

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

5(153)
2009

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

БАЗЫ ДАННЫХ И ЗНАНИЙ

- Шарашов В. В. Методика выбора способа отображения наследования объектов в реляционную базу данных 2
Редреев П. Г. Построение табличных приложений со списочными компонентами. 7
Туманов В. Е. Предметно-ориентированные системы научной осведомленности
Соколов А. В. Методика оценки максимально возможных значений показателей эффективности поиска текстовой информации. 18

ОПТИМИЗАЦИЯ

- Карпенко А. П., Федорук В. Г. Один класс прямых адаптивных методов многокритериальной оптимизации 24
Аввакумов В. Д. Оптимальное размещение плоских объектов произвольной геометрической формы 31

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Князева М. А., Плохих С. А. Концепция системы управления специализированного банка знаний о преобразованиях программ 36
Волченков Е. Я. Система комбинационной генерации тестов вход/выход: основные методы и алгоритмы 41
Губайдуллин И. И., Фрид А. И. Метод экспериментальной оценки надежности программного обеспечения 47

КОМПЬЮТЕРНАЯ ГРАФИКА

- Левашкина А. О., Поршнева С. В. Сравнительный анализ супервизорных критериев оценки качества сегментации изображений 52
Масалович А. А. Численные методы детектирования и удаления геометрических искажений в изображениях текстовых документов. 57

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Архипова О. Е. Концепция региональной эколого-информационной системы мониторинга 62
Бушмелева К. И., Плюснин И. И. Автоматизированная геоинформационная система мониторинга технического состояния магистральных газопроводов. 68

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Гермашев И. В., Васильева В. Д., Дербишер В. Е., Дербишер Е. В., Силина А. Ю. Применение теории нечетких множеств для решения задач химической технологии 72
Воронова Л. М. Моделирование оптимальных планов земледелия с применением адаптированных пакетов технологий 79
Contents 86
Приложение. Кудинов Ю. И. Системный подход к нечеткому моделированию сложных производственных систем.

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ГОРБАТОВ В. А.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НАРИНЬЯНИ А. С.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

Журнал включен в базу данных Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.415.25

В. В. Шарашов, аспирант,
Государственный научно-исследовательский
институт информационных технологий
и телекоммуникаций "Информика", г. Москва,
vladislavvsh@gmail.com

Методика выбора способа отображения наследования объектов в реляционную базу данных

Рассмотрены основные способы отображения наследования объектов в реляционную базу данных. Проведен анализ влияния этих способов на различные аспекты системы. Описана методика выбора оптимального способа отображения на основе требований к системе.

Ключевые слова: объектно-реляционная проекция, сохранение объектов, отображение наследования, системы управления реляционными базами данных, объектно-ориентированное программирование.

Введение

При разработке современного программного обеспечения для хранения данных наиболее часто используют реляционные системы управления базами данных (РСУБД), тогда как для проектирования и программирования логики обработки данных чаще используют объектно-ориентированный подход. Интеграция этих двух моделей открывает широкие возможности как в процессе проектирования программного обеспечения, так и на стадиях эксплуатации и поддержки.

При использовании РСУБД для хранения объектно-ориентированных данных возникает необходимость писать промежуточный слой программного обеспечения, с помощью которого можно было бы как обрабатывать данные в объектно-ориентированном виде, так и сохранять эти данные в реляционной форме. Эта постоянная необходимость в преобразовании между двумя разными моделями данных не только сильно снижает производительность, но и создает трудности для программистов, так как обе формы данных накладывают ограничения друг на друга [1]. Для решения проблемы преобразования данных между двумя моделями используют средства объект-

но-реляционной проекции (ОРП), их англоязычное обозначение — ORM (*Object Relational Mapping*) [2, 3]. Этот промежуточный слой программного обеспечения предназначен для отображения артефактов реляционной модели в артефакты объектной модели и, наоборот, на основе метаданных. Это обеспечивает прозрачную работу с таблицами, полями и связями РСУБД как с объектами, свойствами и коллекциями.

Одной из основных проблем при отображении является то, что в объектной модели есть связи, построенные на основе механизма наследования (*inheritance*). В реляционной модели нет аналогичных механизмов для наследования поведения и данных, которые присутствуют в объектной модели, поэтому для отображения иерархических структур на основе наследования применяются различные методы отображения. Каждый метод влияет на различные аспекты системы, например, на производительность и сложность построения произвольных отчетов. Далее в статье рассматриваются основные методы отображения наследования, анализируется их влияние на различные аспекты системы и приводится методика для выбора оптимального метода исходя из требований к системе.

Способы отображения наследования

Основным вопросом при отображении наследования объектов в реляционную базу данных является следующий: как организовать наследуемые свойства в схеме базы данных? Существуют четыре основных варианта ответа на этот вопрос отображения наследования [4]:

- наследование с одной таблицей (*Single Table Inheritance*);
- наследование с таблицами для каждого класса (*Class Table Inheritance*);
- наследование с таблицами для каждого конкретного класса (*Concrete Table Inheritance*);
- наследование с обобщенной табличной структурой (*Generic Table Structure Inheritance*).

Выбор того или иного способа отображения влияет на различные аспекты системы, такие как время доступа к данным, сложность построения произвольных отчетов и сложность внесения изменений в иерархию классов. Каждое решение представляет собой компромисс между скоростью доступа к данным, дублированием структуры схемы базы данных, сложностью преобразования данных средством ОРП и сложностью построения

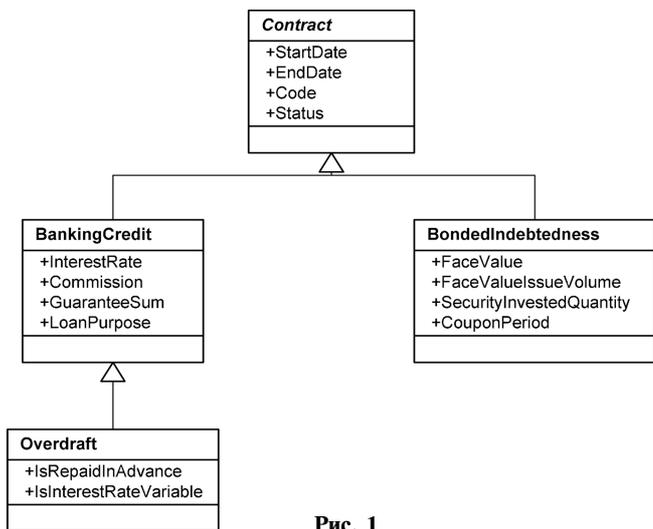


Рис. 1

произвольных запросов на выбор или изменение данных.

В данной статье рассмотрены приведенные выше способы отображения иерархии на основе наследования и описана методика выбора конкретного способа исходя из требований к системе. Для демонстрации каждого подхода будет использована версия иерархии классов, представленная на рис. 1. В этой версии описаны четыре класса: *Contract* (Договор), абстрактный класс, и три конкретных класса — *BankingCredit* (Банковский кредит), *BondedIndebtedness* (Обязательство по облигационному займу) и *Overdraft* (Овердрафт). Для простоты, на рис. 1 не показаны все свойства классов, не указаны полные сигнатуры свойств, отсутствуют методы.

Наследование с одной таблицей

Решение "наследование с одной таблицей" [4] является самым простым в реализации и устойчивым к модификациям иерархии классов вариантом отображения наследования. При данном варианте все объекты иерархии классов отображаются в одну и ту же таблицу, которая содержит столбцы, соответствующие всем свойствам классов, входящих в иерархию. На рис. 2 показан пример физической модели отображения в одну таблицу классов, показанных на рис. 1. Каждому объекту из иерархии соответствует одна запись в таблице. Поля, для которых нет соответствующих свойств в классе, остаются для данной записи пустыми.

Первый столбец (Id) используется в качестве первичного ключа таблицы. Столбец Id хранит значение идентификатора объекта, используе-

мого при отображении, которое является суррогатным ключом. Второй столбец (*ContractType* (Тип договора)) используется для определения типа объекта, хранящегося в записи. При загрузке записи из таблицы необходимо определить, какому классу иерархии соответствует загруженная запись. Для этого к таблице добавляется специальный столбец (*discriminator*), по значению которого и определяется тип объекта. Обычно данный столбец содержит в себе уникальный код класса. Например, значение "С" в столбце *ContractType* будет определять тип объекта, хранимого в записи как *BankingCredit*, а значение "I" — как *BondedIndebtedness*. Если для отображения используется данный метод, то все запросы строятся к одной таблице. Но, если тип объекта, заранее неизвестен, или требуется загрузить множество объектов разных типов иерархии, то необходимо запрашивать значения для всех столбцов и по значению *discriminator* определять тип записи и соответствующие столбцы, в которых сохраняются значения для данного типа, — это единственная сложность при данном методе.

Наследование с таблицами для каждого класса

При данном подходе [4] создается своя таблица для каждого класса с одним столбцом для каждого свойства класса. Свойства класса отображаются на столбцы соответствующей таблицы. На рис. 3 показана физическая модель для отображения классов, показанных на рис. 1.

Информация для объекта класса *BankingCredit* отображена в две таблицы: *Contract* и *BankingCredit*. Следовательно, для того, чтобы считать данную информацию необходимо провести соединение этих двух таблиц (или последовательно считать данные из каждой таблицы, что менее производительно).

Интересным моментом в данной физической модели является использование ключей для связи строк таблиц базы данных. Одно из возможных решений — это использование общего значения

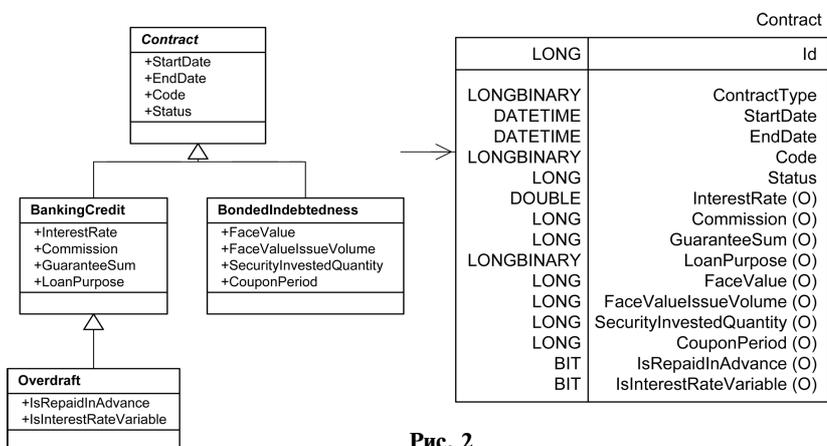


Рис. 2

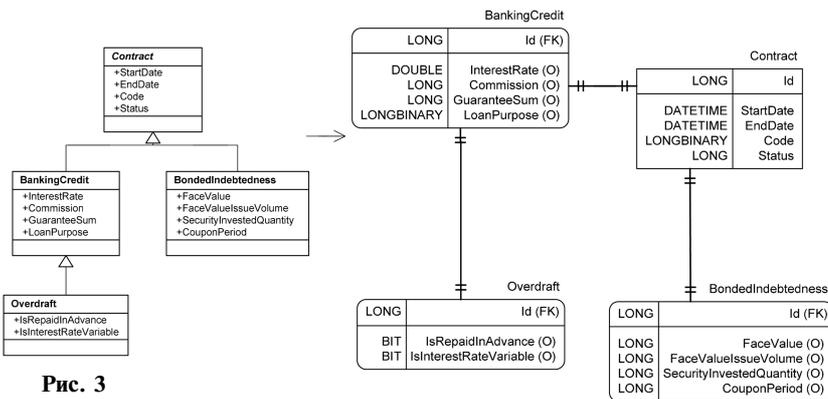


Рис. 3

первичного ключа, чтобы строки с одним значением соответствовали одному объекту. Например, значение столбца Id используется в качестве первичного ключа для всех таблиц. Для таблиц *BankingCredit* и *BondedIndebtedness* значение столбца Id является также и внешним ключом. Так как базовая таблица *Contract* содержит по одной строке для каждой строки всех таблиц производных классов, обеспечивается уникальность первичного ключа (Id) во всех таблицах иерархии.

