантов особей (N), для этого использовалась формула из комбинаторики:

$$N = \prod_{i=1}^n q_i, \quad (2.5)$$

где q – количество возможных вариантов параметра.

Так как параметры были дискретизированы, количество возможных вариантов особей, конечно, поэтому было решено провести полный перебор особей с расчетом целевой функции для того, чтобы найти особь с самым большим ее значением и для того, чтобы сравнивать результаты работы алгоритма с максимально возможным значением целевой функции.

Также было решено проверить работоспособность алгоритма путем сравнения его работы с методом случайных параметров. Метод заключался в том, что формировались случайные особи в количестве, равном количеству

особей в генетическом алгоритме, и сравнивались лучшие особи генетического алгоритма и этого метода. Таким образом, можно было понять, есть ли смысл в применении генетического алгоритма или можно провести такое же количество расчетов без алгоритма с тем же или лучшим результатом.

Также в алгоритм встроена функция многократного выполнения, при которой весь генетический алгоритм повторяется от начала до конца, необходимая для тестирования воспроизводимости результатов работы алгоритма (каждое повторение будем называть итерацией).

Для наглядного представления результатов алгоритм способен строить графики, представленные ниже.

На Рисунке 2.4 представлен пример графика изменения максимального и среднего значения целевой функции в поколении. Также указано максимально возможное значение целевой функции. Заметим, что среднее значение имеет тенденцию к росту, но может локально снижаться в то время, как максимальное значение не снижается. На примере генетический алгоритм достигает максимального значения целевой функции.

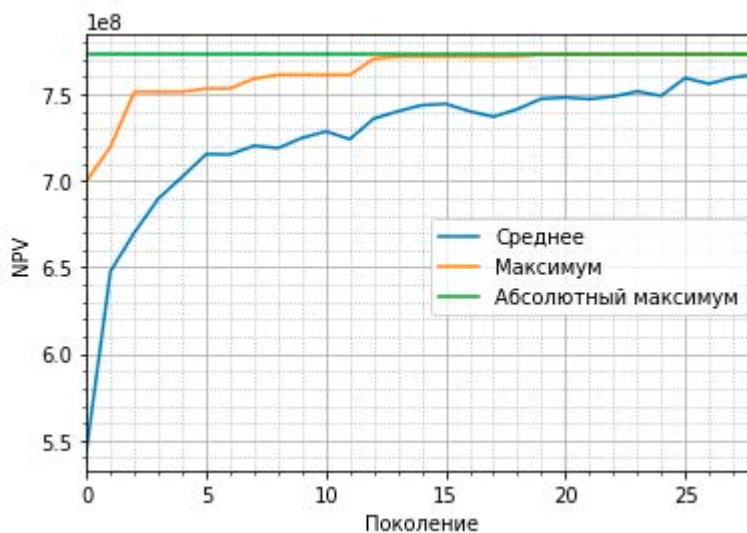


Рисунок 2.4 – Пример графика изменения значения целевой функции в поколении

На Рисунке 2.5 представлен пример столбчатой диаграммы количества поколений в итерации.

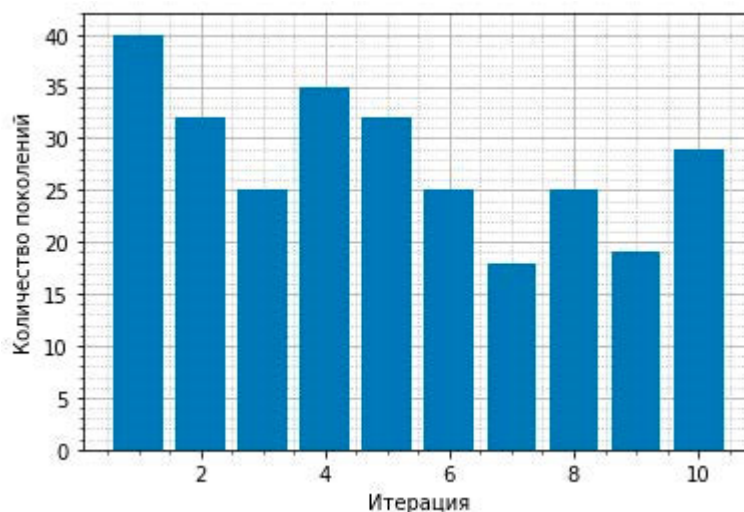


Рисунок 2.5 – Пример столбчатой диаграммы количества поколений в итерации

На Рисунке 2.6 представлен пример столбчатой диаграммы количества расчетов в итерации. Представлено общее количество расчетов и количество расчетов до поколения, в котором появилась лучшая особь в этой итерации.

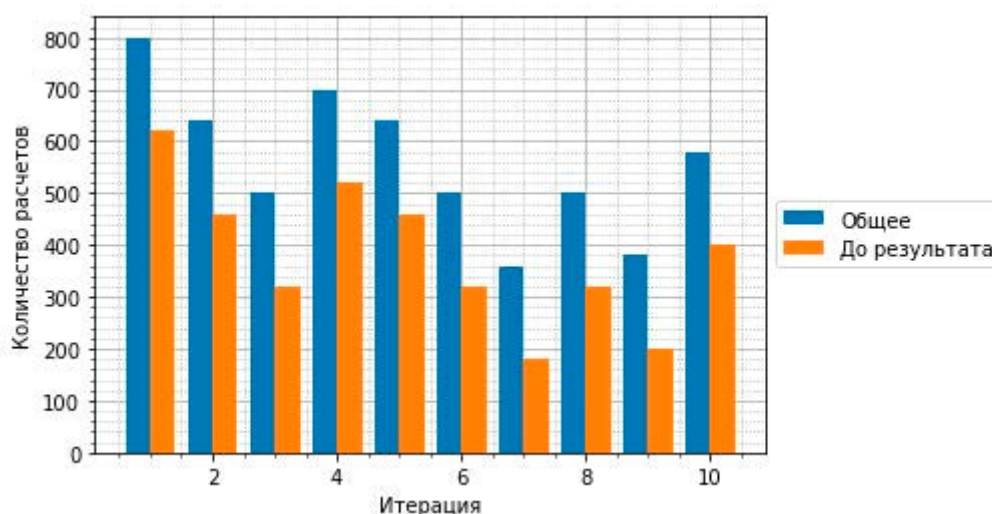


Рисунок 2.6 – Пример столбчатой диаграммы количества расчетов в итерации

На Рисунке 2.7 представлен пример столбчатой диаграммы доли возможных вариантов (частное от количества проведенных расчетов и количества возможных вариантов особей) в итерации. Заметим, что обычно доля не превышает 0,1% от возможных вариантов.

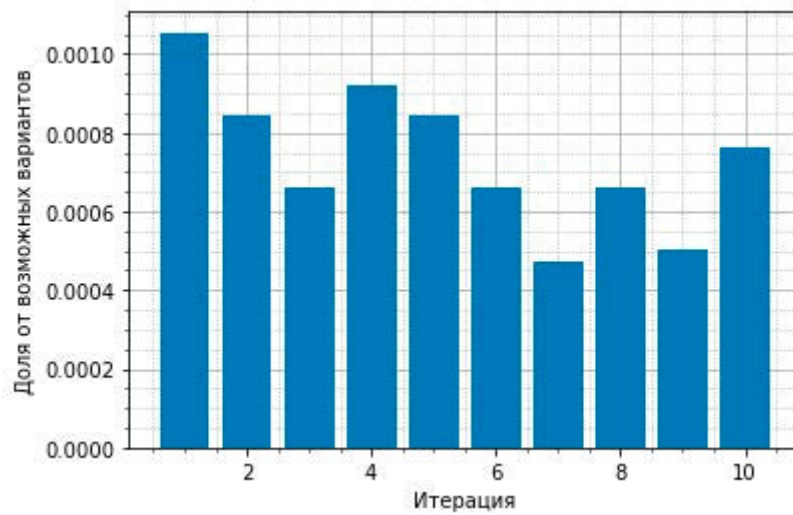


Рисунок 2.7 – Пример столбчатой диаграммы доли возможных вариантов в итерации

На Рисунке 2.8 представлен пример столбчатой диаграммы достигнутых максимумов целевой функции в итерации. Представлены значения генетического алгоритма и метода случайных параметров, а также максимально возможное значение целевой функции. Заметим, что результаты генетического алгоритма лучше результатов метода случайных параметров (в этом случае) и близки к абсолютному максимуму (в этом случае).

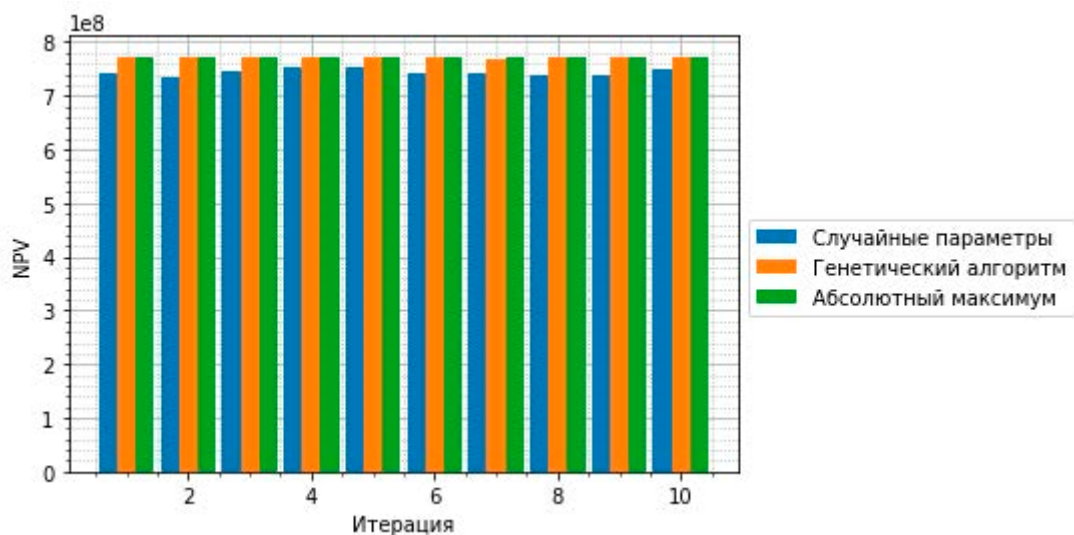


Рисунок 2.8 – Пример столбчатой диаграммы максимального значения целевой функции в итерации

Также выводятся основные параметры расчетов:

- 1) среднее количество поколений в итерации;
- 2) среднее общее количество расчетов в итерации;
- 3) среднее количество расчетов в итерации до результата;
- 4) средняя доля от всех возможных вариантов;
- 5) медианный максимум при методе случайных параметров;
- 6) медианный максимум при генетическом алгоритме;
- 7) доля достижения генетическим алгоритмом максимально возможного значения целевой функции;
- 8) доля случаев, в которых генетический алгоритм справился лучше метода случайных параметров.

Далее рассмотрим результаты, полученные в ходе проведения расчетов.

2.4 Тестовая функция чистой приведенной стоимости

В первую очередь было решено использовать простую тестовую синтетическую функцию чистой приведенной стоимости для тестирования работоспособности алгоритма. Функция представляла собой функцию двух переменных (условных параметров x и y), целевая функция чистой приведенной стоимости обозначена как NPV:

$$NPV = 3x + 6y - x^2 - xy - y^2. \quad (2.6)$$

Функция имеет максимальное значение равное девяти при x равном нулю и y равном трем. График функции представлен на Рисунке 2.9.

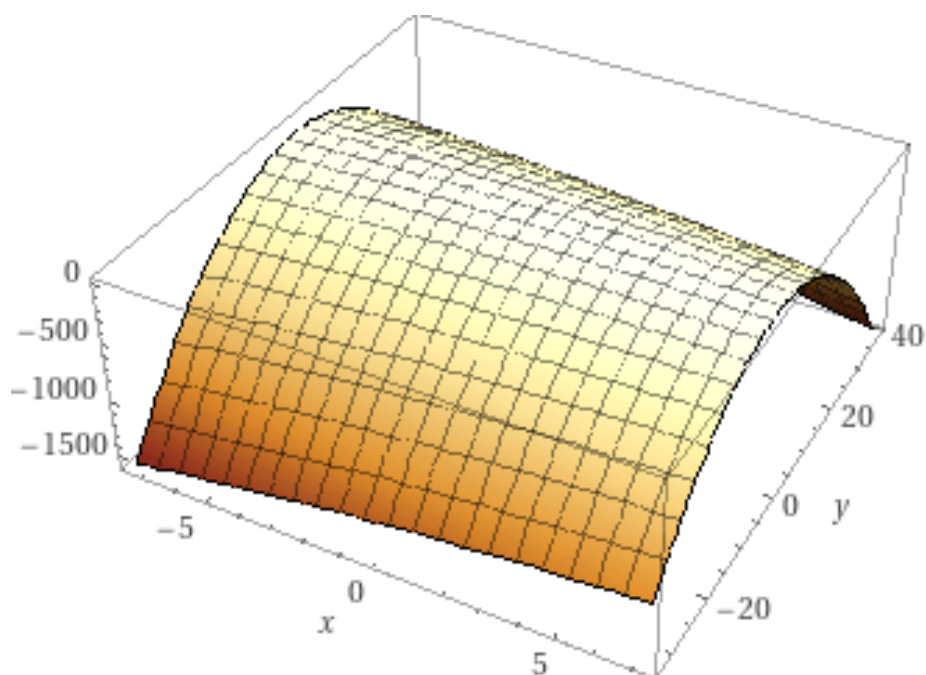


Рисунок 2.9 – График тестовой функции чистой приведенной стоимости

Значения параметров x и y были дискретизированы. Значения параметра x могли меняться от минус 10 до 10 с шагом дискретизации 0,1, значения параметра y могли меняться от минус 100 до 100 с шагом дискретизации один. Всего 402201 возможная комбинация параметров.

Генетический алгоритм был применен на данной функции. Был проведен анализ чувствительности результата работы алгоритма к его гиперпараметрам для определения работоспособности алгоритма и определения оптимальных параметров. Базовым случаем были выбраны значения гиперпараметров:

- 1) размер популяции равный 20 особям;
- 2) вероятность мутации равная 10%;
- 3) условие остановки равное 10 поколениям;
- 4) параметр альфа равный 0,5.

Ниже приведены результаты расчетов.

На Рисунке 2.10 представлена зависимость среднего количества поколений в итерации от гиперпараметров для тестовой функции.

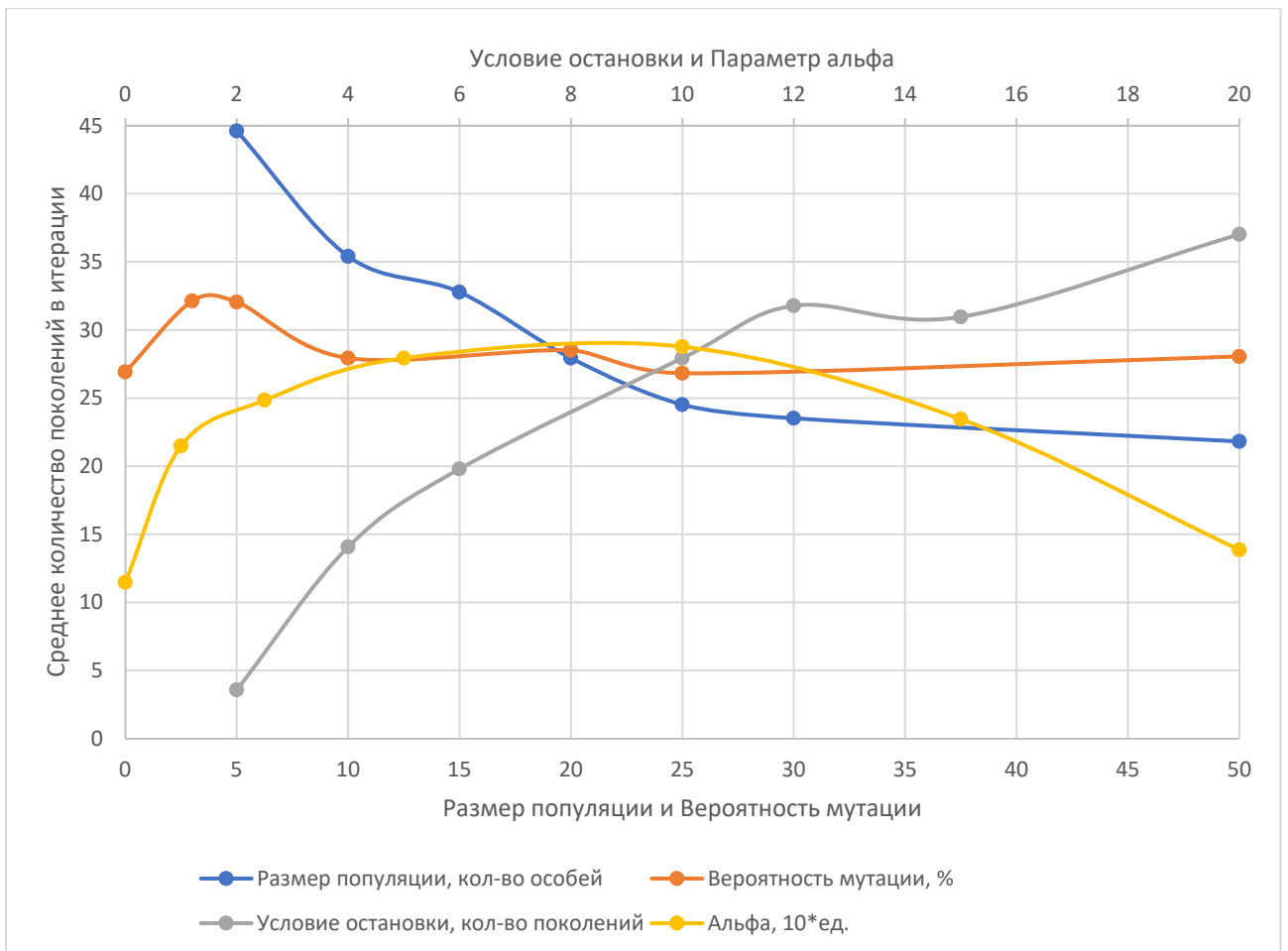


Рисунок 2.10 – Зависимость среднего количества поколений в итерации от гиперпараметров для тестовой функции

Количество поколений снижается при увеличении размера популяции и стабилизируется после определенного значения, так как при большем размере популяции достигается большее исходное разнообразие особей. Вероятность мутации практически не влияет на количество поколений. Количество поколений увеличивается с увеличением условия остановки, так как требуются дополнительные поколения для остановки. Параметр альфа заставляет количество поколений сначала расти, потом снижаться, так как в обоих случаях проявляется преждевременная сходимость.

На Рисунке 2.11 представлена зависимость среднего общего количества расчетов в итерации и количества расчетов до получения максимального результата от гиперпараметров для тестовой функции.

