Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:
Первый проректор
Т. Р. Змызгова
«02» сентября 2022 г.

Рабочая программа учебной дисциплины

СТРУКТУРЫ И АЛГОРИТМЫ ОБРАБОТКИ ДАННЫХ

образовательной программы высшего образования –
программы магистратуры

09.04.04 Программная инженерия

направленность
**Методы и алгоритмы интеллектуальной обработки данных
в информационно-вычислительных системах**

формы обучения – очная

Курган 2022

Рабочая программа дисциплины «Структуры и алгоритмы обработки данных» составлена в соответствии с учебными планами программы магистратуры Программная инженерия (Методы и алгоритмы интеллектуальной обработки данных в информационно-вычислительных системах) очной формы обучения, утвержденными «30» августа 2022 г.

Рабочая программа одобрена на заседании кафедры Программного обеспечения автоматизированных систем «1» сентября 2022 года, протокол № 1.

Рабочую программу составил:

Доцент кафедры
«Программное обеспечение
автоматизированных систем»
к.т.н., доцент ________________ А.М. Семахин

Заведующий кафедрой
«Программное обеспечение
автоматизированных систем»,
руководитель магистерской программой
к.т.н., доцент ________________ В. К. Волк

Согласовано:

Начальник
Управления
образовательной деятельности ________________ И. В. Григоренко

Специалист
по учебно-методической работе
Учебно-методического отдела ________________ Г.В. Казанкова

– базовых форм и механизмов генетической изменчивости организмов, законов и принципов популяционной генетики и эволюционной изменчивости;

– математических моделей процесса эволюции и стратегий генетического поиска;

– базовых принципов и основных подходов к построению совместных схем локального и генетического поиска оптимальных решений;

– архитектуры и стратегии генетического поиска оптимальных решений.

2) практическое освоение:

– среды программирования на языке Python, библиотек NumPy, TensorFlow;

– среды программирования Microsoft Visual Studio Community 2019, языков программирования Visual C++, VisualC#;

– методики решения задач с применением методов эволюционной оптимизации и оптимизации машинного обучения.

Компетенции, формируемые в результате освоения дисциплины:

– способность разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК-2);

– способность разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем (ОПК-5).

В результате освоения дисциплины обучающийся должен демонстрировать следующие ***результаты обучения***:

Должен знать:

– современные интеллектуальные технологии, алгоритмы и методы решения профессиональных задач (ОПК 2);

– методику разработки и модернизации программного и аппаратного обеспечения информационных и автоматизированных систем (ОПК 5).

Должен уметь:

– разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК 2);

– разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем (ОПК 5).

Должен владеть:

– навыками разработки оригинальных алгоритмов и программных средств, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК 2);

– навыками разработки и модернизации программного и аппаратного обеспечения информационных и автоматизированных систем (ОПК 5).

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

Разделы дисциплины		Часов контактной работы с преподавателем	
		Очная форма обучения	
№	Наименование	Лекции	Лабораторные занятия
1	Введение в оптимизацию	2	2
2	Градиентные методы спуска	2	2
3	Сопряжённый градиентный метод	2	2
4	Стохастический градиентный метод	2	2
	Рубежный контроль №1	-	**2**
5	Генетические алгоритмы	4	4
6	Метод симуляции восстановления	4	4
7	Алгоритм оптимизации на основе муравьиной кучи	4	2
8	Алгоритм оптимизации на основе роя частиц	4	2
	Рубежный контроль №2	-	**2**
Всего по дисциплине:		**24**	**24**

4.2 Содержание лекционных занятий

Наименование и содержание лекции	Часов контактной работы с преподавателем
	Очная форма
Раздел №1. Градиентные методы оптимизации	
Лекция 1. Введение в оптимизацию Цели и задачи изучения дисциплины; взаимосвязи с другими дисциплинами учебных планов, обзор рабочей программы и учебно-	**2**

методических материалов. Понятие о задачах оптимизации. Постановка оптимизационной задачи. Виды оптимизаций: численная оптимизация. неограниченная оптимизация. ограниченная оптимизация. многокритериальная оптимизация. комбинаторная оптимизация.	
Лекция 2. Градиентные методы спуска Градиентный метод спуска с постоянным шагом. Формальное описание алгоритма. Этапы градиентного метода. Градиент. Антиградиент. Свойство антиградиента. Теоремы. Следствия. Сходимость градиентных методов. Анализ метода градиентного спуска. Пример. Преимущества и недостатки. Область применения. Метод наискорейшего градиентного спуска. Постановка задачи. Стратегия решения задачи. Алгоритм метода наискорейшего градиентного спуска. Геометрическая интерпретация. Оценка скорости сходимости. Замечания. Пример. Преимущества и недостатки. Область применения.	**2**
Лекция 3. Сопряжённый градиентный метод Понятие сопряжённых градиентов и их свойства. Постановка задачи. Стратегия решения задачи. Алгоритм метода сопряжённых градиентов. Замечания. Пример. Преимущества и недостатки. Область применения.	**2**
Лекция 4. Стохастический градиентный метод Классический стохастический алгоритм градиентного спуска (stochastic gradient descent, SGD). Сходимость метода стохастического градиентного спуска. Пример. Области применения Преимущества и недостатки. Расширения и варианты: классический метод моментов (classical moment, CM), ускоренный градиентный метод Нестерова (Nesterov accelerated gradient, NAG), метод адаптивного градиента (adaptive gradient, AdaGrad), метод скользящего среднего (root mean square propagation, RMSProp), метод адаптивной оценки моментов (adaptive moment estimation, Adam).	**2**
Раздел №2. АЛГОРИТМЫ ЭВОЛЮЦИОННОЙ ОПТИМИЗАЦИИ	
Лекция 5. Генетические алгоритмы Простой бинарный генетический алгоритм; инициализация, оценка, отбор, рекомбинирование (мутации). Регулировочные параметры и примеры генетического алгоритма. Простой непрерывный генетический алгоритм. Преимущества и недостатки. Математические модели генетических алгоритмов. Теория схем. Цепи Маркова. Марковские модели генетических алгоритмов. Системно-динамические модели генетических алгоритмов.	**4**
Лекция 6. Метод симуляции восстановления Алгоритм отжига: начальное решение, оценка решения, случайный поиск решения, критерий допуска. Снижение температуры,	**4**

повтор. Преимущества и недостатки. Режимы охлаждения: линейное, экспоненциальное, обратное, логарифмическое, обратное линейное, размерно-зависимое.	
Лекция 7. Алгоритм оптимизации на основе муравьиной кучи Модели феромона. Муравьиная система. Непрерывная оптимизация. Алгоритм муравья: граф, муравей, начальная популяция, движение муравья, путешествие муравья, Испарение фермента, повторный запуск. Преимущества и недостатки. Другие муравьиные системы: минимаксная муравьиная система, система муравьиной кучи.	**4**
Лекция 8. Алгоритм оптимизации на основе роя частиц Базовый алгоритм оптимизации на основе роя частиц: топология роя частиц. Преимущества и недостатки. Ограничение скорости. Коэффициенты инерционного взвешивания и сужения: инерционное взвешивание, коэффициент сужения, стабильность оптимизации на основе роя частиц. Глобальные обновления скорости. Полноинформированный рой частиц. Самообучение на ошибках.	**4**
Всего часов лекционных занятий	**24**

4.3 Лабораторные работы

Наименование и содержание лабораторной работы	Часов контактной работы с преподавателем
	Очная форма
Раздел №1. Градиентные методы оптимизации	
Лабораторная работа № 1. Метод наискорейшего градиентного спуска. Реализуется алгоритм метода наискорейшего градиентного спуска с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	**2**
Лабораторная работа № 2. Метод градиентного спуска. Реализуется алгоритм метода градиентного спуска с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	**2**
Лабораторная работа № 3. Сопряжённый градиентный метод.	**2**

Реализуется алгоритм сопряженного градиентного метода с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	
Лабораторная работа № 4. Стохастический градиентный метод. Реализуется алгоритм стохастического градиентного метода с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	**2**
Рубежный контроль №1.	**2**
Раздел №2. Алгоритмы эволюционной оптимизации	
Лабораторная работа № 5. Генетические алгоритмы. Реализуется простой бинарный генетический алгоритм с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	**4**
Лабораторная работа № 6. Метод симуляции восстановления. Реализуется алгоритм метода симуляции восстановления в решении задачи N шахматных ферзей (NQP) с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	**4**
Лабораторная работа № 7. Алгоритм оптимизации на основе муравьиной кучи. Реализуется алгоритм метода оптимизации на основе муравьиной кучи в решении задачи коммивояжера (TSP) с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	**2**
Лабораторная работа № 8. Алгоритм оптимизации на основе роя частиц. Реализуется алгоритм метода оптимизации на основе роя частиц с использованием технологии визуального проектирования и событийного программирования на языке программирования Visual C++.	**2**
Рубежный контроль №2.	**2**
Всего часов лабораторных занятий	**24**

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Лекционный курс основывается на методе обучения, использующем технологию, при которой обучающиеся конспектируют теоретический материал, участвуют в опросах и дискуссиях. В этом случае задействованы зрительная, слуховая, моторная и ассоциативная виды памяти.

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующей практической работы.

Преподавателем запланировано использование при чтении лекций технологии учебной дискуссии. Поэтому рекомендуется фиксировать для себя интересные моменты с целью их активного обсуждения на дискуссии в конце лекции.

Залогом качественного выполнения лабораторных работ является самостоятельная подготовка к ним накануне путем повторения материалов лекций. Рекомендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале занятия.

Лабораторные работы выполняются с применением языков программирования Visual C++, Visual C# интегрированной среды программирования Microsoft Visual Studio Community 2022.

Преподавателем запланировано применение на лабораторных занятиях технологий развивающейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно-рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, подготовку к лабораторным занятиям, к рубежным контролям, подготовку к экзамену.

Рекомендуемая трудоемкость самостоятельной работы для очной формы обучения представлена в таблице:

Рекомендуемый режим самостоятельной работы
Очная форма

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. часов
	Очная форма
Самостоятельное изучение тем дисциплины:	**45**
Модифицированный метод адаптивной оценки моментов (Adamax).	3
Ускоренный по Нестерову метод адаптивной оценки моментов (Nesterov accelerated adaptive moment, Nadam)	3
Дифференциальная эволюция	3
Алгоритмы оценивания вероятностных распределений	4
Биогеографическая оптимизация	4
Культурные алгоритмы	4
Алгоритм искусственного косяка рыб	4
Оптимизатор на основе группового поиска	4
Алгоритм перемешанных лягушачьих прыжков	4
Светлячковый алгоритм	4
Оптимизация на основе бактериальной кормодобычи	4
Алгоритм гравитационного поиска	4
Подготовка и выполнение лабораторных работ (по 2 часа на каждое занятие)	**20**
Подготовка к рубежному контролю (по 2 часа на каждый рубеж)	**4**
Подготовка к экзамену	**27**
Всего:	**96**

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся в КГУ (для очной формы обучения).