Другим решением для связи строк является создание своего первичного ключа в каждой таблице и использование внешних ключей в таблицах производных классов.

Часто в базовую таблицу (*Contract*) добавляют логические столбцы или столбцы с кодовым значением для определения типа объекта, который хранится в данной записи. Хотя это и влечет дополнительные расходы на поддержку новых столбцов, данная модификация облегчает некоторые типы запросов.

Важной проблемой при реализации данного метода является эффективное извлечение данных из множества таблиц. Для извлечения объекта *Overdraft* необходимы записи из таблиц *Contract*, *BankingCredit* и *Overdraft*. Выполнение запроса к каждой таблице отдельно является неэффективным. Эффективнее будет использовать соединение таблиц в запросе, но оно тоже непроизводительно, если в нем участвуют много таблиц. К тому же, при выполнении некоторых запросов (например, если требуются все объекты *Contract*, которые включают в себя и *BankingCredit*, и *Overdraft*, и *BondedIndebtedness*) необходимо использовать внешнее соединение (так как не все таблицы содержат данные для каждого конкретного объекта), и определять тип объекта по наличию записей в соответствующих таблицах, что тоже отрицательно сказывается на производительности. Если заранее известен тип загружаемых

объектов, то можно соединять не все таблицы, а только те, которые соответствуют загружаемому объекту, и использовать только внутреннее соединение таблиц.

Наследование с таблицами для каждого конкретного класса

Согласно данному подходу [4] для каждого конкретного (не абстрактного) класса иерархии наследования создается своя таблица. Каждая таблица содержит столбцы, соответствующие свойствам, реализованным в классе, и свойствам, унаследованным классом.

Поэтому поля базового класса дублируются во всех таблицах его производных классов. Каждый объект отображается на отдельную строку базы данных. На рис. 4 показан пример физической модели для отображения классов, показанных на рис. 1, по данному методу. В физической модели для каждого определенного класса, объекты которого можно создавать, существует соответствующая таблица, например, *BankingCredit* или *BondedIndebtedness*. Но для класса *Contract* нет соответствующей таблицы, так как он абстрактный и нельзя создавать объекты данного класса.

Значимой проблемой данного метода являются ключи. Каждая таблица содержит свой собственный первичный ключ, и значения ключей должны быть уникальны не только в рамках одной таблицы, но и в рамках всей иерархии. Данную проблему можно решить, если в качестве значения идентификатора объекта брать составной ключ, включающий как значение первичного ключа, так и имя таблицы, или использовать последовательности, если их поддерживает база данных. Также функцию задания значения первичного ключа можно делегировать средству ОРП, что также решит проблему.

Другой важной проблемой данного подхода является проблема поддержания ссылочной целостности базы данных. Так как в данном при-

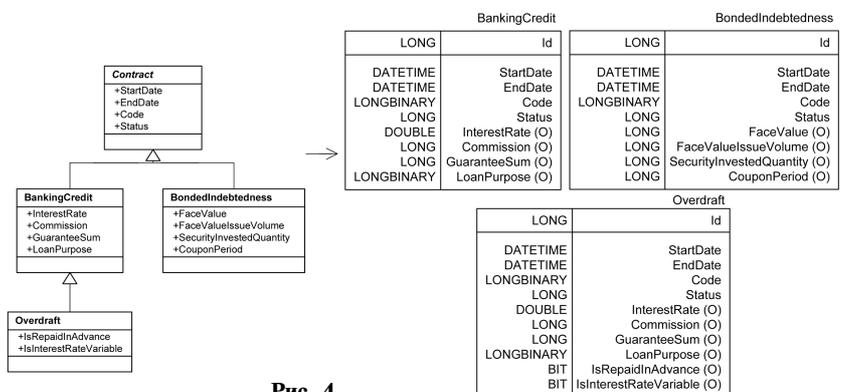


Рис. 4

мере класс *Contract* является абстрактным, то для него не создается таблица. Но все унаследованные классы являются конкретными, и для них создаются таблицы с первичными ключами. В такой ситуации невозможно создать ограничение целостности для внешнего ключа, значение которого ссылалось бы на объекты *Contract* (объекты *BankingCredit*, *BondedIndebtedness* и *Overdraft*, так как они являются подклассами *Contract*), так как нет единой таблицы со значениями первичного ключа. В такой ситуации либо пренебрегают ограничением целостности на уровне базы данных, либо создают по одной таблице для каждого из отношений, моделируя связь "многие ко многим".

При отображении наследования данным методом каждый объект представлен в базе данных одной строкой в конкретной таблице. При загрузке списка объектов, если заранее известен тип загружаемых объектов, можно делать запросы не ко всем таблицам, а только к той, в которой хранится запрашиваемый объект. Если тип объекта заранее не известен, или требуется загрузить множество объектов разных типов иерархии, то необходимо выполнить запросы ко всем таблицам конкретных классов, что снижает производительность.

Наследование с обобщенной табличной структурой

Четвертый подход для отображения иерархических структур в реляционную базу данных заключается в применении общего, управляемого метаданными, способа отображения. Этот подход не привязан к иерархическим структурам с наследованием, он применим во всех случаях отображения. На рис. 5 представлена схема данных для хранения значений свойств объектов и для обхода иерархических структур. Схема, представленная на рис. 5 не полна, она может быть расширена, например, для отображения ассоциаций. Значение каждого отдельного свойства объекта хранится в таблице *Value*, следовательно, при отображении объекта с 10 свойствами будет создано 10 записей, по одной для каждого свойства. Значение столбца *ObjectId* является уникальным идентификатором конкретного объекта (данное утверждение верно, если для всех объектов применяется единый способ назначения идентификаторов). Таблица *Type* содержит список базовых типов, таких как, дата, строка, денежная сумма, целое число и т. д. Эта информация необходима для правильной работы со значением, которое хранится в таблице *Value* в столбце *Value*.

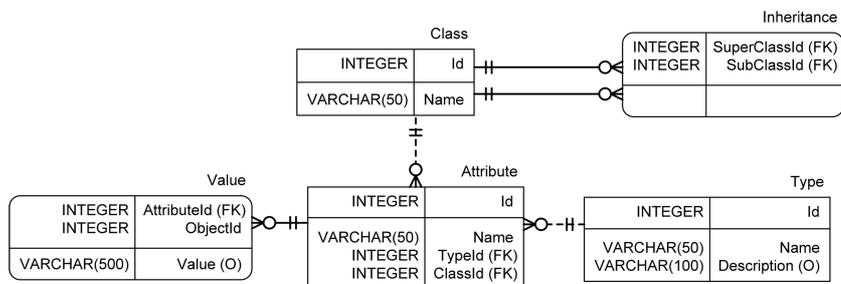


Рис. 5

Рассмотрим пример проекции одного объекта в данную схему. Для проекции объекта *BondedIndebtedness*, представленного на рис. 1, необходимо добавить восемь записей в таблицу *Value* — по одной записи для каждого из свойств, которые определены в классе или наследуются им от базового класса *Contract*. Можно также добавить дополнительные записи для хранения временной метки, если используется оптимистический контроль параллельного доступа. В таблице *Class* будут добавлены две записи: запись для класса *Contract* и запись для класса *BondedIndebtedness*. А в таблице *Attribute* будут записи для каждого свойства, определенного в классе *Contract* и *BondedIndebtedness*, со ссылкой на записи в таблице *Type*.

Таблица *Inheritance* является ключевой в проекции иерархий. Для каждого класса будет создана запись в таблице *Class*. Также будет запись в таблице *Inheritance*, у которой значение *SuperClassId* будет ссылаться на запись в таблице *Class*, которая представляет класс *Contract*. А значение *SubClassId* будет ссылаться на запись в таблице *Class*, которая представляет класс *BondedIndebtedness*. Для проекции остальной части иерархии классов необходимо добавить по одной записи в таблицу *Inheritance* для каждого отношения.

Сравнение стратегий и методика выбора

Ни один из описанных выше подходов к отображению не идеален для всех ситуаций, как это показано в таблице. Опыт показывает, что вначале лучше применить подход с одной таблицей для иерархии, а потом, при необходимости, использовать другой подход. Иногда целесообразно сразу начинать с подхода отображения каждого класса в отдельную таблицу. Данный подход позволяет создавать программный код, который более понятен и гибок к изменениям. Подхода к отображению каждого конкретного класса в отдельную таблицу лучше избегать на ранних этапах проекта, потому что данный подход не обеспечивает гибкости внесения изменений. Подход с использованием обобщенной табличной структуры плохо поддается масштабированию.

Характеристика системы	Способ отображения			
	Наследование с одной таблицей	Наследование с таблицами для каждого класса	Наследование с таблицами для каждого конкретного класса	Наследование с обобщенной табличной структурой
Скорость доступа к данным	Все данные содержатся в одной таблице и отпадает необходимость в соединениях, что увеличивает производительность. Но необходимо учитывать блокировки	Более медленный подход при чтении и записи данных, особенно если глубокая иерархия, потому что необходимо работать с множеством таблиц	Лучшая производительность при доступе к конкретным объектам, если заранее известен их тип. В некоторых случаях необходимо использовать несколько запросов к базе данных, что снижает производительность	Подходит только для небольших объемов данных, так как для доступа к одному объекту необходимо считать большое число записей
Дублирование структуры схемы базы данных	В схеме базы данных создается только одна таблица, куда сохраняется вся иерархия. Нет дублирования схемы	Схема содержит много таблиц, по одной для каждого класса. Нет дублирования схемы	Поля базового класса дублируются во всех таблицах его производных классов	Универсальная схема данных. Дублирования нет
Сложность построения запросов и преобразования данных средством ОРП	Простой в реализации подход. Все запросы строятся к одной таблице, тип объекта определяется по значению в кодовом поле	Необходимо выполнять соединение нескольких таблиц, а при некоторых запросах необходимо использовать внешнее соединение. Тип объекта определяется по наличию записей в соответствующих таблицах	Каждая таблица удобна в использовании, так как содержит данные только об одном объекте, но если неизвестен тип запрашиваемого объекта, то необходимо выполнить запросы ко всем таблицам конкретных классов	Метод прост в реализации. Запросы строятся к универсальной схеме данных. Данный метод может быть расширен различными метаданными
Сложность построения произвольных запросов на выбор или изменение данных	Просто построить произвольный отчет, так как данные хранятся в одной таблице	Относительно сложно построить произвольный отчет, так как данные распределены по разным таблицам и необходимо использовать соединения	Просто построить произвольный отчет, но необходимо строить несколько запросов или использовать объединение	Построение отчетов по данным очень затруднено, так как используется универсальная схема
Сложность внесения изменений в класс	Одна таблица создает единое пространство имен для всех свойств иерархии. Изменения вносятся в одну таблицу	Так как дублирование схемы нет, то изменения вносятся в одну таблицу	При изменении класса необходимо внести изменения в таблицу класса и во все таблицы подклассов	Это очень гибкий метод, так как позволяет быстро вносить изменения, не меняя структуры таблиц
Сложность внесения изменений в иерархию классов	Легко добавить новые классы в иерархию — достаточно добавить новые столбцы для дополнительных данных. Перемещение свойств из класса в суперкласс или в производственный класс не требует внесения изменений в таблицу	Легко вносить изменения в базовый класс и добавлять новые производные классы. Для перемещения свойств необходимо изменять схему базы данных	Для добавления нового класса достаточно добавить таблицу со столбцами, соответствующими всем свойствам класса. Перемещение свойств из класса в суперкласс или в производный класс, если они конкретные, требуют внесения изменений в схему	

Очень важно понимать, что можно комбинировать первые три подхода. Их даже можно комбинировать при отображении одной большой иерархии. Все сказанное можно обобщить в таблице, приведенной выше.