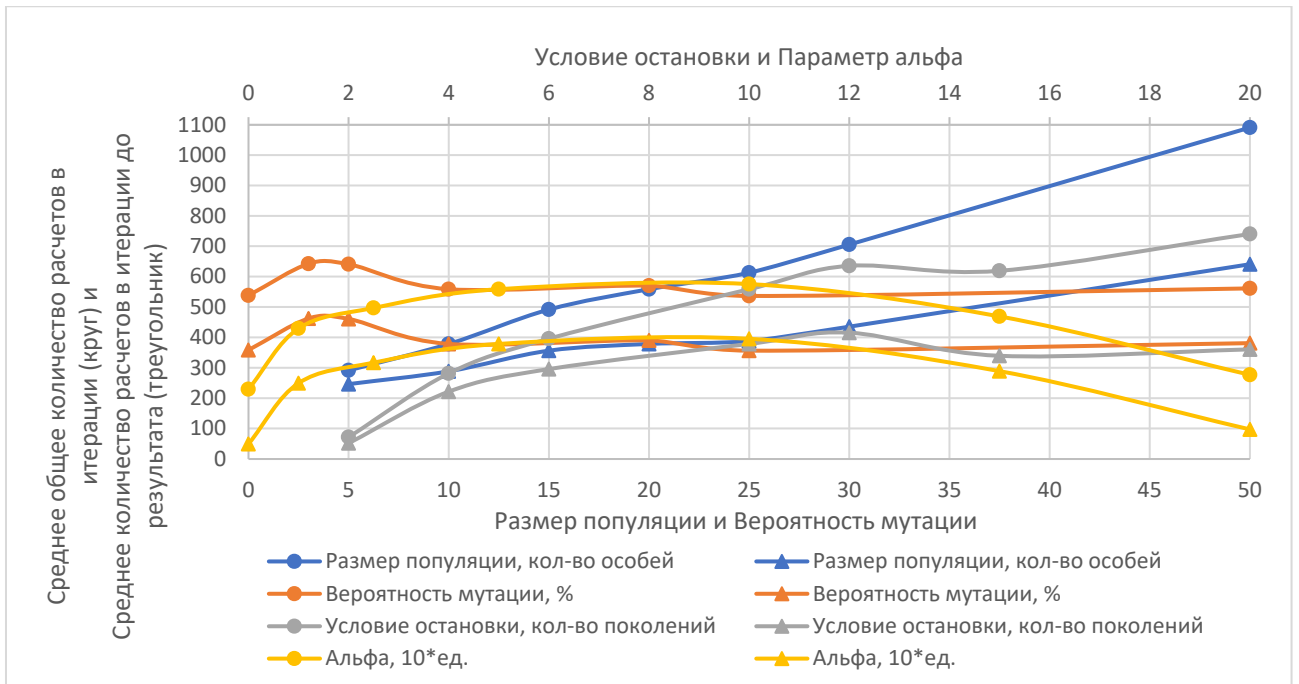


Рисунок 2.11 – Зависимость среднего общего количества расчетов в итерации и количества расчетов до получения максимального результата от гиперпараметров для тестовой функции

По большей части зависимости повторяют таковые для количества поколений. Но при увеличении размера популяции растет количество расчетов.

На Рисунке 2.12 представлена зависимость средней доли от возможных вариантов от гиперпараметров для тестовой функции.

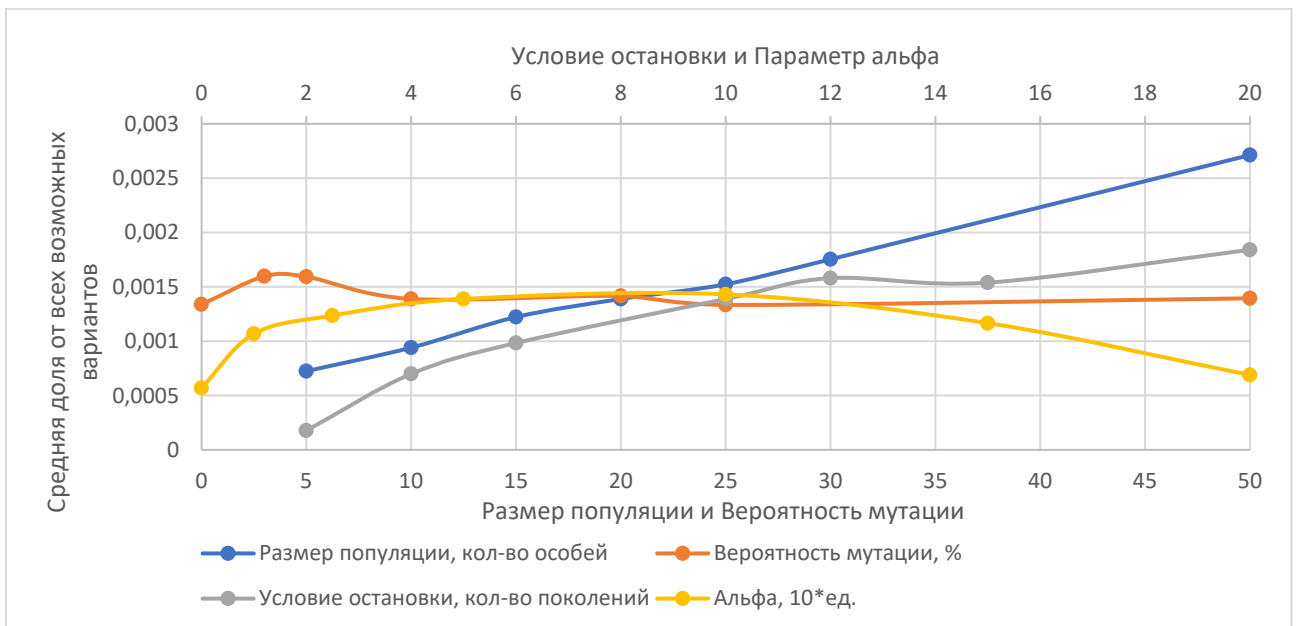


Рисунок 2.12 – Зависимость средней доли от возможных вариантов от гиперпараметров для тестовой функции

Зависимости повторяют таковые для количества расчетов.

На Рисунке 2.13 представлена зависимость медианных максимумов метода случайных чисел и генетического алгоритма от гиперпараметров для тестовой функции.

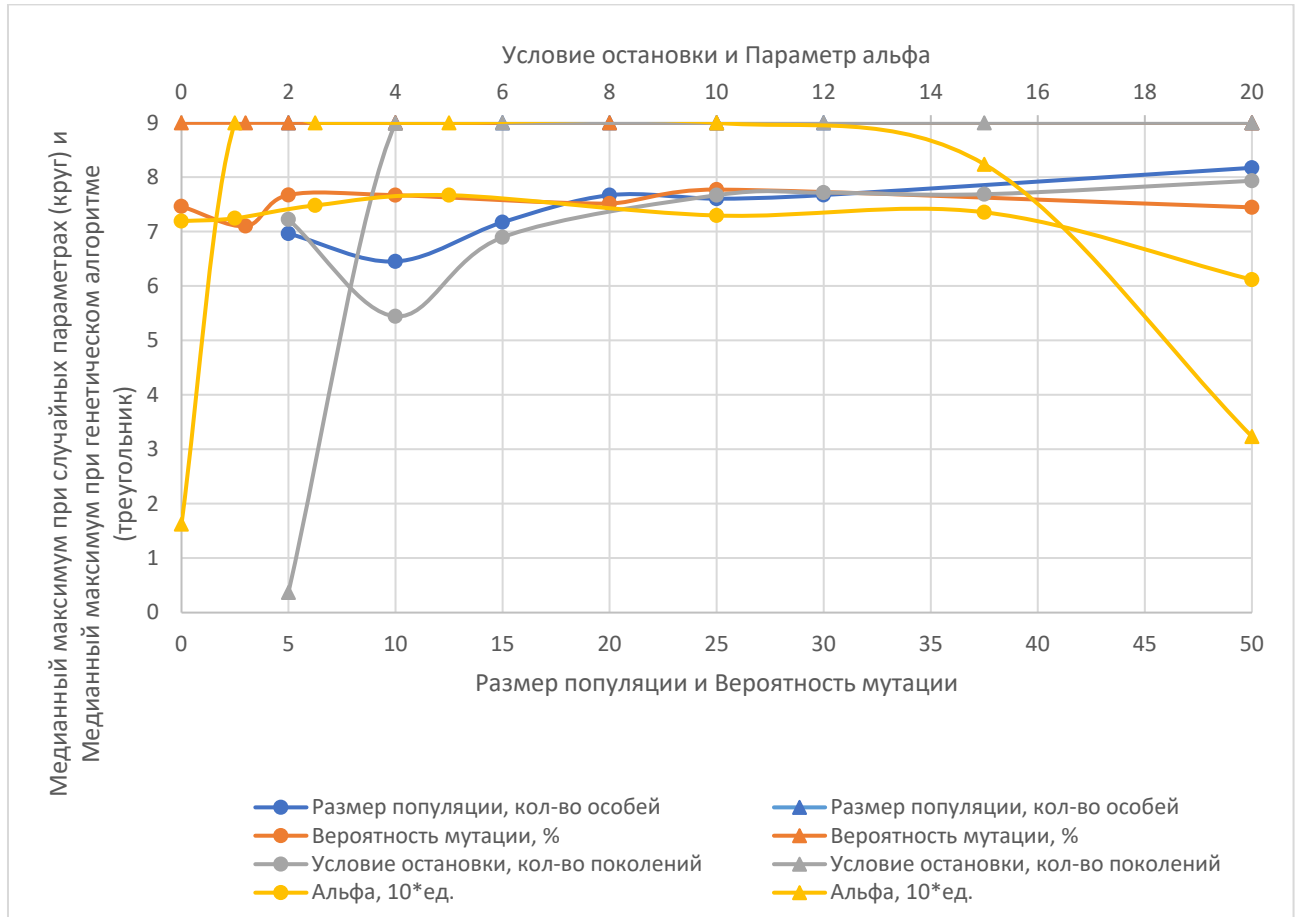


Рисунок 2.13 – Зависимость медианных максимумов метода случайных чисел и генетического алгоритма от гиперпараметров для тестовой функции

Можно заметить, что в абсолютном большинстве случаев генетический алгоритм работает лучше, чем метод случайных чисел, кроме того, в большинстве случаев медианное значение результата генетического алгоритма равно максимально возможному значению.

На Рисунке 2.14 представлена зависимость доли достижения генетическим алгоритмом абсолютного максимума и доли случаев, когда генетический алгоритм работал лучше метода случайных параметров от гиперпараметров для тестовой функции.

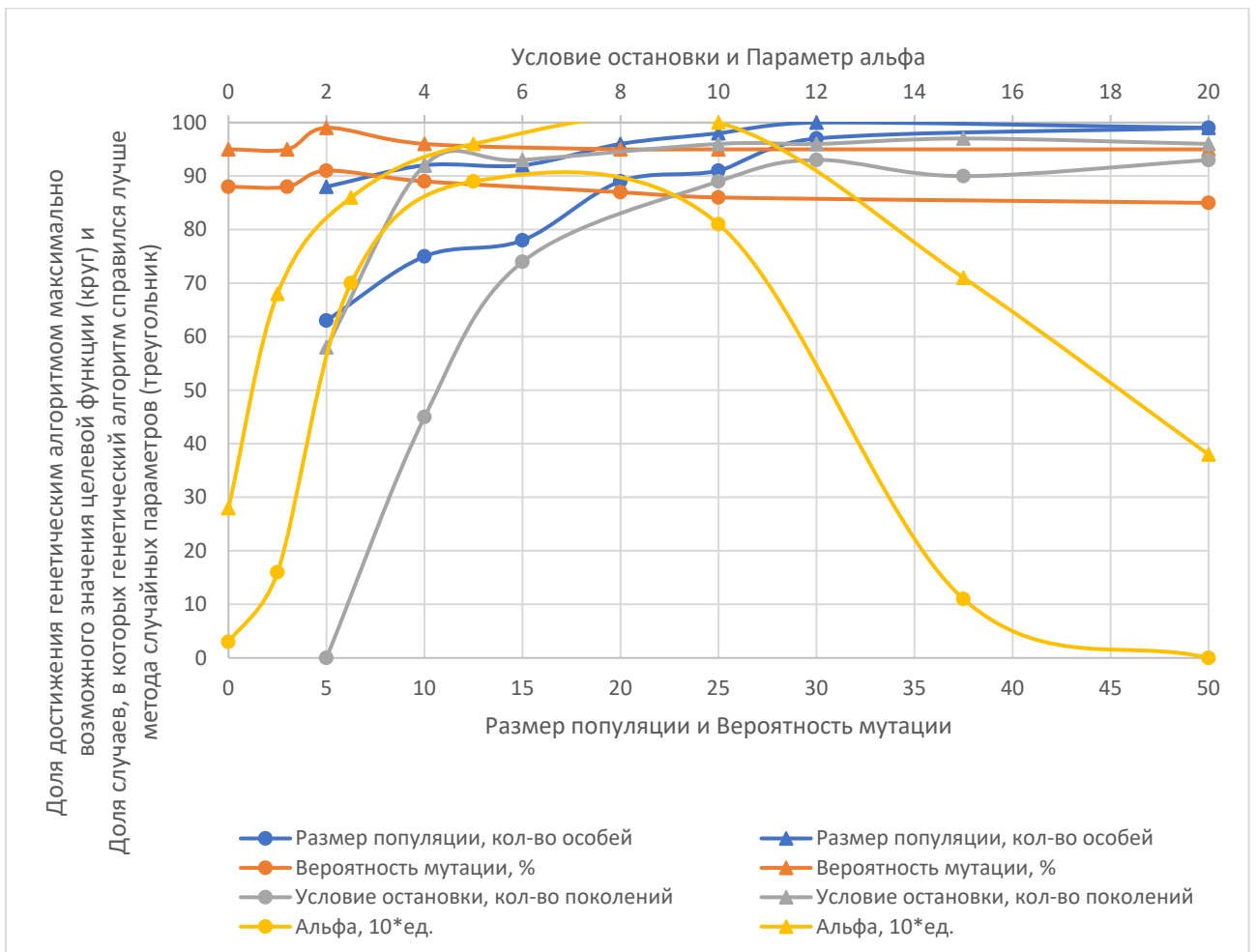


Рисунок 2.14 – Зависимость доли достижения генетическим алгоритмом абсолютного максимума и доли случаев, когда генетический алгоритм работал лучше метода случайных параметров от гиперпараметров для тестовой функции

Генетический алгоритм работает лучше метода случайных чисел в более чем 90% процентах случаев для большинства ситуаций. Достижение максимально возможного значения целевой функции происходит в более чем 85% случаев для большинства ситуаций. При этом вероятность мутации слабо влияет на результат. Алгоритм работает лучше на больших популяциях с высоким значением условия остановки. Параметр альфа имеет оптимальное значение.

Можно заключить, что генетический алгоритм справляется со своей задачей достаточно эффективно. Теперь от тестовой функции можно переходить к функции чистой приведенной стоимости.

2.5 Функция чистой приведенной стоимости

Для моделирования экономического эффекта разработки месторождения была создана финансово-экономическая модель, рассчитывающая чисто-приведенная стоимость (ЧПС) по профилю добычи. Описание финансово-экономической модели приведено в следующем разделе.

Параметры, входящие в расчет функции, представлены в Таблице 2.2. Всего 758670 возможных комбинаций параметров. Был проведен расчет целевой функции для всех этих комбинаций. Лучшая особь – особь с длиной горизонтального ствола 1000 метров, его направлением 90 градусов, расстоянием между скважинами 850 метров, радиусом скважины 30 сантиметров, длиной трещины ГРП 100 метров и без обработки кислотой. ЧПС в таком случае равна 773022903 доллара.

Генетический алгоритм был применен на данной функции аналогично тестовой функции. Ниже приведены результаты расчетов.

На Рисунке 2.15 представлена зависимость среднего количества поколений в итерации от гиперпараметров.

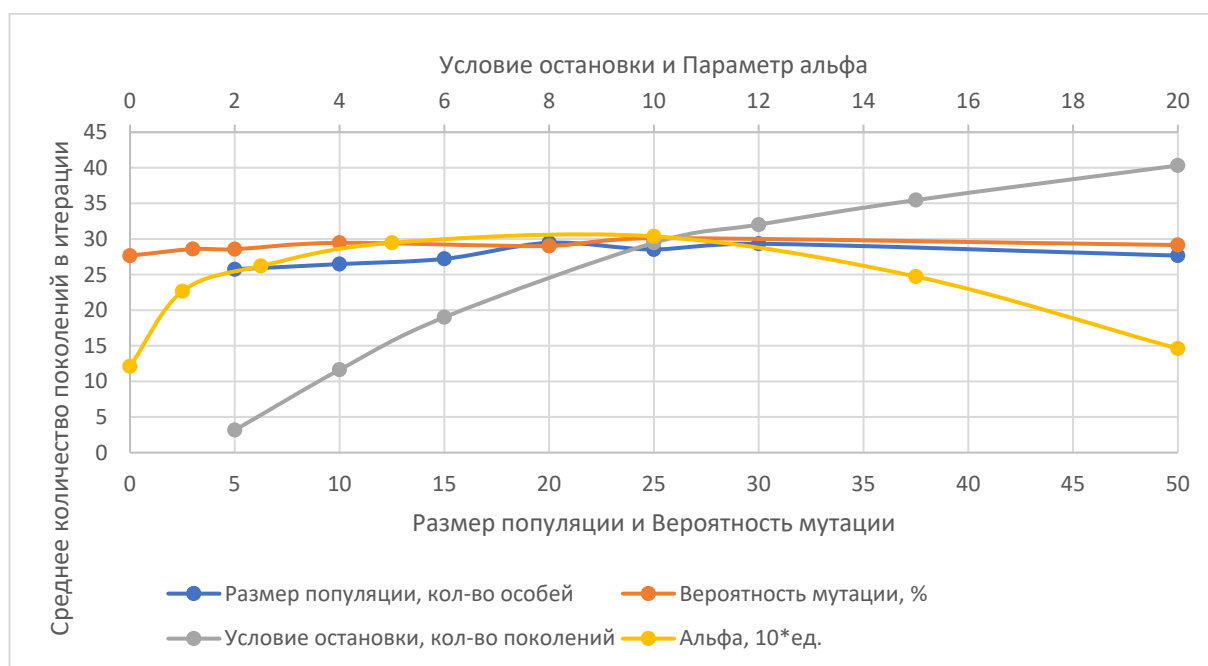


Рисунок 2.15 – Зависимость среднего количества поколений в итерации от гиперпараметров

В данном случае количество поколений почти не меняется с размером популяции. Это происходит из-за того, что изменчивость каждого признака достаточно мала, поэтому нет необходимости в большом размере популяции для обеспечения разнообразия особей.

На Рисунке 2.16 представлена зависимость среднего общего количества расчетов в итерации и количества расчетов до получения максимального результата от гиперпараметров.

