Государственное бюджетное образовательное учреждение
высшего образования Московской области
«Университет «Дубна»
Филиал «Протвино»
Кафедра «Информационные технологии»

В.В. Гусев, М.П. Астафьева

Подготовка и оформление курсовых работ по дисциплине «Теория принятия решений»

Электронное методическое пособие

Рекомендовано
кафедрой информационных технологий
филиала «Протвино» государственного университета «Дубна»
в качестве методического пособия для студентов,
обучающихся по направлениям
«Информатика и вычислительная техника» и
«Прикладная информатика»

Протвино
2017

ББК 22.18я73
Г96

Рецензент:
кандидат физико-математических наук,
начальник сектора прикладной математики отдела математики и вычислительной техники ФГБУ ИФВЭ им. А.А. Логунова НИЦ «Курчатовский институт»
Рябов А.Д.

Гусев, В.В.
Г96 Подготовка и оформление курсовых работ по дисциплине «Теория принятия решений»: электронное методическое пособие / В.В. Гусев, М.П. Астафьева. — Протвино: 2017. — 30с.

Предназначено для студентов очного и заочного отделений направлений «Информатика и вычислительная техника» и «Прикладная информатика».

В пособии рассматриваются правила написания, определяются требования к содержанию, структуре и оформлению курсовых работ, выполняемых на кафедре информационных технологий по дисциплине «Теория принятия решений». В пособии приводятся примеры выполнения курсовых работ.

ББК 22.18я73

© Государственное бюджетное образовательное учреждение высшего образования Московской области «Университет «Дубна», филиал «Протвино», 2017
© Гусев В.В., Астафьева М.П.

СОДЕРЖАНИЕ

Введение

Курсовая работа является важнейшим элементом самостоятельной работы студентов. Ее основная цель — создать и развить навыки исследовательской работы, умение работать с научной литературой, на основе ее изучения делать выводы и обобщения.

Темы курсовых работ разрабатываются преподавателем дисци-плины и утверждаются заведующим кафедрой в течение первого месяца семестра, в котором они должны быть написаны. После этого студенты выбирают тему из предложенного списка. Студент, по согласованию с руководителем, может изменить или предло-жить свою тему курсовой работы. Разработка одной темы не-сколькими студентами допускается, в том случае, если тема носит комплексный характер, и каждый студент работает над отдельной ее частью.

1 Требования к курсовой работе

Курсовая работа не может быть простой компиляцией и состоять из фрагментов различных статей и книг. Она должна быть научным, завершенным материалом, иметь факты и данные, раскрывающие взаимосвязь между явлениями, процессами, аргументами, действиями и содержать нечто новое: обобщение обширной литературы, материалов эмпирических исследований, в которых появляется авторское видение проблемы и ее решение. Этому общетеоретическому положению подчиняется структура курсовой работы, ее цель, задачи, методика исследования и выводы.

1.1 Общая структура курсовой работы

Работу над курсовой работой необходимо начинать с подбора и изучения литературы по выбранной теме. В результате систематизированного изучения соответствующей литературы и Интернет-источников, усваиваются основные понятия, категории, термины, формируются представления о современных средствах создания приложений, проясняются слабые и сильные стороны различных подходов к разработке программного обеспечения, выявляются наиболее актуальные направления, обсуждаемые на текущий момент времени. Одновременно выявляются проблемы, требующие дополнительного осмысления; выясняется то, что еще недоста-

точно изучено. На основе этого определяются направление, цель и задачи курсовой работы, а также составляется список литерату-ры, которую планируется использовать при написании курсовой работы.

Курсовая работа по дисциплине «Методы оптимизации и теория принятия решений» должна иметь следующую структуру:

- Титульный лист
- Оглавление
- Введение
- Теоретическая часть
- Практическая часть
- Заключение
- Библиографический список
- Приложения.

Рассмотрим главные части курсовой работы.

Во ***введении*** необходимо обосновать актуальность выбранной темы, сформулировать цель работы и поставить задачи, которые потребуется решить для ее достижения; описать функциональ - ность методик, используемых при выполнении курсовой работы.

Писать введение целесообразно после завершения работы над основной частью.

В ***первом разделе (Теоретическая часть)*** описываются теоретические аспекты рассматриваемой проблемы. В этом разделе студент должен показать свой уровень подготовки, знание предметной области, умение собирать информацию и систематизировать полученные знания, делать обобщения и выявлять способы решения изучаемых проблем.

В процессе описания необходимы ссылки на использованные источники.

Во ***втором разделе (Практическая часть)*** студент исследует практическую задачу одним или несколькими методами теоретического раздела.

Заключение завершает изложение курсовой работы. В нем подводятся итоги выполненной работы в виде обобщения самых существенных положений. Выводы должны отражать только содержание работы, быть краткими, ясно и четко сформулированными. В данном разделе необходимо показать, каким образом ре-шены задачи, привести основные результаты работы.

1.2 Основные требования к оформлению текста работы

Работа выполняется на компьютере. Предпочтительным является использование стандартов, заложенных в редакторе типа *Word*. Распечатка делается на белом стандартном листе бумаги формата А4 210×297 мм. Ниже приведены основные требования к оформлению стандартного печатного текста.

Требования к оформлению текста, подготовленного с использованием компьютерного набора*:*

1. Установка полей: верхнее — 2 см. нижнее — 2.5 см. ле-вое — 2 см. правое — 2 см.

2. Интервал между строк — полуторный.

3. Шрифт — 14, *Times New Roman*

4. Страницы нумеруют в правом верхнем углу. Первая страница (титульный лист) и вторая (оглавление) не нумеруются, но считаются.

5. Каждый абзац печатается с красной строки.