Выбор того или иного способа отображения обуславливается требованиями к системе. Если требуется высокая производительность и простота внесения изменений, как в иерархию классов, так и в сами классы, и при этом иерархия простая, то предпочтительным является наследование с одной таблицей. Если в системе сложная иерархия и в нее часто вносятся изменения, то предпочтительным является наследование с таблицами для каждого класса, так как метод прост в понимании и нет дублирования схемы. Наследование с таблицами для каждого конкретного класса целесообразно использовать, если уже есть существующая схема

или если структура схемы создается из других соображений, например, для интеграции с другими системами. Наследование с обобщенной табличной структурой является самым простым в реализации, но работает только на небольшом объеме данных. При данном решении построить произвольный отчет очень затруднительно.

Список литературы

1. **Ambler S. W.** The Object-Relational Impedance Mismatch. [Http://www.agiledata.org/essays/impedanceMismatch.html](http://www.agiledata.org/essays/impedanceMismatch.html)
2. **Fussell M. L.** Foundations of Object-Relational Mapping. <http://www.chimu.com/publications/objectRelational/>
3. **Тарасов С.** Обзор средств объектно-реляционной проекции (ORM) для платформы .NET. [Http://www.arbinada.com/main/node/33](http://www.arbinada.com/main/node/33)
4. **Фаулер М.** Архитектура корпоративных программных приложений.: Пер. с англ. — М.: Издательский дом "Вильямс", 2006. 544 с.

П. Г. Редреев, аспирант,
Омский филиал института математики
Сибирского отделения
Российской академии наук им. С. Л. Соболева

Построение табличных приложений со списочными компонентами

Рассматривается технология автоматизации построения сложноструктурированного табличного представления данных со списочными компонентами из исходной реляционной базы данных. При построении формально определены промежуточная и целевая модели данных. Приведены примеры генерации пользовательских приложений.

Ключевые слова: реляционная база данных, аналитическая обработка данных, многомерная модель данных.

Введение

Технология OLAP (Online Analytical Processing) [1—3] при работе с базами данных предназначена для построения представления, удобного для выполнения анализа данных. Далее это представление может быть обработано алгоритмами Data Mining для обнаружения закономерностей в исходных данных. Однако большинство OLAP-представлений используется для умозрительного анализа (сводная информация по медицинским данным, банковским операциям и т. п.). Традиционной моделью исходного представления данных для аналитической обработки является гиперкуб с определенными свойствами (иерархия измерений, суммаризуемость и т. д.) [3]. Формирование гиперкубов осуществляется параллельно с существующими базами данных, что делает необходимым периодическую их актуализацию. Кроме того, структура гиперкубов предполагает заранее предусмотренные виды анализа данных [4], и, если требуется провести какой-либо незапланированный анализ, то хранилище данных в виде гиперкубов будет бесполезно.

В данной работе предполагается, что основой аналитической работы пользователя является формирование новых гиперкубов, а не многократное формирование реализации одного и того же гиперкуба. Следовательно, основное внимание необходимо акцентировать на сокращении времени формирования схемы нового гиперкуба, а формирование представления гиперкуба должно быть выполнено автоматически алгоритмами, соответствующими выбранному классу схем.

В качестве основы для автоматизации формирования представления гиперкуба предлагается использовать формальное определение промежуточной и целевой моделей данных, задающих не только схемы, но и способы формирования представлений. В работе [2] рассматриваются модель гиперкуба, обладающая достаточной общностью, и алгебраические операции над гиперкубами. Указывается на возможность использования модели в рамках реляционной технологии баз данных. Однако не рассматриваются алгоритмы, преобразующие реляционное представление данных в гиперкубическое.

В данной работе инструментарий для построения пользовательского представления формирует приложения без привлечения языка программирования. В работе [5] для автоматизации формирования гиперкуба предлагается использовать последовательность преобразований

$$RRD \Rightarrow TJ \Rightarrow ST,$$

где RRD — реляционное представление данных; TJ — таблица соединений; ST — гиперкуб "семантическая трансформация". В данном случае RRD — представление исходной модели данных, ST — целевой. Представление TJ является промежуточным.

Рассмотрим в качестве целевой модели данных "композиционную таблицу", которая является обобщением модели "семантическая трансформация" на случай списка значений в одной ячейке, разделенных знаками препинания, т. е. списочных компонентов. В качестве промежуточной модели будем использовать "таблицу связанных соединений", накладывающую более жесткие ограничения на представление чем "таблица соединений", целесообразность которых будет рассмотрена ниже.

Примером "композиционной таблицы" является план учебной нагрузки вуза (табл. 1).

Модель данных "Таблица связанных соединений"

В работах, посвященных построению гиперкубического представления данных, в качестве промежуточных моделей используются SQL-таблицы. Средств языка SQL достаточно для формирования любого табличного представления. Однако интерфейс между базой данных и хранилищем данных приходится программировать. Модель данных "таблица связанных соединений" является частным случаем модели "таблица соединений" [5]. Представление этой таблицы формируется автоматически из исходной базы данных, что исключает затраты на программирование. Совокупность свойств этой таблицы является достаточной для формирования большинства приложений.

Рассмотрим преобразование представления реляционной базы данных (БД) со схемой R_1, R_2, \dots ,

План учебной нагрузки

Код	Наименование дисциплины	Трудоемкость										Число недель в семестре			
												17		18	
		Семестр													
		1	1	2	2										
Всего	Лекции	Лаб.	Прак.	Семинар	Сам. раб.	Экзамен	Зачет	Курс. пр.	Курс. раб.	Л	Пр	Л	Пр		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ГСЭ	Общие гуманитарные и социально-экономические дисциплины														
ГСЭ.Ф.00	Федеральный компонент	1260													
ГСЭ.Ф.01	Иностранный язык	340	0		176		164	4	1, 2, 3				3		2
ГСЭ.Ф.02	Физическая культура	408	0		140		268		1, 2, 3, 4				2		2
ГСЭ.Ф.03	Отечественная история	108	34		34		40	1				2	2		
ГСЭ.Ф.06	Правоведение	70	18		36		16	4							
ГСЭ.Ф.07	Психология и педагогика	70	18		18		34	3							
ГСЭ.Ф.10	Философия	108	34		34		40	1				2	2		
ГСЭ.Ф.11	Экономика	156	36		36		84	2			2			2	2
ГСЭ.Р.00	Национально-региональный (вузовский) компонент	270													
ГСЭ.Р.01	Социология	90	18		18		54		3						
ГСЭ.Р.02	Политология	90	34		18		38		3						
ГСЭ.Р.03	Деловой иностранный язык	90	0		36		54		5,6						

Примечание. Трудоемкость для столбцов с 3-го по 8-й измеряется числом учебных часов, для столбцов с 9-го по 12-й — номерами семестров.

R_k в "таблицу связанных соединений" (C, l) , где C — схема отношения, определенная на множестве атрибутов A_1, A_2, \dots, A_n , l — вектор вхождения длины k . Определим принцип формирования кортежей $t \in c$, где c — реализация схемы отношения C . Рассмотрим соединения, удовлетворяющие локальному свойству соединения без потерь информации (ССБПИ) [6] для всевозможных сочетаний из k по m реализаций r_j , $m = 1, \dots, k$. Для каждого кортежа u , в котором атрибуты, являющиеся координатами ячейки "композиционной таблицы", имеют определенные значения каждого соединения, сформируем кортеж t по следующим правилам: $t[A_j] = u[A_j]$, если атрибут A_j принадлежит соединению, и $t[A_j] = \text{emp}$ в противном случае, где emp — пустое значение. Каждому кортежу поставим в соответствие битовый вектор $l(t) = (l_1(t), l_2(t), \dots, l_k(t))$, где $l_j(t) = 1$, если реализация r_j схемы R_j участвует в текущем соединении, и $l_j(t) = 0$ в противном случае.

Рассмотрим отношение частичного порядка над кортежами $t \in c$. Кортеж $t \in c$ является менее определенным или равным кортежу $t' \in c$, когда для любого атрибута A_j выполнено условие: если

$t[A_j] \neq t'[A_j]$, то $t[A_j] = \text{emp}$ и $l_j(t') \geq l_j(t)$, $j = 1, \dots, k$. В этом случае будем писать $t \leq t'$ и назовем кортеж t подчиненным кортежу t' . В представлении c достаточно хранить только кортеж t' , который содержит в себе все менее определенные либо равные кортежи.

Пусть $X(J) = R_{j(1)} \cup R_{j(2)} \cup \dots \cup R_{j(m)}$, где $J = j(1), j(2), \dots, j(m)$. Определим операцию проекции на множестве c : $\pi_{X(J)}(c)$ есть совокупность кортежей $u[X(J)]$, определенных на множестве атрибутов $X(J)$, где для каждого $u[X(J)]$ существует кортеж $t \in c$ такой, что $u[X(J)] = t[X(J)]$ и $l_{j(i)}(t) = 1$, $i = 1, 2, \dots, m$.

Основываясь на способе формирования C -таблицы, сформулируем ее важные свойства.

Свойство 1. Для любого связанного соединения отношений $R_{j(1)}, R_{j(2)}, \dots, R_{j(m)}$, где $m = 1, 2, \dots, k$, выполнено $R_{j(1)} \bowtie R_{j(2)} \bowtie \dots \bowtie R_{j(m)} = \pi_{X(J)}(c)$, где \bowtie — операция естественного соединения.

Это основное свойство "таблицы связанных соединений" показывает, что она содержит все связанные соединения отношений БД, в том числе и исходные отношения. Это позволяет считать

Общий вид "композиционной" таблицы

X_1	X_2	...	X_l						
				Y_1		...		Y_N	
				Z_1		...		Z_N	

"таблицу связанных соединений" обобщением универсального реляционного отношения [6]. Кроме того, операция проекции задает обратное преобразование из "таблицы связанных соединений" в БД.

Свойство 2. Реализация C -таблицы c всегда существует и единственна для любой схемы реляционной БД.

Существование "таблицы связанных соединений" следует из способа построения, единственность доказывается аналогично единственности "таблицы соединений" [7].