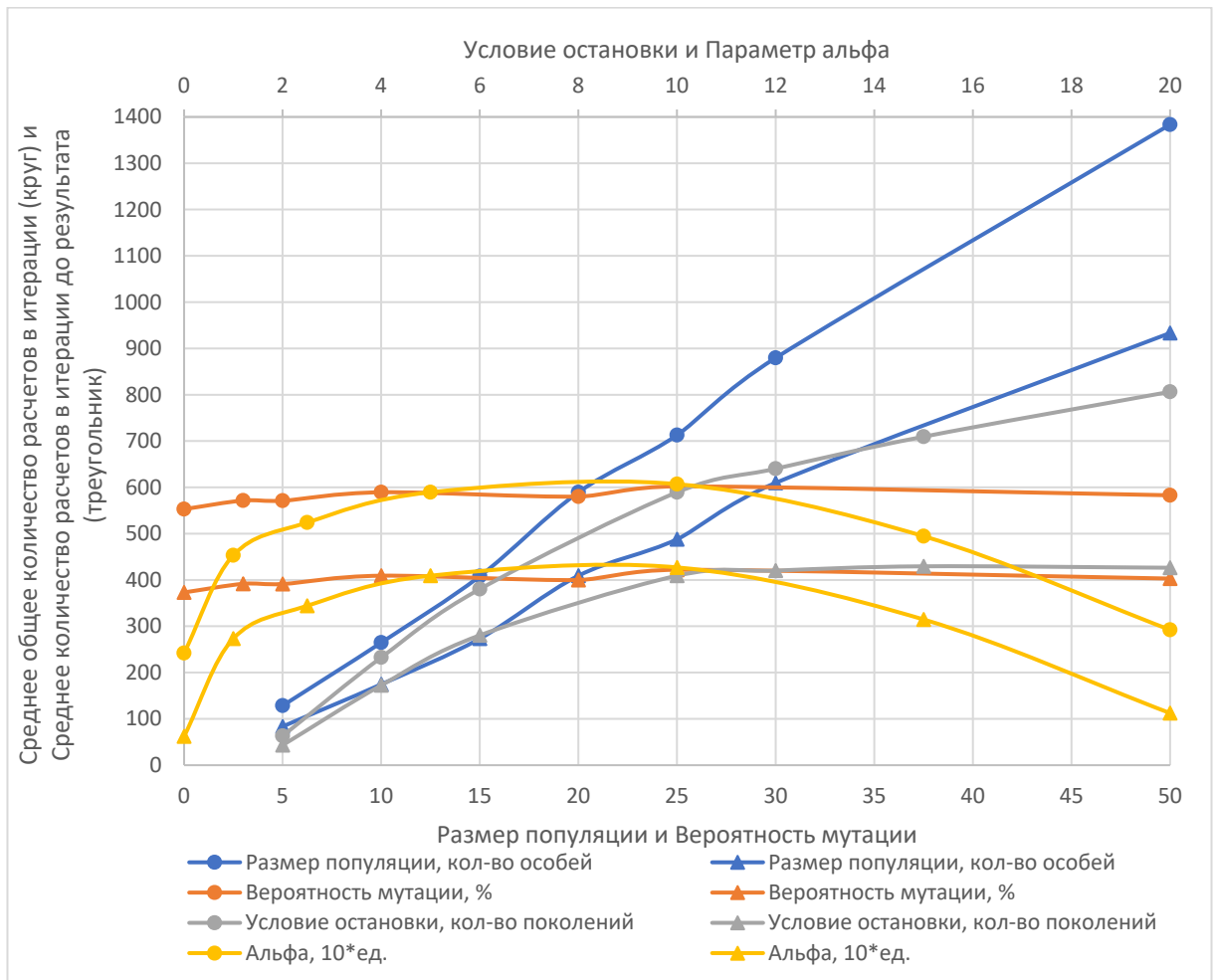


Рисунок 2.16 – Зависимость среднего общего количества расчетов в итерации и количества расчетов до получения максимального результата от гиперпараметров

Тренды тестовой функции повторяются.

На Рисунке 2.17 представлена зависимость средней доли от возможных вариантов от гиперпараметров.

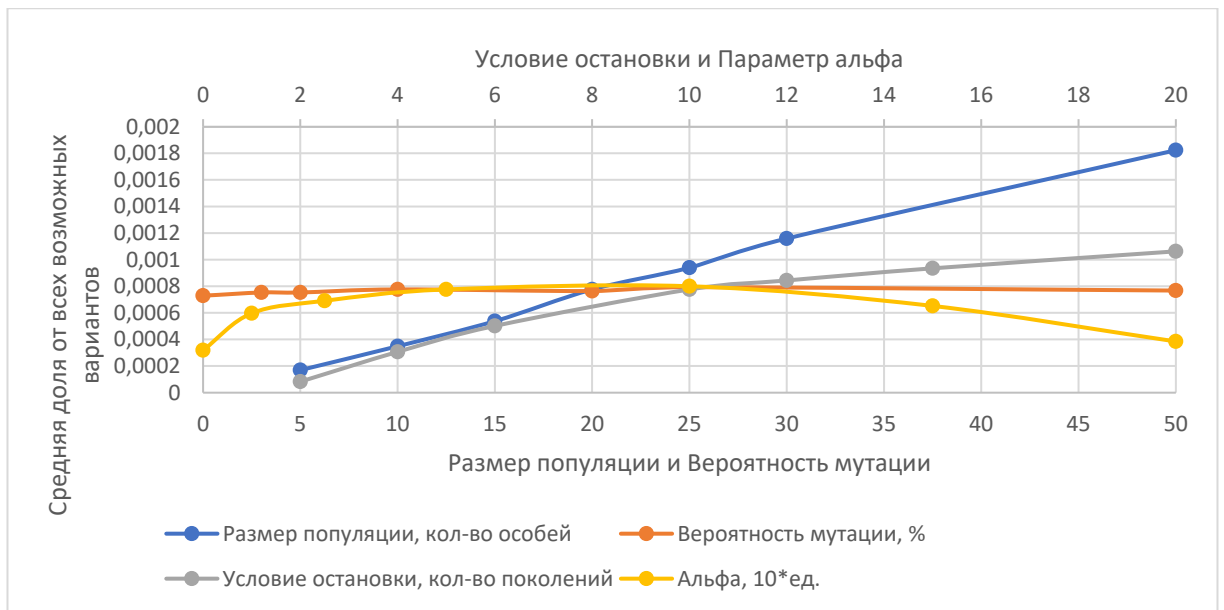


Рисунок 2.17 – Зависимость средней доли от возможных вариантов от гиперпараметров

Тренды тестовой функции повторяются.

На Рисунке 2.18 представлена зависимость медианных максимумов метода случайных чисел и генетического алгоритма от гиперпараметров.

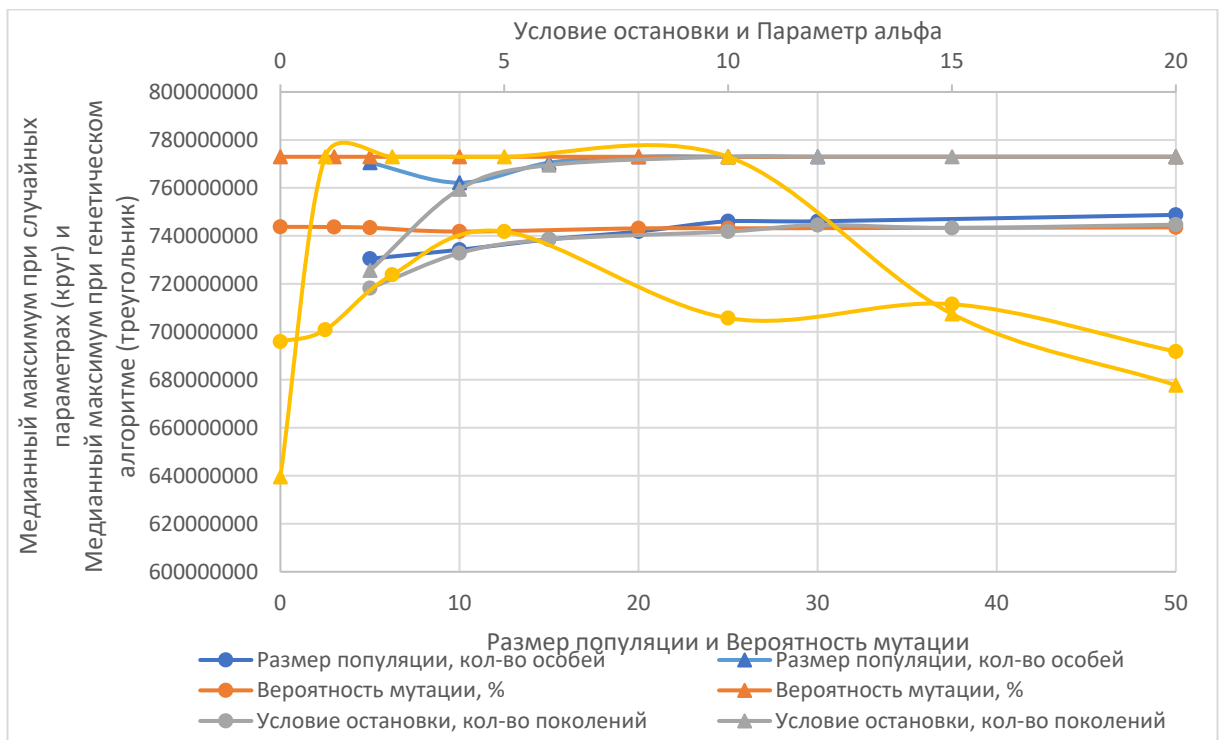


Рисунок 2.18 – Зависимость медианных максимумов метода случайных чисел и генетического алгоритма от гиперпараметров

В целом тренды тестовой функции повторяются.

На Рисунке 2.19 представлена зависимость доли достижения генетическим алгоритмом абсолютного максимума и доли случаев, когда генетический алгоритм работал лучше метода случайных параметров от гиперпараметров.

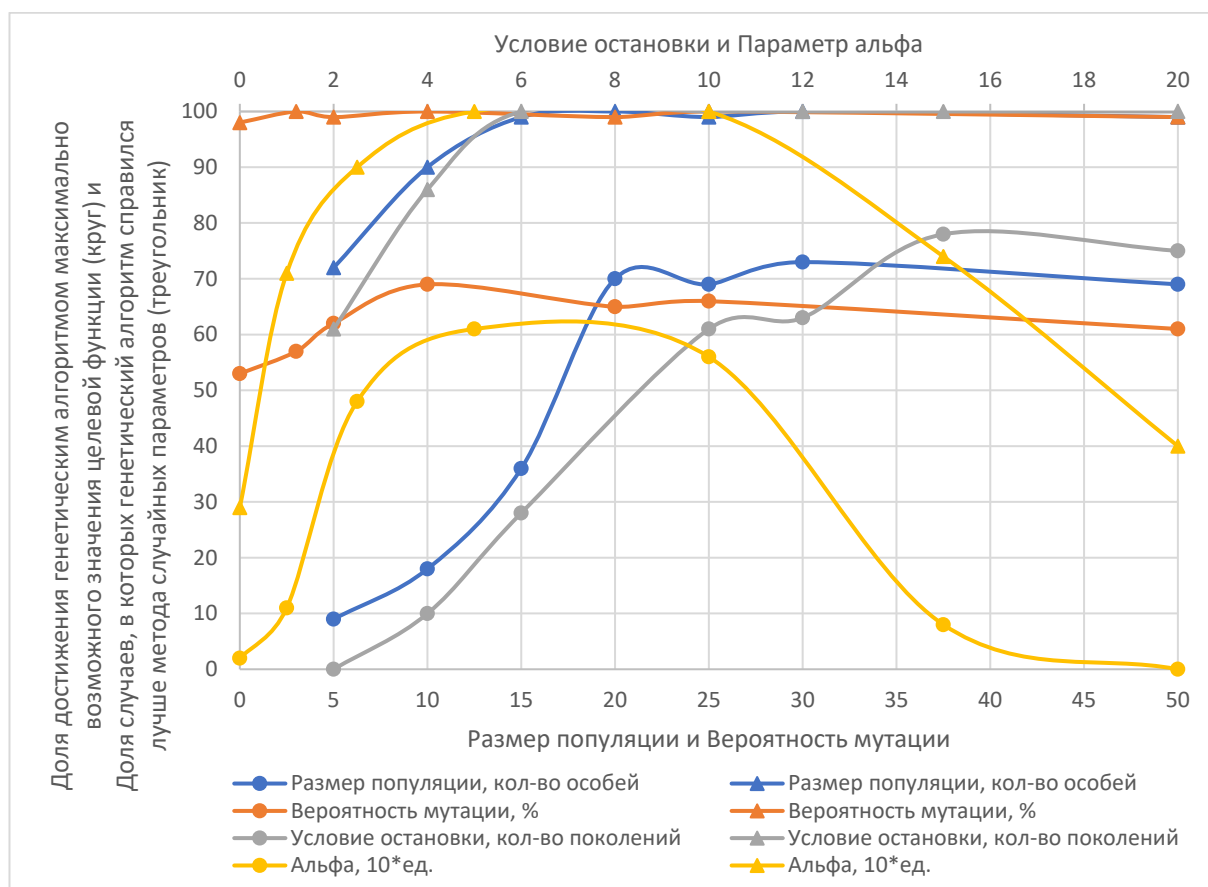


Рисунок 2.19 – Зависимость доли достижения генетическим алгоритмом абсолютного максимума и доли случаев, когда генетический алгоритм работал лучше метода случайных параметров от гиперпараметров

В целом тренды тестовой функции повторяются. Из-за сложности задачи доля достижения абсолютного максимума ниже, но доля случаев, когда генетический алгоритм работает лучше метода случайных параметров в абсолютном большинстве случаев близка к значению в 100%.

Можно сделать вывод, что генетический алгоритм достаточно успешно справляется с поставленной задачей, гиперпараметры, которые были выбраны в качестве базовых, достаточно хорошо подходят для данной задачи.

2.6 Выводы по разделу

Основываясь на проведенных расчетах, можно сделать некоторые выводы и замечания.

Генетический алгоритм показал эффективность в оптимизации системы разработки на синтетической модели месторождения нефти.

Правильно настроенный генетический алгоритм всегда работал лучше, чем метод случайных параметров. Это означает то, что успех работы генетического алгоритма в этой работе не может быть объяснен случайным совпадением параметров.

Более сложные задачи требуют большего количества расчетов и, соответственно, больших временных затрат и вычислительных мощностей.

Гиперпараметры алгоритма должны быть настроены оптимальным образом для каждой конкретной задачи.

В данном случае мерой оптимальности системы разработки является ЧПС. Стоит отметить, что ЧПС – не единственный экономический параметр, который должен быть взят в расчет. Кроме того, оптимальность системы разработки также определяется техническими, технологическими, экологическими и правовыми параметрами.

Генетический алгоритм не гарантирует получение самого лучшего варианта в каждом расчете, поэтому рекомендуется проводить его применение на одной и той же модели несколько раз для получения лучшего варианта с большей вероятностью.

Для более быстрого получения результата рекомендуется использовать технику с высокой вычислительной способностью и пользоваться распараллеливанием задач.

Нельзя использовать генетический алгоритм как некий «черный ящик», получающий входные данные и выдающий какой-то результат. Каждый полученный результат должен быть рассмотрен инженером. Результаты работы

генетического алгоритма рекомендуется использовать лишь в качестве помощи инженеру, а не для замены его работы.

Также рассмотрим некоторые направления развития написанного генетического алгоритма.

Необходимо увеличить быстродействие генетического алгоритма, оптимизировать его работы.

Необходимо протестировать генетического алгоритма на гидродинамических моделях месторождений. На первых этапах на простых гомогенных моделях небольшого размера с малым количеством изменяемых параметров, переходя в итоге к сложным гетерогенным моделям большого размера с большим количеством изменяемых параметров.

Необходимо проверить возможность использования генетического алгоритма для снижения рисков, связанных с неопределенностью. После того, как генетический алгоритм покажет эффективность при применении его на гидродинамических моделях, он может быть использован в качестве средства снижения рисков из-за неопределенностей. Например, он может быть применен на большом количестве различных геологических моделей, далее может быть выбран вариант системы разработки, наиболее подходящий для большего количества из этих геологических моделей.

Необходимо проанализировать использование других оптимизационных алгоритмов (метод роя частиц, алгоритм пчелиной колонии, муравьиный алгоритм и другие) для оптимизации системы разработки месторождений УВ.

3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Было принято, что критерием оптимальности выбора системы разработки является максимальное значение чистой приведенной стоимости. Соответственно, написанная программа считала данный параметр, используя профиль добычи нефти на входе.

Чистая приведенная стоимость относится к концепции денежных потоков. Денежные потоки – это любые доходы и расходы в ходе выполнения проекта, относящиеся к определенному времени (в будущем). Денежные потоки могут быть использованы для оценки активов, в частности проекта разработки месторождения УВ [18].

Чистый денежный поток (NCF) в определенный период времени складывается из положительного денежного потока – дохода (R) и отрицательного денежного потока – капитальных затрат (CAPEX), операционных затрат (OPEX) и налогов (T) [50]:

$$NCF = R - CAPEX - OPEX - T. \quad (3.1)$$

Капитальные затраты – это те затраты, которые используются для создания актива. Операционные затраты – это те затраты, которые используются для поддержания актива [24].

В представленном случае все денежные потоки приводились к шагу времени, равному одному году.

Принято, что инфляция отсутствует, поэтому все денежные потоки выражаются в базисных ценах.

Доход в нашем случае формируется за счет продажи добываемой нефти. Доход от продажи нефти, добытой за определенный период, определяется дебитом нефти (Q) и ценой на нефть (C):

$$R = QC. \quad (3.2)$$

Цена на нефть была принята равной 100 \$/баррель. Также было решено учесть удельные операционные затраты (C_o) на добычу и транспортировку нефти, введя реальную цену (C_t):

$$C_t = C - C_o. \quad (3.3)$$

Величина удельных затрат была принята равной 10 \$/баррель. Тогда реальный доход (R_t) определяется выражением:

$$R_t = QC_t. \quad (3.4)$$

Дебит каждой скважины определялся ежемесячно, после чего дебиты суммировались погодично. Таким образом, формировался профиль добычи нефти с месторождения. Используя формулу 3.4, можно получить профиль дохода с продажи нефти (см. Рисунок 3.1).

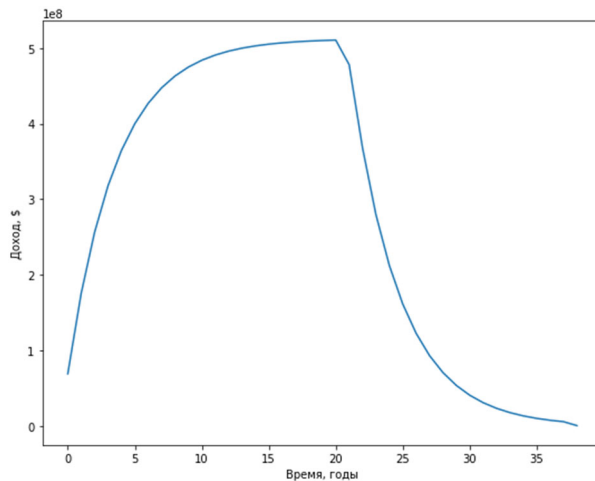


Рисунок 3.1 – Пример профиля дохода с продажи нефти

Капитальные и операционные затраты напрямую зависят от количества и конструкции скважин. Рассмотрим, как меняются эти параметры, опираясь на источники [35, 37].

Принимаем базовое значение капитальных затрат равным 15 млн. \$ на каждую скважину (с учетом поверхностного оборудования). Принимаем базовое значение капитальных затрат равным 1 тыс. \$ в месяц на каждую скважину. Рассмотрим, как параметры системы разработки (из предыдущего раздела) будут влиять на эти параметры.

Капитальные затраты на строительство горизонтального ствола растут с каждым метром. Принимаем, что дополнительные капитальные затраты составляют 500 \$ на каждый метр. Также горизонтальный ствол может нуждаться в больших операционных затратах. Поэтому принимаем, что коэффициент действия длины горизонтального ствола на операционные затраты линейно меняется от 1 до 1,2.

Направление горизонтального ствола и расстояние между скважинами не влияют на капитальные и операционные затраты для одной скважины в нашем случае.

