ПРИЛОЖЕНИЕ А
ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ «Интеллектуальные системы»

1. Перечень оценочных средств для компетенций, формируемых в результате освоения дисциплины

Код контролируемой компетенции	Способ оценивания	Оценочное средство
ОПК-1: Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте	Курсовая работа; экзамен	Контролирующие материалы для защиты курсовой работы; комплект контролирующих материалов для экзамена
ОПК-2: Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	Курсовая работа; экзамен	Контролирующие материалы для защиты курсовой работы; комплект контролирующих материалов для экзамена

2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Оцениваемые компетенции представлены в разделе «Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций» рабочей программы дисциплины «Интеллектуальные системы».

При оценивании сформированности компетенций по дисциплине «Интеллектуальные системы» используется 100-балльная шкала.

Критерий	Оценка по 100-балльной шкале	Оценка по традиционной шкале
Студент освоил изучаемый материал (основной и дополнительный), системно и грамотно излагает его, осуществляет полное и правильное выполнение заданий в соответствии с индикаторами достижения компетенций, способен ответить на дополнительные вопросы.	75-100	*Отлично*
Студент освоил изучаемый материал, осуществляет выполнение заданий в соответствии с индикаторами достижения компетенций с непринципиальными ошибками.	50-74	*Хорошо*
Студент демонстрирует освоение только основного материала, при	25-49	*Удовлетворительно*

выполнении заданий в соответствии с индикаторами достижения компетенций допускает отдельные ошибки, не способен систематизировать материал и делать выводы.		
Студент не освоил основное содержание изучаемого материала, задания в соответствии с индикаторами достижения компетенций не выполнены или выполнены неверно.	<25	*Неудовлетворительно*

3. Типовые контрольные задания или иные материалы, необходимые для оценки уровня достижения компетенций в соответствии с индикаторами

1.Применение интеллектуальных технологий для разработки алгоритмов и программ

Компетенция	Индикатор достижения компетенции
ОПК-1 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте	ОПК-1.1 Приобретает и применяет математические, естественнонаучные, социально-экономические и профессиональные знания для решения задач
ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1 Обосновывает выбор и использует современные информационно-коммуникационные и интеллектуальные технологии для решения профессиональных задач
	ОПК-2.2 Разрабатывает оригинальные алгоритмы и программные средства

Задание 1

Реализовать на системе программирования Python классический генетический алгоритм. Исходные данные вводятся с формы, выводится на график значение приспособленности лучшего организма в текущем поколении. Сгенерировать исходные данные задачи для достаточно большой размерности, чтобы невозможно было найти точное решение. Выполнить отладочный вариант для небольшой размерности с целью проверки работы программы.

Применяя математические и профессиональные знания, выбрать и использовать интеллектуальные технологии для разработки оригинального алгоритма и программы для решения следующей задачи. Дан неориентированный граф. Требуется выбрать из множества всех вершин наименьшее количество, чтобы любое ребро графа было инцидентно хотя бы одной выбранной вершине (опиралось хотя бы одним концом на одну из выбранных вершин). Организмом является строка из n бит 011000,…001. Единица - соответствующая вершина входит выбранный набор, нуль - нет. В приближенном решении требуется минимизировать два показателя: брать как можно меньше вершин (число единиц в строке) и оставлять как можно меньше ребер, не инцидентных и к одной из выбранных вершин. Учитывая возможность различной важности критериев, функцию приспособленности можно записать в виде:

F=A1*K+A2*M. K - число единиц в строке, M - число ребер, не инцидентных ни одной выбранной вершине. Организм F1 более приспособлен чем F2, если у него значение F меньше. A1 и A2 взять по 0,5. Для генерирования задать число вершин V и ребер W, построить симметричную матрицу смежности [VxV] из нулей и единиц, единица означает, что соответствующие вершины соединены ребром. Всего единиц 2W.

Задание 2

Реализовать на системе программирования Python классический генетический алгоритм. Исходные данные вводятся с формы, выводится на график значение приспособленности лучшего организма в текущем поколении. Сгенерировать исходные данные задачи для достаточно большой размерности, чтобы невозможно было найти точное решение. Выполнить отладочный вариант для небольшой размерности с целью проверки работы программы.

Применяя математические и профессиональные знания, выбрать и использовать интеллектуальные технологии для разработки оригинального алгоритма и программы для решения следующей задачи. Дан неориентированный граф. Требуется выбрать из множества всех вершин наименьшее

количество, чтобы любая вершина, не входящая в выбранное множество была соединена по крайней мере одним ребром с какой-нибудь другой из выбранных вершин.

Организмом является строка из n бит 011000,…001. Единица - соответствующая вершина входит выбранный набор, нуль - нет.

В приближенном решении требуется минимизировать два показателя: брать как можно меньше вершин (число единиц в строке) и чтобы при этом как можно меньше оставшихся вершин были не соединены с выбранными.