2. Отчёты обучающихся по лабораторным занятиям.

3. Банк заданий к рубежным контролям № 1, № 2 (для очной формы обучения).

4 Банк заданий к экзамену.

6.2. Система балльно-рейтинговой оценки работы обучающихся по дисциплине

<table>
<tr><th>№</th><th>Наименование</th><th colspan="6">Содержание</th></tr>
<tr><td colspan="8">Очная форма обучения</td></tr>
<tr><td rowspan="3">1</td><td rowspan="3">Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения обучающихся на первом учебном занятии)</td><td colspan="6">Распределение баллов</td></tr>
<tr><td>Вид учебной работы:</td><td>Посещение лекций</td><td>Выполнение и защита отчетов по лабораторным работам</td><td>Рубежный контроль №1</td><td>Рубежный контроль №2</td><td>Экзамен</td></tr>
<tr><td>Балльная оценка:</td><td>1б*12=12б</td><td>6б*8=48б</td><td>5</td><td>5</td><td>30</td></tr>
<tr><td>2</td><td>Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и экзамена</td><td colspan="6">60 и менее баллов – неудовлетворительно;
61…73 – удовлетворительно;
74… 90 – хорошо;
91…100 – отлично</td></tr>
</table>

3	Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов	Для допуска к промежуточной аттестации по дисциплине за семестр обучающийся должен набрать по итогам текущего и рубежного контролей не менее 51 балла. В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается. Для получения экзамена без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся, определяется по количеству баллов, набранных им в ходе текущего и рубежных контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность. Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине , не снижается. За академическую активность в ходе освоения дисциплины, участие в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность составляет 30. Основанием для получения дополнительных баллов являются: - выполнение дополнительных заданий по дисциплине; дополнительные баллы начисляются преподавателем; - участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра	В случае если к промежуточной аттестации (экзамену) набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра. Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли проводятся в виде ответов на вопросы в письменной форме. Экзамен проводится в виде ответов на вопросы билета в устной форме.

Перед проведением рубежного контроля преподаватель прорабатывает с обучающимися основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии.

На выполнение тестовых заданий рубежных контролей обучающемуся отводится 2 часа на практических занятиях.

Варианты заданий для рубежных контролей № 1, № 2 состоят из 20 вопросов. Для определения баллов при проверке рубежных контролей используются интервальные оценки, представленные в таблице

Количество правильных ответов	1-5	6-8	9-11	12-14	15-17	18-20
Количество баллов	0	1	2	3	4	5

Преподаватель оценивает в баллах результаты рубежного контроля каждого обучающегося по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

Билет к экзамену состоит из 2 вопросов. Вопросы к экзамену доводятся до студентов на последней лекции в семестре. На подготовку ответа по вопросам билета к экзамену студенту отводится 1 астрономический час. Каждый вопрос оценивается в 15 баллов.

Результаты текущего контроля успеваемости и экзамена заносятся преподавателем в экзаменационную ведомость, которая сдается в организационный отдел института в день экзамена, а также выставляются в зачетную книжку обучающего-го.

6.4 Примеры оценочных средств для рубежных контролей и экзамена

6.4.1 Примеры вопросов для проведения рубежного контроля №1

Вариант 1_1

1 Какой правильный алгоритм симплекс-метода?

1 Шаг 1. Определение начального допустимого решения.

Шаг 2. Определение включаемой переменной из числа небазисных переменных. Если такой переменной нет, то решение оптимально, иначе осуществляется переход к шагу 3.

Шаг 3. Определение исключаемой переменной из числа базисных переменных.

Шаг 4. Определение нового базисного решения. Переход на шаг 2.

2 Шаг 1. Определение начального допустимого решения.

Шаг 2. Определение включаемой переменной из числа базисных переменных. Если такой переменной нет, то решение оптимально, иначе осуществляется переход к шагу 3.

Шаг 3. Определение исключаемой переменной из числа небазисных переменных.

Шаг 4. Определение нового базисного решения. Переход на шаг 2.

3 Шаг 1. Определение начального допустимого решения.

Шаг 2. Определение исключаемой переменной из числа небазисных переменных. Если такой переменной нет, то решение оптимально, иначе осуществляется переход к шагу 3.

Шаг 3. Определение включаемой переменной из числа базисных переменных.

Шаг 4. Определение нового базисного решения. Переход на шаг 2.

4 Шаг 1. Определение начального допустимого решения.

Шаг 2. Определение включаемой переменной из числа небазисных переменных. Если такой переменной нет, то решение допустимо, иначе осуществляется переход к шагу 3.

Шаг 3. Определение исключаемой переменной из числа базисных переменных.

Шаг 4. Определение нового базисного решения. Переход на шаг 2.

2 Какое правильное определение включаемой переменной?

1 Базисная переменная, которая на следующей итерации подлежит исключению из множества базисных переменных.

2 Небазисная переменная, которая на следующей итерации исключится из множества базисных переменных.

3 Базисная переменная, которая на следующей итерации включится в множество небазисных переменных.

4 Небазисная переменная, которая на следующей итерации включится в множество базисных переменных.

3 Какое правильное определение условия оптимальности?

1 Включаемой переменной в задаче максимизации (минимизации) является базисная переменная, имеющая в Z-уравнении наибольший отрицательный (положительный) коэффициент. В случае равенства коэффициентов для нескольких небазисных переменных выбор делается произвольно. Если все коэффициенты при небазисных переменных в Z-уравнении неотрицательны (неположительны), полученное решение является оптимальным

2 Включаемой переменной в задаче максимизации (минимизации) является небазисная переменная, имеющая в Z-уравнении наибольший отрицательный (положительный) коэффициент. В случае равенства коэффициентов для нескольких небазисных переменных выбор делается произвольно. Если все коэффициенты при небазисных переменных в Z-уравнении неотрицательны (неположительны), полученное решение является оптимальным.

3 Включаемой переменной в задаче максимизации (минимизации) является небазисная переменная, имеющая в Z-уравнении наименьший отрицательный

(положительный) коэффициент. В случае равенства коэффициентов для нескольких небазисных переменных выбор делается произвольно. Если все коэффициенты при небазисных переменных в Z-уравнении неотрицательны (неположительны), полученное решение является оптимальным

4 Включаемой переменной в задаче максимизации (минимизации) является небазисная переменная, имеющая в Z-уравнении наибольший отрицательный (положительный) коэффициент. В случае равенства коэффициентов для нескольких небазисных переменных выбор делается произвольно. Если все коэффициенты при небазисных переменных в Z-уравнении отрицательны (положительны), полученное решение является оптимальным

4 Какое правильное определение условия допустимости?

1 В задачах максимизации и минимизации в качестве исключаемой переменной выбирается базисная переменная, для которой отношение постоянной в правой части соответствующего ограничения к (положительному) коэффициенту ведущего столбца минимально. В случае равенства этого отношения для нескольких базисных переменных выбор делается произвольно.

2 В задачах максимизации и минимизации в качестве исключаемой переменной выбирается небазисная переменная, для которой отношение постоянной в правой части соответствующего ограничения к (положительному) коэффициенту ведущего столбца минимально. В случае равенства этого отношения для нескольких базисных переменных выбор делается произвольно.

3 В задачах максимизации и минимизации в качестве исключаемой переменной выбирается базисная переменная, для которой отношение постоянной в правой части соответствующего ограничения к (положительному) коэффициенту ведущего столбца максимально. В случае равенства этого отношения для нескольких базисных переменных выбор делается произвольно.

4 В задачах максимизации и минимизации в качестве исключаемой переменной выбирается небазисная переменная, для которой отношение постоянной в правой части соответствующего ограничения к (положительному) коэффициенту ведущего столбца макмимально. В случае равенства этого отношения для нескольких базисных переменных выбор делается произвольно.

5 Что такое ведущая строка симплекс таблицы?

1 Строка симплекс таблицы, соответствующая исключаемой небазисной переменной.

2 Строка симплекс таблицы, соответствующая включаемой базисной переменной.

3 Строка симплекс таблицы, соответствующая исключаемой базисной переменной.

4 Строка симплекс таблицы, соответствующая включаемой небазисной переменной.

6 Что такое ведущий столбец симплекс таблицы?

1 Столбец симплекс таблицы, соответствующий включаемой базисной переменной.

2 Столбец симплекс таблицы, соответствующий исключаемой небазисной переменной.

3 Столбец симплекс таблицы, соответствующий включаемой небазисной переменной.

4 Столбец симплекс таблицы, соответствующий исключаемой базисной переменной.

7 Что называется ведущим элементом симплекс таблицы?

1 Элемент симплекс таблицы, находящийся на пересечении ведущего столбца и ведущей строки симплекс таблицы.

2 Элемент ведущей строки симплекс таблицы, соответствующий включаемой переменной.

3 Элемент ведущего столбца симплекс таблицы, соответствующий исключаемой переменной.

4 Элемент симплекс таблицы, соответствующий наименьшему симплексному отношению.

8 Какие правильные вычислительные процедуры метода Гаусса-Жордана?

1 1 Формирование новой ведущей строки.

Новая ведущая строка=Старая ведущая строка/Ведущий элемент.

2 Формирование нового ведущего столбца.

Новый ведущий столбец=Старый ведущий столбец/Ведущий элемент.

3 Формирование остальных новых уравнений.

Новое уравнение=Старое уравнение+(Коэффициент ведущего столбца старого уравнения)*(Новая ведущая строка).

2 1 Формирование новой ведущей строки.

Новая ведущая строка=Старая ведущая строка/Ведущий элемент.

2 Формирование остальных новых уравнений.

Новое уравнение=Старое уравнение+(Коэффициент ведущего столбца старого уравнения)*(Новая ведущая строка).

3 1 Формирование новой ведущей строки.

Новая ведущая строка=Старая ведущая строка/Ведущий элемент.

2 Формирование нового ведущего столбца.

Новый ведущий столбец=Старый ведущий столбец/Ведущий элемент.

3 Формирование остальных новых уравнений.

Новое уравнение=Старое уравнение–(Коэффициент ведущего столбца старого уравнения)*(Новая ведущая строка)

4 1 Формирование новой ведущей строки.

Новая ведущая строка=Старая ведущая строка/Ведущий элемент.

2 Формирование остальных новых уравнений.

Новое уравнение=Старое уравнение–(Коэффициент ведущего столбца старого уравнения)*(Новая ведущая строка).

9 Какой правильный алгоритм метода обыкновенное Жорданово исключение?

1 Этап 1. Разрешающий элемент заменяется обратной величиной.

Этап 2. Остальные элементы разрешающей строки делятся на разрешающий элемент.

Этап 3. Остальные элементы разрешающего столбца делятся на разрешающий элемент и меняют знаки.

Этап 4. Прочие элементы определяются по правилу прямоугольника.

2 Этап 1. Разрешающий элемент заменяется обратной величиной.

Этап 2. Остальные элементы разрешающей строки делятся на разрешающий элемент и меняют знаки.

Этап 3. Остальные элементы разрешающего столбца делятся на разрешающий элемент.