В заключение данного раздела рассмотрим обоснование выбора "таблицы связанных соединений" в качестве промежуточного представления данных. Поскольку таблица содержит все комбинации отношений из числа исходных отношений, в том числе и отдельно взятых, то она будет содержать все данные для построения "композиционной таблицы". Вместе с тем, поскольку для каждого соединения должно быть выполнено ССБПИ, то в таблице не появятся кортежи, которые противоречили бы прикладной области. К сожалению, нельзя привести строгое формальное обоснование для данных утверждений, так как отсутствует представление данных, которое послужит эталоном правильного набора кортежей после наложения на него ограничений. Претендентом на эту роль может служить универсальное реляционное отношение (УРО) [6]. Однако оно существует не для всех реализаций БД и необходимо искать расширение этого отношения, порождая дополнительные проблемы доказательства корректности такого расширения. Обобщением УРО на произвольные реляционные БД является "таблица связанных соединений", и, поскольку она сама используется в качестве промежуточного представления данных, то проблема доказательства корректности построений становится неактуальной.

Модель данных "Композиционная таблица"

Рассмотрим построение представления "композиционной таблицы". Обозначим: R_1, R_2, \dots, R_k — исходные реляционные отношения, C — соответствующая этим отношениям "таблица связанных соединений", CT — результирующее отношение.

Пусть X, Y_i, Z_i — множества атрибутов из R ($i = 1, 2, \dots, N$). Атрибуты X остаются неизменными в CT и являются наименованиями строк, значения атрибутов Y_i становятся именами столбцов в CT , домены атрибутов Z_i , дополненные пустым значением, распределяются между доменами новых атрибутов, введенных для значений Y_i . W_i — дополнительное множество атрибутов, которые

используются в логических формулах-ограничениях, но в CT отсутствуют. Естественными являются ограничения: $X \cap Y_i = \emptyset, X \cap Z_i = \emptyset, Y_i \cap Z_i = \emptyset$ ($i = 1, 2, \dots, N$). $W_i \in R \setminus (X \cup Y_1 \cup \dots \cup Y_N \cup Z_1 \cup \dots \cup Z_N)$, $|\text{Dom}(Y_i)| = L_i, |Z_i| = M_i$. Схема результирующего представления строится из исходных отношений по следующему правилу:

$$\text{Sch}(C) = \{X, Y_1, \dots, Y_N, Z_1, \dots, Z_N, W_1, \dots, W_N\} \Rightarrow \\ \Rightarrow \text{Sch}(CT) = \{X \cup \text{Dom}(Y_i) \times \{Z_i\}, i = 1, 2, \dots, N\},$$

где Sch — схема описания отношения; Dom — область значения атрибута, $\text{Dom}(Y_i) = \text{Dom}(Y_{i1}) \times \text{Dom}(Y_{i2}) \times \dots, Y_{ij} \in Y_i$. Символ \cup обозначает, что "композиционная таблица" состоит из подтаблиц со схемами $\{X, \text{Dom}(Y_i) \times \{Z_i\}\}, i = 1, 2, \dots, N$. Представление CT со схемой $\text{Sch}(CT)$ является плоской таблицей. Затем оно преобразуется в "композиционную таблицу", общий вид которой представлен в табл. 2.

Выбор отношений, участвующих в формировании "таблицы связанных соединений", осуществляется автоматически по множествам атрибутов $X, Y_i, Z_i, i = 1, 2, \dots, N$, с использованием связей между отношениями БД. К сожалению, не найдены условия, которые заменили бы проверку ССБПИ для множества отношений. Поэтому рассмотрим эвристический метод формирования множеств отношений. Эвристики позволят сократить число множеств, для которых выполняется проверка ССБПИ.

В основу эвристических правил положим выбор тех отношений, которые имеют наибольший интервал значений для заданных пользователем атрибутов (атрибуты наиболее информативны). Пусть атрибут $A_l \in X \cup \{Y_i\} \cup \{Z_j\} \cup \{W_j\}$, тогда:

- выбираем отношение R_p , содержащее атрибут A_l в качестве ключевого;
- если несколько отношений содержат атрибут A_l в ключе, то выбираем отношение с минимальным числом атрибутов в ключе;
- в иных случаях выбираем отношение с максимальным числом кортежей.

Для построения множества отношений, с наибольшей вероятностью удовлетворяющего локальному ССБПИ, используем следующее эври-

стическое правило: выбираем отношения, которые являются подчиненными к уже выбранному отношению R_i .

Пусть $R = (R_1, R_2, \dots, R_k)$ — множество всех отношений; V — множество атрибутов из множеств $X, Y_i, Z_i, i = 1, 2, \dots, N$, выбранных для построения "композиционной таблицы". Следующий алгоритм осуществляет построение множества наборов отношений, которые могут быть использованы при формировании "таблицы соединений":

Шаг 1. $P = \emptyset, R^+ = \emptyset$. Для каждого атрибута A_j из множества V :

Если в R имеются отношения, содержащие данный атрибут в качестве ключевого, то добавляем во множество R^+ отношение $R_{m(i)}$ с минимальным числом атрибутов в ключе.

Иначе добавляем во множество R^+ отношение R_j , содержащее атрибут A_j , с максимальным числом кортежей.

Если для результирующего множества R^+ выполнено локальное ССБПИ, то выход. Иначе переходим на шаг 2.

Шаг 2. $R^0 = R^+$ — необходимый набор отношений. Присоединяем к R^+ отношение $R_i \in R$ такое, что $\exists R_j \in R^+, \text{ между } R_i \text{ и } R_j \text{ установлено отношение } 1:M \text{ или } 1:1, R_i \text{ является подчиненным к } R_j$, т. е. выполнена типизированная зависимость включения [8].

Шаг 3. $R' = R^+ \setminus R^0, n = |R'|$. Для каждого сочетания I_m из всевозможных сочетаний отношений множества R' из n по $m = 1, \dots, n$ выполняем проверку:

Если для отношений $\{R_j | j \in I_m\} \cup R^0$ выполнено локальное ССБПИ, то добавляем набор отношений $\{R_j | j \in I_m\} \cup R^0$ во множество P .

Если после перебора всех сочетаний множество $P = \emptyset$, то ошибка: некорректно определены $X, Y_i, Z_i (i = 1, 2, \dots, N)$.

Замечание. На шаге 1 алгоритма: если несколько отношений содержат A_j в качестве неключевого или если несколько отношений содержат его в качестве ключа, то для уточнения семантики приложения выбор из отношений из различных вариантов для включения в R^+ предоставляется пользователю.

Если в результате работы алгоритма получаем несколько наборов отношений, то выбор набора отношений из различных вариантов также предоставляется пользователю для уточнения семантики приложения.

Сравнение "композиционной таблицы" с существующими многомерными моделями

В литературе [2, 3] кубические модели данных трактуют данные как n -мерные кубы. Обычно данные рассматриваются либо как параметры, которые являются числовыми, либо как измерения,

которые, как правило, текстовые и характеризуют факты. Для модели данных, рассмотренной в статье [2], понятие факта может пониматься как точка в многомерном пространстве, каждое измерение многомерного пространства структурировано как иерархия уровней измерения. "Композиционная таблица" со схемой $\{X, \text{Dom}(Y) \times \{Z\}\}, |Z| = 1$ может быть рассмотрена как проекция реализации этой модели с тремя измерениями. Определим данное преобразование. Пусть x, y, z — множества уровней измерений многомерного объекта M с тремя измерениями; ct — представление "композиционной таблицы", $ct = \emptyset$. Каждому факту $f \in M$ соответствуют множества значений уровней измерений $x_1 \in x, y_1 \in y, z_1 \in z$. Пусть строка $ct_1 \in ct: ct_1[X] = x_1, ct_1[y_1.Z_1] \neq \text{emp}$. Дополняем список $ct_1[y_1.Z_1]$ строкой, состоящей из значений z_1 с разделителями. Если в таблице ct нет соответствующей строки для ввода значений, то формируется новая строка $ct_1: ct_1[X] = x_1, ct_1[y_1.Z_1] = \text{строка}$, состоящая из значений z_1 с разделителями.

Оценим модель "композиционная таблица" по одиннадцати требованиям к многомерным моделям данных [2]:

1. Явные иерархии в измерениях.
2. Симметричное обращение с измерениями и мерами.
3. Множественные иерархии в каждом измерении.
4. Поддержка семантики агрегирования.
5. Нестрогие иерархии.
6. Несюръективные иерархии.
7. Непокрывающие иерархии.
8. Отношения многие ко многим между фактами и измерениями.
9. Обработка изменения и времени.
10. Обработка различных уровней гранулярности.
11. Обработка неточности.

Для модели "композиционная таблица" иерархии возможны только во множествах, являющихся координатами. Иерархии поддерживаются явно по схеме измерений. Например, для измерения "Время" должна поддерживаться иерархия День < Месяц < Год. Возможны множественные, нестрогие, несюръективные, непокрывающие иерархии. В случае несюръективных и непокрывающих иерархий отсутствующему значению уровня иерархии соответствует пустая ячейка измерения X или Y_i "композиционной таблицы". В различных приложениях атрибуты-меры могут трактоваться как атрибуты-измерения, и наоборот. "Композиционная таблица" не поддерживает семантику агрегации данных. Такие аспекты агрегации данных, как исключение двойного подсчета данных и исключение добавления неаддитивных данных, могут быть реализованы как операции преобразования представления "композиционной таблицы".

Между фактами и измерениями допустимо наличие отношений многие ко многим. Это обусловлено тем, что в ячейках "композиционной таблицы" содержится несколько значений, разделенных знаками препинания. Выполнение временного анализа данных и обработка неточных данных в рассматриваемой модели не поддерживаются. Анализ данных с различной гранулярностью может быть реализован с помощью введения соответствующей операции преобразования представления "композиционной таблицы".

Программная реализация

Для формирования интерфейса между реляционной БД и представлением "композиционной таблицы" используется технология на основе метода межмодельного преобразования данных [9]. Разработанные на ее основе алгоритмы реализованы в виде программного обеспечения. При этом была реализована последовательность преобразований данных

$$RRD \Rightarrow C \Rightarrow CT,$$

где RRD — реляционное представление данных; C — "таблица связанных соединений"; CT — "композиционная таблица".

Разработаны два приложения, использующие базу данных в формате MDB.

Приложение 1 содержит следующие компоненты:

1. Формирование множества атрибутов X . Общий список атрибутов формируется автоматически объединением всех атрибутов всех таблиц базы данных. Пользователю необходимо выбрать из списка нужные атрибуты.

2. Выбор атрибута Y из общего списка атрибутов.

3. Формирование множества атрибутов Z .

4. Выбор атрибутов сортировки, по которым сортируются строки в "композиционной таблице".

5. Кнопка для сохранения информации о выбранных множествах атрибутов в файл.

Приложение 1 позволяет задать необходимые значения параметров "композиционной таблицы" со схемой $\{X, \text{Dom}(Y) \times \{Z\}\}$. Работу с этим приложением осуществляет администратор.

Приложение 2 содержит следующие компоненты:

1. Электронная таблица для отображения "композиционной таблицы".

2. Кнопка "Передать данные", предназначенная для передачи данных из базы данных в электронную таблицу.

Приложение 2 формирует "композиционную таблицу" по заданным в файле параметрам.