Учитывая возможность различной важности критериев, функцию приспособленности можно записать в виде:

F=A1*K+A2*M. K - число единиц в строке, M - число вершин из оставшихся, не соединенных ни с одной из выбранных. Организм F1 более приспособлен чем F2, если у него значение F меньше. A1 и A2 взять по 0,5. Для генерирования задать число вершин V и ребер W, построить симметричную матрицу смежности [VxV] из нулей и единиц, единица означает, что соответствующие вершины соединены ребром. Всего единиц 2W.

Задание 3

Реализовать на системе программирования Python классический генетический алгоритм. Исходные данные вводятся с формы, выводится на график значение приспособленности лучшего организма в текущем поколении. Сгенерировать исходные данные задачи для достаточно большой размерности, чтобы невозможно было найти точное решение. Выполнить отладочный вариант для небольшой размерности с целью проверки работы программы.

Применяя математические и профессиональные знания, выбрать и использовать интеллектуальные технологии для разработки оригинального алгоритма и программы для решения следующей задачи. Дан неориентированный граф. Требуется выбрать из множества всех вершин наибольшее количество, чтобы любые две вершины из выбранных были соединены ребром.

Организмом является строка из n бит 011000,…001. Единица - соответствующая вершина входит выбранный набор, нуль - нет.

В приближенном решении требуется максимизировать разность числа взятых вершин и числа пар вершин, не соединенных ребром. Учитывая возможность различной важности критериев, функцию приспособленности можно записать в виде:

F=A1*K-A2*M. K - число единиц в строке, M - число пар вершин из выбранных, не соединенных между собой. Организм F1 более приспособлен чем F2, если у него значение F больше. A1 и A2 взять по 0,5. Для генерирования задать число вершин V и ребер W, построить симметричную матрицу смежности [VxV] из нулей и единиц, единица означает, что соответствующие вершины соединены ребром. Всего единиц 2W.

Задание 4

Применяя математические и профессиональные знания, выбрать и использовать интеллектуальные технологии для разработки оригинального алгоритма и программы для решения следующей задачи. Реализовать на системе программирования Python алгоритм Мамдами нечеткого вывода. Алгоритм содержит этапы:

1. фаззификация
2. агрегирование
3. активизация
4. аккумуляция
5. дефаззификация.

В итоге получается управляющее воздействи при заданных входных значениях.

Модель движения автомобиля по трассе.

Исходные данные для моделирования:

1. Входные величины – расстояние от автомобиля до препятствия (изменяется в пределах от 0до 500); скорость автомобиля (0-200).
2. Выходная величина – сила торможения (0-100).

Расстояние и скорость измеряются в лингвистических переменных очень мало (VS), мало(S), средне (M), велико (B), очень велико (VB). Сила торможения задается лингвистическими переменными близка к нулю (Z), четверть (OQ), половина (H), три четверти (TQ), полная (FU). Параметры входных и выходной переменных представлены в табл.

Параметры переменной «Скорость»

Имя переменной	Тип функции принадлежности	Диапазон изменения
Очень малая (VS)	Трапецеидальная	[0, 0, 20, 60]

Малая (S)	Треугольная	[20, 60, 100]
Средняя (M)	Треугольная	[60, 100, 140]
Большая (B)	Треугольная	[100, 140, 180]
Очень большая (VB)	Трапецеидальная	[140, 180, 200, 200]

Параметры переменной «Расстояние»

Имя переменной	Тип функции принадлежности	Диапазон изменения
Очень малое (VS)	Трапецеидальная	[0, 0, 50, 150]
Малое (S)	Треугольная	[50, 150, 250]
Среднее (M)	Треугольная	[150, 250, 350]
Большое (B)	Треугольная	[250, 350, 450]
Очень большое (VB)	Трапецеидальная	[350, 450, 500, 500]

Параметры переменной «Сила торможения»

Имя переменной	Тип функции принадлежности	Диапазон изменения
Близка к нулю (Z)	Трапецеидальная	[0, 0, 10, 30]
Четверть (OQ)	Треугольная	[10, 30, 50]
Половина (H)	Треугольная	[30, 50, 70]
Три четверти (TQ)	Треугольная	[50, 70, 90]
Полная (FU)	Трапецеидальная	[70, 80, 100, 100]

Список правил:

1. Если скорость очень малая и расстояние очень малое, то сила торможения равна половине.
2. Если скорость малая, а расстояние очень малое, то сила торможения равна трём четвяртям.
3. Если скорость средняя, а расстояние очень малое, то сила торможения полная.
4. Если скорость большая, а расстояние очень малое, то сила торможения полная.
5. Если скорость очень большая, а расстояние очень малое, то сила торможения полная.
6. Если скорость очень малая, а расстояние малое, то сила торможения равна одной четверти.
7. Если скорость малая и расстояние малое, то сила торможения равна половине.
8. Если скорость средняя, а расстояние малое, то сила торможения равна трём четвяртям.
9. Если скорость большая, а расстояние малое, то сила торможения полная.
10. Если скорость очень большая, а расстояние малое, то сила торможения полная.
11. Если скорость очень малая, а расстояние среднее, то сила торможения близка к нулю.
12. Если скорость малая, а расстояние среднее, то сила торможения равна одной четверти.
13. Если скорость средняя и расстояние среднее, то сила торможения равна половине.
14. Если скорость большая, а расстояние среднее, то сила торможения равна трём четвяртям.
15. Если скорость очень большая, а расстояние среднее, то сила торможения полная.
16. Если скорость очень малая, а расстояние большое, то сила торможения близка к нулю.
17. Если скорость малая, а расстояние большое, то сила торможения близка к нулю.
18. Если скорость средняя, а расстояние большое, то сила торможения равна одной четверти.

19. Если скорость большая и расстояние большое, то сила торможения равна половине.
20. Если скорость очень большая, а расстояние большое, то сила торможения равна трём четвяртям.
21. Если скорость очень малая, а расстояние очень большое, то сила торможения близка к нулю.
22. Если скорость малая, а расстояние очень большое, то сила торможения близка к нулю.
23. Если скорость средняя, а расстояние очень большое, то сила торможения близка к нулю.
24. Если скорость большая, а расстояние очень большое, то сила торможения равна одной четверти.
25. Если скорость очень большая и расстояние очень большое, то сила торможения равна половине.
26.
Входные значения
Скорость 80 Расстояние 300

Задание 5

Применяя математические и профессиональные знания, выбрать и использовать интеллектуальные технологии для разработки оригинального алгоритма и программы для решения следующей задачи. Реализовать на системе программирования Python алгоритм распознавания образов на основе байесовского метода.

Даны 3,4,5 изображений 6x6 символов F, L, C и два шаблона. Пусть поданы 4 образа, в которых отсутствуют шаблоны A и B, присутствует один из шаблонов и присутствуют оба шаблоа. Вычислить все апостериорные вероятности AB, A~B ~AB, ~A~B, в каждом случае по максимуму из них найти ответ экспертной системы.

F

```
. 1 1 1     . 1 1 1     . . 1 1     . 1 1 .
. 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 1 1     . 1 1 1     . 1 1 1     . 1 1 .
. 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 . .     . . . .     . 1 . .     . 1 . .
```

L

```
. 1 . .     . 1 . .     . 1 . .     . 1 . .     . . . .
. 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 . .     . 1 . .     . . . .     . 1 . .     . 1 . .
. 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 1 1     . . 1 1     . 1 1 1     . 1 1 .     . . 1 1
```

C

```
. 1 1 1     . 1 1 1     . 1 1 1     . . 1 1     . 1 1 .     . 1 1 .
. 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 . .     . 1 . .     . . . .     . 1 . .     . 1 . .     . 1 . .
. 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .
. 1 . .     . 1 . .     . 1 . .     . 1 . .     . 1 . .     . . . .
. 1 1 1     . . 1 1     . 1 1 1     . . 1 1     . 1 1 1     . 1 1 1
```

Шаблон A

```
1
1
1 1
```

Шаблон B
111
Первый образец, где элементы A и B отсутствуют.

```
. . . .
. 1 . .
. 1 . .
. 1 . .
. 1 . .
. . 1 1
```

Второй образец, где элемент A присутствует в образце, элемент B – нет.

```
. 1 1 .
. 1 . .
. 1 1 .
. 1 . .
. 1 . .
. 1 . .
```

Третий образец, где элемент B присутствует в образце, элемент A – нет.

```
. 1 1 1
. 1 . .
. 1 . .
. 1 . .
. 1 . .
. . 1 1
```

Четвертый образец, где присутствуют оба элемента в образце.

```
. 1 1 1
. 1 . .
. . . .
. 1 . .
. 1 . .
. 1 1 1
```

Задание 6

Применяя математические и профессиональные знания, выбрать и использовать интеллектуальные технологии для разработки оригинального алгоритма и программы для решения следующей задачи. На системе программирования Python реализовать экспертную систему, основанную на фактах и правилах по оценке продолжительности жизни. После чтения файлов factvalues.txt, questions.txt, rules.txt программа задает вопросы, ответы выбираются и система выводит доказанные утверждения с указанием правил, по которым они получены

Задание 7

Применяя математические и профессиональные знания, выбрать и использовать интеллектуальные технологии для разработки оригинального алгоритма и программы для решения следующей задачи. Реализовать на системе программирования Python алгоритм обратного распространения ошибки для битовых изображений 6x6 трех классов. После

обучения выводятся изображения с подписью истинного номера изображения всех классов и активности выходного слоя.
Символы: T X Z

4. Файл и/или БТЗ с полным комплектом оценочных материалов прилагается.