Увеличение радиуса скважины увеличивает капитальные и операционные затраты. Поэтому принимаем, что коэффициент действия увеличения радиуса скважины на капитальные и операционные затраты линейно меняется от 1 до 1,1.

Проведение ГРП увеличивает капитальные затраты. Принимаем, что дополнительные капитальные затраты составляют 2000 \$ на каждый метр длины трещины. Также и операционные затраты увеличиваются. Поэтому принимаем, что коэффициент действия длины трещины ГРП на операционные затраты линейно меняется от 1 до 1,5.

Проведение кислотной обработки призабойной зоны пласта увеличивает капитальные затраты на скважину. Принимаем стоимость раствора (с учетом работ) равной 100 \$ на кубический метр. Также и операционные затраты могут

увеличиваться. Поэтому принимаем, что коэффициент действия радиуса кислотной обработки на операционные затраты линейно меняется от 1 до 1,1.

Капитальные и операционные затраты учитывались погодично. Имея доход от продажи нефти, капитальные и операционные затраты, может быть найдена полученная прибыль (P), необходимая для выплаты налогов:

$$P = R - CAPEX - OPEX. \quad (3.5)$$

Принято, что единственным видом налога является налог на прибыль. Принято, что его величина составляет 50% от прибыли. После этого может быть рассчитан чистый денежный поток на каждый год.

Заметим, что затраты на ликвидацию месторождения уже учтены в капитальных затратах. Для этих целей используется амортизационный фонд.

Для расчета чистой приведенной стоимости сначала необходимо рассчитать дисконтированный денежный поток (DCF) на каждый год от базового года (n) с учетом ставки дисконтирования (DR):

$$DCF = \frac{NCF}{(1 + DR)^n}. \quad (3.6)$$

Сумма всех дисконтированных потоков дает чистую приведенную стоимость (NPV), которую и требовалось найти:

$$NPV = \sum DCF. \quad (3.7)$$

В работе ставка дисконтирования принималась равной 15%. На Рисунке 3.2 приведен пример графика накопленного дисконтированного денежного потока.

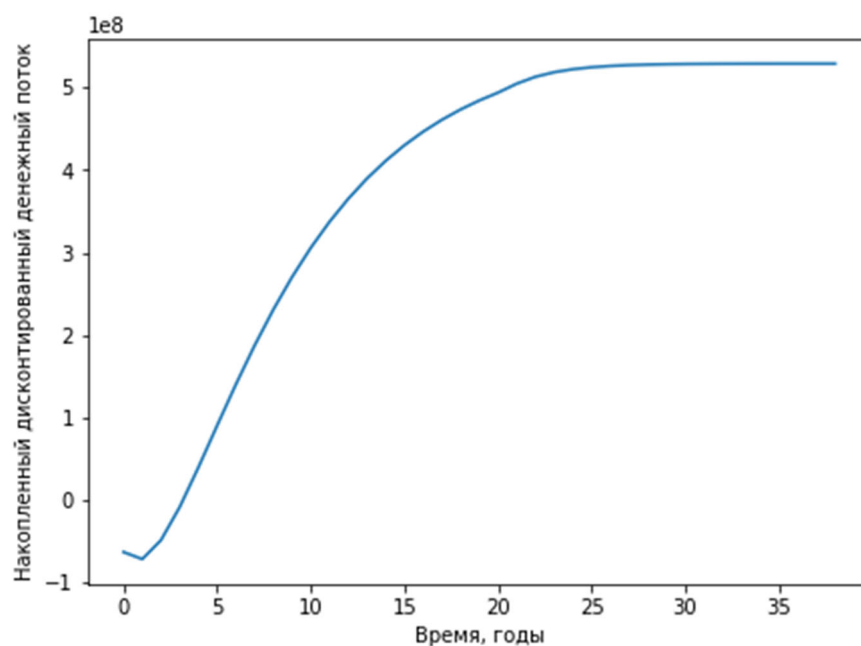


Рисунок 3.2 – Пример графика накопленного дисконтированного денежного потока

Стоит также отметить, что ЧПС является далеко не единственным экономическим параметром, который должен быть учтен при планировании инвестиционного проекта. Также весьма важными экономическими показателями являются следующие параметры [20, 50].

Период окупаемости. Данный параметр показывает, через какое количество времени инвестированные средства будут возвращены. Параметр является важным индикатором риска инвестиций, но ничего не говорит о будущем проекта.

Чистый доход. Параметр показывает, сколько денежных средств будет получено после завершения проекта (конечная точка кривой накопленного денежного потока). Чистая приведенная стоимость совпадает с этим параметром, но при этом учитывает дисконтирование.

Максимальная потребность в капитале (МПК). Параметр показывает, сколько средств должно быть инвестировано в проект. Обычно проекты с большим доходом требуют больших вложений капитала.

Индекс доходности. Параметр представляет собой частное чистого дохода и максимальной потребности в капитале. Параметр показывает, сколько финансов заработано на каждую вложенную единицу инвестиций.

Максимальное вложение капитала. Параметр совпадает с параметром максимальной потребности в капитале, но в условиях дисконтирования.

Дисконтированный индекс доходности. Параметр совпадает с параметром индекса доходности, но в условиях дисконтирования.

Внутренняя норма рентабельности. Параметр представляет собой ставку дисконтирования, при которой чистая приведенная стоимость равна нулю. Чем выше данный параметр, тем привлекательнее проект.

Также заметим, что при любом инвестировании средств необходимо учитывать возможные риски. В нефтегазовых проектах экономические риски связаны, прежде всего с параметрами, описанными ниже [40].

Геологический риск. Существует риск в оценке (переоценке) запасов УВ. Также из-за ограниченного количества информации существует риск выбора неоптимальной системы разработки.

Риск, связанный с выходом оборудования из строя. Выход из строя какого-либо оборудования чаще всего происходит неожиданно (часто из-за человеческого фактора), что приводит к потерям времени и, соответственно, потере денежных средств.

Политический риск. Налоги являются одной из самых важных статей расходов нефтегазовых компаний. Всегда существует риск, что государство может изменить налоговую политику.

Экономический риск. Цены на УВ и курсы валют (и другие экономические параметры) все время меняются, что сильно влияет на доходы, получаемые с продажи УВ.

Партнерский риск. Нефтегазовые компании работают с другими компаниями-партнерами. Если компания-партнер не выполнит какие-либо свои обязательства, это может приводить к срывам сроков и экономическим потерям.

Итак, в данном разделе был рассмотрен процесс расчета чистой приведенной стоимости инвестиционного проекта на примере системы разработки синтетической модели месторождения УВ. Также обсуждались другие экономические параметры, используемые для оценки инвестиционной привлекательности проектов и риски, связанные с ними.

4 Социальная ответственность

В данном разделе диссертации рассматриваются вопросы производственной безопасности, экологической безопасности и безопасности в чрезвычайных ситуациях при разработке и использовании генетических алгоритмов с использованием персональных компьютеров (ПК).

4.1 Правовые и организационные вопросы обеспечения безопасности

4.1.1 Специальные (характерные для рабочей зоны) правовые нормы трудового законодательства

Правовые нормы трудового законодательства (в том числе характерные для рабочей зоны – офиса) регламентируются Трудовым кодексом Российской Федерации [12]. Согласно нему работник офиса имеет право на:

- 1) рабочее место, соответствующее требованиям охраны труда;
- 2) обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний;
- 3) получение достоверной информации об условиях и охране труда на рабочем месте, о существующем риске повреждения здоровья, а также о мерах по защите от воздействия вредных и (или) опасных производственных факторов;
- 4) отказ от выполнения работ в случае возникновения опасности для его жизни и здоровья вследствие нарушения требований охраны труда до устранения такой опасности;
- 5) обеспечение средствами индивидуальной и коллективной защиты в соответствии с требованиями охраны труда;
- 6) обучение безопасным методам и приемам труда;

- 7) дополнительное профессиональное образование в случае ликвидации рабочего места вследствие нарушения требований охраны труда;
- 8) запрос о проведении проверки условий и охраны труда на его рабочем месте;
- 9) обращение в различные органы власти по вопросам охраны труда;
- 10) участие в рассмотрении вопросов, связанных с обеспечением безопасных условий труда на его рабочем месте, и в расследовании происшедшего с ним несчастного случая на производстве или профессионального заболевания;
- 11) внеочередной медицинский осмотр в соответствии с медицинскими рекомендациями с сохранением за ним места работы (должности) и среднего заработка во время прохождения указанного медицинского осмотра;
- 12) гарантии и компенсации, если он занят на работах с вредными и (или) опасными условиями труда.

4.1.2 Организационные мероприятия при компоновке рабочей зоны

Описываемые далее организационные мероприятия при компоновке рабочей зоны основаны на следующих документах:

- 1) ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования [4];
- 2) СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы [8];
- 3) ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре [5];
- 4) ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора [6];

5) ГОСТ Р ИСО 9241-7-2007. Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов (ВДТ). Часть 7. Требования к дисплеям при наличии отражений [7].

Ниже представлены основные требования к рабочему месту в офисе.

Высота над уровнем пола рабочей поверхности, за которой работает оператор, должна составлять 720 мм. Оптимальные размеры поверхности стола 1600 на 1000 кв. мм. Под столом должно иметься пространство для ног с размерами по глубине 650 мм. Рабочий стол должен также иметь подставку для ног, расположенную под углом 15 градусов к поверхности стола. Длина подставки 400 мм, ширина – 350 мм. Удаленность клавиатуры от края стола должна быть не более 300 мм, что обеспечит удобную опору для предплечий. Расстояние между глазами оператора и экраном видеодисплея должно составлять от 40 до 80 см. Так же рабочий стол должен быть устойчивым, иметь однотонное неметаллическое покрытие, не обладающее способностью накапливать статическое электричество. Рабочий стул должен иметь дизайн, исключаящий онемение тела из-за нарушения кровообращения при продолжительной работе на рабочем месте.

4.2 Производственная безопасность

Далее будут рассматриваться вредные и опасные производственные факторы, связанные с использованием компьютера в офисном помещении.

4.2.1 Анализ выявленных вредных и опасных производственных факторов

ГОСТ 12.0.003-2015 Опасные и вредные производственные факторы. Классификация [1] был использован для выявления вредных и опасных

производственных факторов в рассматриваемой ситуации. Выявленные вредные и опасные производственные факторы представлены в Таблице 4.1.

Таблица 4.1 – Выявленные вредные и опасные производственные факторы

Вредный или опасный производственный фактор	Нормативный документ
а) Перенапряжение зрительного анализатора	СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы [8]
б) Умственное перенапряжение	Трудовой кодекс Российской Федерации [12]
в) Повышенный уровень шума на рабочем месте	СН 2.2.4/2.1.8.562-96. Санитарные нормы. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки [10]
г) Повышенный уровень электромагнитных излучений	СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы [8]
д) Повышенная или пониженная температура воздуха рабочей зоны	СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений [9]
е) Недостаточная освещенность рабочей зоны	СП 52.13330.2016 Естественное и искусственное освещение [11]
ж) Электрический ток, под действие которого может попасть персонал	ГОСТ 12.1.038-82 Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов [3]

а) Перенапряжение зрительного анализатора.

Компьютерный монитор является постоянным источником света. Работник офиса проводит большую часть рабочего дня, смотря в монитор, соответственно, зрительный анализатор находится в постоянном напряжении. Это может приводить к ухудшению зрения, сухости глаз, головным болям. Важным элементом профилактики данных явлений является периодический отдых. Кроме того, дисплей компьютера должен отвечать требованиям, представленным в Таблице 4.2.

Таблица 4.2 – Допустимые визуальные параметры устройств отображения информации

Параметр	Допустимые значения
Яркость белого поля	Не менее 35 кд/м ²
Неравномерность яркости рабочего поля	Не более примерно 20%
Контрастность (для монохромного режима)	Не менее 3 к 1
Частота обновления изображения	Не менее 60 Гц

б) Умственное перенапряжение.

Умственное перенапряжение может возникнуть вследствие выполнения сложной производственной задачи. Это может приводить к состоянию стресса, психологическим проблемам, усталости, снижению концентрации и работоспособности. Для профилактики данного явления следует соблюдать режим работы и отдыха, а также следить за сном и питанием и заниматься физическими упражнениями.

в) Повышенный уровень шума на рабочем месте.

Громкие или (и) постоянные шумы на рабочем месте могут возбуждать нервную систему, что может приводить к состоянию стресса, психологическим проблемам, усталости, снижению концентрации и работоспособности, головным болям. В условиях офиса шум не должен превышать значений, представленных в Таблице 4.3. Для этого можно использовать регламентирование использования офисного оборудования, вызывающего громкие звуки, а также использовать оборудование с низким уровнем шума и звукоизолирующие устройства.

Таблица 4.3 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для офисных помещений

Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентные уровни звука (в дБА)
31,5	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	50

г) Повышенный уровень электромагнитных излучений.

ПК, использующийся в офисном помещении, создает электромагнитные поля, которые могут оказывать влияние на организм человека. Превышение допустимых значений может приводить к нарушениям нервной и сердечно-сосудистой систем организма человека.

ПК, используемые во время работы, не должны превышать допустимые значения, представленные в Таблице 4.4. Для этого ПК должны быть

проверены на наличие положительных результатов (от аккредитованных лабораторий) модели на соответствие норм по уровню электромагнитного поля.

Таблица 4.4 – Допустимые уровни электромагнитного поля, создаваемых персональной электронно-вычислительной машиной

Наименование параметров		Допустимый уровень электромагнитного поля
Напряженность электрического поля	в диапазоне частот от 5 Гц до 2 кГц	25 В/м
	в диапазоне частот от 2 кГц до кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот от 5 Гц до 2 кГц	250 нТл
	в диапазоне частот от 2 кГц до кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

д) Повышенная или пониженная температура воздуха рабочей зоны

Повышенная или пониженная температура на рабочем месте может приводить к снижению работоспособности, повышенной утомляемости, в крайних случаях к перегреву или переохлаждению. В Таблице 4.5 представлены оптимальные величины показателей микроклимата в офисном помещении (категория работ по уровню энергозатрат Ib). Для поддержания оптимального микроклимата могут быть использованы системы кондиционированию в теплый период года и системы водяного отопления в холодный период года.

Таблица 4.5 – Оптимальные величины показателей микроклимата в офисном помещении

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	от 21 до 23	от 20 до 24	от 60 до 40	0,1
Теплый	от 22 до 24	от 21 до 25	от 60 до 40	0,1

е) Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны может приводить к перенапряжению зрительного анализатора (о чем говорилось ранее).

Офисные помещения должны располагать системой общего равномерного освещения. В случае работы с печатными документами дополнительно используются системы местного освещения рабочего стола.

Освещенность на поверхности стола в зоне размещения рабочих документов должна быть от 300 до 500 лк. Освещение не должно создавать бликов на поверхности монитора. Освещенность поверхности экрана должна быть не более 300 лк.

В Таблице 4.6 приведены нормативные показатели освещения офисных помещений.

Таблица 4.6 – Нормативные показатели освещения в офисном помещении

Искусственное освещение						Естественное освещение		Совмещенное освещение	
Освещенность рабочих поверхностей, лк		Цилиндрическая освещенность, лк	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, %, не более	Индекс цветопередачи источников света	коэффициент естественной освещенности, %		коэффициент естественной освещенности, %	
при комбинированном освещении	при общем освещении					при верхнем или комбинированном освещении	при боковом освещении	при верхнем или комбинированном освещении	при боковом освещении
400/200	300	-	21	15	80	3	1	1,8	0,6

ж) Электрический ток, под действие которого может попасть персонал

Основным опасным производственным фактором в офисном помещении является возможность поражения электрическим током. Основными причинами поражения током в условиях офиса являются прикосновение к токоведущим частям, находящимся под напряжением и использование неисправного электрического оборудования или небезопасная эксплуатация исправного электрического оборудования.

Человек под воздействием электрического тока может получить ожоги, могут возникать нарушения работы нервной, сердечно-сосудистой и мышечной систем, в крайнем случае возможны летальные исходы.

Для защиты от поражения электрическим током следует выполнять общие правила безопасности при работе с электричеством: не использовать неисправное оборудование, использовать оборудование только по назначению и придерживаться инструкций, проверять оборудование на исправность перед использованием, а также не производить ремонт техники без соответствующей квалификации.

4.2.2 Расчет системы искусственного освещения

Проведем расчет системы искусственного освещения для офисной комнаты, в которой проводилось написание магистерской диссертации. Расчет проводится в соответствии с источниками [14, 21, 22].

Для выбора типа светильника необходимо определить коэффициенты отражения потолка ($\rho_{\text{п}}$) и стен ($\rho_{\text{с}}$). В комнате чистый бетонный потолок и бетонные стены с окнами, коэффициенты отражения потолка и стен равняются 50 и 30% соответственно. Так как помещение нормальное, с хорошим отражением потолка и стен, выбираем открытый светильник типа ОДО с двумя люминесцентными лампами мощностью 40 Вт.

Определим высоту подвеса ($h_{\text{п}}$) светильников. Высота потолка (H) комнаты равна 4,4 м. Свес светильников ($h_{\text{с}}$) принимаем равным 0,6 м. Тогда:

$$h_{\text{п}} = H - h_{\text{с}} = 4,4 - 0,6 = 3,8 \text{ м.} \quad (4.1)$$

Данное значение больше минимально допустимого (3,5 м), соответственно, высота подвеса соответствует нормам.

Определим расчетную высоту (h). Высота рабочей поверхности ($h_{рп}$) равна 0,8 м. Тогда:

$$h = h_{п} - h_{рп} = 3,8 - 0,8 = 3 \text{ м.} \quad (4.2)$$

Все вертикальные расстояния посчитаны, схему можно увидеть на Рисунке 4.1. Относительные масштабы соблюдены.

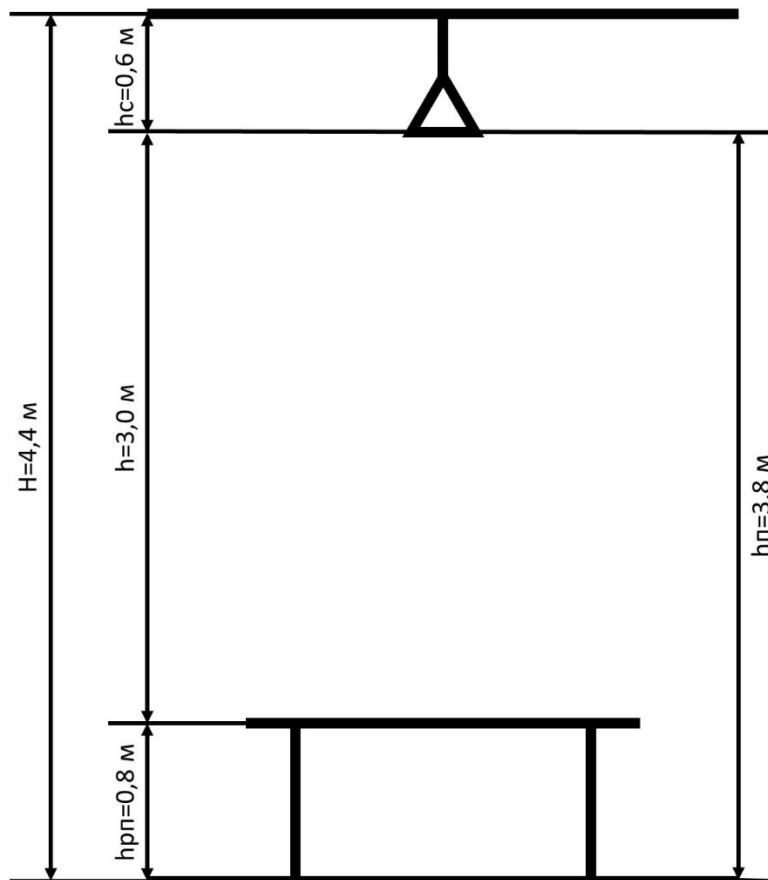


Рисунок 4.1 – Схема вертикальных расстояний для светильников

Определим расстояние между светильниками (L). Учтем, что интегральный критерий оптимальности расположения светильников (λ) для выбранного типа равен 1,4. Тогда:

$$L = \lambda \times h = 1,4 \times 3 = 4,2 \text{ м.} \quad (4.3)$$

Определим количество рядов ($n_{\text{ряд}}$). Учтем, что ширина комнаты (B) равна 6 м. Тогда:

$$n_{\text{ряд}} = \frac{\left(B - \frac{2}{3}L\right)}{L} + 1 = \frac{\left(6 - \frac{2}{3} \times 4,2\right)}{4,2} + 1 = 1,76. \quad (4.4)$$

Округлим данное значение до целого (2).