6. В случае использования таблиц и иллюстраций следует учитывать, что:

• единственная иллюстрация и таблица не нумеруются;

• нумерация иллюстраций и таблиц допускается как сквозная (таб. 1, таб. 2 и т. д.), так и по главам (рис. 4.1. рис. 5.2 и т. п.);

• в графах таблицы нельзя оставлять свободные места. Следует заполнять их либо знаком "-" либо писать "нет", "нет данных".

7. Для редактирования математических формул рекомендуется использовать соответствующие приложения компьютерных программ. Каждая формула нумеруется арабскими цифрами. Принципы нумерации аналогичны нумерации таблиц. Номер указывается рядом с формулой в круглых скобках. В тексте должно быть четко указано, что обозначает каждый используемый символ.

1.3 Защита и оценка курсовой работы

К защите должны быть подготовлены:

- презентация, которая предоставляется на диске и прикладывается к работе;

- отчёт о проделанной работе.

Завершенный текст курсовой работы должен быть представлен руководителю не позднее, чем за две недели до установлен-

ного срока защиты курсовой работы. Срок защиты устанавливается до зачётной недели.

К защите не допускаются и возвращаются для повторного написания курсовые работы, полностью или в значительной степени, выполненные не самостоятельно или работы, в которых содержание и оформление, как в целом, так и разделов, не соответствуют выбранной теме, не удовлетворяют требованиям, описанным в данном учебно-методическом пособии и предъявляемым руководителем.

До защиты студент должен продемонстрировать преподавателю работоспособность.

На защите преподаватель и другие студенты, защищающие курсовые работы, могут задавать отвечающему вопросы, касающиеся теоретической и практической частей проекта. Студент должен дать краткие, четко аргументированные ответы и доказать, что проект выполнен им самостоятельно.

Преподаватель оценивает содержание и качество выполненной курсовой работы, уровень теоретической и практической подготовки студента и выставляет оценку следующим критериям*:*

– **«отлично»** выставляется студенту, показавшему глубокие знания, примененные им при самостоятельной разработке избранной темы, способному обобщить практический материал, сделать на основе анализа выводы и представившему качественные презентацию и доклад;

– **«хорошо»** выставляется студенту, показавшему в работе и при ее защите полное знание материала, всесторонне осветившему вопросы темы, но не в полной мере проявившему самостоятельность в исследовании;

– **«удовлетворительно»** выставляется студенту, раскрывшему в работе основные вопросы избранной темы, но не проявившему самостоятельности в анализе или допустившему отдельные неточности в содержании работы, неуверенность при ответе на вопросы;

– **«неудовлетворительно»** выставляется студенту, не раскрывшему основные положения избранной темы и допустившему грубые ошибки в содержании работы, а также допустившему плагиат.

2 Выбор темы курсовой работы

Темы курсовых работ могут быть как теоретической, так и практической направленности. В работах практической направленности студент может продемонстрировать свои навыки владения теоретическим материалом предметной области дисциплины. В качестве тем курсовых работ могут быть следующие:

1. Линейное программирование Симплексный метод Двойственные задачи линейного программирования Модели линейного программирования

2. Модели сетевой оптимизации

Задача о кратчайшем пути

Задача о максимальном потоке

Задача о коммивояжере

3. Модели транспортного типа

Транспортная задача линейного программировании

Задача распределения ресурсов на транспортных сетях

4. Специальные задачи линейного программирования

Целочисленные задачи линейного программирования Задачи параметрического и дробно-линейного программирования

5. Нелинейное программирование

Методы поиска экстремума для функций одной переменной

Градиентные методы безусловной оптимизации

Методы условной оптимизации. Конечномерные гладкие задачи с равенствами

Методы условной оптимизации. Конечномерные гладкие задачи с равенствами и неравенствами

6. Теория игры Антагонистические игры двух лиц

Неантагонистические игры двух лиц

Некооперативные игры двух лиц с ненулевой суммой Задача о переговорах Равновесие по Нэшу в неантагонистических играх

Выбор оптимальной стратегии в условиях неопределенности (игры с природой)

Специальные разделы методов оптимизации Генетические алгоритмы поиска глобального экстремума Динамическое программирование. Задача оптимального распределения ресурсов

Динамическое программирование

Сетевой анализ проектов Системы массового обслуживания

Имитационное моделирование

Задачи дискретной оптимизации

3 Рекомендуемая литература

1. *Волков, И. К.* «Исследование операций» / И. К. Волков, Е. А. Загоруйко. – Издательство МГТУ им. Н. Э. Баумана, Москва 2002.

2. *Аттетков, А. В.* «Методы оптимизации» / А. В. Аттетков, С. В. Галкин, В. С. Зарубин. – Издательство МГТУ им. Н. Э. Баумана, Москва 2003.

3. *Косоруков, О. А.* Исследование операций: Учебник / О. А. Косоруков, А. В. Мищенко // Под общ. ред. д.э.н., проф. Н. П. Тихомирова. – М.: Издательство «Экзамен», 2003.

4. *Ларичев, О. И.* «Теория и методы принятия решений» / О. И. Ларичев. – Физматкнига, Москва 2006.

5. *Вапник, В. Н.* «Теория распознавания образов» / В. Н. Вапник, А. Я. Червоненкис. – Издательство «Высшая школа», Москва 1974.

6. *Акулич, И. Л.* «Математическое программирование в примерах и задачах» / И. Л. Акулич. – Издательство «Высшая школа», Москва 1986.

7. *Петросян, Л. А.* «Теория игр» / Л. А. Петросян, Н. А. Зенкевич, Е. А. Сёмина. – Издательство «Высшая школа», Москва 1998.