Рассмотрим примеры генерации приложений. Для этого используем следующую схему БД [5]:

R_1 = Студенты (№ студента, № группы, ФИО студента)

R_2 = Список_групп (№ группы, Код группы)

R_3 = Предметы (№ предмета, Предмет)

R_4 = Преподаватели (№ преподавателя, ФИО преподавателя)

R_5 = Неделя (№ дня недели, День недели)

R_6 = Начало_занятий (№ пары, Время начала пары)

R_7 = Оценки (№ студента, № группы, № предмета, Оценка)

R_8 = Расписание (№ группы, № дня недели, № пары, № предмета, № преподавателя, № аудитории).

Для формирования приложения "Обобщенное расписание" используем следующие параметры:

Атрибуты множества X : № дня недели, День недели;

Атрибуты множества Y : Код группы;

Атрибуты множества Z : Предмет;

Атрибуты сортировки: № дня недели.

Приложение "Обобщенное расписание" представлено в виде табл. 3.

Для формирования приложения "Расписание преподавателей" используем следующие параметры:

Атрибуты множества X : ФИО преподавателя;

Атрибуты множества Y : Код группы;

Атрибуты множества Z : День недели;

Атрибуты сортировки: ФИО преподавателя.

Приложение "Расписание преподавателей" представлено в виде табл. 4.

Заключение

Рассмотрено формирование приложений из исходного реляционного представления данных.

Таблица 3

Приложение "Обобщенное расписание"

№ дня недели	День недели	1-330	2-110
1	Понедельник	История, Физика, Философия, Химия	История
2	Вторник	Философия, Химия	
3	Среда		
4	Четверг		
5	Пятница		
6	Суббота		

Таблица 4

Приложение "Расписание преподавателей"

ФИО преподавателя	М-210	М-220
Артамонов В. А.	Понедельник	Понедельник
Дергачев А. С.	Понедельник, Четверг	
Иванов А. А.	Понедельник	Четверг
Ильичев П. А.		
Петров Д. А.		
Чигишев О. М.	Понедельник	

Использование в качестве промежуточной модели "таблицы связанных соединений" позволило сделать автоматическим выбор отношений, участвующих в ее построении, при генерации приложений.

В дальнейшем предполагается разработка алгоритма обратного преобразования данных при их передаче из приложения в базу данных. При этом будут реализованы функции редактирования данных в табличных приложениях. Существенно расширить способы обработки данных, полученных в приложении, позволит организация интерфейса по данным с SPSS.

Список литературы

1. Педерсен Т. Б., Йенсен К. С. Технологии многомерных баз данных // Открытые системы. 2002. № 1.

2. Pedersen T. B., Jensen C. S., Dyreson C. E. A foundation for capturing and querying complex multidimensional data // Information Systems. 2001. № 26 (5). P. 383—423.

3. Lechtenborger J., Vossen G. Multidimensional normal forms for data warehouse design // Information Systems. 2003. Vol. 28. N 5. P. 415—434.

4. Armstrong R. Seven Steps to Optimizing Data Warehouse Performance // Computer. 2001. Vol. 34. N 12. P. 76—79.

5. Зыкин С. В. Формирование гиперкубического представления реляционной базы данных // Программирование. 2006. № 6. С. 348—354.

6. Ульман Дж. Основы систем баз данных. М.: Финансы и статистика, 1983. 334 с.

7. Зыкин С. В. Построение отображения реляционной базы данных в списковую модель данных // Управляющие машины и системы. 2001. № 3. С. 42—63.

8. Levene M., Vincent M. W. Justification for Inclusion Dependency Normal Form // IEEE Transactions on Knowledge and Data Engineering. 2000. Vol. 12. N 2. P. 281—291.

9. Зыкин С. В. Метод формирования представлений данных для работы с информационными ресурсами // Омский научный вестник. 2006. № 3 (36). С. 124—126.

УДК 004.3.01

В. Е. Туманов, канд. хим. наук,
зав. сектором информационного обеспечения
научных исследований,
Институт проблем химической физики РАН,
e-mail: tve@icp.ac.ru

Предметно-ориентированные системы научной осведомленности

Обсуждается понятие предметно-ориентированных систем научной осведомленности, основные требования к таким системам и их общей программно-технологической архитектуре. Показаны место и роль систем научной осведомленности в научных информационных ресурсах. Рассмотрены программные механизмы формирования хранилищ знаний, их архитектура для предметно-ориентированных систем научной осведомленности. Приведен пример прототипа предметно-ориентированной системы научной осведомленности в физической химии радикальных реакций.

Ключевые слова: научные базы данных, научные информационные системы, системы научной осведомленности, предметно-ориентированные системы научной осведомленности, хранилища знаний, встроенные экспертные системы, искусственные нейронные сети.

Введение

В результате научных экспериментов и компьютерного моделирования постоянно накапливаются большие объемы данных, которые организуются в электронные информационные ресурсы: базы и

хранилища данных, электронные информационные и вычислительные системы, научные центры данных. Такие информационные ресурсы становятся местом накопления, хранения, верификации, извлечения, использования и распространения профессиональных и корпоративных знаний.

Эффективное развитие науки и высоких технологий требует интенсивной обработки и анализа фундаментальных знаний, накопленных в различных исследовательских организациях, что приводит к потребности в развитии информационных технологий накопления, извлечения и анализа предметно-ориентированных профессиональных знаний на основе разработки универсальных и специализированных моделей организации и представления научных данных и знаний в электронных ресурсах.

Целью настоящей статьи является попытка определить класс предметно-ориентированных систем научной осведомленности и сформулировать основные требования к таким системам, их общей программно-технологической архитектуре, показать эффективность их применения на практике.

Научные базы данных и научные информационные системы

Разработка и создание научных баз данных являются одним из направлений применения информационных технологий для компьютерной поддержки научных исследований. *Научными базами данных* [1] называют такие базы данных, которые содержат информацию о какой-либо предметной области науки, т. е. информацию, исполь-

зуюмую или получаемую в процессе научных исследований, научных экспериментов или компьютерного моделирования. Согласно каталогу [2] уже в середине 80-х годов прошлого века насчитывалось около 7500 научных баз данных.

Исторически при разработке научных баз данных использовались различные модели данных и средства их компьютерной реализации. Поэтому при коллективном использовании коллекций научных баз данных актуальным является вопрос стандартизации доступа пользователей к таким коллекциям. Например, в одной из крупнейших глобальных информационных систем в области науки и технологий STN International Databases Science and Technology [3] для онлайн-доступа к базам данных разработан специальный языковой интерфейс.

Развитие науки и технологий приводит к интенсивному обмену и коллективному использованию знаний и технологических достижений, полученных научным сообществом и исследовательскими центрами компаний. Компьютерные системы для доступа к научной и академической информации называются *научными информационными системами* (или в англоязычной литературе — Current Research Information System, CRIS). Международная профессиональная ассоциация euroCRIS (<http://www.eurocris.org/>) рекомендует переводить термин CRIS на русский язык как *информационные системы по актуальным научным исследованиям*¹.

В обзоре [4] дано определение термина научной информационной системы, приведен критический обзор исследований в области разработки таких систем и показана их важность для современной науки и технологии.

Основные требования к научным информационным системам следующие [5]:

- *полнота информации*, что приводит к необходимости вовлечения в систему как можно большее количество информационных ресурсов, относящихся к определенной области знаний;
- *релевантность документов*. Получаемые при автоматическом сборе документов из сети Интернет выборки данных могут оказаться шире, чем это определено предметной областью системы, поэтому в таких системах необходимы процедуры верификации поступающих данных на соответствие предметной области;
- *актуальность, полнота и достоверность происхождения документов*;
- *наличие интеллектуальных служб обслуживания пользователей*, что приводит к необходимости

¹ Устное сообщение Президента ассоциации euroCRIS Keith Jeffery, сделанное на Семинаре ассоциации euroCRIS в Москве 8–11.10.2008, проведенного совместно с ЦЭМИ РАН, ВАК РФ и компанией Парк-медиа.

поддерживать полнотекстовый поиск по атрибутам документов. Для ряда систем определена поддержка семантического поиска;

- *распределенность программно-технологической архитектуры*. Такое требование является необходимым условием для обеспечения полноты и актуальности представленной в системе информации;
- *структурированность информационного пространства*. Это требование является необходимым условием обеспечения сложных функций поиска, когда в системе недостаточно хранить только полнотекстовые описания;
- *историчность информации*. Для значительной части научной информации характерен короткий срок жизни и актуальности, поэтому в системе необходимо иметь описание жизненного цикла используемых информационных ресурсов в целях их восстановления на любой момент времени (из архива);
- *поддержка различных уровней абстракции* позволяет ускорять поиск информации пользователем без потери скорости извлечения и точности подбора информации;
- *ведение архива документов*. Несмотря на то, что большая часть научной информации быстро устаревает, система должна поддерживать возможность длительного хранения информации в электронных архивах в целях обеспечения ретроспективного поиска.

Большинство исследовательских учреждений имеют свои информационные системы, в которых аккумулируются данные о результатах ведущихся в них научных и научно-практических работ. Однако ни одно научное учреждение не имеет достаточно ресурсов, чтобы собирать и хранить информацию, полученную научным сообществом в какой-либо предметной области в целом. Поэтому для развития научных информационных систем актуальными задачами являются автоматизация процесса извлечения и накопления информации из различных источников и интеграция данных из научных информационных систем различных организаций.

Основными проблемами при разработке научных информационных систем являются:

- сбор информации (ученые не желают тратить время и силы на ввод информации);
- использование научных информационных систем (пока частота использования таких систем невысока, поскольку пользователи хотят иметь гарантии полноты и надежности представленной информации);
- отсутствие эффективных моделей представления неструктурированной информации и организации доступа к ней.

Одним из возможных путей решения этих проблем является создание узконаправленных, специализированных и сложных научных информационных систем с последующим объединением их на основе разработанных стандартов представления научной информации и развитой модели метаданных. Такими специализированными информационными научными системами могут стать системы научной осведомленности, о которых пойдет речь ниже.

Вместе с тем, как отмечено в [6], на задачи анализа данных в научных информационных системах существенно влияют такие факторы, как алгоритмическая сложность, нелинейность и пропускная способность программно-аппаратных средств, что требует разработки эффективных алгоритмов анализа и моделей представления данных, предназначенных для науки.

Системы научной осведомленности

В последнее время интенсивное развитие получили системы деловой осведомленности или бизнес-аналитики (*Business Intelligence System, BI System*), в основе которых лежат технологии складирования данных (*Data Warehousing*), анализа и извлечения знаний (*Data Mining*) [7]. Успех в использовании систем деловой осведомленности в бизнесе привел к идее разработки и создания систем научной осведомленности (*Science Intelligence System, SI System*).

Одна из первых попыток дать определение систем научной осведомленности была предпринята в статье R. Nackathorn [8]. В этой работе *система научной осведомленности* определена "как информационная инфраструктура, которая обеспечивает принятие решений и совместную работу научного сообщества в рамках выделенной предметной области знаний". Там же рассмотрена общая архитектура таких систем в разрезе категорий основных пользователей и использования современных информационных технологий (рис. 1).

Как видно на рисунке, в системах научной осведомленности научные решения, методология и методы исследований интегрируются в общую библиотеку решений, а данные из разнородных источников — в общее хранилище данных, которое через предметно-ориентированные информационные ресурсы поставляют информацию пользователям: ученым, технологам, представителям промышленности и государства. Особенностью систем научной осведомленности является предоставление пользователям, помимо собственно профес-

сиональной информации, инструментария для анализа данных.

К основным требованиям, которые предъявляются к таким системам, относятся следующие.