Этап 4. Прочие элементы определяются по правилу прямоугольника.

3 Этап 1. Разрешающий элемент заменяется обратной величиной.

Этап 2. Остальные элементы разрешающей строки делятся на разрешающий элемент и меняют знаки.

Этап 3. Остальные элементы разрешающего столбца делятся на разрешающий элемент и меняют знаки.

Этап 4. Прочие элементы определяются по правилу прямоугольника.

4 Этап 1. Разрешающий элемент заменяется обратной величиной и меняет знак.

Этап 2. Остальные элементы разрешающей строки делятся на разрешающий элемент и меняют знаки.

Этап 3. Остальные элементы разрешающего столбца делятся на разрешающий элемент.

Этап 4. Прочие элементы определяются по правилу прямоугольника.

10 Какой правильный алгоритм метода золотое сечение?

1 Исходные данные: $[a, b]$ – отрезок, содержащий точку максимума, ε – параметр окончания счёта.

Этап 1. $\lambda = \frac{\sqrt{5+1}}{2}$; $k=1$, $a_k = a$, $b_k = b$;

$y = (\lambda-1) * a_k + (\lambda-1)^2 * b_k$; $A = f(y)$;

$z = (\lambda-1)^2 * a_k + (\lambda-1) * b_k$; $B = f(z)$;

Этап 2. Если $A > B$, то перейти на 4 этап.

Этап 3. $a_{k+1} = y$; $b_{k+1} = b_k$;

$y = z$; $A = B$;

$z = (\lambda-1)^2 * a_{k+1} + (\lambda-1) * b_{k+1}$;

$B = f(z)$, перейти на 5 этап.

Этап 4. $a_{k+1} = a_k$; $b_{k+1} = z$; $z = y$; $B = A$;

$y = (\lambda-1) * a_{k+1} + (\lambda-1)^2 * b_{k+1}$;

$A = f(y)$

Этап 5. Если $b_{k+1} - a_{k+1} < \varepsilon$, то $X^* \in [a_{k+1}, b_{k+1}]$, конец.

Этап 6. $k=k+1$, перейти на этап 2

2 Исходные данные: $[a, \mathrm{b}]$ – отрезок, содержащий точку максимума, ε – параметр окончания счёта.

Этап 1. $\lambda=\frac{\sqrt{5+1}}{2}$; $k=1$, $a_k=a$, $b_k=b$;

$y=(\lambda-1)*a_k+(\lambda-1)^2*b_k$; $A=f(y)$;

$z=(\lambda-1)^2*a_k+(\lambda-1)*b_k$; $B=f(z)$;

Этап 2. Если $A<B$, то перейти на 4 этап.

Этап 3. $a_{k+1}=y$; $b_{k+1}=b_k$;

$y=z$; $A=B$;

$z=(\lambda-1)^2*a_{k+1}+(\lambda-1)*b_{k+1}$;

$B=f(z)$, перейти на 5 этап.

Этап 4. $a_{k+1}=a_k$; $b_{k+1}=z$; $z=y$; $B=A$;

$y=(\lambda-1)*a_{k+1}+(\lambda-1)^2*b_{k+1}$;

$A=f(y)$

Этап 5. Если $b_{k+1}-a_{k+1}<\varepsilon$, то $X^*\in[a_{k+1},b_{k+1}]$, конец.

Этап 6. $k=k+1$, перейти на этап 2.

3 Исходные данные: $[a, \mathrm{b}]$ – отрезок, содержащий точку максимума, ε – параметр окончания счёта.

Этап 1. $\lambda=\frac{\sqrt{5+1}}{2}$; $k=1$, $a_k=a$, $b_k=b$;

$y=(\lambda-1)*a_k+(\lambda-1)^2*b_k$; $A=f(y)$;

$z=(\lambda-1)^2*a_k+(\lambda-1)*b_k$; $B=f(z)$;

Этап 2. Если $A>B$, то перейти на 4 этап.

Этап 3. $a_{k+1}=y$; $b_{k+1}=b_k$;

$z=y$; $A=B$;

$z=(\lambda-1)^2*a_{k+1}+(\lambda-1)*b_{k+1}$;

$B=f(z)$, перейти на 5 этап.

Этап 4. $a_{k+1}=a_k$; $b_{k+1}=z$; $z=y$; $B=A$;

$y=(\lambda-1)*a_{k+1}+(\lambda-1)^2*b_{k+1}$;

$A=f(y)$

Этап 5. Если $b_{k+1}-a_{k+1}<\varepsilon$, то $X^*\in[a_{k+1},b_{k+1}]$, конец.

Этап 6. $k=k+1$, перейти на этап 2.

4 Исходные данные: $[a, \mathrm{b}]$ – отрезок, содержащий точку максимума, ε – параметр окончания счёта.

Этап 1. $\lambda = \frac{\sqrt{5+1}}{2}$; $k=1$, $a_k = a$, $b_k = b$;

$y = (\lambda - 1) * a_k + (\lambda - 1)^2 * b_k$; $A = f(y)$;

$z = (\lambda - 1)^2 * a_k + (\lambda - 1) * b_k$; $B = f(z)$;

Этап 2. Если $A > B$, то перейти на 4 этап.

Этап 3. $a_{k+1} = y$; $b_{k+1} = b_k$;

$y = z$; $A = B$;

$z = (\lambda - 1)^2 * a_{k+1} + (\lambda - 1) * b_{k+1}$;

$B = f(z)$, перейти на 5 этап.

Этап 4. $a_{k+1} = a_k$; $b_{k+1} = z$; $y = z$; $A = B$;

$y = (\lambda - 1) * a_{k+1} + (\lambda - 1)^2 * b_{k+1}$;

$A = f(y)$

Этап 5. Если $b_{k+1} - a_{k+1} < \varepsilon$, то $X^* \in [a_{k+1}, b_{k+1}]$, конец.

Этап 6. $k = k+1$, перейти на этап 2.

11 Что такое конгруэнтность чисел?

1 Два числа конгруэнтны, если разность чисел дробное число.

2 Два числа конгруэнтны, если разность чисел целое число.

3 Два числа конгруэнтны, если сумма чисел дробное число.

4 Два числа конгруэнтны, если сумма чисел целое число.

12 Что понимается под дробной частью числа?

1 Наибольшее неотрицательное число, конгруэнтное исходному числу.

2 Наименьшее отрицательное число, конгруэнтное исходному числу.

3 Наименьшее неотрицательное число, конгруэнтное исходному числу.

4 Наибольшее отрицательное число, конгруэнтное исходному числу.

13 Какой правильный алгоритм метода отсекающих плоскостей?

1 Этап 1. Ослабленная задача. Целочисленность искомых переменных игнорируется, симплекс-методом определяется оптимальный план.

Этап 2. Расширенная задача. Если план нецелочисленный, составляется дополнительное ограничение, “отсекающее” дробную часть искомой переменной. Дополнительное ограничение включается в систему ограничений и решается расширенная задача.

2 1 Этап 1. Ослабленная задача. Неотрицательность искомых переменных игнорируется, симплекс-методом определяется оптимальный план.

Этап 2. Расширенная задача. Если план нецелочисленный, составляется дополнительное ограничение, “отсекающее” дробную часть искомой переменной. Дополнительное ограничение включается в систему ограничений и решается расширенная задача.

3 1 Этап 1. Ослабленная задача. Целочисленность искомых переменных игнорируется, симплекс-методом определяется оптимальный план.

Этап 2. Расширенная задача. Если план целочисленный, составляется дополнительное ограничение, “отсекающее” дробную часть искомой переменной. Дополнительное ограничение включается в систему ограничений и решается расширенная задача.

4 1 Этап 1. Ослабленная задача. Целочисленность искомых переменных игнорируется, симплекс-методом определяется оптимальный план.

Этап 2. Расширенная задача. Если план нецелочисленный, составляется дополнительное ограничение, “отсекающее” дробную часть дополнитнльной переменной. Дополнительное ограничение включается в систему ограничений и решается расширенная задача.

14 Градиент – вектор $\nabla f(x)$ непрерывно дифференцируемой функции $f(x)$ в точке x называется вектор-столбец элементами которого являются частные производные первого порядка, вычисленные в данной точке?

1 Да

2 Нет

15 Решением (точкой минимума), задачи безусловной оптимизации называется вектор $\bar{x}^* \in R^n$, такой что $f(\bar{x}^*) \le f(\bar{x})$ для всех $\bar{x} \in R^n$, $f(\bar{x}^*) = \min f(\bar{x})$, $\bar{x} \in R^n$?

1 Нет

2 Да

16 Какой правильный алгоритм метода спуска?

1 ***Начальный этап.*** Задать $\bar{x}1 \in R^n$ – начальную точку, $\varepsilon > 0$ – параметр окончания счёта. Положить $k = 1$.

Основной этап включает шаги:

Шаг 1. В точке $\bar{x}k$ проверить условие окончания счёта, если оно выполняется, то положить $\bar{x}^* = \bar{x}k$ и остановиться.

Шаг 2. В точке $\bar{x}k$ выбрать направление спуска $\bar{S}k$.

Шаг 3. Положить $\bar{x}k{+}1 = \bar{x}k - \lambda_k \bar{S}k$, где λ_k – длина шага вдоль направления $\bar{S}k$. Положить $k = k + 1$. Перейти к 1 шагу.

2 ***Начальный этап.*** Задать $\bar{x}1 \in R^n$ – начальную точку, $\varepsilon > 0$ – параметр окончания счёта. Положить $k = 1$.

Основной этап включает шаги:

Шаг 1. В точке $\bar{x}k$ проверить условие окончания счёта, если оно выполняется, то положить $\bar{x}^* = \bar{x}k$ и остановиться.

Шаг 2. В точке $\bar{x}k$ выбрать направление спуска $\bar{S}k$.

Шаг 3. Положить $\overline{x}k{+}1 = \overline{x}k + \lambda_k \overline{S}k$, где λk – длина шага вдоль направления $\overline{S}k$. Положить $k = k - 1$. Перейти к 1 шагу.

3 ***Начальный этап.*** Задать $\overline{x}1 \in R^n$ – начальную точку, $\varepsilon > 0$ – параметр окончания счёта. Положить $k = 1$.

Основной этап включает шаги:

Шаг 1. В точке $\overline{x}k$ проверить условие окончания счёта, если оно выполняется, то положить $\overline{x}^* = \overline{x}k{+}1$ и остановиться.

Шаг 2. В точке $\overline{x}k$ выбрать направление спуска $\overline{S}k$.

Шаг 3. Положить $\overline{x}k{+}1 = \overline{x}k + \lambda_k \overline{S}k$, где λk – длина шага вдоль направления $\overline{S}k$. Положить $k = k + 1$. Перейти к 1 шагу.

4 ***Начальный этап.*** Задать $\overline{x}1 \in R^n$ – начальную точку, $\varepsilon > 0$ – параметр окончания счёта. Положить $k = 1$.

Основной этап включает шаги:

Шаг 1. В точке $\overline{x}k$ проверить условие окончания счёта, если оно выполняется, то положить $\overline{x}^* = \overline{x}k$ и остановиться.