8. *Ту, Дж.* «Принципы распознавания образов», том 1 / Дж. Ту, Р. Гон-салес. – Издательство «Мир», Москва 1978.

9. *Ларичев, О. И.* Теория и методы принятия решений, а также Хроника событий в Волшебных страницах: Учебник / О. И. Ларичев – М.: Университетская книга, Логос, 2006.

10. *Черноморов, Г. А.* Теория принятия решений: Учебное пособие / Г. А. Черноморов. – Юж.-Рос. гос. техн. ун-т.-Новочеркасск: Ред. журн. «Изв. вузов. Электротехника». 2005.

11. *Казанов, Ю. К.* Методы поддержки принятия решений. Учебное пособие / Ю. К. Казанов. – М.: МГУ – ПИ. 2009.

12. *Хедми, А. Таха.* Введение в исследование операций: Пер. с англ. – М.: Вильямс, 2007.

13. *Шикин, Е. В.* Исследование операций: учеб. / Е. В. Шикина, Г. Е. Шикина. – М.: ТК Велби, Изд-во Проспект, 2006.

14. *Воробьев, Н. Н.* Теория игр для экономистов-кибернетиков. – М.: Наука, Главная редакция физико-математической литературы, 1985.

15. *Пантелеев, А. В.* Методы оптимизации в примерах и задачах: Учеб. пособие / А. В. Пантелеев, Т. А. Летова. – 2-е изд., исправл. – М.: Высш. шк., 2005.

16. *Блюмин, С. Л.* Модели и методы принятия решений в условиях неопределенности / С. Л. Блюмин, И. А. Шуйкова. – Липецк: ЛЭГИ, 2001.

17. *Крушевский, А. В.* Теория игр / А. В. Крушевский. – Киев, Издательское объединение «Вища школа», 1977.

18. *Haupt, Randy L.* Practical genetic algorithms / Randy L. Haupt, Sue Ellen Haupt. – 2nd Edition, ISBN: 978-0-471-45565-3, June 2004, WILEY.

19. *Амосов, А. А.* Вычислительные методы для инженеров: Учеб. посо-бие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. – М.: Высш. шк., 1994. ил. ISBN 5-06-000625-5.

ПРИЛОЖЕНИЕ

Образец титульного листа курсовой работы

Государственное бюджетное образовательное учреждение
высшего образования Московской области
«Университет «Дубна»
Филиал «Протвино»

Филиал **«Протвино»**
Кафедра **«Информационные технологии»**
(наименование кафедры)

КУРСОВАЯ РАБОТА
по
Теория принятия решений
(наименование учебной дисциплины)

«Нахождение глобального экстремума
методом генетического алгоритма»
(наименование темы)

Выполнил: студент
_____ группы
_____ курса
_____
(Ф.И.О.)
Руководитель:
_____
(ученая степень, ученое звание, занимаемая должность)

Дата защиты: _____
Оценка: _____
_____
(подпись руководителя)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Эволюция в природе показала себя как мощный механизм развития и приспособления живых организмов к окружающей среде и удивляет многообразием и эффективностью решений. Поэтому исследователи в области компьютерных технологий обратились к природе в поисках новых алгоритмов.

Группа алгоритмов, использующих в своей основе идею эволюции Дарвина, называется *эволюционными алгоритмами*. В ней выделяют следующие направления:

– генетические алгоритмы (ГА);
– эволюционные стратегии;
– генетическое программирование;
– эволюционное программирование.

Генетические алгоритмы применяются для решения таких задач, как:

– поиск глобального экстремума многопараметрической функции;
– аппроксимация функций;
– задачи о кратчайшем пути;
– задачи размещения;
– настройка искусственной нейронной сети;
– игровые стратегии;
– обучение машин.

Фактически, генетические алгоритмы максимизируют многопараметрические функции. Поэтому их область применения столь широка. Все приведенные задачи решаются именно путем фор-мирования функции, зависящей от некоторого числа параметров, глобальный максимум которой будет соответствовать решению задачи.

Генетические алгоритмы работают по аналогии с природой. Они оперируют с совокупностью «особей», представляющих со-бой строки, каждая из которых кодирует одно из решений задачи. Приспособленность особи оценивается с помощью специальной функции. Наиболее приспособленные получают шанс скрещиваться и давать потомство. Наихудшие особи удаляются и не дают потомства. Таким образом, приспособленность нового поколения в среднем выше предыдущего (рис. 1).

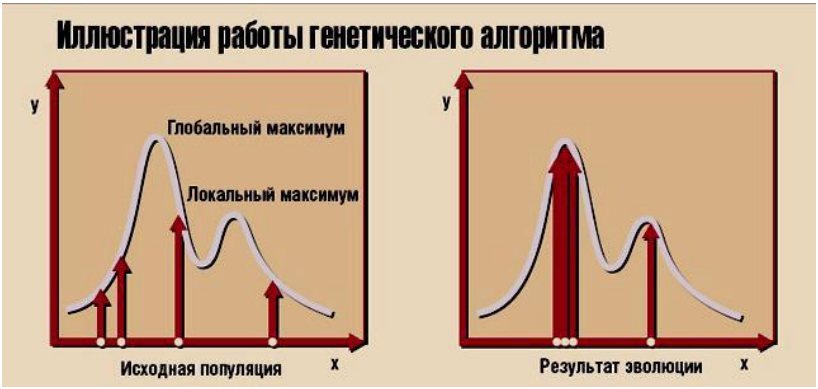


Рис. 1 Иллюстрация работы генетического алгоритма

Живые существа характеризуются их внешними параметрами (фенотипом). Некоторые из параметров оказываются полезными для выживания и размножения, другие скорее вредят. Все внешние данные особи кодируются ее цепью ДНК (генотипом). Отдельные участки этой цепи (гены) определяют различные параметры особи.