Поскольку интенсивность потоков получаемых научных данных непрерывно растет, особенно в крупномасштабных научных проектах, используемые информационные технологии должны быть способны обработать эти потоки данных, что приводит к требованию *масштабируемости данных* в таких системах.

Источники данных зачастую имеют территориальную распределенность, что приводит к требованию наличия *стандартов представления данных и открытости* систем научной осведомленности для приема и обработки научных данных.

Как правило, полученные электронные массивы научных данных требуют применения сложных методов анализа, многократно и под различными углами зрения, что требует представления данных в форме, пригодной для применения различных *методов интеллектуального анализа данных (Data Mining)*.

От систем научной осведомленности требуется также наличие *механизмов своевременной доставки данных конечному потребителю* по первому требованию, включая также поиск опубликованных данных в системе научных библиотек.

Поскольку системы научной осведомленности предназначены для коллективного использования и совместной работы, они должны *предоставлять быстрый доступ к данным и простые способы обмена информацией* между работающими, возможно, в разных странах специалистами.

Перечисленные общие требования к системам научной осведомленности приводят к необходимости использования таких информационных

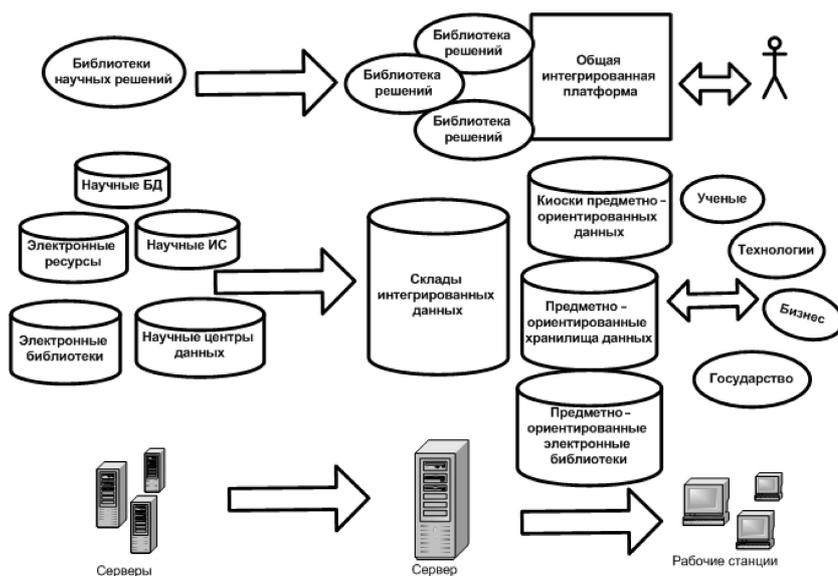


Рис. 1. Общая архитектура систем научной осведомленности

технологий, как порталные технологии, технологии складирования данных, XML-технологии, вычислительных кластеров серверов высокой готовности и ряда других. Реализация указанных требований в полном масштабе возможна лишь в крупномасштабных, хорошо финансируемых научных проектах.

Как видно из сказанного выше, системы научной осведомленности можно рассматривать как класс научных информационных систем, ориентированных на интеллектуальный анализ данных в целях получения новых знаний.

Предметно-ориентированные системы научной осведомленности

В [9] был рассмотрен общий подход к построению систем управления фундаментальными знаниями на примере физико-химических данных. В [10] был использован аналогичный подход для создания интеллектуальной информационной системы по физической химии радикальных реакций, при этом модель системы рассматривалась с точки зрения внедрения в нее элементов прикладного искусственного интеллекта для производства новых предметных знаний.

В национальном стандарте США ("*Proposed Draft American National Standard. Knowledge Management Vocabulary*", April, 2003) термин "производство знаний" определяется как:

- разработка и обеспечение новых знаний (OECD 1966:2);
- обстоятельства, при которых люди, группы людей и организации успешно генерируют новые знания и практики (OECD 2000:39).

Далее в тексте статьи под производством знаний понимается извлечение новых знаний из эмпирических данных в рамках компьютерной системы с участием человека методами прикладного искусственного интеллекта.

В [11] было дано определение *предметно-ориентированных систем научной осведомленности*, как узкоспециализированных систем научной осведомленности, которые кроме возможности решения задач интеллектуального анализа данных наделены способностью производства новых профессиональных знаний. Общая программная архитектура такой системы показана на рис. 2.

Как показано на рисунке, основными компонентами предметно-ориентированной системы научной осведомленности являются, помимо инструментария интеллектуального анализа данных, хранилище знаний и подсистема производства профес-

сиональных знаний, включающая набор встроенных экспертных систем и искусственных нейронных сетей.

С учетом требования анализа и производства знаний в программно-технологическую архитектуру предметно-ориентированных систем научной осведомленности добавляются следующие программные компоненты:

- **компонент фактографических научных баз данных**, содержащих экспериментальные или модельные данные, в частности, фундаментальные константы, числовые и лингвистические характеристики химических или физических процессов;
- **компонент интеллектуального анализа данных**. Поскольку новые инструменты научных исследований обладают исключительной точностью, увеличиваются точность и качество фактографических данных. Для анализа таких данных в целях нахождения тонких эффектов, упущенных в предыдущих исследованиях, требуется набор алгоритмов, позволяющий проводить сложный анализ данных;
- **компонент производства новых знаний**. Найденные в результате интеллектуального анализа данных эмпирические закономерности позволяют строить прогнозы значений физических или химических процессов и оценивать значение фундаментальных характеристик материалов. Это создает предпосылки для встраивания в предметно-ориентированные системы научной осведомленности элементов прикладного искусственного интеллекта, например, экспертных систем для производства новых знаний и их сохранения в системе;
- **компонент дистанционного обучения**. Наличие такого компонента в системе делает ее более привлекательной для использования и распространения предметно-ориентированных знаний, а также служит привлечению заинтересованного круга профессиональных пользователей к производству новых знаний.



Рис. 2. Общая программная архитектура предметно-ориентированной системы научной осведомленности

Для разработки и создания предметно-ориентированных систем научной осведомленности целесообразно использовать технологии мультиагентных систем. Компонентом такой системы становится интеллектуальный агент, который можно представить в виде веб-приложения, наделенного искусственным интеллектом и расположенного за некоторым внешним порталом. При этом сам агент ориентирован на обработку научных данных в узкоспециализированном разделе предметной области. При наличии протокола взаимодействия между такими агентами система научной осведомленности в целом строится поэтапно.

Предметно-ориентированные системы научной осведомленности могут быть созданы в научных проектах меньшего масштаба. При хорошо организованной и спроектированной системе метаданных они легко могут интегрированы в более крупные системы научной осведомленности с учетом территориальной распределенности последних.

Хранилище знаний предметно-ориентированных систем научной осведомленности

Центральным компонентом предметно-ориентированных систем научной осведомленности является хранилище знаний.

Перед тем как дать определение хранилища знаний, напомним определение хранилища данных, данное в [12]. Под *хранилищем данных* понимается предметно-ориентированная, интегрированная, неизменяемая и поддерживающая хронологию электронная коллекция данных для поддержки принятия решений. Подобно хранилищу данных, хранилище знаний может рассматриваться как предметно-ориентированная, интегрированная, содержащая временные ряды и поддерживающая процесс принятия решений электронная коллекция. Однако, в отличие от хранилища данных, хранилище знаний является комбинацией неизменяемых и изменяемых объектов и компонентов, и должна хранить не только данные, но и знания предметной области. В процессе исследования данных методами интеллектуального анализа полученные знания могут изменяться. Так, могут измениться параметры центра кластера в зависимости от поступления новых данных или используемого алгоритма кластеризации. Или может быть уточнен вид зависимости в данных. Например, параболический тренд может оказаться справедливым только в определенном интервале, а вне этого интервала будет иметь место линейный тренд. Компонент производства знаний может генерировать данные, которые являются прогнозом значений



Рис. 3. Общая программно-аппаратная архитектура хранилища знаний

некоторого параметра, но экспериментального подтверждения не имеют, т. е. знания, хранимые в хранилище знаний, могут изменяться и дополняться в результате его эксплуатации.

Таким образом, *хранилище знаний* есть предметно-ориентированная, интегрированная, поддерживающая временные ряды данных электронная коллекция, которая содержит данные, знания, процедуры генерирования знаний и используется для анализа и исследования данных, производства новых знаний и поддержки принятия решений.

Определение, приведенное выше, является расширением определения, данного в [13]. Существенное отличие настоящего определения состоит в добавлении в хранилище знаний компонентов интеллектуального анализа данных и средств генерирования знаний.

Общая программно-аппаратная архитектура хранилища знаний приведена на рис. 3. Как видно на рисунке, хранилище знаний включает хранилище данных для исследования (*Exploration Data Warehouse*), которое содержит неизменяемые во времени данные (экстенциональный фрагмент хранилища данных), базу знаний выделенной предметной области, встроенные механизмы производства профессиональных знаний, хранилище производных данных (интенциональный фрагмент хранилища данных), интерфейс пользователя и интерфейс эксперта. Интерфейс пользователя предназначен для выборки и распространения знаний, а интерфейс эксперта — для производства новых знаний.

Предметно-ориентированная система научной осведомленности по физической химии радикальных реакций

Исследования [10, 14–15] позволили приступить к разработке предметно-ориентированной системы научной осведомленности по физической химии радикальных жидкофазных реакций

на основе порталных и мультиагентных технологий. Разрабатываемая система рассматривается как интеллектуальная система в сети Интернет, назначением которой являются сбор, хранение, верификация, извлечение, распространение и производство новых предметно-ориентированных знаний по физической химии радикальных реакций.

В основе модели данных системы лежит совокупность связанных киосков данных и база знаний о свойствах классов реакций и параметров реагентов (хранилище знаний). Система предназначена для решения следующих задач:

- поиск данных о реакционной способности реагентов в бимолекулярных радикальных реакциях в жидкой фазе;
- поиск значений энергий диссоциации связей органических молекул, а также энтальпий радикалов и молекул;
- поиск в базе данных библиографических ссылок;
- оценка реакционной способности реагентов в радикальных бимолекулярных реакциях в жидкой фазе по термодинамическим и кинетическим данным;
- оценка энергий диссоциации связей органических молекул по кинетическим данным.

На рис. 4 показано главное меню системы. Каждая URL-ссылка меню соответствует варианту использования системы. Так, ссылка "E & K" отсылает пользователя к подсистеме оценки реакционной способности элементарной реакции, которая включает в себя встроенную экспертную систему и набор искусственных нейронных сетей для производства новых значений констант скорости и энергий активации химической реакции. Ссылка "E-learning" отправляет пользователя в подсистему дистанционного обучения, которая включает в себя электронные курсы лекций, электронные задачки и тесты контроля знаний в удаленном режиме.

Активными элементами (агентами) производства новых знаний в системе являются встроенные экспертные системы (по прогнозированию констант скорости и энергии диссоциации связей), а также экспертная система контекстного управления навигацией пользователя.

Полученные в результате применения пользователем экспертных систем константы скоростей реакций или энергии диссоциации связей органических молекул могут быть сохранены в хранилище знаний. Такая возможность делает систему активной и позволяет заинтересованному научному сообществу накапливать в ней новые знания.