Шаг 2. В точке $\overline{x}k$ выбрать направление спуска $\overline{S}k$.

Шаг 3. Положить $\overline{x}k{+}1 = \overline{x}k + \lambda_k \overline{S}k$, где λk – длина шага вдоль направления $\overline{S}k$. Положить $k = k + 1$. Перейти к 1 шагу.

17 Какой правильный алгоритм Марквардта?

1 **Этап 1.** Задать исходные данные x^0, $\varepsilon_1 > 0$, M – предельное число итераций. Найти градиент $\nabla f(x)$, матрицу Гессе $H(x)$.

Этап 2. Положить $k = 0$.

Этап 3. Вычислить градиент $\nabla f(x^k)$.

Этап 4. Проверить выполнения условия $\left\|\nabla f(x^k)\right\| < \varepsilon_1$:

a) если условие выполняется, то расчёт окончен и $x^* = x^k$;
b) иначе перейти на этап 5.

Этап 5. Проверить выполнение неравенства $k \geq M$:

a) если неравенство выполнено, то расчёт окончен и $x^* = x^k$;
b) иначе перейти на этап 6.

Этап 6. Вычислить матрицу $H(x)$.

Этап 7. Вычислить $H(x^k) + \mu^k E$.

Этап 8. Вычислить $[H(x^k)+\mu^k E]^{-1}$.

Этап 9. Вычислить $d_k = -[H(x^k)-\mu^k E]^1 \nabla f(x^k)$.

Этап 10. Вычислить $x^{k+1} = x^k - [H(x^k)+\mu^k E]^{-1} \nabla f(x^k)$.

Этап 11. Проверить выполнение условия $f(x^{k+1}) < f(x^k)$:

a) если неравенство выполняется, то перейти на этап 12;

b) иначе, перейти на этап 13.

Этап 12. Положить $k = k+1$, $\mu^{k+1} = \dfrac{\mu^k}{2}$ и перейти на этап 3.

Этап 13. Положить $\mu^k = 2\mu^k$ и перейти на этап 7.

2 **Этап 1.** Задать исходные данные x^0, $\varepsilon_1 > 0$, M – предельное число итераций. Найти градиент $\nabla f(x)$, матрицу Гессе $H(x)$.

Этап 2. Положить $k = 0$.

Этап 3. Вычислить градиент $\nabla f(x^k)$.

Этап 4. Проверить выполнения условия $\left\| \nabla f(x^k) \right\| < \varepsilon_1$:

c) если условие выполняется, то расчёт окончен и $x^* = x^k$;

d) иначе перейти на этап 5.

Этап 5. Проверить выполнение неравенства $k \ge M$:

c) если неравенство выполнено, то расчёт окончен и $x^* = x^k$;

d) иначе перейти на этап 6.

Этап 6. Вычислить матрицу $H(x)$.

Этап 7. Вычислить $H(x^k)+\mu^k E$.

Этап 8. Вычислить $[H(x^k)+\mu^k E]^{-1}$.

Этап 9. Вычислить $d_k = -[H(x^k)+\mu^k E]^{-1} \nabla f(x^k)$.

Этап 10. Вычислить $x^{k+1} = x^k - [H(x^k)+\mu^k E]^{-1} \nabla f(x^k)$.

Этап 11. Проверить выполнение условия $f(x^{k+1}) < f(x^k)$:

c) если неравенство выполняется, то перейти на этап 12;

d) иначе, перейти на этап 13.

Этап 12. Положить $k = k+1$, $\mu^{k+1} = \dfrac{\mu^k}{2}$ и перейти на этап 3.

Этап 13. Положить $\mu^k = 2\mu^k$ и перейти на этап 7.

3 **Этап 1.** Задать исходные данные x^0, $\varepsilon_1 > 0$, M – предельное число итераций. Найти градиент $\nabla f(x)$, матрицу Гессе $H(x)$.

Этап 2. Положить $k = 0$.

Этап 3. Вычислить градиент $\nabla f(x^k)$.

Этап 4. Проверить выполнения условия $\left\|\nabla f(x^k)\right\| < \varepsilon_1$:

e) если условие выполняется, то расчёт окончен и $x^* = x^k$;
f) иначе перейти на этап 5.

Этап 5. Проверить выполнение неравенства $k \geq M$:

e) если неравенство выполнено, то расчёт окончен и $x^* = x^k$;
f) иначе перейти на этап 6.

Этап 6. Вычислить матрицу $H(x)$.

Этап 7. Вычислить $H(x^k) + \mu^k E$.

Этап 8. Вычислить $[H(x^k) + \mu^k E]^{-1}$.

Этап 9. Вычислить $d_k = -[H(x^k) + \mu^k E]^{-1} \nabla f(x^k)$.

Этап 10. Вычислить $x^{k+1} = x^k - [H(x^k) + \mu^k E]^{-1} \nabla f(x^k)$.

Этап 11. Проверить выполнение условия $f(x^{k+1}) < f(x^k)$:

e) если неравенство выполняется, то перейти на этап 12;
f) иначе, перейти на этап 13.

Этап 12. Положить $k = k + 1$, $\mu^{k+1} = \dfrac{\mu^k}{4}$ и перейти на этап 3.

Этап 13. Положить $\mu^k = 2\mu^k$ и перейти на этап 7.

4 **Этап 1.** Задать исходные данные x^0, $\varepsilon_1 > 0$, M – предельное число итераций. Найти градиент $\nabla f(x)$, матрицу Гессе $H(x)$.

Этап 2. Положить $k = 0$.

Этап 3. Вычислить градиент $\nabla f(x^k)$.

Этап 4. Проверить выполнения условия $\left\|\nabla f(x^k)\right\| < \varepsilon_1$:

g) если условие выполняется, то расчёт окончен и $x^* = x^k$;
h) иначе перейти на этап 5.

Этап 5. Проверить выполнение неравенства $k \geq M$:

g) если неравенство выполнено, то расчёт окончен и $x^* = x^k$;
h) иначе перейти на этап 6.

Этап 6. Вычислить матрицу $H(x)$.

Этап 7. Вычислить $H(x^k)+\mu^k E$.

Этап 8. Вычислить $[H(x^k)+\mu^k E]^{-1}$.

Этап 9. Вычислить $d_k=-[H(x^k)+\mu^k E]^{-1}\nabla f(x^k)$.

Этап 10. Вычислить $x^{k+1}=x^k-[H(x^k)+\mu^k E]^{-1}\nabla f(x^k)$.

Этап 11. Проверить выполнение условия $f(x^{k+1})<f(x^k)$:

g) если неравенство выполняется, то перейти на этап 12;

h) иначе, перейти на этап 13.

Этап 12. Положить $k=k+1$, $\mu^{k+1}=\dfrac{\mu^k}{2}$ и перейти на этап 3.

Этап 13. Положить $\mu^k=4\mu^k$ и перейти на этап 7.

18 С какой скоростью сходится последовательность $\{x^k\}$ к точке минимума в методе Ньютона?

1 Квадратичная

2 Линейная

3 Логарифмическая

4 Кубическая

19 Какой правильный алгоритм метода штрафов?

1 **Этап 1.** Задать начальную точку x^0, начальное значение параметра штрафа $r^k\ge 0$, число $C>1$ для увеличения параметра, малое число $\varepsilon>0$ для окончания алгоритма. Положить $k=0$.

Этап 2. Составить вспомогательную функцию

$$F(x,r^k)=\frac{r^k}{2}\left\{\sum_{j=1}^{m}[g_j(x)]^2-\sum_{j=m+1}^{p}[g_j^+(x)]^2\right\}$$

Этап 3. Найти точку $x^*(r^k)$ безусловного минимума функции $F(x,r^k)$ по x с помощью какого-либо метода (нулевого, первого или второго порядка)

$$F(x^*(r^k),r^k)=\min_{x\in R^n}F(x,r^k)$$

При этом задать все требуемые выбранным методом параметры. В качестве начальной точки взять x^k. Вычислить $P(x^*(r^k),r^k)$.

Этап 4. Проверить условие окончания:

а) если $P(x^*(r^k),r^k)\le\varepsilon$, процесс поиска закончить:

$x^*=x^*(r^k),\ f(x^*)=f(x^*(r^k))$

b) если $P(x^*(r^k), r^k) > \varepsilon$, положить : $r^{k+1} = Cr^k$, $x^{k+1} = x^*(r^k)$, $k = k+1$.
Перейти на 2 этап.

2 **Этап 1.** Задать начальную точку x^0, начальное значение параметра штрафа $r^k \geq 0$, число $C > 1$ для увеличения параметра, малое число $\varepsilon > 0$ для окончания алгоритма. Положить $k = 0$.

Этап 2. Составить вспомогательную функцию

$$F(x, r^k) = \frac{r^k}{2}\left\{\sum_{j=1}^{m}[g_j(x)]^2 + \sum_{j=m+1}^{p}[g_j^+(x)]^2\right\}$$

Этап 3. Найти точку $x^*(r^k)$ безусловного минимума функции $F(x, r^k)$ по x с помощью какого-либо метода (нулевого, первого или второго порядка)

$$F(x^*(r^k), r^k) = \max_{x \in R^n} F(x, r^k)$$

При этом задать все требуемые выбранным методом параметры. В качестве начальной точки взять x^k. Вычислить $P(x^*(r^k), r^k)$.

Этап 4. Проверить условие окончания:

c) если $P(x^*(r^k), r^k) \leq \varepsilon$, процесс поиска закончить:
$x^* = x^*(r^k)$, $f(x^*) = f(x^*(r^k))$

d) если $P(x^*(r^k), r^k) > \varepsilon$, положить : $r^{k+1} = Cr^k$, $x^{k+1} = x^*(r^k)$, $k = k+1$.
Перейти на 2 этап.

3 **Этап 1.** Задать начальную точку x^0, начальное значение параметра штрафа $r^k \geq 0$, число $C > 1$ для увеличения параметра, малое число $\varepsilon > 0$ для окончания алгоритма. Положить $k = 0$.

Этап 2. Составить вспомогательную функцию

$$F(x, r^k) = \frac{r^k}{2}\left\{\sum_{j=1}^{m}[g_j(x)]^2 + \sum_{j=m+1}^{p}[g_j^+(x)]^2\right\}$$

Этап 3. Найти точку $x^*(r^k)$ безусловного минимума функции $F(x, r^k)$ по x с помощью какого-либо метода (нулевого, первого или второго порядка)

$$F(x^*(r^k), r^k) = \min_{x \in R^n} F(x, r^k)$$

При этом задать все требуемые выбранным методом параметры. В качестве начальной точки взять x^k. Вычислить $P(x^*(r^k), r^k)$.

Этап 4. Проверить условие окончания:

e) если $P(x^*(r^k), r^k) \ge \varepsilon$, процесс поиска закончить:

$x^* = x^*(r^k), f(x^*) = f(x^*(r^k))$

f) если $P(x^*(r^k), r^k) > \varepsilon$, положить : $r^{k+1} = Cr^k$, $x^{k+1} = x^*(r^k)$, $k = k+1$. Перейти на 2 этап.