Те особи, которые не обладают качествами, способствующими их выживанию, с большой вероятностью не проживут долго и не смогут создать многочисленное потомство. Кроме того, им сложнее будет найти хорошую пару для скрещивания, поэтому с боль-шой вероятностью генотип таких особей исчезнет из генофонда популяции.

Изредка происходит мутация: некоторый случайный нуклеотид цепи ДНК особи может измениться на другой. Если полученная цепь будет использоваться для создания потомства, то возможно появление у детей совершенно новых качеств.

Естественный отбор, скрещивание и мутация обеспечивают развитие популяции . Каждое новое поколение в среднем более приспособлено, чем предыдущее, т. е. оно лучше удовлетворяет требованиям внешней среды. Этот процесс называется эволюцией.

Рассматривая эволюцию в природе, возникает мысль о том, что можно искусственно отбирать особи, подходящие нам по некоторым параметрам, создавая таким образом искусственные внеш-

ние условия. Это называется селекцией и используется людьми для получения новых пород животных, к примеру, коров, дающих больше молока или овец с более густой шерстью. Но почему бы не устроить собственную эволюцию с помощью компьютера? Действительно, пусть есть функция, которая по заданному набору численных параметров возвращает некоторое значение (многопараметрическая функция). Создадим множество строк, каждая из которых будет кодировать вектор чисел (длина вектора равна количеству параметров функции). По заданному вектору можно высчитать соответствующее ему значение функции. Те строки, для которых это значение велико, будем считать более приспособленными, чем те, для которых оно мало. Запуская эволюцию на строках по подобию природной, на каждом поколении будем получать строки со все большими значениями функции. Таким образом, такого рода эволюция решает задачу максимизации многопараметрической функции.

1. ПОСТАНОВКА ЗАДАЧИ

Пусть имеется функция от многих переменных, у которой необходимо найти глобальный максимум или минимум:

$$F(x_1, x_2, x_3, \ldots, x_n).$$

Первым шагом работы с генетическим алгоритмом будет преобразование независимых переменных в хромосомы, которые будут содержать всю необходимую информацию о каждой создаваемой особи. Имеется два варианта кодирования параметров:

– в двоичном формате;

– в формате с плавающей запятой.

В случае, если мы применяем двоичное кодирование, мы используем ***n*** бит для каждого параметра, причем ***n*** может быть различным для каждого параметра. Если параметр может изменяться между минимальным значением ***min*** и максимальным ***max***, возьмем следующие формулы для преобразования:

$$r = \frac{g(\max - \min)}{2^n - 1} + \min \qquad (1)$$

$$g = \frac{r - \min}{(\max - \min)(2^n - 1)} \qquad (2),$$

где $\boldsymbol{g}$ — целочисленные двоичные гены, — эквивалент генов в формате с плавающей запятой.

Хромосомы в формате с плавающей запятой создаются при помощи размещения закодированных параметров один за другим.

Если сравнивать эти два способа представления, то лучшие результаты дает вариант представления в двоичном формате. Правда, в этом случае приходится мириться с постоянным кодировани-ем/декодированием параметров.

В общем случае генетический алгоритм работает следующим образом. В первом поколении все хромосомы генерируются случайно. Определяется их «полезность». Начиная с этой точки гене-тический алгоритм может начинать генерировать новую популя-цию. Обычно размер популяции постоянен.

Репродукция состоит из четырех шагов:

– **селекции**;

– **кроссовера** (скрещивания);

– **мутациии**;

– **инверсии**.

1) ***Селекция хромосом*** — выбор (по рассчитанным значениям функции приспособленности) тех хромосом, которые будут участвовать в создании потомков для следующей популяции, то есть, для очередного поколения. Такой выбор производится согласно принципу естественного отбора, по которому наибольшие шан-сы на участие в создании новых особей имеют хромосомы с наибольшими значениями функции приспособленности. Существуют различные методы селекции. Наиболее популярным считается так называемый *метод рулетки,* который свое название получил по аналогии с известной азартной игрой. Каждой хромосоме мо-жет быть сопоставлен сектор колеса рулетки, величина которого устанавливается пропорциональной значению функции приспособленности данной хромосомы. Поэтому чем больше значение функции приспособленности, тем больше сектор на колесе рулетки. Все колесо рулетки соответствует сумме значений функции приспособленности всех хромосом рассматриваемой популяции.

Каждой хромосоме, обозначаемой χ_i для i =1,2, ..., n (где n обозна-чает численность популяции) соответствует сектор колеса v(χ_i), выраженный в процентах согласно формуле

$$v(\chi_i) = p_s(\chi_i)*100\%, \quad (3),$$

где

$$p_s(\chi_i)=\frac{F(x_i)}{\sum_{i=1}^{n} F(x_i)} \quad (4)$$

причем $F(\chi_i)$ — значение функции приспособленности хромосомы χ_i, а $p_s(\chi_i)$ — вероятность селекции хромосомы χ_i. Селекция хромосомы может быть представлена как результат поворота колеса рулетки, поскольку "выигравшая" (то есть , выбранная) хромосома относится к выпавшему сектору этого колеса. Очевидно, что чем больше сектор, тем больше вероятность "победы" соответствующей хромосомы. Поэтому вероятность выбора данной хромосомы оказывается пропорциональной значению ее функции приспособленности. Если всю окружность колеса рулетки представить в виде цифрового интервала [0, 100], то выбор хромосомы можно отождествить с выбором числа из интервала [a, b], где a и b обозначают соответственно начало и окончание фрагмента окружности, соответствующего этому сектору колеса; очевидно, что $0 \le a < b \le 100$. В этом случае выбор с помощью колеса рулетки сводится к выбору числа из интервала [0, 100], которое соответствует конкретной точке на окружности колеса. В результате процесса селекции создается родительская популяция, также называемая с численностью N, равной численности текущей популяции.