Возможность пополнения системы предметными знаниями накладывает на ее функционал

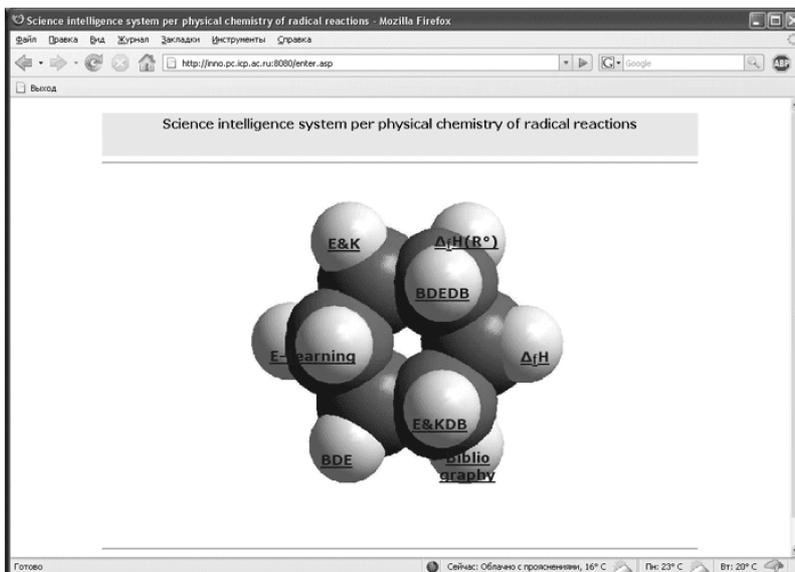


Рис. 4. Главное меню системы научной осведомленности для физической химии жидкофазных радикальных реакций

определенные ограничения. Такие ограничения связаны с необходимостью обеспечить достоверность заносимых в нее данных. Поэтому хранилище знаний системы состоит из двух разделов: базового раздела, составленного экспертами по данным научных публикаций, и раздела, произведенного пользователями системы.

При попытке занесения новых данных экспертная система контекстного управления на основе нечетких рассуждений делает ряд проверок и выводов о достоверности этих данных, а затем принимает решение либо о занесении данных с определенным показателем их надежности, либо об отказе в запоминании данных. При сохранении данных система просит пользователя заполнить анкету.

Включение в систему полнотекстовых электронных вариантов статей сталкивается с вопросами обеспечения авторских прав научных издательств и авторов, поэтому в системе не предусмотрен соответствующий функционал.

Одной из особенностей архитектуры системы, что делает ее привлекательной для использования в профессиональном образовании химиков и физиков, является наличие подсистемы дистанционного обучения. Основу подсистемы дистанционного обучения составляют веб-ориентированные курсы лекций, разработанные для обучения студентов и аспирантов. Курсы включают в себя тексты лекций, электронные задачки, разнесенные по лекциям, и тесты.

Подсистема объяснений экспертной системы контекстного управления обращается в случае необходимости к фрагментам лекций и управляющего тезауруса основных терминов при выполнении вычислений.

Заключение

Разработка и публикация в сети Интернет предметно-ориентированных систем научной осведомленности на основе использования хранилищ знаний позволят научному сообществу создавать распределенные сети для сбора, хранения, извлечения, интеллектуального анализа, распространения и производства знаний в узкоспециализированных областях исследований и технологий. Создание таких систем возможно небольшими коллективами ученых.

Включение в такие системы подсистемы дистанционного обучения предметно-ориентированным знаниям значительно расширяет круг ее потенциальных пользователей (студентов и аспирантов), что способствует самостоятельному формированию у них профессиональных знаний, а преподавательскому составу высших учебных заведений предоставляет дополнительный учебный материал и электронный ресурс-справочник.

Работа выполнена при поддержке гранта РФФИ 07-07-00343а.

Список литературы

1. **Когаловский М. Р.** Статистические базы данных // Программирование. 1995. № 2. С. 37.
2. **R & D DATABASE HANDBOOK.** A Worldwide Guide to Key Scientific and Technical Databases // Ed. L. Davidson. Technical Insights, Inc. 1984. 194 p.
3. **STN International Databases in Science & Technology.** FIZ Karlsruhe STN Service Center Europe. 1994. 44 p.
4. **Лопатенко А. С.** Современные научные информационные системы. Перспективы использования. 2001. http://derpi.tuwien.ac.at/~andrei/CRIS_DOC.htm

5. **Кулагин М. В., Лопатенко А. С.** Научные информационные системы и электронные библиотеки. Потребность в интеграции // Сб. тр. Третьей Всероссийской конференции по электронным библиотекам. RCDL'2001, Петрозаводск, 11–13 сентября 2001 г. С. 14–19.

6. **Grey J., Lui D. T., Nieto-Santisteban M., Szalay A., DeWitt D. J., Heber G.** Scientific Data Management in the Coming Decade // SIGMOD Record. 2005. V. 34. N 4.

7. **Thierauf R. J.** Effective Business Intelligence Systems. Westport. London: Quorum Books. 2001. 392 p.

8. **Hackathorn R.** Science Intelligence. Can a Business Intelligence Approach Enable "Smart" Science? DM Review. 2005. (<http://www.DMReview.com>).

9. **Dong Q., Yan X., Chirico R. D., Wilhoit R. C., Frenkel M.** Database Infrastructure to Support Knowledge Management in Physicochemical Data // 18-th CODATA Conference. 2002, Sep. 29–3 Oct., Montreal, Canada. 2002. 36 p.

10. **Tumanov V. E.** Data Warehousing and Data Mining in Thermochemistry of Free Radical Reactions // Fourth Winter Symposium on Chemometrics "Modern Methods of Data Analysis". Russia. Chernogolovka. February 15–18. 2005. P. 28–29.

11. **Туманов В. Е.** Использование систем научной осведомленности в профессиональном образовании // XVII Международная конф.-выставка "Информационные технологии в образовании". II Всероссийская конференция по мероприятию "Развитие технической основы современных информационных образовательных технологий" приоритетного национального проекта "Образование": Сб. тр. участников конференции. Часть IV. М.: "БИТ про". 2007. 9–11 ноября, Москва. С. 72–74.

12. **Inmon W. H.** Building the Data Warehouse. Third Edition. John Wiley & Sons, New York. 2002. 412 p.

13. **Firestone J. M.** Enterprise Information Portals and Knowledge Management. Butterworth-Heinemann. 2002. 422 p.

14. **Кромкин Е. А., Туманов В. Е.** Разработка базы знаний для оценки энергий диссоциации связей по кинетическим данным радикальных реакций отрыва // Башкирский химический журнал. 2001. Т. 8. № 2. С. 32–40.

15. **Туманов В. Е.** Интеллектуальный анализ кинетических и термодимических данных радикальных реакций // Техника машиностроения. 2006. № 3. С. 7–15.

УДК 004.51

А. В. Соколов, канд. физ.-мат. наук, доц.,
Государственный университет—Высшая школа экономики,
e-mail: avsookolov2003@list.ru

Методика оценки максимально возможных значений показателей эффективности поиска текстовой информации

На базе экспертной оценки релевантности как нечеткого отношения предложен метод построения кривой зависимости максимально возможных (для данной предметной области) значений точности от полноты поиска, а также подход к оценке эффективности поискового механизма информационно-поисковой системы на основе сравнения реальных значений этих показателей с принципиально достижимыми идеальными значениями.

Ключевые слова: текстовая информация, эффективность поиска, релевантность, полнота, точность, методика оценки, оптимизация.

Введение

При проектировании средств тематического поиска текстовой информации в документальных автоматизированных информационных системах (АИС) одним из важных является вопрос о воз-

можности достижения приемлемых характеристик эффективности поиска — степени соответствия множества найденных документов поисковому запросу. До настоящего времени для оценки качества поискового механизма АИС использу-

ются ставшие уже классическими показатели полноты и точности поиска [1, 2].

Известно [2], что при тематическом поиске текстовой информации невозможно создать механизм, способный выдавать всю информацию, соответствующую запросу, и при этом не выдавать ничего лишнего. В связи с этим возникает вопрос, до каких пределов можно совершенствовать поисковый механизм, чтобы получить максимально возможные значения показателей полноты и точности.

Показатели полноты и точности находятся в некотором смысле в обратной зависимости друг от друга: меры, предпринимаемые для повышения полноты поиска, обычно приводят к снижению точности, и наоборот. Таким образом, перебирая всевозможные варианты механизмов поиска и выбирая наилучшие из них при заданной полноте, можно построить кривую зависимости максимальной точности от полноты. Но для этого необходимо перебрать все варианты механизмов поиска, что практически невозможно. Перебрав же только часть вариантов, мы получим не самую лучшую зависимость, а лишь некоторое приближение к ней снизу.

Что для пользователя важнее: добиться максимальной полноты за счет уменьшения точности или максимальной точности за счет уменьшения полноты — вопрос не простой. На заре информатизации, когда документов в электронном виде было относительно мало, считалось, что нужно максимизировать полноту. Однако с развитием информационных технологий, появлением сети Интернет рост объемов массивов документов принял лавинообразный характер. В этих условиях пользователь часто не в состоянии просмотреть всю выдаваемую информацию, и речь уже может идти об увеличении точности в ущерб полноте, тем более, что часто информация многократно дублируется в разных документах.

К настоящему времени разработано несколько технологических схем для улучшения показателей поиска. Это итеративный поиск, реализуемый в виде интерактивной процедуры (уточнение запроса, поиск в уже найденном), а также ранжирование найденных документов по убыванию степени их релевантности. Тем не менее, не теряет актуальности вопрос о качестве поиска при однократном запрашивании, поскольку интерактивный поиск базируется на серии однократных поисковых процедур. А от их качества зависит, сколько релевантной информации было уже потеряно при каждом единичном запрашивании, а также сколько итеративных шагов придется совершить. Что касается упорядочения выдачи по степени релевантности, то в этом случае обычно выдается довольно большой объем информации,

который пользователь все равно не в состоянии охватить, и ему приходится ограничиваться просмотром лишь части выданных документов.

Формализация понятия релевантности в нечеткой постановке

Рассмотрим модель АИС под углом зрения однократного тематического поиска текстовых документов, при котором на запрос s выдается вполне определенный массив $D(s)$ документов без их упорядочения по степени релевантности. Массив документов M , накопленный к данному моменту времени в АИС, будем считать фиксированным.

Документ $d \in M$ называется релевантным запросу s , если он соответствует основному смысловому содержанию запроса (см., например, [1, 2]). Пару (d, s) будем называть релевантной парой, если документ d релевантен запросу s .

На массиве всех документов M и запросов S , т. е. на их декартовом произведении $M \times S$, введем два отношения: отношение релевантности r и отношение выдачи φ :

$$r(d, s) = \begin{cases} 1, & \text{если документ } d \text{ релевантен запросу } s; \\ 0, & \text{если документ } d \text{ не релевантен запросу } s, \end{cases} \quad (1)$$

$$\varphi(d, s) = \begin{cases} 1, & \text{если АИС выдает документ } d \text{ на запрос } s; \\ 0, & \text{если АИС не выдает документ } d \text{ на запрос } s. \end{cases} \quad (2)$$

В приведенном определении релевантности предполагается, что пользователь может однозначно определить, соответствует ли данный документ запросу или нет. Однако это не всегда так: разные пользователи могут по-разному оценивать соответствие документа запросу. Вообще, следует отметить, что в зависимости от характера предметной области разброс мнений относительно релевантности меняется. Так, для предметной области, содержащей четкую терминологию, например научно-технического плана, разброс мнений относительно релевантности меньше, чем для предметной области с размытыми понятиями типа политической или бытовой предметной области.