4 **Этап 1.** Задать начальную точку x^0, начальное значение параметра штрафа $r^k \ge 0$, число $C > 1$ для увеличения параметра, малое число $\varepsilon > 0$ для окончания алгоритма. Положить $k = 0$.

Этап 2. Составить вспомогательную функцию

$$F(x,r^k) = \frac{r^k}{2}\left\{\sum_{j=1}^{m}[g_j(x)]^2 + \sum_{j=m+1}^{p}[g_j^+(x)]^2\right\}$$

Этап 3. Найти точку $x^*(r^k)$ безусловного минимума функции $F(x,r^k)$ по x с помощью какого-либо метода (нулевого, первого или второго порядка)

$$F(x^*(r^k), r^k) = \min_{x\in R^n} F(x, r^k)$$

При этом задать все требуемые выбранным методом параметры. В качестве начальной точки взять x^k. Вычислить $P(x^*(r^k), r^k)$.

Этап 4. Проверить условие окончания:

g) если $P(x^*(r^k), r^k) \le \varepsilon$, процесс поиска закончить:

$x^* = x^*(r^k), f(x^*) = f(x^*(r^k))$

h) если $P(x^*(r^k), r^k) > \varepsilon$, положить : $r^{k+1} = Cr^k$, $x^{k+1} = x^*(r^k)$, $k = k+1$. Перейти на 2 этап.

20 Какое условие должно выполняться в построении последовательности допустимых точек $\{x^k\}$ в стратегии решения задач методом Зойтендейка?

1 $f(x^{k+1}) > f(x^k)$, $k = 0,1,...$

2 $f(x^{k+1}) < f(x^k)$, $k = 0,1,...$

3 $f(x^{k+1}) == f(x^k)$, $k = 0,1,...$

4 $f(x^{k+1}) != f(x^k)$, $k = 0,1,...$

6.4.2 Примеры вопросов для проведения рубежного контроля №2

Вариант 2_1

1 Генетический алгоритм – адаптивный поисковый метод, основанный на селекции лучших элементов в популяции?

1 Да

2 Нет

2 Что понимается под оператором кроссинговера?

1 Языковая конструкция, позволяющая на основе преобразования родительской хромосомы(или ее части) создавать хромосому потомка.

2 Языковая конструкция, которая определяет, как новая генерация хромосом будет построена из родителей и потомков.

3 Языковая конструкция, позволяющая на основе инвертирования родительской хромосомы(или ее части) создавать хромосому потомка.

4 Языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков.

3 Что понимается под оператором инверсии?

1 Языковая конструкция, позволяющая на основе преобразования родительской хромосомы(или ее части) создавать хромосому потомка.

2 Языковая конструкция, которая определяет, как новая генерация хромосом будет построена из родителей и потомков.

3 Языковая конструкция, позволяющая на основе инвертирования родительской хромосомы(или ее части) создавать хромосому потомка.

4 Языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков.

4 Что понимается под оператором мутации?

1 Языковая конструкция, позволяющая на основе преобразования родительской хромосомы(или ее части) создавать хромосому потомка.

2 Языковая конструкция, которая определяет, как новая генерация хромосом будет построена из родителей и потомков.

3 Языковая конструкция, позволяющая на основе инвертирования родительской хромосомы(или ее части) создавать хромосому потомка.

4 Языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков.

5 Что понимается под оператором рекомбинации?

1 Языковая конструкция, позволяющая на основе преобразования родительской хромосомы(или ее части) создавать хромосому потомка.

2 Языковая конструкция, которая определяет, как новая генерация хромосом будет построена из родителей и потомков.

3 Языковая конструкция, позволяющая на основе инвертирования родительской хромосомы(или ее части) создавать хромосому потомка.

4 Языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков.

6 Селекция – процесс, посредством которого хромосомы (альтернативные решения), имеющие более высокое значение целевой функции, получают большую возможность для воспроизводства (репродукции) потомков?

1 Нет

2 Да

7 Какой правильный простой генетический алгоритм?

1 *Родители<– {случайно сгенерированная популяция }*

While not *(критерий останова)*

Рассчитать приспособленность каждого родителя в популяции

Дети $\leftarrow$ *0*

While *|Дети| > |Родители|*

Применить приспособленности для вероятностного отбора пары

Родителей с целью их спаривания

Спарить родителей для создания детей $c1$ *и* $c2$

Дети $\leftarrow$ *Дети* $\cup$ $\{c1, c2\}$

Loop

Случайно мутировать несколько детей

Родители$\leftarrow$*Дети*

Next *поколение*

2 *Родители<– {случайно сгенерированная популяция }*

While not *(критерий останова)*

Рассчитать приспособленность каждого родителя в популяции

Дети $\leftarrow$ *0*

While *|Дети| < |Родители|*

Применить приспособленности для вероятностного отбора пары

Родителей с целью их спаривания

Спарить родителей для создания детей $c1$ *и* $c2$

Дети $\leftarrow$ *Дети* $\cap$ $\{c1, c2\}$

Loop

Случайно мутировать несколько детей

Родители$\leftarrow$*Дети*

Next *поколение*

3 *Родители<– {случайно сгенерированная популяция }*

While not *(критерий останова)*

Рассчитать приспособленность каждого родителя в популяции

Дети $\leftarrow$ *0*

While *|Дети| < |Родители|*

Применить приспособленности для вероятностного отбора пары

Родителей с целью их спаривания

Спарить родителей для создания детей $c1$ *и* $c2$

Дети ← *Дети* ∪ $\{c1, c2\}$

Loop

Случайно мутировать несколько детей

Родители← *Дети*

***Next** поколение*

4 *Родители<– {случайно сгенерированная популяция }*

***While not** (критерий останова)*

Рассчитать приспособленность каждого родителя в популяции

Дети ← *0*

***While** |Дети| > |Родители|*

Применить приспособленности для вероятностного отбора пары

Родителей с целью их спаривания

Спарить родителей для создания детей $c1$ *и* $c2$

Дети ← *Дети* ∩ $\{c1, c2\}$

Loop

Случайно мутировать несколько детей

Родители← *Дети*

***Next** поколение*

8 Как правильно формулируется правило репродукции Д. Холланда?

1 В простом генетическом алгоритму шаблон со значением целевой функции ниже среднего копируется в следующую генерацию, а шаблон с целевой функцией выше среднего – устраняется.

2 В простом генетическом алгоритму шаблон со значением целевой функции выше среднего копируется в следующую генерацию, а шаблон с целевой функцией ниже среднего – устраняется.

3 В простом генетическом алгоритму шаблон со значением целевой функции выше медианы копируется в следующую генерацию, а шаблон с целевой функцией ниже медианы – устраняется.

4 В простом генетическом алгоритму шаблон со значением целевой функции выше нуля копируется в следующую генерацию, а шаблон с целевой функцией ниже нуля – устраняется.

9 Алгоритм отжига (Simulated annealing, SA, симулированное закаливание) – алгоритм оптимизации, основанный на охлаждающем и кристаллизующем поведении химических веществ?

1 Да

2 Нет

10 Какой правильный простой алгоритм симулированного закаливания?

1 *T=исходная температура>0*

α *(T)=функция охлаждения* $\alpha(T) \in [0, T]$ для всех T

Инициализировать кандидатное решение x_0 для минимизационной задачи $f(x)$

While not *(критерий останова)*
Сгенерировать кандидатное решение x
if *($f(x) > f(x_0)$)*
$x_0 \leftarrow x$
else
$r \leftarrow U[0,1]$
if($r < [(f(x_0) - f(x))/T]$***) then***
$x_0 \leftarrow x$
end if
end if
$T \leftarrow \alpha(T)$
Next *итерация*
2 *T=исходная температура>0*
α *(T)=функция охлаждения* $\alpha(T) \in [0,T]$ для всех T
Инициализировать кандидатное решение x_0 для максимизационной задачи $f(x)$
While not *(критерий останова)*
Сгенерировать кандидатное решение x
if *(*$f(x) < f(x_0)$*)*
$x_0 \leftarrow x$
else
$r \leftarrow U[0,1]$
if($r < [(f(x_0) - f(x))/T]$***) then***
$x_0 \leftarrow x$
end if
end if
$T \leftarrow \alpha(T)$
Next *итерация*
3 *T=исходная температура>0*
α *(T)=функция охлаждения* $\alpha(T) \in [0,T]$ для всех T
Инициализировать кандидатное решение x_0 для минимизационной задачи $f(x)$
While not *(критерий останова)*
Сгенерировать кандидатное решение x
if *(*$f(x) < f(x_0)$*)*
$x_0 \leftarrow x$
else
$r \leftarrow U[0,1]$
if($r < exp[(f(x_0) - f(x))/T]$***) then***
$x_0 \leftarrow x$
end if

end if
$T \leftarrow \alpha(T)$

Next *итерация*

4 *T=исходная температура>0*

α *(T)=функция охлаждения* $\alpha(T) \in [0,T]$ для всех T

Инициализировать кандидатное решение x_0 для минимизационной задачи $f(x)$

While not *(критерий останова)*

Сгенерировать кандидатное решение x

if *(*$f(x) < f(x_0)$*)*

$x_0 \leftarrow x$

else

$r \leftarrow U[0,1]$

if($r < [(f(x_0) - f(x))/T]$***) then***

then

$x_0 \leftarrow x$

end if

end if

$T \leftarrow \alpha(T)$

Next *итерация*

11 Какое правильное математическое описание режима обратного линейного охлаждения?

1 $\alpha(T) = T_0 / k$

2 $\alpha(T) = T/(1+\beta T)$

3 $\alpha(T) = \max(T_0 - \eta k, \mathrm{T_{min}})$

4 $\alpha(T) = c/\ln(k+d)$

12 Какое правильное математическое описание режима размерно-зависимого охлаждения?

1 $$f(x) = 20 + e - 20\exp(-0.2\sum_{i=1}^{n} y_i^2/n) + \exp(\sum_{i=1}^{n}\cos 2\pi y_i)/n$$

$$y_i = \begin{cases} x_i \text{ для нечётного } i \\ x_i/4 \text{ для чётного } i \end{cases}$$

2 $$f(x) = 20 + e - 20\exp(-0.2\sum_{i=1}^{n} y_i^2/n) - \exp(\sum_{i=1}^{n}\cos 2\pi y_i)/n$$

$$y_i = \begin{cases} x_i \text{ для нечётного } i \\ x_i/4 \text{ для чётного } i \end{cases}$$

3 $$f(x) = 20 + e + 20\exp(-0.2\sum_{i=1}^{n} y_i^2/n) - \exp(\sum_{i=1}^{n}\cos 2\pi y_i)/n$$

$$y_i = \begin{cases} x_i \text{ для нечётого } i \\ x_i / 4 \text{ для чётного } i \end{cases}$$

4 $$f(x) = 20 - e - 20\exp(-0.2 \sum_{i=1}^{n} y_i^2 / n) - \exp(\sum_{i=1}^{n} \cos 2\pi y_i) / n$$

$$y_i = \begin{cases} x_i \text{ для нечётого } i \\ x_i / 4 \text{ для чётного } i \end{cases}$$

13 Какое правильное описание математической модели нанесения и испарения феромонов?