2) ***Скрещивание*** (кроссовер) является наиболее важным генетическим оператором. Оно генерирует новую хромосому, объединяя генетический материал двух родительских. Существует несколько вариантов скрещивания. Наиболее простым является одноточечный. В этом варианте просто берутся две хромосомы и перерезаются в случайно выбранной точке. Результирующая хро-мосома получается из начала одной и конца другой родительских хромосом.

001100101110010 |11000⟩

⟩ – – – – > 001100101110010 |11100

3) ***Мутация*** представляет собой случайное изменение хромосомы (обычно простым изменением состояния одного из битов на противоположное). Данный оператор позволяет более быстро находить локальные экстремумы, с одной стороны, и «перескаки-вать» на другой локальный экстремум — с другой.

00110010111001**0**11000 – – – – > 00110010111001**1**11000

4) ***Инверсия*** инвертирует (изменяет) порядок бит в хромосоме путем циклической перестановки (случайное количество раз). Многие модификации генетического алгоритма обходятся без данного генетического оператора.

00110010111001011000 – – – – > 11000001100101110010

В результате применения генетических операторов к хромосомам временной родительской популяции, создается новая популяция. При завершении работы генетического алгоритма проверяется условие его остановки. Если условие остановки алгоритма выполнено, то следует вывести результат работы, то есть представить искомое решение задачи. Лучшим решением считается хромосома с наибольшим значением функции приспособленности.

2. МЕТОДЫ РЕШЕНИЯ

Классический генетический алгоритм (также называемый элементарным или простым генетическим алгоритмом) состоит из следующих шагов:

1) **инициализация**, или выбор исходной популяции хромосом;

2) **оценка** приспособленности хромосом в популяции – расчет функции приспособленности для каждой хромосомы;

3) **проверка** условия остановки алгоритма;

4) **селекция** хромосом — выбор тех хромосом, которые будут участвовать в создании потомков для следующей популяции;

5) **применение генетических операторов** – мутации и скрещивания;

6) **формирование** новой популяции;

7) **выбор** «наилучшей» хромосомы.

Выбор исходной популяции хромосом осуществляется путем преобразования в них независимых переменных. Об этом сказано в п. 1.

Функция приспособленности — это функция, зависящая от переменных, входящих в данную хромосому и характеризующая приспособленность данной хромосомы к выживанию. Задача оптимизации состоит в максимизации функции приспособленности. В процессе эволюции в результате отбора, рекомбинаций и мутаций геномов особей происходит поиск особей с высоким уровнем приспособленности.

Определение условия остановки генетического алгоритма зави-сит от его конкретного применения. В оптимизационных задачах, если известно максимальное (или минимальное) значение функ-ции приспособленности, то остановка алгоритма может произой-ти после достижения ожидаемого оптимального значения, воз-можно — с заданной точностью. Остановка алгоритма также мо-жет произойти, когда его выполнение не приводит к улучшению уже достигнутого значения или произошло истечение определен-ного времени выполнения, либо — после выполнения заданного количества итераций. Если условие остановки выполнено, то про-изводится переход к завершающему этапу выбора «наилучшей» особи.

Селекция хромосом заключается в выборе (по рассчитанным на втором этапе значениям функции приспособленности) тех хромо-сом, которые будут участвовать в создании потомков для следую-щей популяции. Подробно об этом сказано в п. 1.

Применение генетических операторов к хромосомам, отобранным с помощью селекции, приводит к формированию новой попу-ляции потомков от созданной на предыдущем шаге родительской популяции.

В классическом генетическом алгоритме применяются два основных генетических оператора: оператор скрещивания (крос-

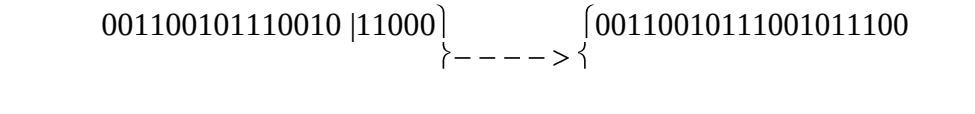
синговер) и оператор мутации. Однако следует отметить, что оператор мутации играет явно второстепенную роль по сравнению с оператором скрещивания. Это означает, что скрещивание в классическом генетическом алгоритме производится практически всегда, тогда как мутация — достаточно редко. Вероятность скрещивания, как правило, достаточно велика (обычно $0{,}5 \le p_c \le 1$), тогда как вероятность мутации устанавливается весьма малой (чаще всего $0 \le p_m \le 0{,}1$). Это следует из аналогии с миром живых организмов, где мутации происходят чрезвычайно редко.

В генетическом алгоритме мутация хромосом может выполняться на популяции родителей перед скрещиванием либо на популяции потомков, образованных в результате скрещивания.