Определим количество светильников в ряду ($n_{\text{св}}$). Учтем, что длина комнаты (A) равна 6 м и длина светильника ($l_{\text{св}}$) равна 1,23 м. Тогда:

$$n_{\text{св}} = \frac{\left(A - \frac{2}{3}L\right)}{l_{\text{св}} + 0,5} + 1 = \frac{\left(6 - \frac{2}{3} \times 4,2\right)}{1,23 + 0,5} + 1 = 1,85. \quad (4.5)$$

Округлим данное значение до целого (2).

Схему размещения светильников на плане можно увидеть на Рисунке 4.2. Относительные масштабы соблюдены.

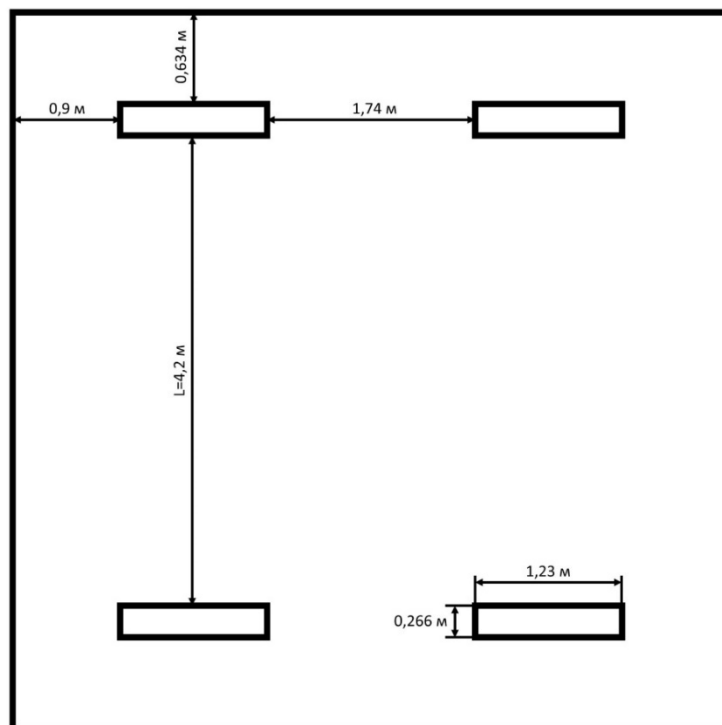


Рисунок 4.2 – Схема размещения светильников на плане

Определим число ламп в осветительной системе (N). Учтем, что количество ламп (n) в одном светильнике равно 2. Тогда:

$$N = n \times n_{\text{ряд}} \times n_{\text{св}} = 2 \times 2 \times 2 = 8. \quad (4.6)$$

Определим индекс помещения (i):

$$i = \frac{AB}{h(A+B)} = \frac{6 \times 6}{3(6+6)} = 1. \quad (4.7)$$

Тогда коэффициент использования светового потока (η) равен 44%.

Так как помещение является офисным, считаем его помещением с малым выделением пыли, тогда коэффициент запаса (K_3) равен 1,5. Нормативная освещенность (E_n) для искусственного освещения офисного помещения равна 200 лк.

Определим световой поток (Φ). Учтем, что коэффициент неравномерности освещения (Z) для люминесцентных ламп равен 1,1. Тогда:

$$\Phi = \frac{E_n A B K_3 Z}{N \eta} = \frac{200 \times 6 \times 6 \times 1,5 \times 1,1}{8 \times 0,44} = 3375 \text{ лм.} \quad (4.8)$$

Ближайшая стандартная лампа – лампа ЛД мощностью (p) 65 Вт со световым потоком (ϕ) 3750 лм. Выполним проверку:

$$\begin{aligned} -10\% \leq \frac{\phi - \Phi}{\Phi} 100\% \leq +20\% & \Rightarrow \\ \Rightarrow -10\% \leq \frac{3750 - 3375}{3750} 100\% \leq +20\% & \Rightarrow \\ \Rightarrow -10\% \leq +10\% \leq +20\%. & \end{aligned} \quad (4.9)$$

Условие выполняется. Можно заключить, что для рассматриваемого помещения подходит установка четырех светильников типа ОДО с двумя лампами типа ЛД мощностью 65 Вт и световым потоком 3750 лм в каждом.

Также определим номинальную мощность осветительной системы (P):

$$P = Np = 8 \times 65 = 520 \text{ Вт.} \quad (4.10)$$

Таким образом, была рассчитана система искусственного освещения рабочей комнаты. Выбраны четыре светильника типа ОДО с двумя лампами типа ЛД мощностью 65 Вт и световым потоком 3750 лм в каждом. Размещение светильников показано на Рисунках 4.1 и 4.2.

4.3 Экологическая безопасность

Разработка и использование генетического алгоритма проходит с использованием ПК. Именно поэтому в данном подразделе будет рассматриваться (прежде всего) утилизация компьютерной техники с опорой на источник [17].

Компьютерная техника и периферийные устройства содержат в себе свинец, ртуть, бериллий, кадмий, мышьяк, охлаждающие жидкости и другие токсичные вещества, которые могут загрязнять литосферу (твердые отходы компьютерной техники) и гидросферу (жидкие отходы компьютерной техники) при неправильной утилизации – при вывозе на свалки (возможно, стихийные) бытовых отходов.

Кроме того, компьютерная техника содержит материалы, которые могут и должны быть использованы повторно (пластик, редкие и ценные металлы и так далее).

По перечисленным выше причинам, компьютерная техника должна быть утилизирована особым образом. Самый очевидный и правильный способ – это заключение договора с фирмами, профессионально занимающимися утилизацией подобного рода техники. Кроме того, используемая бумага, люминесцентные лампы, аккумуляторы, которые также могут использоваться в офисе, также должны быть утилизированы подобным образом. При этом нанимаемая фирма верным образом утилизирует токсичные отходы и отправляет на вторичную переработку материалы, которые могут быть переработаны.

Также отметим, что некоторые сетевые производители и продавцы электроники имеют программы утилизации старой техники, в ходе которых можно обменять устаревший или сломанный ПК на денежные средства или скидку на покупку других товаров. Сданные таким образом ПК также отправляются на вторичную переработку.

4.4 Безопасность в чрезвычайных ситуациях

Основные чрезвычайные ситуации, которые могут произойти в офисном помещении, включают в себя:

- 1) геологические воздействия (землетрясения, провалы и обвалы территорий);
- 2) опасные явления метеорологического характера;
- 3) природные катастрофы (наводнения, цунами, ураганы);
- 4) пожары.

Так как разработка и использование генетического алгоритма происходит в офисном здании с использованием ПК, наиболее вероятная чрезвычайная ситуация – это пожар (например, по причине возгорания компьютерной техники из-за какой-либо неисправности). Приведем меры по предупреждению данной чрезвычайной ситуации с опорой на источник [2].

Пожар в офисном здании может появиться по причине перегрузке электрической сети, короткого замыкания, неисправной проводки, использования неисправного электрического оборудования.

Здание, в котором выполняются работы, должно соответствовать требованиям пожарной безопасности, иметь план эвакуации (все работники должны быть с ним ознакомлены), иметь указатели путей эвакуации. Также должна действовать пожарная сигнализация, должны иметься огнетушители и пожарные рукава.

Также весь персонал должен быть информирован о действиях в случае пожара, должны проводиться периодические учения. Нельзя курить в помещении и пользоваться открытым огнем, необходимо пользоваться только исправным электрическим оборудованием.

4.5 Выводы по разделу

Таким образом, в разделе «Социальная ответственность» были рассмотрены правовые и организационные вопросы обеспечения безопасности (рассмотрены специальные правовые нормы трудового законодательства и организационные мероприятия при компоновке рабочей зоны), производственная безопасность (проанализированы вредные и опасные производственные факторы), экологическая безопасность (воздействие на окружающую среду) и безопасность в чрезвычайных ситуациях. Кроме того, был проведен расчет системы искусственного освещения для офисного помещения. Полученные результаты могут быть использованы на производстве.

Заключение

В ходе выполнения данной магистерской диссертации были изучены генетические алгоритмы (их схема работы, применение, преимущества и недостатки и используемые методы), после чего написан и применен генетический алгоритм, который позволил оптимизировать сначала тестовую функцию, а затем и систему разработки синтетической модели месторождения нефти. Результаты были проанализированы и было доказано, что генетический алгоритм может быть использован для оптимизации системы разработки, так как позволяет находить такие параметры системы разработки, которые позволяют максимизировать такой экономический параметр как чистая приведенная стоимость. Таким образом, цель и задачи были выполнены.

Тем не менее, рекомендуется продолжить изучение применимости алгоритма на более сложных системах, таких как гидродинамические модели месторождений нефти и газа различной сложности. Также рекомендуется проанализировать эффективность других оптимизационных алгоритмов при решении той же задачи.

При успешном применении данного генетического алгоритма на более сложных системах алгоритм может быть использован на предприятиях добычи нефти и газа инженерами-разработчикам месторождений углеводородов и специалистами по геологическому и гидродинамическому моделированию в качестве помощи при принятии решений. Это позволит подбирать более эффективные системы разработки месторождений, что приведет к росту экономической эффективности проектов разработки месторождений. Также алгоритм может быть использован для экономии времени специалистов.

Кроме того, в работе была представлена финансово-экономическая модель, которая использовалась для расчета чистой приведенной стоимости из профиля добычи нефти и проанализированы другие экономические параметры, используемые для оценки проектов, и риски, которые должны быть учтены. Также рассматривались вопросы социальной ответственности при создании и

использовании генетических алгоритмов (правовые и организационные вопросы обеспечения безопасности, производственная безопасность, экологическая безопасность и безопасность в чрезвычайных ситуациях).

Список использованных источников

Нормативные документы

1. ГОСТ 12.0.003-2015 Опасные и вредные производственные факторы. Классификация [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 2015. – 3 с.

2. ГОСТ 12.1.004-91 Пожарная безопасность. Общие требования. [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 1991. – 66 с.

3. ГОСТ 12.1.038-82 Система стандартов безопасности труда. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 1982. – 7 с.

4. ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 1978. – 9 с.

5. ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 2009. – 40 с.

6. ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 2009. – 38 с.

7. ГОСТ Р ИСО 9241-7-2007. Эргономические требования при выполнении офисных работ с использованием видеодисплейных терминалов

(ВДТ). Часть 7. Требования к дисплеям при наличии отражений [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 2007. – 28 с.

8. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 2003. – 54 с.

9. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 1996. – 20 с.

10. СН 2.2.4/2.1.8.562-96. Санитарные нормы. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 1996. – 8 с.

11. СП 52.13330.2016 Естественное и искусственное освещение [Электронный ресурс] – Доступ из справ.-правовой системы «КонсультантПлюс», 2016. – 116 с.

12. Трудовой кодекс Российской Федерации [Электронный ресурс]: федер. закон от 30.12.2001 № 197-ФЗ – Доступ из справ.-правовой системы «КонсультантПлюс», 2001. – 210 с.

Опубликованные источники

13. Батищев Д. И. Применение генетических алгоритмов к решению задач дискретной оптимизации / Д. И. Батищев, Е. А. Неймарк, Н. В. Старостин. – Нижний Новгород: Нижегородский государственный университет им. Н.И. Лобачевского, 2007. – 347 с.

14. Беспалько Н. Е. Расчет общего равномерного искусственного освещения производственного помещения: метод. указания / Н. Е. Беспалько. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 12 с.

15. Бойко В. С. Разработка и эксплуатация нефтяных месторождений / В. С. Бойко. – М.: Недра, 1990. – 427 с.

16. Гуреева М. А. Экономика нефтяной и газовой промышленности: учебник для студ. учреждений сред. проф. образования / М. А. Гуреева. – М.: Издательский центр «Академия», 2012. – 240 с.
17. Куда сдать старый компьютер [Электронный ресурс] / Point metal recycling. URL: <https://pointmetal.ru/60-kuda-sdat-staryi-kompyuter.html#pochemu-nelzya-vybrosit-na-svalku>. Дата обращения: 25.05.2022.
18. Кузьмин Т. Г. Экономика инвестиционного проекта в нефтегазовой отрасли: учебное пособие. / Т. Г. Кузьмин, П. В. Молодых. – Томск: Изд-во Томского политехнического университета, 2012. – 261 с.
19. Панченко Т. В. Генетические алгоритмы: учебно-методическое пособие / Т. В. Панченко под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с.
20. Ример М. И. Экономическая оценка инвестиций / М. И. Ример, А. Д. Касатов, Н. Н. Матиенко, под общ. ред. М. И. Римера – СПб.: Питер, 2008. – 480 с.
21. Романенко С. В. Методика расчета системы общего равномерного искусственного освещения С. В. Романенко, Ю. В. Анищенко. – Томск: Изд-во Томского политехнического университета, 2016. – 7 с.
22. Романенко С. В. Методические указания по разработке раздела «Социальная ответственность» выпускной квалификационной работы магистра, специалиста и бакалавра всех направлений (специальностей) и форм обучения ТПУ / С. В. Романенко, Ю. В. Анищенко. – Томск: Изд-во Томского политехнического университета, 2016. – 211 с.
23. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Телеком, 2006. – 452 с.
24. Сироткин С. А. Экономическая оценка инвестиционных проектов: учебное пособие / С. А. Сироткин, Н. Р. Кельчевская. – М.: ИНФРА-М, 2020. – 274 с.

25. Скобцов В. Ю. Интеллектуальный анализ данных: генетические алгоритмы: учеб.-метод. пособие / В. Ю. Скобцов, Н. В. Лапицкая, С. Н. Нестеренков. – Минск: БГУИР, 2018. – 92 с.
26. Тер-Саркисов Р. М. Разработка и добыча трудноизвлекаемых запасов углеводородов / Р. М. Тер-Саркисов. – М.: Недра, 2005. – 407 с.
27. Цой Ю. Р. О поисковых способностях эволюционных алгоритмов / Ю. Р. Цой // Генетические алгоритмы. – 2006. Т. 8. – С. 28-37.
28. Ahmed, T. Reservoir Engineering Handbook / T. Ahmed. – Burlington: Gulf Professional Publishing, 2006. – 1360 p.
29. Albon, C. Machine Learning with Python Cookbook / C. Albon. – Sebastopol: O'Reilly Media, Inc., 2018. – 350 p.
30. Austin, E. H. Drilling Engineering Handbook / E. H. Austin. – Boston: Reidel Publishing Company, 1983. – 302 p.
31. Chambers, L. D. The Practical Handbook of Genetic Algorithms Applications / L. D. Chambers. – Boca Raton: Chapman & Hall/CRC, 2001. – 535 p.
32. Chin, W. Reservoir Engineering in Modern Oilfields. Vertical, Deviated, Horizontal and Multilateral Well Systems / W. Chin. – Beverly: Scrivener Publishing, 2016. – 246 p.
33. Chollet, F. Deep Learning with Python / F. Chollet. – N-Y: Manning Publications Co., 2021. – 478 p.
34. Dake, L. P. Fundamentals of reservoir engineering / L. P. Dake. – Amsterdam: Elsevier Science B.V., 1978. – 438 p.
35. Davies, D. Production Technology Manual / D. Davies. – Edinburgh: Heriot Watt University, 2016. – 816 p.
36. Fogel, D. B. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence / D. B. Fogel // Complexity. – 1998. – V. 11. – P. 28-30.
37. Ford, J. Production Technology Manual / J. Ford. – Edinburgh: Heriot Watt University, 2017. – 610 p.
38. Fundamentals of petroleum / The University of Texas at Austin Petroleum Extension Service. – Austin: Petex, 2017. – 679 p.

39. Genetic algorithm [Электронный ресурс] / Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Genetic_algorithm. Дата обращения: 02.04.2022.
40. Guidelines for the Economic Analysis of Projects / Asian Development Bank. – Manila: Asian Development Bank, 2017. – 156 p.
41. Guo, B. Petroleum Production Engineering / B. Guo, W. C. Lyons, A. Ghalambor. – Burlington: Gulf Professional Publishing, 2007. – 288 p.
42. Iba, H. Deep Neural Evolution. Deep Learning with Evolutionary Computation / H. Iba, N. Noman. – Singapore: Springer Nature Singapore Pte Ltd., 2020. – 438 p.
43. Joshi, P. Artificial Intelligence with Python / P. Joshi. – Birmingham: Packt Publishing, 2017. – 423 p.
44. Kapoor, A. Hands-On Artificial Intelligence for IoT / A. Kapoor. – Birmingham: Packt Publishing, 2019. – 340 p.
45. Ketlar, N. Deep Learning with Python: A Hands-on Introduction / N. Ketlar. – Bangalore: Apress, 2017. – 160 p.
46. Koza, J. R. Genetic Programming / Koza J. R., Poli R. – Stanford: Stanford University Press, 2005. – 356 p.
47. Luke, S. Essentials of Metaheuristics / S. Luke. – Fairfax: Lulu Publishing, 2013. – 261 p.
48. Melanie, M. An Introduction to Genetic Algorithms / M. Melanie. – Cambridge: The Massachusetts Institute of Technology Press, 1999. – 158 p.
49. Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs / Z. Michalewicz. – Charlotte: Springer, 1996. – 387 p.
50. Petroleum Economics Manual / Petroleum Economics Course Team. – Edinburgh: Heriot Watt University, 2016. – 556 p.
51. Petroleum Geoscience Manual / P. Corbett, G. Couples, A. Gardiner, H. Lever. Edinburgh: Heriot Watt University, 2017. – 640 p.
52. Raschka, S. Python: Deeper Insights into Machine Learning / S. Raschka, D. Julian. J. Hearty. – Birmingham: Packt Publishing, 2016. – 881 p.