1 $$P_1 = \frac{(m_1 - k)^h}{(m_1 + k)^h - (m_2 + k)^h}$$

2 $$P_1 = \frac{(m_1 - k)^h}{(m_1 - k)^h - (m_2 - k)^h}$$

3 $$P_1 = \frac{(m_1 + k)^h}{(m_1 - k)^h + (m_2 - k)^h}$$

4 $$P_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h}$$

14 Какой правильный простой алгоритм муравьиной системы?

1 *n=число городов*

α, β = относительная важность феромонов относительно эвристической информации

Q=константа выделения феромона

ρ =скорость испарения $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон между городами i и j для* $i \in [1,n]$ *и* $j \in [1,n]$

d_{ij} *= расстояние между городами i и j для* $i \in [1,n]$ *и* $j \in [1,n]$

While not *(критерий останова)*

For *q=1* ***to*** *n*

For each *муравей* $k \in [1,n]$

Инициализировать стартовый город c_{k1} *каждого муравья* $k \in [1,n]$

Инициализировать множество городов, посещённых муравьём k:

$C_k \leftarrow \{c_{k1}\}, k \in [1,n]$

For each *город* $j \in [1,n]$, $j \notin C_k$

Вероятность $$p_{ij}^k \leftarrow (\tau_{ij}^{\alpha} / d_{ij}^{\beta}) / (\sum_{m=1, m \in C_k}^{n} \tau_{im}^{\alpha} / d_{im}^{\beta})$$

Next *j*

Пусть муравей k идёт в город j с вероятностью p_{ij}^k

Использовать $c_{k,q+1}$ *для обозначения города, отобранного в предыдущей строке* $C_k \leftarrow C_k \cup \{c_{k,q+1}\}$

***Next** муравей*

***Next** q*

L_k *<– длина общего пути, сконструированного муравьём k, для* $k \in [1, N]$

***For each** город* $i \in [1, n]$ *и город* $j \in [1, n]$

***For each** муравей* $k \in [1, N]$

***if** муравей* $k \in [1, N]$ *прошел из города i в город j*

$\Delta\tau_{ij}^{k} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{k} \leftarrow 0$

End if

***Next** муравей*

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$$

***Next** пара городов*

***Next** поколение*

2 *n=число городов*

α, β = относительная важность феромонов относительно эвристической информации

Q=константа выделения феромона

ρ =скорость испарения $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон между городами i и j для* $i \in [1, n]$ *и* $j \in [1, n]$

d_{ij} *= расстояние между городами i и j для* $i \in [1, n]$ *и* $j \in [1, n]$

***While not** (критерий останова)*

***For** q=1 **to** n*

***For each** муравей* $k \in [1, n]$

Инициализировать стартовый город c_{k1} *каждого муравья* $k \in [1, n]$

Инициализировать множество городов, посещённых муравьём k:

$C_k \leftarrow \{c_{k1}\}, k \in [1, n]$

***For each** город* $j \in [1, n]$, $j \notin C_k$.

$$p_{ij}^{k} \leftarrow (\tau_{ij}^{\alpha} / d_{ij}^{\beta}) / (\sum_{m=1, m \in C_k}^{n} \tau_{im}^{\alpha} / d_{im}^{\beta})$$

Вероятность

***Next** j*

Пусть муравей k идёт в город j с вероятностью p_{ij}^{k}

Использовать $c_{k,q+1}$ *для обозначения города, отобранного в предыдущей строке* $C_k \leftarrow C_k \cup \{c_{k,q+1}\}$

Next *муравей*

Next *q*

L_k *<– длина общего пути, сконструированного муравьём k, для* $k \in [1, N]$

For each *город* $i \in [1, n]$ *и город* $j \in [1, n]$

For each *муравей* $k \in [1, N]$

if *муравей* $k \in [1, N]$ *прошел из города i в город j*

$\Delta\tau_{ij}^{k} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{k} \leftarrow 0$

End if

Next *муравей*

$$\tau_{ij} \leftarrow (1+\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$$

Next *пара городов*

Next *поколение*

3 *n=число городов*

α, β = относительная важность феромонов относительно эвристической информации

Q=константа выделения феромона

ρ =скорость испарения $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон между городами i и j для* $i \in [1, n]$ *и* $j \in [1, n]$

d_{ij} *= расстояние между городами i и j для* $i \in [1, n]$ *и* $j \in [1, n]$

While not *(критерий останова)*

For *q=1* ***to*** *n-1*

For each *муравей* $k \in [1, n]$

Инициализировать стартовый город c_{k1} *каждого муравья* $k \in [1, n]$

Инициализировать множество городов, посещённых муравьём k:

$C_k \leftarrow \{c_{k1}\}, k \in [1, n]$

For each *город* $j \in [1, n]$, $j \notin C_k$

Вероятность $$p_{ij}^{k} \leftarrow (\tau_{ij}^{\alpha} / d_{ij}^{\beta}) / (\sum_{m=1, m \in C_k}^{n} \tau_{im}^{\alpha} / d_{im}^{\beta})$$

Next *j*

Пусть муравей k идёт в город j с вероятностью p_{ij}^{k}

Использовать $c_{k,q+1}$ *для обозначения города, отобранного в предыдущей строке* $C_k \leftarrow C_k \cup \{c_{k,q+1}\}$

***Next** муравей*

***Next** q*

L_k *<– длина общего пути, сконструированного муравьём k, для* $k \in [1, N]$

***For each** город* $i \in [1, n]$ *и город* $j \in [1, n]$

***For each** муравей* $k \in [1, N]$

***if** муравей* $k \in [1, N]$ *прошел из города i в город j*

$\Delta\tau_{ij}^{k} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{k} \leftarrow 0$

End if

***Next** муравей*

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$$

***Next** пара городов*

***Next** поколение*

4 *n=число городов*

α, β = относительная важность феромонов относительно эвристической информации

Q=константа выделения феромона

ρ =скорость испарения $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон между городами i и j для* $i \in [1, n]$ *и* $j \in [1, n]$

d_{ij} *= расстояние между городами i и j для* $i \in [1, n]$ *и* $j \in [1, n]$

***While not** (критерий останова)*

***For** q=1 **to** n-1*

***For each** муравей* $k \in [1, n]$

Инициализировать стартовый город c_{k1} *каждого муравья* $k \in [1, n]$

Инициализировать множество городов, посещённых муравьём k:

$C_k \leftarrow \{c_{k1}\}, k \in [1, n]$

***For each** город* $j \in [1, n]$, $j \notin C_k$

$$p_{ij}^{k} \leftarrow (\tau_{ij}^{\alpha} / d_{ij}^{\beta}) / (\sum_{m=1, m \in C_k}^{n} \tau_{im}^{\alpha} / d_{im}^{\beta})$$

Вероятность

***Next** i*

Пусть муравей k идёт в город j с вероятностью p_{ij}^{k}

Использовать $c_{k,q+1}$ *для обозначения города, отобранного в предыдущей строке* $C_k \leftarrow C_k \bigcup \{c_{k,q+1}\}$

Next *муравей*

Next *q*

L_k *<– длина общего пути, сконструированного муравьём k, для* $k \in [1, N]$

For each *город* $i \in [1, n]$ *и город* $j \in [1, n]$

For each *муравей* $k \in [1, N]$

if *муравей* $k \in [1, N]$ *прошел из города i в город j*

$\Delta\tau_{ij}^{k} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{k} \leftarrow 0$

End if

Next *муравей*

$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$

Next *пара городов*

Next *поколение*

15 Какой правильный алгоритм непрерывной оптимизации на основе муравьиной кучи?

1 *n=число размерностей*

Разделить i-ю размерность на Bi–1 интервала согласно уравнению $X_i \in [x_{i\,\min}, x_{i,\max}]$; $X_{i,\min} = b_{i1} < b_{i2} < \ldots < b_{i,B_i} = X_{i,\max}$

α *=важность количества феромона*

Q=константа выделения феромона

ρ *скорость испарения,* $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон для* $i \in [1, n]$ $j \in [1, B_i - 1]$

Случайно инициализировать популяцию муравьев (кандидатные решения) ak $k \in [1, N]$

While not *(критерий останова)*

For each *муравей ak*, $k \in [1, N]$

For each *размерность* $i \in [1, n]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], \mathrm{j} \in [1, \mathrm{B_i} - 1]$

Вероятность $p_{ij}^{(k)} \leftarrow \tau_{ij}^{\alpha} / \sum_{m=1}^{B_i - 1} \tau_{im}^{\alpha}$

Next *дискретизированный интервал*

$a_k(x) \leftarrow \bigcup [b_{ij}, b_{ij+1}]$ *с вероятностью* $p_{ij}^{(k)}$

Next *размерность*

Next *муравей*

$L_k \leftarrow$ *стоимость решения, сконструированного муравьём* a_k, $k \in [1, N]$

For each *размерность* $i \in [1, N]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], j \in [1, B_i - 1]$

For each *муравей* a_k, $k \in [1, N]$

if $a_k(x_i) \in [b_{ij}, b_{ij+1}]$

$\Delta\tau_{ij}^{(k)} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{(k)} \leftarrow 0$

End if *муравей*

Next *муравей*

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$$

Next *дискретизированный интервал*

Next *размерность*

Next *поколение*

2 *n=число размерностей*

Разделить i-ю размерность на Bi–1 интервала согласно уравнению $X_i \in [x_{i\min}, x_{i,\max}]$; $X_{i,\min} = b_{i1} < b_{i2} < \ldots < b_{i,B_i} = X_{i,\max}$

α *=важность количества феромона*

Q=константа выделения феромона

ρ *скорость испарения,* $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон для* $i \in [1, n]$ $j \in [1, B_i - 1]$

Случайно инициализировать популяцию муравьев (кандидатные решения) ak $k \in [1, N]$

While not *(критерий останова)*

For each *муравей ak*, $k \in [1, N]$

For each *размерность* $j \in [1, n]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], j \in [1, B_i - 1]$

Вероятность $p_{ij}^{(k)} \leftarrow \tau_{ij}^{\alpha} / \sum_{m=1}^{B_i - 1} \tau_{im}^{\beta}$

Next *дискретизированный интервал*

$a_k(x) \leftarrow \bigcup [b_{ij}, b_{ij+1}]$ *с вероятностью* $p_{ij}^{(k)}$

Next *размерность*

Next *муравей*

$L_k \leftarrow$ *стоимость решения, сконструированного муравьём* a_k, $k \in [1, N]$

For each *размерность* $i \in [1, N]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], j \in [1, B_i - 1]$

For each *муравей* a_k, $k \in [1, N]$

if $a_k(x_i) \in [b_{ij}, b_{ij+1}]$

$\Delta\tau_{ij}^{(k)} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{(k)} \leftarrow 0$