На первом этапе скрещивания выбираются пары хромосом из родительской популяции. Это временная популяция, состоящая из хромосом, отобранных в результате селекции и предназначенных для дальнейших преобразований операторами скрещивания и мутации с целью формирования новой популяции потомков. На данном этапе хромосомы из родительской популяции объединяются в пары. Это производится случайным способом в соответствии с вероятностью скрещивания P_c. Далее для каждой пары отобранных таким образом родителей разыгрывается позиция гена в хромосоме, определяющая так называемую точку скрещивания. Если хромосома каждого из родителей состоит из $\boldsymbol{L}$ генов, то очевидно, что точка скрещивания l_k представляет собой натуральное число, меньшее $\boldsymbol{L}$. Поэтому фиксация точки скрещивания сводится к случайному выбору числа из интервала [1, $\boldsymbol{L}$-1]. В результате скрещивания пары родительских хромосом получается следующая пара потомков:

1) потомок, хромосома которого на позициях от 1 до l_k состоит из генов первого родителя, а на позициях от l_k+1 до $\boldsymbol{L}$ — из генов второго родителя;

2) потомок, хромосома которого на позициях от 1 до l_k состоит из генов второго родителя, а на позициях от l_k+1 до L — из генов первого родителя.



Оператор мутации с вероятностью P_m изменяет значение гена в хромосоме на противоположное (то есть,. с 0 на 1 или наоборот).

0011001011100101100 0 – – – – > 0011001011100111100 0

Как уже упоминалось выше, вероятность мутации обычно очень мала, и именно от нее зависит, будет данный ген мутировать или нет. Вероятность P_m мутации может эмулироваться, напри-мер, случайным выбором числа из интервала [0,1] для каждого гена и отбором для выполнения этой операции тех генов, для ко-торых разыгранное число оказывается меньшим или равным зна-чению P_m.

Хромосомы, полученные в результате применения генетиче-ских операторов к хромосомам временной родительской популя-ции, включаются в состав новой популяции. Она становится так называемой текущей популяцией для данной итерации генетиче-ского алгоритма. На каждой очередной итерации рассчитываются значения функции приспособленности для всех хромосом этой популяции, после чего проверяется условие остановки алгоритма и — либо фиксируется результат в виде хромосомы с наибольшим значением функции приспособленности, либо осуществляется переход к следующему шагу генетического алгоритма, то есть к селекции. В классическом генетическом алгоритме вся предше-ствующая популяция хромосом замещается новой популяцией по-томков, имеющей ту же численность.

Если условие остановки алгоритма выполнено, то следует вы-вести результат работы, то есть представить искомое решение за-дачи. Лучшим решением считается хромосома с наибольшим зна-чением функции приспособленности.

Стоит заметить, что в настоящее время генетические алгорит-мы представляют собой целый класс алгоритмов, направленных на решение разнообразных задач (рис. 2), в том числе, задачи по-иска глобального экстремума функции.

В зависимости от конкретного алгоритма, применение генети-ческих операторов может отличаться, например: во многих ал-горитмах появляется оператор инверсии, который отсутствует в классическом алгоритме. Для решения каждой конкретной задачи необходимо подбирать наиболее подходящий для нее вид алгорит-

ма. Зачастую, выбор параметров генетического алгоритма и конкретных генетических операторов производится на интуитивном уровне, так как пока не существует объективных доказательств преимущества тех или иных настроек и операторов.

Рис. 2 Эффективность методов минимизации

3. ПРИМЕР РЕШЕНИЯ

3.1. Генетический алгоритм нахождения максимума функции двух переменных в заданной области их изменения.

Пусть о функции неизвестно ничего, кроме того, что нам доступен алгоритм, позволяющей вычислить её значение в любой произвольно выбранной точке.

• Возьмём наугад n значений $x_{i(1)}$, принадлежащих области определения. Вычислим (отыщем) значения $y_i^{(1)} = f(x_i^{(1)})$ во всех этих точках, а далее будем рассуждать и действовать следующим образом.

• Расположим $x_{i(1)}$ в порядке возрастания соответствующих зна-чений $y_i^{(1)}$.

• Будем считать, что значения $x_{i(1)}$, соответствующие большим значениям $f(x_{i(1)})$, находятся ближе чем другие к точке искомого экстремума нашей функции, и будем считать их «хорошими» (от-метим, что это предположение есть гипотеза, основанная только на интуиции).

• Сформируем новую последовательность $x_{j(2)}$ той же длины n, но уже не совсем случайным образом, а так, чтобы часть но-вых $x_i^{(1)}$ находились поблизости от «хороших» точек $x_j^{(2)}$, а другая часть, во избежание попадания в локальный экстремум, охватыва-ла бы другие, «далёкие» области определения функции.

• Получив новую последовательность $x_{j(2)}$, которая в сложившейся терминологии называется «популяцией», мы будем генери-ровать таким же образом всё новые и новые популяции.

Этот процесс построения все новых и новых популяций $x_{j(k)}$ будем продолжать в надежде, что на каком-то шаге мы найдём точку, в которой функция $f(x)$ достигнет своего экстремума. Чем можно подтвердить то, что наша надежда оправдалась, что мы действительно на некой итерации получили значение координаты искомого экстремума? Например, тем, что новые итерации не дают уже улучшения результатов, или тем, что, с точки зрения эксперта, вся область определения функции, экстремум которой мы ищем, достаточно плотно обследована. Эти и многие другие критерии завершения процесса итераций эволюционного алгоритма, очевид-но, не имеют ничего общего с формальными строгими математическими доказательствами полученных результатов.

Для получения точек новой популяции на основе анализа точек предыдущей популяции используются операции ***скрещивания*** и ***мутации***. Выбор хороших точек и отбрасывание плохих называется ***селекцией***.

Обратимся к нашему простому примеру. Заметим, что если мы будем каждый раз выбирать для получения новой популяции только хорошие точки — геномы предыдущей популяции, то велика вероятность зациклиться около точки локального экстремума и не найти популяцию, содержащую глобальный экстремум (рис. 3).

Во избежание такого явления живая природа придумала мутацию — случайное изменение какого-то генома, приводящее к изменчивости отдельной особи либо в лучшую, либо в худшую сторону. В соответствии с этим механизмом изменчивости в генетических алгоритмах используют операцию мутации, которая обеспечивает, как правило, выход из зацикливания около хороших геномов. Более сложную операцию скрещивания можно интер-претировать как попытку сформировать новую особь смешением признаков двух других особей (родителей).