53. Ravichandiran, S. Deep Reinforcement Learning with Python / S. Ravichandiran. – Birmingham: Packt Publishing, 2020. – 730 p.
54. Reservoir Engineering Manual / Reservoir Engineering Course Team. – Edinburgh: Heriot Watt University, 2018. – 814 p.
55. Review and Classification of Bio-inspired Algorithms and Their Applications / X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, H. Chizari. // Journal of Bionic Engineering. – 2020. – V. 17. – P. 611-631.
56. Terry, R. E. Applied petroleum reservoir engineering / R. E. Terry, J. B. Rogers. – Westford: Pearson Education, 2015. – 493 p.
57. tNavigator 21.4 User Manual / Rock Flow Dynamics, 2022. – 518 p.
58. Whitley, D. A Genetic Algorithm Tutorial / D. Whitley // Statistics and Computing. – 1998. – P. 1-40. DOI: 10.1007/BF00175354.
59. Wirsansky, E. Hands-On Genetic Algorithms with Python / E. Wirsansky. – Birmingham: Packt Publishing, 2020. – 320 p.
60. Xiao, P. Artificial Intelligence Programming with Python from Zero to Hero / P. Xiao. – Hoboken: John Wiley & Sons, Inc, 2022. – 682 p.

Приложение А
(справочное)

Application of a genetic algorithm to optimise the hydrocarbon field development system using a synthetic analogue as an example

Literature Review

Студент

Группа	ФИО	Подпись	Дата
2ТМ01	Новак Максим Викторович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Матвеев И.В.	к.ф.-м.н.		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Макаровских А.В.	-		

1 Genetic algorithms

Genetic algorithms are a concept related to information technology. Since most of the work is related to them, it is absolutely necessary to have an understanding of genetic algorithms in order to understand the work.

1.1 Introduction to genetic algorithms

Genetic algorithms are a family of search algorithms that are based on the evolution of living things occurring in nature [58].

Genetic algorithms use a simplified version of so-called Darwinian evolution, the basic principles of which can be formulated as [48]:

- The principle of variation. The genes of individuals belonging to a population may differ from individual to individual, and the individuals themselves may differ accordingly.

- The principle of inheritance. Certain traits are constantly passed on from generation to generation. Accordingly, descendants resemble their ancestors more than other individuals in the population.

- The principle of selection. More adapted individuals have a higher chance of surviving and of producing offspring.

Thus, evolution contributes to bringing more adapted individuals [43].

Here is an analogy between the evolution of living things and genetic algorithms [60]:

- The individual is a possible solution to the problem being solved.

- A population (a set of individuals) is a set of possible solutions to the problem being solved.

- A gene (a particular trait of an individual) is an input parameter inherent in the possible solution to the problem being solved.

- Genotype (the set of genes of an individual) is all the input parameters inherent to the possible solution to the problem being solved.

- Fitness (otherwise fitness function, the ability of an individual to survive and reproduce, genotype-dependent) is the value of the objective function to be maximised or minimised, as determined by the input parameters.

- Selection (choosing of individuals who participate in reproduction and pass on their genotype to their offspring) is choosing possible solutions whose parameters will be used to form new possible solutions.

- Crossover (the emergence of new individuals from existing individuals) is the use of existing possible solutions to form new possible solutions.

- Offspring (an individual produced by previous individuals, ancestors) is a possible solution derived from previous possible solutions.

- Ancestor (otherwise parent, an individual who participates in reproduction and produces a new individual, the offspring) is a possible solution from which new possible solutions are formed.

- Mutation (random change in the genotype of an individual) is a random change in the parameters of a possible solution.

- Reduction (decrease in the number of individuals in a population) is decrease in the number of possible solutions in a set of given possible solutions.

Thus, there are individuals that are a part of a population that have certain traits (a genotype made up of genes). Individuals strive to survive and reproduce, and the greater the chance of survival and reproduction, the greater the fitness of the individual, as determined by a process of selection. By reproducing, individuals produce new individuals (offspring) that inherit certain traits of the parental individuals (ancestors). Also, the offspring's traits can change randomly through the process of mutation. Since the chance of survival and reproduction is the highest for the fittest individuals, “good” genes are passed on to the offspring and the fitness of new generations must increase. This is the basis of the evolution of living organisms and genetic algorithms [33].

In what follows, when referring to terms of natural evolution, we will understand the meaning of these terms within genetic algorithms.

1.2 History of the development and use of genetic algorithms

In the section the history of the development and use of genetic algorithms, based on the sources, are briefly reviewed [45, 48].

The study of evolutionary systems began in the 1950s and 1960s by several scientists independently. It was assumed that these evolutionary systems could be used to optimise solutions to various engineering problems. As already mentioned, the concept was to evolve a certain population of candidate solutions for a problem, using operators inspired by the evolution of life.

In 1964, the German scientist Ingo Rechenberg introduced the concept of evolutionary strategies, a method he used to optimise the parameters of various devices (e.g., aircraft wings). This concept was further developed by the German scientist Hans-Paul Schwefel. It should be noted that the sphere of evolutionary strategies remains a sphere of active research, developing in parallel with genetic algorithms (although the two fields have recently begun to interact actively).

British scientists Vogel, Owens and Walsh at about the same time developed 'evolutionary programming', a method in which candidates for assigned tasks were represented as mechanisms that solve certain problems, which evolved by randomly changing their parameters and selecting the most adaptable ones. Evolutionary programming also remains an area of active research. Together, evolutionary strategies, evolutionary programming, and genetic algorithms form the basis of the sphere of evolutionary computing.

Other scientists working in the 1950s and 1960s developed optimisation and machine learning algorithms, also inspired by evolution. In addition, a number of evolutionary biologists used computers to simulate natural evolution in order to conduct experiments in evolutionary biology.

Genetic algorithms themselves were invented by John Holland in the 1960s and developed by him and his students and colleagues at the University of Michigan in the 1960s and 1970s. Unlike evolutionary strategies and evolutionary programming, Holland did not pursue the goal of creating algorithms to solve specific problems, but studied the phenomenon of adaptation itself, as it occurs in nature. Later, Holland moved on to the development of ways to transfer the mechanisms of natural adaptation to computer systems. As a result, Holland was able to present the genetic algorithm as a model for biological evolution in computer systems and put forward a theoretical framework for adaptation within genetic algorithms. Holland's genetic algorithms are a method of moving from one population of "chromosomes" (such as strings of ones and zeros, or "bits") to a new population by means of "natural selection" together with the genetics-inspired crossover, mutation, and inversion operators.

Holland's creation of such an algorithm was a major innovation. Moreover, Holland was the first to try to put "computational evolution" on a solid theoretical basis. This theoretical foundation was and still is the basis for almost all subsequent theoretical works on genetic algorithms.

Over the last few years, there has been widespread interaction between researchers studying different methods of evolutionary computation, and the boundaries between genetic algorithms, evolutionary strategies, evolutionary programming, and other evolutionary approaches have collapsed to some extent. Today, researchers often use the term "genetic algorithm" to describe all these concepts.

1.3 Application of genetic algorithms

Genetic algorithms are capable of providing solutions to a variety of optimisation, learning, and search problems. Moreover, genetic algorithms can cope with problems that cannot be solved using classical search and optimisation

algorithms, especially with problems with complex mathematical representations and a large number of parameters. Let's consider the main applications of genetic algorithms based on the sources [42, 47].

Genetic algorithms can be useful for two largely different purposes. The first purpose is the selection of parameters to optimise system performance. These systems can be of various kinds: gas distribution piping systems, traffic lights, commissaries, project allocation, scheduling, material handling and mixing, and so on. Such systems usually depend on input parameters on which decisions are based, chosen by the developer or operator of the system. The right or wrong choice of decision parameters will cause the system to perform better or worse, as measured by the achievement of some relevant goal. In real large systems, the interaction between parameters is usually not amenable to analytical treatment, and the researcher has to resort to appropriate search methods. Most of the published work on genetic algorithms focuses on using them to optimise systems, or at least to improve them by approaching an optimum. It should be noted that this paper uses genetic algorithms in this direction.

The second potential application of genetic algorithms is less discussed, but is also a very important and promising area. It lies in the area of testing and fitting quantitative models. Scientific investigation of a problem domain can be described as an iterative process. A descriptive model is built, data are collected and used to test the model. If discrepancies are found, the models are changed. The process is repeated until the problem is solved (if the genetic algorithm works correctly). In this case, in contrast to the use of genetic algorithms for maximising system performance, the search for parameters that minimise the discrepancy between the model and the real data takes place. Optimisation involves finding parameter values for the model that minimise the inconsistency.

Two main areas of application for genetic algorithms: system optimisation or quantitative model fitting have been identified. This can be compared to the difference between maximising system performance and minimising the discrepancy between the model and the observed data set. This distinction is actually

insignificant, as maximisation and minimisation can always be interchanged. Maximising the performance of an operating system is equivalent to minimising its lag from some optimum. Conversely, minimisation of a non-convex function is an equivalent to maximisation of a negative value of the correspondence function.

It is worth noting that genetic algorithms are already used in software products used in the oil and gas industry, primarily in geological and simulation modelling. For example, the tNavigator software package from Rock Flow Dynamics uses genetic algorithms for hydrodynamic model auto-adaptation, i.e., reducing inconsistency between historical and calculated parameters. It is noteworthy that in this software other machine learning algorithms can be used along with genetic algorithm. The genetic algorithms used can be different, their parameters can vary depending on the objectives pursued [57].

1.4 Differences from traditional algorithms

Genetic algorithms are quite different from traditional search and optimisation algorithms (e.g., gradient methods). Here are the main distinguishing features [19, 36]:

- Using a genetic representation of the solutions. The parameters of individuals are represented as chromosomes (which are certain coded values). This representation allows the use of evolutionary operations. However, it is worth realising that because of this representation there is some separation from the domain of the problem to be solved, i.e., the chromosomes must be fine-tuned to be interpretable within the framework of a particular problem.

- Exhibiting a probabilistic behavior. Traditional algorithms are often deterministic, whereas genetic algorithms are based on a probabilistic approach. Evolutionary operations performed on individuals are determined by probabilities. It is worth noting, however, that genetic algorithms are not purely probabilistic; the

probabilistic approach serves to guide individuals towards an increase in fitness, that is, to optimise the function.

- Utilizing the outcome of a fitness function. GAs aim to maximise the fitness of individuals. GAs do not use differentiation or integration (as many other traditional algorithms do), allowing GAs to be used in cases where differentiation or integration is too difficult or impossible.

- Maintaining a population of solutions. GAs do not just deal with individuals, but with populations of individuals. At any given time, an entire population (at least one) of individuals is stored in memory. Most often, traditional algorithms deal with only one possible solution, which iteratively changes.

1.5 Advantages and limitations of genetic algorithms

Genetic algorithms, like any other approach, have certain strengths and weaknesses [25, 59].

The merits of genetic algorithms:

- Handling problems with a complex mathematical representation. As mentioned earlier, genetic algorithms only need fitness functions for each individual to work, there is no need for derived or first-order functions. Accordingly, as mentioned earlier, GAs can be used for functions with a complex mathematical representation. Also, GAs can be used for functions with a large number of variables and different types of variables (e.g., discrete and continuous parameters).

- Handling problems that lack mathematical representation. Some problems being solved cannot be represented mathematically (at least directly). For example, when the fitness function represents the opinion of individuals. Also, GAs can be used even when the value of the function cannot be obtained directly, but the relative position of the fitness of individuals can be revealed.

- Support for parallelism and distributed processing. Most often, calculations in genetic algorithms can be easily paralleled, i.e. fitness for each individual can be

calculated separately, also evolutionary operators can be applied in parallel for individuals and pairs of individuals.

- Global optimisation capability. Often functions have several local maxima (minima) – see Figure 1.1. Most traditional algorithms cannot “escape” from a found local maximum (minimum). Genetic algorithms use population of individuals and probability-based operators, because of which local maxima (minima) can be overcome.

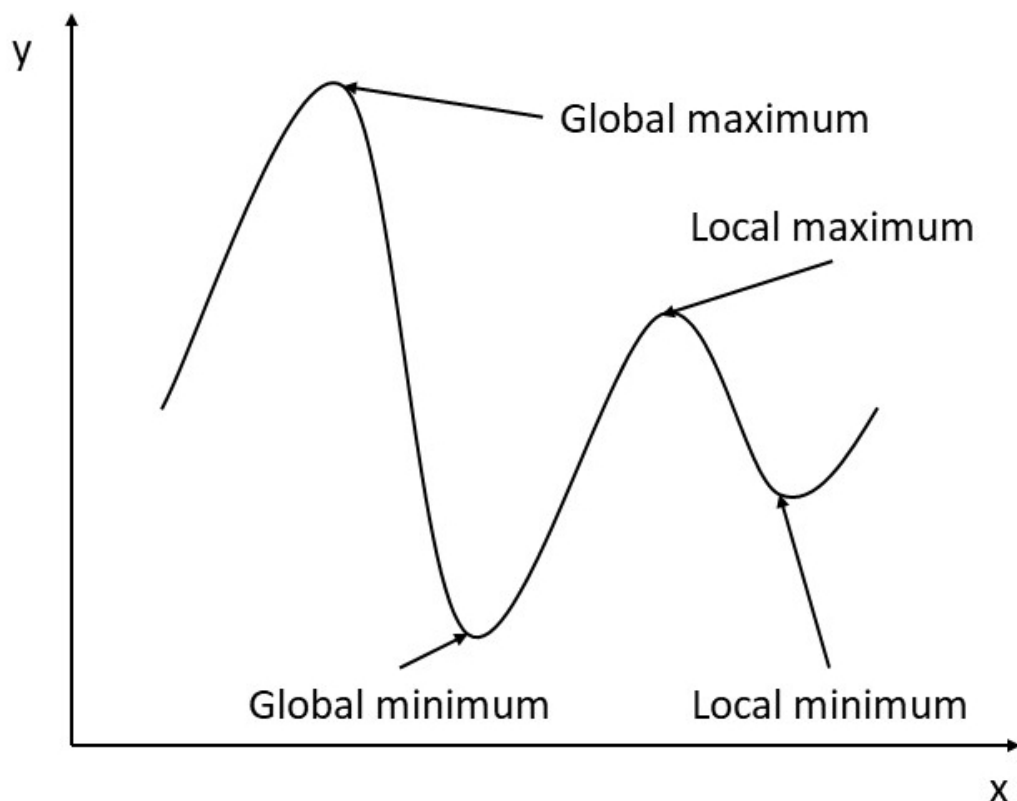


Figure 1.1 – Difference between global and local extrema of a function

- Resilience to noise. Sometimes the fitness function may produce different output values for the same input values (e.g., due to measurements error). If the overall trend is unchanged, genetic algorithms are able to deal with such noise.

- Suitability for continuous learning. Genetic algorithms are capable of adjusting to changing input and output conditions.

Disadvantages and limitations of genetic algorithms:

- Computationally-intensive operations. Because populations can be quite large and the fitness functions are quite complex (and must be calculated for each

individual), GA work can be costly in terms of the number of calculations and the time to reach the result.

- The need for hyperparameter tuning. Hyperparameters, such as the number of individuals in the population, mutation probabilities and so on, need to be tuned to the specific task and are difficult to scale.

- The need for special definitions. For each problem being solved, the type of representation of input parameters, the fitness functions, and the evolutionary operators will be different, so must be chosen for each specific problem. It is noteworthy that at the current level of GA development it is almost always possible to find a similar problem already solved using genetic algorithms.

- No guaranteed solution. If the hyperparameters are misconfigured or if there is an unfortunate match, genetic algorithms may be prone to premature convergence, i.e. finding local rather than global maxima. Therefore, the result can almost never be certain to become a correct solution. It is worth noting that these problems are common to all similar algorithms.

1.6 General scheme of genetic algorithms

All genetic algorithms have unique scheme, yet they all rely on one initial common scheme [23]:

- An initial population of a certain number of individuals is created.
- A fitness function value is calculated for each individual and assigned to it.
- Individuals who will participate in breeding are selected.
- Selected individuals are crossed and offspring is created.
- The offspring may be susceptible to mutation.
- The resulting individuals are added to the old population or form a completely new one.

- If new individuals have been added to the old population, the population is reduced, i.e. the number of individuals in the population is decreased to the original number (this step is not always performed).

- The algorithm's stop criterion is checked. If it is not met, the cycle starts again by evaluating the fitness function for each individual. If the criterion is met, the best individual is found, which is considered the solution to the problem.

The general flowchart of the GA is shown in Figure 1.2.

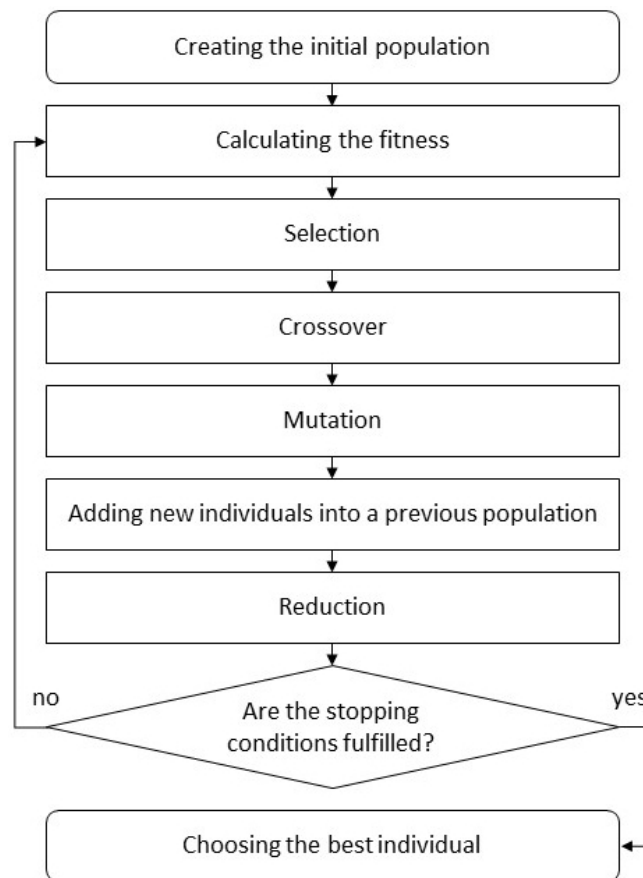


Figure 1.2 – General scheme of genetic algorithms

Below each step of the algorithm is briefly reviewed.

1.7 Creating the initial population

The initial population is a group of individuals whose parameters are generated randomly. The types of parameters can be different (integers from a

continuous distribution, decimals, binary numbers, and so on). At this stage, a hyperparameter such as population size is laid down [46].

It is worth noting that in order to obtain a good solution, the initial population must contain a sufficient number of individuals, distributed as evenly as possible over the search area. If the sample is too small or if the initial population is made up of too similar individuals, the GA may encounter premature convergence problems (although the mutation operator can cope with this problem). This is why the initial population must be checked before each GA run, or “similarity” constraints must be imposed on the initial population. As the establishment of the initial population is usually a process that does not require complex and time-consuming calculations, it can be repeated until an acceptable sample is obtained [46].

In some cases, an initial population can be created in two stages: the first stage creates an initial population uniformly covering the entire search area, the second stage selects sub-areas based on high values of the fitness function, and it is in these areas that new initial populations are created and the GA is launched. This approach often speeds up the finding of a solution, but it is not suitable for all problems (in some cases this approach can only give high values of the fitness function in local maxima, respectively, the subareas selected will exclude the global maximum, which may lead to premature convergence). Usually, this approach works well for problems where the desired solution is sufficiently well known [46].

1.8 Calculating the fitness function

A fitness function is a function of several variables to be optimised, i.e. maximised or minimised. The fitness function can be specified explicitly or implicitly [44].

In fact, the fitness function can be any expression or way of comparing individuals with each other. The only condition is that each individual (in the initial

and all subsequent generations) must have an intrinsic value of the fitness function [44].