End if *муравей*

Next *муравей*

$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$

Next *дискретизированный интервал*

Next *размерность*

Next *поколение*

3 *n=число размерностей*

Разделить i-ю размерность на Bi–1 интервала согласно уравнению $X_i \in [x_{i\min}, x_{i,\max}]$; $X_{i,\min} = b_{i1} < b_{i2} < \ldots < b_{i,B_i} = X_{i,\max}$

α *=важность количества феромона*

Q=константа выделения феромона

ρ *скорость испарения,* $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон для* $i \in [1, n]$ $j \in [1, B_i - 1]$

Случайно инициализировать популяцию муравьев (кандидатные решения) ak $k \in [1, N]$

While not *(критерий останова)*

For each *муравей ak*, $k \in [1, N]$

For each *размерность* $i \in [1, n]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], j \in [1, B_i - 1]$

Вероятность $p_{ij}^{(k)} \leftarrow \tau_{ij}^{\alpha} / \sum_{m=1}^{B_i - 1} \tau_{im}^{\alpha}$

Next *дискретизированный интервал*

$a_k(x) \leftarrow \cap[b_{ij}, b_{ij+1}]$ *с вероятностью* $p_{ij}^{(k)}$

Next *размерность*

Next *муравей*

$L_k \leftarrow$ *стоимость решения, сконструированного муравьём* a_k, $k \in [1, N]$

For each *размерность* $i \in [1, N]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], \mathrm{j} \in [1, \mathrm{B_i} - 1]$

For each *муравей* $a_k{}'$ $k \in [1, N]$

if $a_k(x_i) \in [b_{ij}, b_{ij+1}]$

$\Delta\tau_{ij}^{(k)} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{(k)} \leftarrow 0$

End if *муравей*

Next *муравей*

$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} * \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$

Next *дискретизированный интервал*

Next *размерность*

Next *поколение*

4 *n=число размерностей*

Разделить i-ю размерность на Bi–1 интервала согласно уравнению $X_i \in [x_{i\min}, x_{i,\max}]$; $X_{i,\min} = b_{i1} < b_{i2} < \ldots < b_{i,B_i} = X_{i,\max}$

α *=важность количества феромона*

Q=константа выделения феромона

ρ *скорость испарения,* $\rho \in (0,1)$

$\tau_{ij} = \tau_0$ *исходный феромон для* $i \in [1, n]$ $j \in [1, B_i - 1]$

Случайно инициализировать популяцию муравьев (кандидатные решения) ak $k \in [1, N]$

While not *(критерий останова)*

For each *муравей ak'* $k \in [1, N]$

For each *размерность* $i \in [1, n]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], \mathrm{j} \in [1, \mathrm{B_i} - 1]$

Вероятность $p_{ij}^{(k)} \leftarrow \tau_{ij}^{\alpha} / \sum_{m=1}^{B_i+1} \tau_{im}^{\alpha}$

Next *дискретизированный интервал*

$a_k(x) \leftarrow \bigcup[b_{ij}, b_{ij+1}]$ *с вероятностью* $p_{ij}^{(k)}$

Next *размерность*

Next *муравей*

$L_k \leftarrow$ *стоимость решения, сконструированного муравьём* $a_k{}'$ $k \in [1, N]$

For each *размерность* $i \in [1, N]$

For each *дискретизированный интервал* $[b_{ij}, b_{ij+1}], \mathrm{j} \in [1, \mathrm{B_i} - 1]$

For each *муравей* a_k, $k \in [1, N]$

if $a_k(x_i) \in [b_{ij}, b_{ij+1}]$

$\Delta\tau_{ij}^{(k)} \leftarrow Q / L_k$

else

$\Delta\tau_{ij}^{(k)} \leftarrow 0$

End if *муравей*

Next *муравей*

$\tau_{ij} \leftarrow (1+\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij}^{(k)}$

Next *дискретизированный интервал*

Next *размерность*

Next *поколение*

16 Какой правильный базовый алгоритм оптимизации на основе роя частиц?

1 *Инициализировать случайную популяцию особей* $\{x_i\}$, $i \in [1, N]$

Инициализировать n-элементный скоростной вектор каждой особи v_i, $i \in [1, N]$

Инициализировать лучшее к настоящему моменту положение каждой особи $b_i \leftarrow x_i$ $i \in [1, N]$

Задать размер окрестности $\sigma > N$

Задать значение максимального влияния $\phi_{1,\max}$ *и* $\phi_{2,\max}$

Задать максимальную скорость $v_{\max}$

While not *(критерий останова)*

For each *особь* x_i $i \in [1, N]$

$H_i \leftarrow \{\sigma \text{ ближайших соседей для } x_i\}$

$h_i \leftarrow \arg\min_x \{f(x) : x \in H\}$

Сгенерировать случайный вектор ϕ_1, *где* $\phi_1(k) \sim U[0, \phi_{1,\max}]$ *для* $i \in [1, N]$

Сгенерировать случайный вектор ϕ_2, *где* $\phi_2(k) \sim U[0, \phi_{2,\max}]$ *для* $i \in [1, N]$

$v_i \leftarrow v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i)$

If $|v_i| > v_{\max}$ ***then***

$v_i \leftarrow v_i v_{\max} / |v_i|$

End if

$x_i \leftarrow x_i + v_i$

$b_i \leftarrow \arg\min_x \{f(x_i) : f(b)\}$

Next *особь*

Next *поколение*

2 *Инициализировать случайную популяцию особей* $\{x_i\}$, $i \in [1, N]$

Инициализировать n-элементный скоростной вектор каждой особи v_i, $i \in [1, N]$

Инициализировать лучшее к настоящему моменту положение каждой особи $b_i \leftarrow x_i$ $i \in [1, N]$

Задать размер окрестности $\sigma < N$

Задать значение максимального влияния $\phi_{1,\max}$ *и* $\phi_{2,\max}$

Задать максимальную скорость $v_{\max}$

While not *(критерий останова)*

For each *особь* x_i $i \in [1, N]$

$H_i \leftarrow \{\sigma$ ближайших соседей для $x_i\}$

$h_i \leftarrow \arg\min_x \{f(x) : x \in H\}$

Сгенерировать случайный вектор ϕ_1, *где* $\phi_1(k) \sim U[0, \phi_{1,\max}]$ *для* $i \in [1, N]$

Сгенерировать случайный вектор ϕ_2, *где* $\phi_2(k) \sim U[0, \phi_{2,\max}]$ *для* $i \in [1, N]$

$v_i \leftarrow v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i)$

If $|v_i| > v_{\max}$ ***then***

$v_i \leftarrow v_i v_{\max} / |v_i|$

End if

$x_i \leftarrow x_i + v_i$

$b_i \leftarrow \arg\min_x \{f(x_i) : f(b)\}$

Next *особь*

Next *поколение*

3 *Инициализировать случайную популяцию особей* $\{x_i\}$, $i \in [1, N]$

Инициализировать n-элементный скоростной вектор каждой особи v_i, $i \in [1, N]$

Инициализировать лучшее к настоящему моменту положение каждой особи $b_i \leftarrow x_i$ $i \in [1, N]$

Задать размер окрестности $\sigma < N$

Задать значение максимального влияния $\phi_{1,\max}$ *и* $\phi_{2,\max}$

Задать максимальную скорость $v_{\max}$

While not *(критерий останова)*

For each *особь* x_i $i \in [1, N]$

$H_i \leftarrow \{\sigma$ ближайших соседей для $x_i\}$

$h_i \leftarrow \arg\min_x \{f(x) : x \in H\}$

Сгенерировать случайный вектор ϕ_1, *где* $\phi_1(k) \sim U[0, \phi_{1,\max}]$ *для* $i \in [1, N]$

Сгенерировать случайный вектор ϕ_2*, где* $\phi 2(k) \sim U[0, \phi_{2,\max}]$ *для* $i \in [1, N]$

$v_i \leftarrow v_i + \phi_1(b_i - x_i) + \phi_2(h_i + x_i)$

If $|v_i| > v_{\max}$ ***then***

$v_i \leftarrow v_i v_{\max} / |v_i|$

End if

$x_i \leftarrow x_i + v_i$

$b_i \leftarrow \arg\min_x \{f(x_i) : f(b)\}$

Next *особь*

Next *поколение*

4 *Инициализировать случайную популяцию особей* $\{x_i\}$, $i \in [1, N]$

Инициализировать n-элементный скоростной вектор каждой особи v_i, $i \in [1, N]$

Инициализировать лучшее к настоящему моменту положение каждой особи $b_i \leftarrow x_i$ $i \in [1, N]$

Задать размер окрестности $\sigma < N$

Задать значение максимального влияния $\phi_{1,\max}$ *и* $\phi_{2,\max}$

Задать максимальную скорость $v_{\max}$

While not *(критерий останова)*

For each *особь* x_i $i \in [1, N]$

$H_i \leftarrow \{\sigma \text{ ближайших соседей для } x_i\}$

$h_i \leftarrow \arg\min_x \{f(x) : x \in H\}$

Сгенерировать случайный вектор ϕ_1*, где* $\phi_1(k) \sim U[0, \phi_{1,\max}]$ *для* $i \in [1, N]$

Сгенерировать случайный вектор ϕ_2*, где* $\phi 2(k) \sim U[0, \phi_{2,\max}]$ *для* $i \in [1, N]$

$v_i \leftarrow v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i)$

If $|v_i| > v_{\max}$ ***then***

$v_i \leftarrow v_i v_{\max} / |v_i|$

End if

$x_i \leftarrow x_i - v_i$

$b_i \leftarrow \arg\min_x \{f(x_i) : f(b)\}$

Next *особь*

Next *поколение*

17 Что понимается под колёсной (wheel topology) топологией?

1 Топология, в которой каждая частица связана с четырьмя соседними частицами.

2 Топология, в которой в которой каждая частица полносвязна внутри своего кластера, в то время как несколько частиц в каждом кластере также связаны с дополнительной частицей в другом кластере.

3 Топология, в которой каждая частица соединена с двумя другими частицами.

4 Топология, в которой фокальная частица соединена со всеми другими частицами

18 Что понимается под кольцевой (ring topology) топологией?

1 Топология, в которой каждая частица связана с четырьмя соседними частицами.

2 Топология, в которой в которой каждая частица полносвязна внутри своего кластера, в то время как несколько частиц в каждом кластере также связаны с дополнительной частицей в другом кластере.

3 Топология, в которой каждая частица соединена с двумя другими частицами.

4 Топология, в которой фокальная частица соединена со всеми другими частицами

19 Какая правильная формула коррекции скорости в модифицированном алгоритме оптимизации на основе роя части с учётом инерционного взвешивания?

1 $v_i(k) \leftarrow wv_i(k) + \phi_1(k)(b_i(k) - x_i(k)) + \phi_2(k)(h_i(k) - x_i(k))$

2 $v_i(k) \leftarrow wv_i(k) + \phi_1(k)(b_i(k) - x_i(k)) - \phi_2(k)(h_i(k) - x_i(k))$

3 $v_i(k) \leftarrow wv_i(k) + \phi_1(k)(b_i(k) + x_i(k)) + \phi_2(k)(h_i(k) - x_i(k))$

4 $v_i(k) \leftarrow wv_i(k) - \phi_1(k)(b_i(k) - x_i(k)) + \phi_2(k)(h_i(k) - x_i(k))$

20 Какая правильная формула коррекции скорости в модифицированном алгоритме оптимизации на основе роя части с учётом коэффициента сужения?