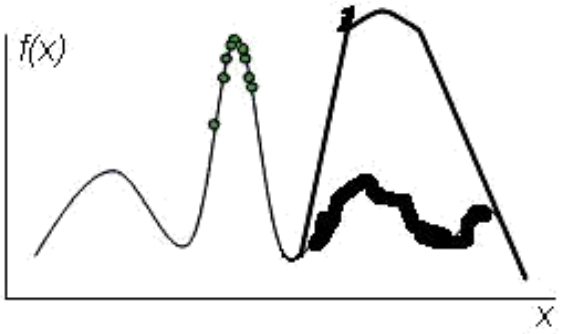


Рис. 3 Пример локализации локального экстемума

Общую ситуацию, в которой действует генетический алгоритм нашей простой задачи отыскания экстремума функции, следует представлять следующим образом.

Имеется чёрный ящик, устройства которого мы не знаем, но ко-торый, получая на вход некоторые данные, на выходе выдаёт не-которые значения, сравнимые между собой, если их ранжировать по величине или значимости. На вход могут поступать характери-стики некоторых объектов, которые мы будем называть геномом особи. Геном должен быть закодирован некоторой последователь-ностью символов. С вычислительной точки зрения любой код для компьютера представляет собой структурированную последова-тельность двоичных разрядов.

Двоичные элементы такой последовательности часто называют генами, а смысловые поля последовательности иногда называют хромосомами. Сразу же отметим, что в формальной постановке задач, решаемых генетическими алгоритмами, эти термины не связаны с их исходным биологическим смыслом. В общем случае можно считать, что на вход чёрного ящика поступают данные в двоичном коде, на выходе мы получаем закодированные в двоич-ном виде результаты, с помощью которых можно упорядочить ге-номы популяции, генерируемой на каждом шаге работы алгорит-ма. Операция скрещивания формально состоит в том, что из попу-ляции выбираются геномы двух родителей и из них формируются геномы нескольких потомков. По некоторым правилам, напоми-нающим перекрёстное опыление (для потомков часть хромосом

(блоков генов) берётся от одного родителя, часть хромосом — от другого), они объединяются, но так, чтобы их общее число, а ино-гда и место расположения, при этом оставалось таким же, как у родителей.

Одноместная операция мутации состоит в том, что у *случайно* выбранного представителя популяции *случайно* меняется один двоичный разряд (ген) на противоположный — так получают значение нового представителя. Слово случайно означает, что этот процесс производится с некоторой заранее заданной вероятностью.

Операция селекции состоит в выборе геномов, полученных в результате скрещиваний и мутаций, для формирования новой популяции следующего шага итерации генетического алгоритма. При организации селекции также используется вероятностный подход, то есть задаются вероятности выбора новых геномов и оставления в популяции старых особей.

Перечисленные выше вероятности являются параметрами генетического алгоритма, и от их выбора в значительной степени зависит качество работы самого алгоритма и успех в решении поставленной задачи. Входными параметрами генетического алгоритма являются также размер генома, заданный размер популяции и число шагов итерационного процесса, необходимых для получе-ния приемлемого решения.

3.2. Генетический алгоритм нахождения минимума функции нескольких переменных в заданной области их изменения

Как правило, задача отыскания минимума функционала сводит-ся к отысканию глобального минимума функции нескольких переменных в некотором конечномерном подпространстве (рис. 4).

Последовательность оптимизируемых параметров (особь) записывается в виде хромосомы конечномерного вектора. Для представления хромосом использовано двоичное кодирование (код Грэя, который предпочтительнее стандартного двоичного коди-рования в силу сохранения непрерывности бинарной операции) с последовательным расположением генов (векторов) фиксированной длины; при этом каждому параметру соответствует один ген и все гены имеют одинаковую длину. Предполагается, что область изменения (гиперпараллелепипед в $\boldsymbol{R}_n$, где $\boldsymbol{n}$ — число перемен-

ных) каждого из разыскиваемых параметров известна. Область линейно отображается в гиперкубе с ребром длины 2_s. Таким образом, каждый параметр переводится в целочисленную точку на отрезке $[0... 2_s]$, содержащем $2_s + 1$ точек. Затем это целое число переводится в двоичное представление с помощью кода Грэя. Код Грея — система нумерования, в которой два соседних значения различаются только в одном разряде.

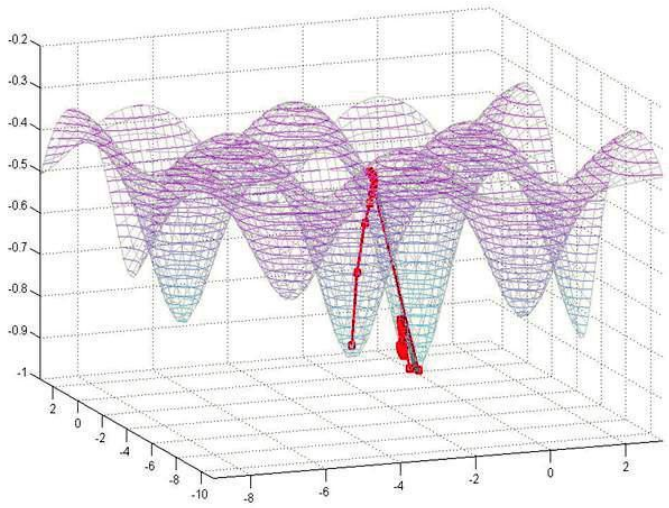


Рис. 4 Поиск глобального минимума функции нескольких переменных

В **алгоритме использованы традиционные генетические операторы.**

1. **Двухточечный кроссовер** (скрещивание): случайно выбира-ются две позиции — i_1 и i_2 в хромосоме, хромосомы обмениваются своими частями, которые расположены между i_1 и i_2. В частности, если позиция i_2 равна длине хромосомы, то фактически выполня-ется одноточечный кроссовер.