In some cases, the calculation of the fitness function may take a considerable amount of time. The value of the fitness function of one individual is independent on the values of the fitness function of all other individuals. For this reason, the calculation of the fitness function can be carried out in parallel for each individual, which can significantly reduce the time required to calculate the fitness function for a generation and, consequently, speed up the algorithm and the result [44].

1.9 Selection

The selection function is the selection of individuals from the population who will then participate in reproduction. Again, that selection is probabilistic. For this reason, each individual in a generation has a chance of being selected, but the probability of selection depends on the fitness of the individual, i.e. the higher the fitness, the higher the probability of selection. It is worth noting that the chance of selection of individuals with a relatively low value of the fitness function is an important element of genetic algorithms, as such individuals may carry “useful” genes that can be used to solve the problem, otherwise, these genes may be lost [29].

There are different selection methods (depending on the problem being solved), here are the main ones [29]:

- Roulette wheel selection. This is the simplest and the most commonly used method of selection. The probability of selection of an individual is directly proportional to the fitness function. Selection using this method is similar to the use of a roulette wheel used in a casino. The angle of the roulette sector is proportional to the proportion of the fitness function value in the sum of the fitness function values for all individuals in the generation. A selection point is conventionally fixed, and the roulette wheel spins and stops at a random moment. Such roulette spins continue until the required number of individuals are selected for reproduction. Obviously, the same

individual may then be selected more than once. It should be noted that with an apparently higher value of the fitness function in one of the individuals, it may be selected a large number of times, which may lead to premature convergence. This method is difficult to use if there is the possibility of a negative value of the fitness function.

- Stochastic universal sampling. This method is similar to the roulette wheel method. The angle of the roulette sector is also proportional to the proportion of the fitness function value in the sum of the fitness function values for all individuals in the generation, but there are multiple sampling points evenly distributed throughout the roulette. The tape measure may be rotated several times or the number of selection points may correspond to the number of individuals to be selected. With this method, the population of individuals sampled is more heterogeneous. The chance of selection for individuals with a relatively low value of the fitness function is greatly increased, which may help to solve the problem of premature convergence. This method is difficult to use if there is the possibility of a negative value of the fitness function.

- Rank-based selection. The method is very similar to the roulette wheel selection method, but in this case the angle of the roulette wheel sector is not proportional to the value of the fitness function itself in the individual, but proportional to the rank. That is, the values of the fitness function are only needed to sort out all the individuals. Each individual is assigned a rank (from the number of individuals to one). Accordingly, the angle of the roulette sector is also proportional to the fraction of the rank in the sum of the ranks for all individuals in the generation. In this case, selection can take place according to the roulette method scheme or according to the stochastic universal sampling scheme. It should be noted that negative values of the fitness function are not a problem in this case.

- Fitness scaling. In order to reduce the outnumbering of individuals with high values of the fitness function, scaling of the fitness function and, accordingly, scaling the sector angle on the tape measure can be used. To carry out scaling, the scaling limits should be chosen, then the worst individual will correspond to the lower

scaling limit and the best individual will correspond to the upper scaling limit. This increases the chances of selecting individuals with a low value of the fitness function. In this case, the selection can take place using the roulette method or stochastic universal sampling scheme. It should be noted that the scaling limits will differ from task to task.

- Tournament selection. This method involves comparing several individuals. A certain number of individuals are randomly selected from the population, then the best one is selected for breeding. The higher the number of individuals selected, the higher the probability of selecting the best individual. This method is interesting in that it does not require the value of the fitness function itself, the only prerequisite being the ability to compare individuals with each other.

1.10 Crossover

Crossover is the process by which a certain number of individuals (usually two) create an individual with a mixed genotype of parent individuals. In some cases, probability of interbreeding is applied. In some (rare) cases, no interbreeding occurs and the ancestors pass into the new generation unchanged [49].

There are various methods of crossover (depending on the problem being solved), here are the main ones [52]:

- Single-point crossover. This method is used (primarily) for binary numbers. In this method, some space between two characters of a number is randomly chosen, and the left and right parts are swapped. In this case, we get two descendants that have the parameters of the two parent individuals.

- Multiple-point crossover. In this method, several gaps between the number signs are randomly selected and then exchanged between the parents in a certain way. In the classic version, two gaps are used, the genes between which are to be exchanged. If more points are used, the characters between them are exchanged. This method is also applicable (primarily) for binary numbers.

- Uniform crossover. In this method, the decision to exchange genes is made for each gene individually. Each gene has a probability (usually 50%) of remaining unchanged and a probability of being changed. The probability may also be greater or less than the standard 50%. The offspring can be formed by either selecting the parents' genes (then one offspring is formed) or by exchanging the parents' genes (then two offspring are formed). This method is suitable for any type of data. This method achieves greater diversity in the resulting generation, because the change occurs in a chromosomal manner.

- Ordered crossover. In some cases, it is necessary that a gene value is present in each individual exactly once, a similar method is used for such cases. In it, a certain gap is randomly selected from each parent, which is passed on to the offspring. The remaining gaps are then successively filled with the remaining values from the parent's genes.

- Blend crossover. The method is used primarily for real numbers from a continuous distribution. In this case, the value of the offspring parameter is selected randomly from the interval between the values of the parameter of the parental individuals. Sometimes a widening (narrowing) of the selection interval is also used. Expanding the interval encourages more “randomness” in value selection, but may also allow the diversity of offspring genes to be increased.

- Simulated binary crossover. The method is used primarily for real numbers from a continuous distribution. Single-point crossover is simulated. The values of the offspring parameter change so that the offspring represent the values of the parents that are distant (closer) from each other (the average value of the parameter for the parents) by a certain random percentage.

1.11 Mutation

Mutation is a very important component of genetic algorithms because it allows new genes to be produced as the algorithm runs, thus preventing premature

convergence. Mutation is performed on created individuals with a certain (usually low) probability, which is a hyperparameter of the algorithm. Too high a mutation probability can degrade the population and even turn the genetic algorithm into a random solution search. In some cases, the mutation probability increases from generation to generation, as convergence is usually observed in later generations [13].

There are various mutation methods (depending on the problem being solved), here are the main ones [55]:

- Flip bit mutation. When mutation is performed on binary numbers, it can be done by changing the values of one or more bits from one to zero or vice versa.

- Swap mutation. For binary and integer numbers mutation can be done by shuffling the parameters at certain positions. This method is suitable for ordered lists.

- Inversion mutation. In this method, a randomly determined portion of the parameters is inverted. This method is suitable for ordered lists.

- Scramble mutation. In this case, a randomly determined fraction of the parameters takes on a random order. This method is suitable for ordered lists.

- Real mutation. For real numbers, mutation can be used by replacing any parameters with random parameters. Since this approach carries the danger that the resulting individual may not be similar to an ancestor, mutation can be done using a distribution (e.g., normal) from the value of the ancestor parameter with a certain standard deviation. In such a case, the probability of selecting a value is the higher the closer it is to the value of the ancestor parameter.

1.12 Population expansion and reduction

There are four principal approaches to the expansion and reduction appearing in the process of population reproduction [53]:

- The individuals obtained by breeding are added to the preceding population. The reduction mechanism then works. The individuals with the lowest values of the

fitness function are removed from the population until the number of individuals in the population is equal to the original target.

- All of the individuals resulting from the interbreeding fall into the population, and reduction does not work. The number of individuals in the population increases with each generation. This method is convenient because the number of fitness function calculations is reduced, as its value is known for all preceding individuals.

- Only the new generation received is retained. The ancestral generation is deleted. The advantage of this method is that it saves the memory of the computing device.

- The latter method is based on elitism, i.e. a certain number or a certain proportion of the best individuals from the past generation are always passed on to the new generation. This ensures that the achievements of past generations are not lost.

It is also worth noting that it is possible to store fitness function values for certain parameter values. This prevents the fitness function from being calculated again in the event that a similar individual has appeared earlier [31].

1.13 Checking the stopping conditions and shutdown

The algorithm is terminated when the stopping condition is met. There are different stopping conditions, the main ones are given below [27]:

- Achieving the maximum number of fitness function calculations or the maximum number of generations. This method is used to reduce the running time of the algorithm or reduce the computational resources consumed, but can relatively often result in the algorithm ‘failing’ to find a global solution to the problem in question.

- No significant changes in the maximum value of the fitness function within a generation over several generations. To use this method, it is necessary to store the

maximum values of the fitness function in past generations. This approach maximises the probability of obtaining a global solution, but can be relatively time consuming and computationally intensive.

- A predetermined amount of time has elapsed. In some cases (when something needs to be optimised as quickly as possible), the time constraint is a more important criterion than finding a global solution.

- Reaching a certain amount of occupied device memory. The device memory may be full when storing information about the algorithm operations.

- The best individual occupies a certain proportion of all individuals in the population. In this approach, it is assumed that a better solution cannot be found due to the loss of diversity of individuals.

The individual with the highest value of the fitness function at the end of the algorithm is taken as the solution to the problem under study [27].

1.14 Conclusions on the section

In this section the basic principles of genetic algorithms used for optimisation were presented. The basic concepts used in genetic algorithms, history of their development and application, main applications and differences from traditional optimization algorithms, advantages and disadvantages, general scheme and main steps (including methods) of genetic algorithms were discussed.

It is worth noting that this section does not aim to cover all the theoretical background used in genetic algorithms, but aims to build the reader's understanding of all the work that follows.

Приложение Б

(справочное)

Код программы

Импорт пакетов

```
In [1]: import random
import statistics
import math
from matplotlib import pyplot as plt
```

Входные данные

```
In [2]: # Параметры ГА
razmer_populyacii = 20 # Количество особей в каждой популяции
veroyatnost_mutacii = 0.1 # Вероятность замены одного из потомков на случайно сгенерированного
uslovie_ostanovki = 10 # Через сколько поколений расчет остановится при неизменном максимальном полученном NPV
alpha = 0.5 # Доля интервала между параметрами при скрещиваниях, добавляемая с каждой стороны
kolichество_povtoreniy = 1 # Количество повторов всего цикла

# Длина ГС - все в метрах
min_dlina_GS = 0 # Минимальное значение длины ГС
max_dlina_GS = 1000 # Максимальное значение длины ГС
shag_dlina_GS = 100 # Шаг дискретизации длины ГС

# Направление ГС - все в градусах
min_napravlenie_GS = 0 # Минимальное значение направления ГС
max_napravlenie_GS = 180 # Максимальное значение направления ГС
shag_napravlenie_GS = 10 # Шаг дискретизации направления ГС

# Расстояние между скважинами - все в метрах
min_rasstoyanie_mezhdu_skvazhinami = 500 # Минимальное значение расстояния между скважинами
max_rasstoyanie_mezhdu_skvazhinami = 1000 # Максимальное значение расстояния между скважинами
shag_rasstoyanie_mezhdu_skvazhinami = 50 # Шаг дискретизации расстояния между скважинами

# Радиус скважины - все в сантиметрах
min_radius_skvazhiny = 10 # Минимальное значение радиуса скважины
max_radius_skvazhiny = 30 # Максимальное значение радиуса скважины
shag_radius_skvazhiny = 5 # Шаг дискретизации радиуса скважины

# Длина трещины ГРП - все в метрах
min_dlina_treshchiny_GRP = 0 # Минимальное значение длины трещины ГРП
max_dlina_treshchiny_GRP = 100 # Максимальное значение длины трещины ГРП
shag_dlina_treshchiny_GRP = 10 # Шаг дискретизации длины трещины ГРП

# Радиус обработки кислотой - все в дециметрах
min_radius_obrabotki_kislotoj = 0 # Минимальное значение радиуса обработки кислотой
max_radius_obrabotki_kislotoj = 50 # Максимальное значение радиуса обработки кислотой
shag_radius_obrabotki_kislotoj = 10 # Шаг дискретизации радиуса обработки кислотой

# Вывод результатов
raschet_vozmozhnyj_max = False # Нужно ли находить особь с максимально возможной NPV или взять из явно указанной ниже
vozmozhnyj_max = [{'dlina_GS': 1000, 'napravlenie_GS': 90, 'rasstoyanie_mezhdu_skvazhinami': 850, 'radius_skvazhiny': 30,
                    'dlina_treshchiny_GRP': 100, 'radius_obrabotki_kislotoj': 0, 'NPV': 773022902.5445597}]
grafiki_NPV = False # Нужно ли рисовать графики NPV от поколений для всех итераций
diagrammy = False # Нужно ли рисовать диаграммы
status = False # Нужно ли показывать, сколько процентов итераций прошло
rezultaty = False # Нужно ли выводить основные результаты
```

Словарь из списков всех возможных вариантов параметров

```
In [3]: vse = {'dlina_GS': [i for i in range(min_dlina_GS, max_dlina_GS + 1, shag_dlina_GS)],
              'napravlenie_GS': [i for i in range(min_napravlenie_GS, max_napravlenie_GS + 1, shag_napravlenie_GS)],
              'rasstoyanie_mezhdu_skvazhinami': [i for i in range(min_rasstoyanie_mezhdu_skvazhinami,
                                                                max_rasstoyanie_mezhdu_skvazhinami + 1,
                                                                shag_rasstoyanie_mezhdu_skvazhinami)],
              'radius_skvazhiny': [i for i in range(min_radius_skvazhiny, max_radius_skvazhiny + 1, shag_radius_skvazhiny)],
              'dlina_treshchiny_GRP': [i for i in range(min_dlina_treshchiny_GRP, max_dlina_treshchiny_GRP + 1,
                                                       shag_dlina_treshchiny_GRP)],
              'radius_obrabotki_kislotoj': [i for i in range(min_radius_obrabotki_kislotoj, max_radius_obrabotki_kislotoj + 1,
                                                           shag_radius_obrabotki_kislotoj)]}
```

Считаем и выводим количество возможных вариантов

```
In [4]: kolichество_vozmozhnyh_variantov = 1
for key in vse.keys():
    kolichество_vozmozhnyh_variantov *= len(vse[key])
print('Количество возможных вариантов CP:', kolichество_vozmozhnyh_variantov)
```

Количество возможных вариантов CP: 758670

Создание прототипа системы разработки

```
In [5]: # Прототип системы разработки с параметрами СР и рассчитанным NPV, а также вероятностями отбора для размножения
SR = {'dlina_GS': 0, 'napravlenie_GS': 0, 'rasstoyanie mezhdru skvazhinami': 0, 'radius_skvazhiny': 0,
      'dlina_treshchiny_GRP': 0, 'radius_obraботki_kislotoj': 0, 'NPV': 0, 'rang': 0, 'veroyatnost_otbora': 0}
```

Создание файлов лога

```
In [6]: # avg_NPV и max_NPV являются списками средних и максимальных значений NPV в пополениях
# Log является списком всех поколений
# Индекс в списке соответствует поколению, начиная с исходного - нулевого
avg_NPV = []
max_NPV = []
log = []
```