1 $v_i(k) \leftarrow K[v_i + \phi_1(b_i - x_i) - \phi_2(h_i - x_i)]$

2 $v_i(k) \leftarrow K[v_i - \phi_1(b_i - x_i) + \phi_2(h_i - x_i)]$

3 $v_i(k) \leftarrow K[v_i + \phi_1(b_i - x_i) + \phi_2(h_i - x_i)]$

4 $v_i(k) \leftarrow K[v_i + \phi_1(b_i - x_i) + \phi_2(h_i + x_i)]$

6.4.3 Таблица ответов

№ вопроса	Правильные ответы	
	Вариант 1_1	Вариант 2_1
1	1	1
2	4	4
3	2	3
4	1	1
5	3	2
6	3	2
7	1	3
8	4	2
9	2	1

10	1	3
11	2	1
12	3	2
13	1	4
14	1	3
15	2	1
16	4	2
17	2	4
18	1	3
19	4	1
20	1	3

6.4.4 Вопросы к экзамену

1 Предмет и задачи дисциплины «Структуры и алгоритмы обработки» данных. Основные понятия и определения. Классификация математических моделей. Этапы математического моделирования.

2 Общая постановка задачи линейного программирования. Формы записи математической модели задачи линейного программирования.

3 Геометрический метод решения задачи линейного программирования. Пример.

4 Симплекс метод в решении задач линейного программирования. Метод Гаусса-Жордана. Пример.

5 Общая постановка задачи целочисленного линейного программирования. Основные понятия и определения. Метод отсечения плоскостей (Р. Гомори).

6 Общая постановка задачи целочисленного линейного программирования. Основные понятия и определения. Метод ветвей и границ.

7 Общая постановка задачи нелинейного программирования. Основные понятия и определения Теорема Куна-Такера.

8 Безусловная оптимизация. Методы нулевого порядка. Метод деления интервала пополам. Постановка задачи. Стратегия поиска. Алгоритм метода. Сходимость.

9 Безусловная оптимизация. Методы нулевого порядка. Алгоритм Свенна. Метод золотого сечения. Постановка задачи. Стратегия поиска. Алгоритм метода. Сходимость.

10 Безусловная оптимизация. Методы нулевого порядка. Метод сопряжённых направлений. Постановка задачи. Стратегия поиска. Алгоритм метода.

11 Безусловная оптимизация. Методы первого порядка. Метод градиентного спуска с постоянным шагом. Постановка задачи. Стратегия поиска. Алгоритм метода. Сходимость. Скорость сходимости.

12 Безусловная оптимизация. Методы первого порядка. Метод наискорейшего градиентного спуска. Постановка задачи. Стратегия поиска. Алгоритм метода. Сходимость. Скорость сходимости. Модификации метода Ньютона.

13 Безусловная оптимизация. Методы второго порядка. Метод Ньютона. Постановка задачи. Стратегия поиска. Алгоритм метода. Сходимость.

14 Безусловная оптимизация. Методы второго порядка. Метод Ньютона-Рафсона. Постановка задачи. Стратегия поиска. Алгоритм метода. Сходимость.

15 Безусловная оптимизация. Методы второго порядка. Метод Марквардта. Постановка задачи. Стратегия поиска. Алгоритм метода.

16 Принципы построения численных методов поиска условного экстремума. Метод штрафов. Постановка задачи. Стратегия поиска. Алгоритм метода. Сходимость.

17 Принципы построения численных методов поиска условного экстремума. Метод Зойтендейка. Постановка задачи. Стратегия поиска. Алгоритм метода.

18 Решение задач с помощью методов численной оптимизации. Среда программирования. Технология программирования. Пример.

19 Эволюционная оптимизация. Основные понятия и определения. Простой генетический алгоритм. Теорема о схемах Д. Холланда. Отличия от традиционных алгоритмов оптимизации. Преимущества и недостатки. Область применения.

20 Эволюционная оптимизация. Основные компоненты генетических алгоритмов. Базовая структура генетического алгоритма. Методы отбора. Методы скрещивания. Методы мутации. Элитизм.

21 Метод отжига (Simulated annealing, SA). Алгоритм. Режимы охлаждения. Модификация базового алгоритма симулированного закаливания. Преимущества и недостатки. Область применения.

22 Алгоритмы муравья (Ant algorithms). Модели феромона. Алгоритм. Модификации алгоритма муравья: минимаксная муравьиная система (max-min ant system, MMAS) и система муравьиной кучи (ant colony system, ACS). Преимущества и недостатки. Область применения.

23 Оптимизация на основе роя частиц (Particle swarm optimization, PSO). Базовый алгоритм. Параметры алгоритма. Топология роя частиц. Инерционное взвешивание. Коэффициент сужения. Модификации алгоритма оптимизации на основе роя частиц: алгоритм оптимизации на основе роя частиц с отрицательным подкреплением (negative reinforcement, NPSO) и алгоритм оптимизации на основе роя частиц с использованием стимулятора-зубатки (catfish PSO). Преимущества и недостатки. Область применения.

24 Решение задач с помощью генетических алгоритмов. Библиотека DEAP. Модули creator, toolbox. Пример.

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная учебная литература

1. . Дэн Саймон Алгоритмы эволюционной оптимизации./ пер с англ. А. В. логунова. – Москва: ДМК Пресс, 2020. – 940 с.
2. Гладков А. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы/Под ред. В. М. Курейчика. – Москва: Физматлит, 2016. – 368 с.
3. Optimization for machine learning. Edited by Suvrit Sra, Sebastian Nowozin, Stephen J. Wright. The MIT Press, 2012, 494 p.
4. .Эйал Вирсански. Генетические алгоритмы на Python. – Москва: ДМК Пресс, 2020. – 286 с
5. . М. Т. Джонс. Программирование искусственного интеллекта в приложениях/Пер. с англ. Осипов А. И. – Москва: ДМК Пресс, 2013. – 312 с.

7.2. Дополнительная учебная литература

1. Пантелеев А. В. Методы оптимизации. Практический курс: учебное пособие/ А. В. Пантелеев, Т. А. Летова. – Москва: Логос, 2017. – 424 с.
2. Матрёнин П. В. Методы стохастической оптимизации: учебное пособие/П. В. Матрёнин, М. Г. Гриф, В. Г. Секаев. – Новосибирск: Из-во НГТУ, 2016. – 67 с.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

1. Семахин А.М. Структуры и алгоритмы обработки данных. Методические указания к выполнению лабораторных и контрольных работ для магистрантов направления подготовки 09.04.04 «Программная инженерия». Курган, КГУ, 2020. – 46 с. (электронный).
2. Семахин А. М. Структуры и алгоритмы обработки данных: учебное пособие. – Курган : Изд-во КГУ, 2020 – 54 с. (электронный).

9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Федеральный портал «Российское образование» URL: http://www.edu.ru/
2. Сайт дистанционного обучения в НОУ «ИНТУИТ». URL: http://www.intuit.ru/

10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

При чтении лекций используются слайдовые презентации.
Минимальные требования к операционной системе и программному обеспечению компьютера, используемого при показе слайдовых презентаций: Windows XP, Foxit PDF Reader 12.1, 13.12.2022 г., свободное программное обеспечение (free software), GNU License.

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

11.1 Техническое обеспечение

№	Наименование	Использование
1	Комплект: ноутбук, медиа-проектор, экран	Для демонстрации иллюстративного материала при чтении лекций.
2	Персональный компьютер стандартной комплектации	Используется в качестве инструмента и объекта исследования при выполнении лабораторных работ.

11.2 Программное обеспечение

№	Наименование	Использование
1	Операционная система Windows 10	Управление устройствами компьютерной системы и обеспечение удобного интерфейса для работы.
2	Интегрированная среда программирования Microsoft Visual Studio 2019 Community, языки программирования Visual C++, Visual C#	Формализация алгоритмов решения задач при выполнении лабораторных работ
3	Среда программирования на языке Python, библиотеки NumPy, TensorFlow, Mayavi	Формализация алгоритмов решения задач при выполнении лабораторных работ

12. ДЛЯ ОБУЧАЮЩИХСЯ С ИСПОЛЬЗОВАНИЕМ ДИСТАНЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) занятия полностью или частично проводятся в режиме онлайн. Объем дисциплины и распределение нагрузки по видам работ соответствует п. 4.1. Распределение баллов соответствует п. 6.2 либо может быть изменено в соответствии с решением кафедры, в случае перехода на ЭО и ДОТ в процессе обучения. Решение кафедры об используемых технологиях и системе оценивания достижений обучающихся принимается с учетом мнения ведущего преподавателя и доводится до обучающихся.

ПРИЛОЖЕНИЕ 1

Аннотация
рабочей программы учебной дисциплины

Структуры и алгоритмы обработки данных

образовательной программы высшего образования – программы магистратуры

09.04.04 Программная инженерия
направленность
Методы и алгоритмы интеллектуальной обработки данных в информационно-вычислительных системах

формы обучения – очная

Трудоемкость освоения дисциплины – 4 зач. ед. (144 акад. часа)

Семестры: 1-й (для очной формы обучения)

Форма промежуточной аттестации: Экзамен

Содержание дисциплины

Дисциплина «Структуры и алгоритмы обработки данных» включена в модуль «Анализ данных и машинное обучение» обязательной части блока 1 учебного плана. Для освоения дисциплины необходимы компетенции в области программирования, типовых структур данных и анализа алгоритмов, формируемые соответствующими дисциплинами программ бакалавриата или специалитета. Результаты изучения дисциплины используются при освоении профильных дисциплин, включенных в модули «Анализ данных и машинное обучение», «Технологии распределённой обработки данных» и «Прикладные задачи интеллектуального анализа данных».

Основная цель изучения дисциплины – формирование теоретических знаний алгоритмов методов эволюционной оптимизации и оптимизации машинного обучения и приобретение практических навыков формализации их на ПЭВМ с помощью языков высокого уровня.

Задачи дисциплины:
изучение:

- алгоритмов градиентных методов оптимизации;
- алгоритмов эволюционной оптимизации;

- базовых форм и механизмов генетической изменчивости организмов, законов и принципов популяционной генетики и эволюционной изменчивости;
- математических моделей процесса эволюции и стратегий генетического поиска;
- базовых принципов и основных подходов к построению совместных схем локального и генетического поиска оптимальных решений;
- архитектуры и стратегии генетического поиска оптимальных решений.

практическое освоение:

- среды программирования на языке Python, библиотек NumPy, TensorFlow;
- среды программирования Microsoft Visual Studio Community 2019, языков программирования Visual C++, VisualC#;
- методики решения задач с применением методов эволюционной оптимизации и оптимизации машинного обучения.