2. ***k*-точечная мутация.** Случайно выбирается позиция в хромосоме, инвертируется значение бита, причем процедура повторяется k раз; в серии вычислительных экспериментов принято $k = 3$.

3. **Инверсия.** В хромосоме случайно выбирается позиция i, а затем меняются местами части хромосомы, расположенные до и после этой позиции.

Разработанный оригинальный генетический алгоритм состоит из нескольких этапов.

1. Случайным образом (здесь и далее использовано равномерное распределение) генерируется начальная популяция, содержащая n хромосом (нулевое или начальное поколение).

2. Хромосомы сортируются в порядке от лучшей к худшей в соответствии со значением целевой функции на хромосоме. Счет-чику $\boldsymbol{i}$ присваивается значение 1.

3. Осуществляется случайный выбор двух хромосом.

4. С вероятностью $\boldsymbol{p_c}$ они скрещиваются и порождают двух потомков, причем если скрещивания не происходит, то потомками считаются исходные хромосомы.

5. К каждому из потомков с вероятностью $\boldsymbol{p_m}$ применяется опе-ратор мутации.

6. К каждому из потомков п. 5 с вероятностью $\boldsymbol{p_i}$ применяется оператор инверсии.

7. Лучшая из этих двух хромосом отбирается и помещается в поколение вместо ($\frac{n}{2} + i$)-й хромосомы. Счетчик i увеличивается на единицу.

8. Пункты 3–7 повторяются $\frac{n}{2}$ раз.

9. Пункты 2–8 повторяются до выполнения критерия остановки. Критерием остановки всего процесса является стягивание ядра популяции (некоторого процентного числа $\boldsymbol{Q}$ лучших элитных особей) в сферу заданного достаточно малого радиуса $\boldsymbol{R_e}$.

В предложенном алгоритме, в отличие от традиционной рулет-ки с секторами, площадь которых пропорциональна приспособленности особи, применена “суперэлитная” стратегия, которая обеспечивает, с одной стороны, гарантированное сохранение лучшего из найденных решений, а с другой — достаточно быструю сходимость процесса. Вероятность кроссовера p_c выбирается ~0.9. Чтобы уравновесить сильное давление отбора, вероятности мутации и инверсии, в отличие от традиционных схем, нужно на-значать достаточно большими: $\boldsymbol{p_m} \sim 0.7$, а $\boldsymbol{p_i} \sim 0.2$. При этом отме-тим, что мутация обеспечивает локальный поиск — уточнение, а инверсия — большие броски в области поиска. Таким образом,

настроечными параметрами алгоритма являются число особей в популяции n, число мутирующих бит k, параметры Q и R_e, а также вероятности p_c, p_m и p_i. По своим свойствам данный алгоритм занимает некое промежуточное положение между детерминированными и глобальными случайными методами поиска экстремума.

ЗАКЛЮЧЕНИЕ

Таким образом генетические алгоритмы являются универсальным методом оптимизации многопараметрических функций, и поэтому способны решать широкий спектр задач. Генетические алгоритмы предоставляют огромный материал для исследований за счет большого количества модификаций и параметров . Зачастую небольшое изменение одного из них может привести к неожиданному улучшению результата. В то же время следует помнить, что применение генетических алгоритмов полезно лишь в тех случа-ях, когда для данной задачи нет подходящего специального алго-ритма решения. По сравнению с таким алгоритмом генетический алгоритм будет работать, по крайней мере, не лучше.

Во многих случаях генетический алгоритм на несколько поряд-ков превосходит по быстроте случайный поиск.

Библиографический список

1. *Емельянов, В. В.* Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик. — М: Физматлит, 2003, С. 432. ISBN 5-9221-0337-7

2. *Курейчик, В. М.* Поисковая адаптация: теория и практика. / В. М. Ку-рейчик, Б К. Лебедев, О К. Лебедев. — М: Физматлит, 2006, С. 272, ISBN 5-9221-0749-6

3. *Гладков, Л. А.* Генетические алгоритмы: Учебное пособие. / Л. А. Глад-ков, В. В. Курейчик., В. М. Курейчик. — М: Физматлит, 2006, С. 320. — ISBN 5-9221-0510-8

4. *Гладков, Л. А.* Биоинспирированные методы в оптимизации: монография. / Л. А. Гладков, В. В. Курейчик., В. М. Курейчик и др. -— М: Физматлит, 2009, С. 384, ISBN 978-5-9221-1101-0

5. *Рутковская, Д.* Нейронные сети, генетические алгоритмы и нечеткие системы. / Д. Рутковская., М. Пилиньский., Л. Рутковский — М: Горячая линия-Телеком, 2008, С. 452, ISBN 5-93517-103-1

Электронное учебное издание

Гусев Виктор Владимирович
Астафьева Марина Петровна

ПОДГОТОВКА И ОФОРМЛЕНИЕ КУРСОВЫХ РАБОТ ПО ДИСЦИПЛИНЕ «ТЕОРИЯ ПРИНЯТИЯ РЕШЕНИЙ»

ЭЛЕКТРОННОЕ МЕТОДИЧЕСКОЕ ПОСОБИЕ

Филиал «Протвино»
государственного университета «Дубна
142281 г. Протвино Московской обл.,
Северный проезд, 9