Функция расчета NPV

```
In [7]: def raschet_NPV(populyaciya):
        spisok_NPV = []
        for osob in populyaciya:

            # Расчет конечного количества скважин
            def kolichestvo_skvazhin(rasstoyanie):
                ploshchad_mestorozhdeniya = 100000000 # В метрах квадратных
                return int((ploshchad_mestorozhdeniya ** 0.5 // rasstoyanie) ** 2)

            obshchee_kolichestvo_skvazhin = kolichestvo_skvazhin(osob['rasstoyanie mezhdru skvazhinami'])

            nachalnyj_debit = 10000 # В стандартных кубических метрах в месяц в базовом случае
            # Действие длины ГС
            nachalnyj_debit = nachalnyj_debit * (1 + 0.2 * (osob['dlina_GS'] - min_dlina_GS) / (max_dlina_GS - min_dlina_GS))
            # Действие направления ГС
            nachalnyj_debit = nachalnyj_debit * (1 + 0.05 * math.sin(math.radians(osob['napravlenie_GS'])))
            # Действие расстояния между скважинами и радиуса скважины
            nachalnyj_debit = nachalnyj_debit / math.log(50 * osob['rasstoyanie mezhdru skvazhinami'] / osob['radius_skvazhiny'],
                math.e)

            # Действие длины трещины ГРП
            nachalnyj_debit = nachalnyj_debit * (1 + 0.1 * (osob['dlina_treshchiny_GRP'] - min_dlina_treshchiny_GRP) /
                (max_dlina_treshchiny_GRP - min_dlina_treshchiny_GRP))

            # Действие радиуса обработки кислотой
            nachalnyj_debit = nachalnyj_debit * (1 + 0.01 * (osob['radius_obraботki_kislotoj'] - min_radius_obraботki_kislotoj) /
                (max_radius_obraботki_kislotoj - min_radius_obraботki_kislotoj))

            skorost_padeniya = 0.01 # Безразмерный коэффициент в степени экспоненты в базовом случае
            # Действие длины ГС
            skorost_padeniya = skorost_padeniya * (1 + 0.2 * (osob['dlina_GS'] - min_dlina_GS) / (max_dlina_GS - min_dlina_GS))
            # Действие направления ГС
            skorost_padeniya = skorost_padeniya
            # Действие расстояния между скважинами
            skorost_padeniya = skorost_padeniya * (1 + 1.5 * (obshchee_kolichestvo_skvazhin
                - kolichestvo_skvazhin(max_rasstoyanie mezhdru skvazhinami))
                / (kolichestvo_skvazhin(min_rasstoyanie mezhdru skvazhinami)
                - kolichestvo_skvazhin(max_rasstoyanie mezhdru skvazhinami)))

            # Действие радиуса скважины
            skorost_padeniya = skorost_padeniya * (1 + 0.1 * (osob['radius_skvazhiny'] - min_radius_skvazhiny) /
                (max_radius_skvazhiny - min_radius_skvazhiny))

            # Действие длины трещины ГРП
            skorost_padeniya = skorost_padeniya * (1 + 0.1 * (osob['dlina_treshchiny_GRP'] - min_dlina_treshchiny_GRP) /
                (max_dlina_treshchiny_GRP - min_dlina_treshchiny_GRP))

            # Действие радиуса обработки кислотой
            skorost_padeniya = skorost_padeniya * (1 + 0.05 * (osob['radius_obraботki_kislotoj'] - min_radius_obraботki_kislotoj) /
                (max_radius_obraботki_kislotoj - min_radius_obraботki_kislotoj))

            # Задаем функция дебита в стандартных кубических метрах в месяц от времени в месяцах
            def debit_skvazhiny(t):
                return nachalnyj_debit / math.exp(t * skorost_padeniya)

            CAPEX = 15000000 # Базовый CAPEX строительства одной скважины и связанных расходов в долларах
            # Действие длины ГС
            CAPEX = CAPEX + 500 * osob['dlina_GS']
            # Действие направления ГС
            CAPEX = CAPEX
            # Действие расстояния между скважинами
            CAPEX = CAPEX
            # Действие радиуса скважины
            CAPEX = CAPEX * (1 + 0.1 * (osob['radius_skvazhiny'] - min_radius_skvazhiny) /
                (max_radius_skvazhiny - min_radius_skvazhiny))
            # Действие длины трещины ГРП
            CAPEX = CAPEX + 2000 * osob['dlina_treshchiny_GRP']
            # Действие радиуса обработки кислотой
            stoimost_kisloty = 100 # Стоимость кислоты в долларах за стандартный кубический метр
            moshchnost_plasta = 20 # Мощность пласта в метрах
            CAPEX = CAPEX + stoimost_kisloty * moshchnost_plasta * math.pi * (osob['radius_obraботki_kislotoj'] / 10) ** 2

            OPEX = 1000 # Базовый OPEX обслуживания одной скважины в долларах в месяц
            # Действие длины ГС
            OPEX = OPEX * (1 + 0.2 * (osob['dlina_GS'] - min_dlina_GS) / (max_dlina_GS - min_dlina_GS))
            # Действие направления ГС
            OPEX = OPEX
```

```

# Действие расстояния между скважинами
OPEX = OPEX
# Действие радиуса скважины
OPEX = OPEX * (1 + 0.1 * (osob['radius_skvazhiny'] - min_radius_skvazhiny) /
               (max_radius_skvazhiny - min_radius_skvazhiny))
# Действие длины трещины ГРП
OPEX = OPEX * (1 + 0.5 * (osob['dlina_treshchiny_GRP'] - min_dlina_treshchiny_GRP)
               / (max_dlina_treshchiny_GRP - min_dlina_treshchiny_GRP))
# Действие радиуса обработки кислотой
OPEX = OPEX * (1 + 0.1 * (osob['radius_obrabotki_kislotoj'] - min_radius_obrabotki_kislotoj)
               / (max_radius_obrabotki_kislotoj - min_radius_obrabotki_kislotoj))

# Параметры для расчетов
stoimost_nefti = 90 # Долларов за баррель с учетом удельных затрат
koefficient = 6.29 # Перевод кубических метров в баррели
nalogovaya_stavka = 0.5 # Какая доля прибыли идет в налоги
stavka_diskontirovaniya = 0.15 # Безразмерная ставка дисконтирования

# Создаем списки нужных параметров
# Дебит каждой скважины ежемесячно в стандартных кубических метрах в месяц
debit_po_skvazhinam = [0] * obshchee_kolichestvo_skvazhin
OPEX_po_skvazhinam = [0] * obshchee_kolichestvo_skvazhin # OPEX каждой скважины ежемесячно в долларах
vremya_po_skvazhinam = [-1] * obshchee_kolichestvo_skvazhin # Относительное время каждой скважины в месяцах
vyruchka_po_mesyacam = [] # Список выручек по месяцам
OPEX_po_mesyacam = [] # Список OPEX по месяцам

time = 0 # Время в месяцах

# Пока выручка больше операционных затрат, добываем нефть
# Рассчитывается дебит и OPEX каждой скважины в каждый момент времени
# Получаем дебит и OPEX всего месторождения в каждый момент времени
while True:
    if sum(debit_po_skvazhinam) * koefficient * stoimost_nefti >= sum(OPEX_po_skvazhinam):
        if time < obshchee_kolichestvo_skvazhin:
            for i in range(time + 1):
                vremya_po_skvazhinam[i] += 1
            for num in range(obshchee_kolichestvo_skvazhin):
                if vremya_po_skvazhinam[num] > -1:
                    debit_po_skvazhinam[num] = debit_skvazhiny(vremya_po_skvazhinam[num])
                    OPEX_po_skvazhinam[num] = OPEX
            else:
                for i in range(obshchee_kolichestvo_skvazhin):
                    vremya_po_skvazhinam[i] += 1
                    debit_po_skvazhinam[i] = debit_skvazhiny(vremya_po_skvazhinam[i])
                    OPEX_po_skvazhinam[i] = OPEX
                vyruchka_po_mesyacam.append(sum(debit_po_skvazhinam) * koefficient * stoimost_nefti)
                OPEX_po_mesyacam.append(sum(OPEX_po_skvazhinam))
                time += 1
        else:
            break

# Удаляем последний месяц - месяц, в который OPEX превышает выручку
vyruchka_po_mesyacam = vyruchka_po_mesyacam[:-1]
OPEX_po_mesyacam = OPEX_po_mesyacam[:-1]

polnye_gody = (time - 1) // 12 # Количество полных лет разработки

vyruchka_po_godam = [0] * (polnye_gody + 1) # Список выручек по годам
OPEX_po_godam = [0] * (polnye_gody + 1) # Список OPEX по годам
# Суммируем дебиты и OPEX по годам
for god in range(polnye_gody):
    vyruchka_po_godam[god] = sum(vyruchka_po_mesyacam[0 + 12 * god:12 + 12 * god])
    OPEX_po_godam[god] = sum(OPEX_po_mesyacam[0 + 12 * god:12 + 12 * god])
    vyruchka_po_godam[-1] = sum(vyruchka_po_mesyacam[12 * polnye_gody:])
    OPEX_po_godam[-1] = sum(OPEX_po_mesyacam[12 * polnye_gody:])

# Находим CAPEX по годам
CAPEX_po_godam = [0] * (polnye_gody + 1) # Список CAPEX по годам
for god in range(obshchee_kolichestvo_skvazhin // 12):
    CAPEX_po_godam[god] = 12 * CAPEX
CAPEX_po_godam[obshchee_kolichestvo_skvazhin // 12] = obshchee_kolichestvo_skvazhin * 12 * CAPEX

denezhnyj_potok = [0] * (polnye_gody + 1) # Список денежных потоков (выручка минус CAPEX, OPEX и налоги) по годам
for god in range(polnye_gody + 1):
    denezhnyj_potok[god] = (vyruchka_po_godam[god] - OPEX_po_godam[god] - CAPEX_po_godam[god]) * (1 - nalogovaya_stavka)

diskontirovannyj_denezhnyj_potok = [0] * (polnye_gody + 1) # Список дисконтированных денежных потоков
for god in range(polnye_gody + 1):
    diskontirovannyj_denezhnyj_potok[god] = denezhnyj_potok[god] / (1 + stavka_diskontirovaniya) ** god

nakoplennyj_diskontirovannyj_denezhnyj_potok = [0] * (polnye_gody + 1)
# Список накопленных дисконтированных денежных потоков
for god in range(polnye_gody + 1):
    nakoplennyj_diskontirovannyj_denezhnyj_potok[god] = sum(diskontirovannyj_denezhnyj_potok[:god + 1])

osob['NPV'] = nakoplennyj_diskontirovannyj_denezhnyj_potok[-1]
spisok_NPV.append(osob['NPV'])

avg_NPV.append(sum(spisok_NPV) / len(spisok_NPV))
max_NPV.append(max(spisok_NPV))
log.append(populyaciya)

```

Расчет максимально возможного NPV

```
In [8]: if raschet_vozmozhnyj_max:
# Создаем список из одного словаря (нужно для входа на функцию расчета NPV) со значением NPV минус бесконечность
vozmozhnyj_max = [{'NPV': float('-inf')}]

# Для визуализации прогресса
procent = 0.1
nomer_rascheta = 1

# Перебираем все возможные (дискретно) CP и считаем для них NPV, находим максимально возможный
for dg in vse['dlina_GS']:
    for ng in vse['napravlenie_GS']:
        for rms in vse['rasstoyanie_mezhdu_skvazhinami']:
            for rs in vse['radius_skvazhiny']:
                for dtg in vse['dlina_treshchiny_GRP']:
                    for rok in vse['radius_obraботki_kislotoj']:
                        variant_SR = [{'dlina_GS': dg,
                                        'napravlenie_GS': ng,
                                        'rasstoyanie_mezhdu_skvazhinami': rms,
                                        'radius_skvazhiny': rs,
                                        'dlina_treshchiny_GRP': dtg,
                                        'radius_obraботki_kislotoj': rok}]
                        raschet_NPV(variant_SR)
                        if variant_SR[0]['NPV'] > vozmozhnyj_max[0]['NPV']:
                            vozmozhnyj_max = variant_SR

# Для визуализации прогресса
if kolichество_vozmozhnyh_variantov * procent / 100 <= nomer_rascheta:
    print(procent, '%', sep='')
    procent += 0.1
    nomer_rascheta += 1

# Зануляем логи для экономии памяти
avg_NPV = []
max_NPV = []
log = []

print('Максимально хорошая особь:', vozmozhnyj_max)
```

Функция создания исходной популяции

```
In [9]: def sozdanie_iskhodnoj_populyacii():
global iskhodnaya_populyaciya
iskhodnaya_populyaciya = [] # Создаем пустой список, в который будем добавлять новые особи
# Добавляем в созданный ранее список CP с случайными параметрами и считаем для них NPV
for _ in range(0, razmer_populyacii):
    osob = {}
    for key in vse.keys():
        osob[key] = random.choice(vse[key])
    iskhodnaya_populyaciya.append(osob)
    raschet_NPV(iskhodnaya_populyaciya)
```

Функция реализации генетического алгоритма

```
In [10]: def geneticheskij_algoritm(populyaciya):

# Определение вероятности отбора методом ранжирования
spisok_NPV = [] # Создаем список всех NPV для нахождения ранга
for osob in populyaciya:
    spisok_NPV.append(osob['NPV'])
sortirovannyj_spisok_NPV = sorted(spisok_NPV) # Сортируем список всех NPV для нахождения ранга
for osob in populyaciya:
    osob['rang'] = sortirovannyj_spisok_NPV.index(osob['NPV']) + 1 # Находим ранг для определения вероятности отбора
spisok_rangov = [] # Создаем список рангов для нахождения суммы рангов для определения вероятности отбора
for osob in populyaciya:
    spisok_rangov.append(osob['rang'])
summa_rangov = sum(spisok_rangov)
for osob in populyaciya:
    osob['veroyatnost_otbora'] = osob['rang'] / summa_rangov

# Создание родительских пар
spisok_veroyatnost_otbora = [] # Создаем список вероятностей отбора для выбора родителей с учетом весов
for osob in populyaciya:
    spisok_veroyatnost_otbora.append(osob['veroyatnost_otbora'])
roditel'skie_pary = []
# Пара из двух лучших особей всегда участвует в размножении
for osob in populyaciya:
    if osob['rang'] == max(spisok_rangov):
        roditel'skie_pary.append([osob, osob])
        break

# Создаем родительские пары до тех пор, пока их количество не станет равно размеру популяции
while len(roditel'skie_pary) < razmer_populyacii:
    vremennyj_spisok = [] # Во временный список добавляются два родителя, то есть формируется пара
    vremennyj_spisok.append(random.choices(populyaciya, spisok_veroyatnost_otbora)[0])
```

```

vremennyj_spisok.append(random.choices(populyaciya, spisok_veroyatnost_otbora)[0])
# Сформированная пара действительно участвует в размножении только если такой пары еще нет
if vremennyj_spisok not in roditelskie_pary and reversed(vremennyj_spisok) not in roditelskie_pary:
    roditelskie_pary.append(vremennyj_spisok)

# Скрещивание
novaya_populyaciya = []
for para in roditelskie_pary: # Новое значение выбирается из интервала значений родителей
    potomok = {}
    for key in vse.keys():
        min_v_pare = min(para[0][key], para[1][key])
        max_v_pare = max(para[0][key], para[1][key])
        interval = vse[key].index(max_v_pare) + 1 - vse[key].index(min_v_pare)
        if vse[key].index(min_v_pare) - int(alpha * interval) < 0:
            l_gran = 0
        else:
            l_gran = vse[key].index(min_v_pare) - int(alpha * interval)
        if vse[key].index(max_v_pare) + int(alpha * interval) > len(vse[key]) + 1:
            p_gran = len(vse[key]) + 1
        else:
            p_gran = vse[key].index(max_v_pare) + int(alpha * interval) + 1
        potomok[key] = random.choice(vse[key][l_gran:p_gran])
    novaya_populyaciya.append(potomok)

# Мутация
for osob in novaya_populyaciya:
    if random.random() < veroyatnost_mutacii: # С определенной вероятностью новая особь меняется на случайно сгенерированную
        osob = {}
        for key in vse.keys():
            osob[key] = random.choice(vse[key])

# Расчет NPV особей в новой популяции
raschet_NPV(novaya_populyaciya)

while len(max_NPV) < uslovie_ostanovki:
    geneticheskij_algoritm(novaya_populyaciya)
while len(set(max_NPV[-uslovie_ostanovki:1])) != 1:
    geneticheskij_algoritm(novaya_populyaciya)

```

Метод случайных параметров

```

In [11]: def metod_sluchajnyh_parametrov(kolichestvo_pokolenij):

    zapis_itogov = [] # Используется для дальнейшего получения графиков

    sluchajnyj_max = float('-inf') # Создаем переменную со значением минус бесконечность для поиска максимума

    # Создаем случайную особь столько раз, сколько расчетов NPV проводили
    # Рассчитываем NPV и проверяем, выше ли оно, чем уже встреченное NPV, если да, то считаем его максимальным
    for _ in range(0, razmer_populyacii * kolichestvo_pokolenij):
        sluchajnaya_osob = {}
        for key in vse.keys():
            sluchajnaya_osob[key] = random.choice(vse[key])
        raschet_NPV(sluchajnaya_osob)
        if sluchajnaya_osob[0]['NPV'] > sluchajnyj_max:
            sluchajnyj_max = sluchajnaya_osob[0]['NPV']

    # Переменные, показывающие то, сколько расчетов NPV было проведено в генетическом алгоритме всего и до завершения алгоритма
    provedeno_raschetov = razmer_populyacii * kolichestvo_pokolenij
    raschetov_do_maksimuma = razmer_populyacii * (kolichestvo_pokolenij - (uslovie_ostanovki - 1))

    zapis_itogov.append(sluchajnyj_max)
    zapis_itogov.append(provedeno_raschetov)
    zapis_itogov.append(raschetov_do_maksimuma)

    return zapis_itogov

```

Функция запуска на многократный расчет

```

In [12]: def mnogokratnye_povtoreniya():

    # Объявляем, что будем менять глобальные логи
    global log
    global avg_NPV
    global max_NPV

    spisok_kolichestv_pokolenij = []
    spisok_sluchajnyj_max = []
    spisok_provedeno_raschetov = []
    spisok_raschetov_do_maksimuma = []
    spisok_maksimumov_GA = []

    for _ in range(kolichestvo_povtoreniya):
        sozdanie_iskhodnoj_populyacii() # Создаем исходную популяцию
        geneticheskij_algoritm(iskhodnaya_populyaciya) # Запускаем генетический алгоритм

        kolichestvo_pokolenij = len(log) # Записываем количество поколений для метода случайных параметров
        spisok_kolichestv_pokolenij.append(kolichestvo_pokolenij) # Добавляем количество поколений в этой итерации в список

```

```

spisok_maksimumov_GA.append(max_NPV[-1]) # Добавлен в список максимумов ГА результат ГА

if status:
    print(100 * len(spisok_kolichestv_pokolenij) / kolichestvo_povtorenij, '%', sep='')

# Рисуем графики NPV от поколений для всех итераций, если нужно
if grafiki_NPV:
    plt.xlabel('Поколение')
    plt.ylabel('NPV')
    plt.xlim(0, len(max_NPV) - 1)
    plt.grid()
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.plot(range(len(max_NPV)), avg_NPV, label='Среднее')
    plt.plot(range(len(max_NPV)), max_NPV, label='Максимум')
    plt.plot(range(len(max_NPV)), [vozmozhnyj_max[0]['NPV']] * len(max_NPV), label='Абсолютный максимум')
    plt.legend(bbox_to_anchor=(1, 0.6))
    plt.show()

dlya_rezultatov = metod_sluchajnyh_parametrov(kolichestvo_pokolenij) # Запускаем метод случайных параметров
spisok_sluchajnyj_max.append(dlya_rezultatov[0]) # Записываем случайный максимум в итерации
spisok_provedeno_raschetov.append(dlya_rezultatov[1]) # Записываем общее количество расчетов в итерации
spisok_raschetov_do_maksimuma.append(dlya_rezultatov[2]) # Записываем количество расчетов до результата ГА в итерации

# Зануляем логи
log = []
avg_NPV = []
max_NPV = []

if diagrammy:
    # Рисуем диаграмму количества поколений в итерации
    plt.xlabel('Итерация')
    plt.ylabel('Количество поколений')
    plt.grid(zorder=0)
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.bar(range(1, len(spisok_kolichestv_pokolenij) + 1), spisok_kolichestv_pokolenij, zorder=10)
    plt.show()

    # Рисуем диаграмму количества расчетов
    plt.xlabel('Итерация')
    plt.ylabel('Количество расчетов')
    plt.grid(zorder=0)
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.bar([i - 0.2 for i in range(1, len(spisok_provedeno_raschetov) + 1)], spisok_provedeno_raschetov, zorder=10,
            width = 0.4, label='Общее')
    plt.bar([i + 0.2 for i in range(1, len(spisok_raschetov_do_maksimuma) + 1)], spisok_raschetov_do_maksimuma, zorder=10,
            width = 0.4, label='До результата')
    plt.legend(bbox_to_anchor=(1, 0.6))
    plt.show()

    # Рисуем диаграмму долей от возможного количества расчетов
    dolya_vozmozhnyh_iskhodov = []
    for kolvo_raschetov in spisok_provedeno_raschetov:
        dolya_vozmozhnyh_iskhodov.append(kolvo_raschetov / kolichestvo_vozmozhnyh_variantov)
    plt.xlabel('Итерация')
    plt.ylabel('Доля от возможных вариантов')
    plt.grid(zorder=0)
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.bar(range(1, len(dolya_vozmozhnyh_iskhodov) + 1), dolya_vozmozhnyh_iskhodov, zorder=10)
    plt.show()

    # Рисуем диаграмму максимумов NPV в двух подходах
    plt.xlabel('Итерация')
    plt.ylabel('NPV')
    plt.grid(zorder=0)
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.bar([i - 0.3 for i in range(1, len(spisok_sluchajnyj_max) + 1)], spisok_sluchajnyj_max, zorder=10,
            width = 0.3, label='Случайные параметры')
    plt.bar([i for i in range(1, len(spisok_maksimumov_GA) + 1)], spisok_maksimumov_GA, zorder=10,
            width = 0.3, label='Генетический алгоритм')
    plt.bar([i + 0.3 for i in range(1, kolichestvo_povtorenij + 1)], [vozmozhnyj_max[0]['NPV']] * kolichestvo_povtorenij,
            zorder=10, width = 0.3, label='Абсолютный максимум')
    plt.legend(bbox_to_anchor=(1, 0.6))
    plt.show()

if rezultaty:
    print('Среднее количество поколений в итерации:', sum(spisok_kolichestv_pokolenij) / len(spisok_kolichestv_pokolenij))
    print('Среднее общее количество расчетов в итерации:',
          sum(spisok_provedeno_raschetov) / len(spisok_provedeno_raschetov))
    print('Среднее количество расчетов в итерации до результата:',
          sum(spisok_raschetov_do_maksimuma) / len(spisok_raschetov_do_maksimuma))
    print('Средняя доля от всех возможных вариантов:', sum(dolya_vozmozhnyh_iskhodov) / len(dolya_vozmozhnyh_iskhodov))
    print('Медианный максимум при случайных параметрах:', statistics.median(spisok_sluchajnyj_max))
    print('Медианный максимум при генетическом алгоритме:', statistics.median(spisok_maksimumov_GA))
    print('Генетический алгоритм достиг максимально возможного NPV в ',
          100 * spisok_maksimumov_GA.count(vozmozhnyj_max[0]['NPV']) / kolichestvo_povtorenij, '% случаев', sep='')
    counter = 0
    for i in range(kolichestvo_povtorenij):

```

```
if spisok_maksimov_GA[i] >= spisok_sluchajnyj_max[i]:
    counter += 1
print('Генетический алгоритм справился лучше метода случайных параметров в ', 100 * counter / kolichestvo_povtorenij,
      '% случаев', sep='')
```

Запуск многократных повторений

In [13]: `mnogokratnye_povtoreniya()`