Поскольку мнения различных пользователей могут различаться, для оценки качества поискового механизма АИС следует ввести усредненную оценку релевантности. Для этого необходимо иметь представительное жюри экспертов, моделирующее параметры массового пользователя при оценке релевантности. Понятие релевантности при этом размывается. Оно уже не является четким, и его можно назвать *степенью релевантности*.

Итак, пусть мы имеем жюри, состоящее из m экспертов, причем каждый эксперт оценивает релевантность каждого документа $d \in M$ каждому запросу $s \in S$.

Пусть $\eta_j(d, s)$, $j = 1, \dots, m$, — результат оценки j -м экспертом релевантности документа d запросу s .

Эта оценка может проводиться как по двухбалльной шкале (да/нет или 0/1), как в формуле (1), так и по нечеткой шкале [3], содержащей значения из отрезка [0;1], например, по 11-балльной шкале 0; 0,1; 0,2; ...; 1 с шагом $h = 0,1$. Оценка по нечеткой шкале предпочтительнее, поскольку для достижения той же точности результатов требуется меньшее число экспертов.

В случае использования нечеткой шкалы $\eta_j(d, s) = 0$ означает, что эксперт абсолютно уверен, что пара (d, s) нерелевантна, $\eta_j(d, s) = 1$ означает, что эксперт абсолютно уверен в релевантности пары (d, s) , $\eta_j(d, s) = 0,5$ означает, что эксперт не может отдать предпочтение ни одному из решений о релевантности или нерелевантности пары (d, s) , и т. д.

Таким образом, степень релевантности пары (d, s) можно оценить по формуле

$$\eta(d, s) = \frac{1}{m} \sum_{j=1}^m \eta_j(d, s). \quad (3)$$

Очевидно, значение величины $\eta(d, s)$ лежит в пределах от нуля до единицы и, как правило, будет дробной величиной.

Формализация понятий полноты и точности поиска в нечеткой постановке

Как уже упоминалось, для оценки качества поискового механизма в информатике вводятся два основных показателя: коэффициенты полноты и точности поиска.

Коэффициентом полноты (поиска, выдачи) называется отношение числа выданных релевантных документов $n_{вр}$ к числу всех релевантных n_p :

$$k_{п} = \frac{n_{вр}}{n_p}.$$

Коэффициентом точности (поиска, выдачи) называется отношение числа выданных релевантных документов $n_{вр}$ к числу всех выданных n_b :

$$k_{т} = \frac{n_{вр}}{n_b}.$$

Из (1) и (2) следует, что при поиске по запросу s

$$k_{п}(s) = \frac{\sum_{d \in M} r(d, s)\varphi(d, s)}{\sum_{d \in M} r(d, s)};$$

$$k_{т}(s) = \frac{\sum_{d \in M} r(d, s)\varphi(d, s)}{\sum_{d \in M} \varphi(d, s)}.$$

Для экспертной оценки этих коэффициентов в нечетком случае функцию $r(d, s)$ следует заменить на η из (3).

Чтобы оценить качество поискового механизма АИС в целом, необходимо провести оценку

для всех запросов и каким-то образом усреднить результаты. Усреднение можно выполнять по-разному. Все варианты отличаются друг от друга не существенно. Здесь будет использоваться способ усреднения, который более удобен для дальнейших построений. А именно, коэффициент полноты будем рассчитывать как отношение усредненного (по всем запросам) числа выданных релевантных документов к усредненному числу релевантных документов, а коэффициент точности — как отношение усредненного числа выданных релевантных документов к усредненному числу выданных документов:

$$k_{п} = \frac{\frac{1}{|S|} \sum_{s \in S} \sum_{d \in M} \eta(d, s)\varphi(d, s)}{\frac{1}{|S|} \sum_{s \in S} \sum_{d \in M} \eta(d, s)} = \frac{\sum_{(d, s) \in M \times S} \eta(d, s)\varphi(d, s)}{\sum_{(d, s) \in M \times S} \eta(d, s)}; \quad (4)$$

$$k_{т} = \frac{\frac{1}{|S|} \sum_{s \in S} \sum_{d \in M} \eta(d, s)\varphi(d, s)}{\frac{1}{|S|} \sum_{s \in S} \sum_{d \in M} \varphi(d, s)} = \frac{\sum_{(d, s) \in M \times S} \eta(d, s)\varphi(d, s)}{\sum_{(d, s) \in M \times S} \varphi(d, s)}. \quad (5)$$

Коэффициент полноты равен 1 в том и только том случае, когда числитель и знаменатель совпадают. А это возможно лишь тогда, когда при $\eta(d, s) \neq 0$ обязательно $\varphi(d, s) = 1$, т. е. все сколько-нибудь релевантные документы выдаются системой. Заметим, что единичного значения полноты можно добиться всегда, выдавая весь массив документов на любой запрос.

Коэффициент точности будет единичным, если выдавать только документы со степенью релевантности 1. Однако может случиться, что таких документов ни для одного запроса нет. И тогда единичного значения коэффициента точности достичь невозможно. Но чем меньше пар с низкой степенью релевантности будет в выдаче, тем больше будут значения коэффициента точности.

Метод построения паретовой границы в критериальном пространстве показателей эффективности поиска

Зададимся целью решить двухкритериальную оптимизационную задачу

$$k_{п} \rightarrow \max, k_{т} \rightarrow \max$$

по $\varphi(d, s) \in 2^{|D \times S|}, d \in D, s \in S$.

Пусть мы имеем представительный массив документов M , запросов S и экспертную оценку релевантности на декартовом произведении $M \times S$.

Изменяя состав выдачи, т. е. функцию $\varphi(d, s)$, мы будем получать различные значения показателей полноты и точности поиска. Изменение функции $\varphi(d, s)$ означает изменение поискового механизма. Возможность произвольного задания функции выдачи означает, что мы можем перебрать все (реализуемые и практически не реализуемые) варианты механизмов поиска. И если при каждом фиксированном значении полноты найдем вариант с максимально возможным значением точности, то это будет означать, что лучшего результата мы в принципе получить не можем (при заданных массивах документов и запросов и экспертной оценке релевантности, проведенной на них).

Найдя для каждого значения полноты наибольшее значение точности, мы тем самым построим паретову границу для множества значений полноты-точности поиска. При достаточной плотности задаваемых точек k_{Π} полученные оптимальные точки (k_{Π}, k_T) можно соединить плавной линией — кривой оптимальных характеристик поиска, выше которой невозможно получить значения ни при каких механизмах поиска.

Построенная таким образом кривая только в исключительных случаях может вырождаться в точку (1; 1). Это произойдет, если все эксперты оценивают релевантность по четкой шкале (да/нет), причем оценки всех экспертов совпадают. Если же мнения экспертов расходятся, вырождения в точку не произойдет. Формальная причина этого заключается в том, что значения степени релевантности выражаются дробными числами, а функции выдачи — целыми.

Для построения паретовой границы можно, конечно, перебрать все варианты значений функции выдачи $\varphi(d, s)$. Однако их очень много ($2^{|M| \times |S|}$), причем подавляющее большинство из них будет доминировано по Парето.

В данной статье предлагается способ, позволяющий строить точки, непосредственно принадлежащие паретовой границе. Этот способ состоит в следующем.

Пусть k_{Π} фиксировано. Тогда из формул (4) и (5) получим

$$\sum_{(d, s) \in M \times S} \eta(d, s)\varphi(d, s) = k_{\Pi} \sum_{(d, s) \in M \times S} \eta(d, s) = C = \text{const}; \quad (6)$$

$$k_T = \frac{C}{\sum_{(d, s) \in M \times S} \varphi(d, s)}. \quad (7)$$

Для максимизации k_T нужно минимизировать знаменатель дроби (7) при условии (6):

$$\sum_{(d, s) \in M \times S} \varphi(d, s) \Rightarrow \min \text{ по } \varphi$$

при $\sum_{(d, s) \in M \times S} \eta(d, s)\varphi(d, s) = C.$

Поскольку $\sum_{(d, s) \in M \times S} \varphi(d, s)$ — это сумма единиц и нулей, минимизация ее будет достигнута при наименьшем возможном числе единиц, но при соблюдении условия (6). А это реализуется тогда, когда мы положим $\varphi(d, s) = 1$ для пар (d, s) с наибольшим значением $\eta(d, s)$.

Отсюда следует метод построения паретовой границы для множества пар (k_{Π}, k_T) .

Значения $\eta(d, s)$ упорядочиваются по убыванию. Затем строится точка (k_{Π}, k_T) при $k_{\Pi} = 1$. Для этого в выдачу определяются все пары, для которых $\eta(d, s) \neq 0$, т. е. для них полагается $\varphi(d, s) = 1$, и по формуле (5) подсчитывается k_T . Затем из выдачи удаляются наименее релевантные пары и для них подсчитываются коэффициенты полноты и точности и т. д. до тех пор, пока в выдаче останутся пары, для которых $\eta(d, s) = 1$ или, если таковых нет, — пары с наибольшим значением $\eta(d, s)$ из имеющихся. Полученные точки отмечаются на плоскости в системе координат (k_{Π}, k_T) и соединяются плавной кривой $k_T = f(k_{\Pi})$.

Автором был проведен небольшой макетный эксперимент на 25 текстах и 10 запросах к массиву новостной политической информации. В оценке участвовало четыре эксперта.

На основе полученных данных построена кривая оптимальных характеристик (рис. 1).

По построенной подобным образом кривой можно сориентироваться, выше каких значений полноты и точности для данной предметной области добиться невозможно. Так, если мы хотим

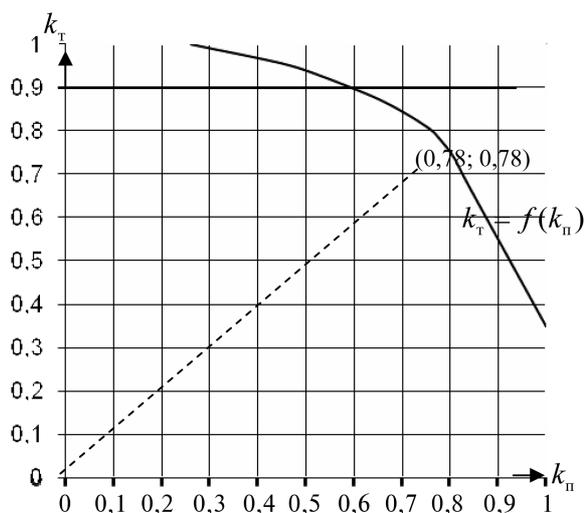


Рис. 1. Кривая оптимальных характеристик качества поиска

получить $k_{\Pi} = 1$, то мы должны смириться с тем, что точность будет не более 0,35. А это значит, что 65 % выдачи будет состоять из "шума" — нерелевантной информации. Если мы хотим получить $k_T = 1$, т. е. чтобы не было "шума", не следует ожидать полноты более 26 %. Это значит, что 74 % релевантной информации мы потеряем. Если требования пользователя к полноте и точности примерно одинаковы, то, проведя биссектрису координатного угла, получим $k_{\Pi} = k_T = 0,78$. Это значит, что выдача будет содержать 78 % релевантной информации и 22 % "шума". Потери тоже составят 22 %.

Оценка достоверности модели

Кривая оптимальных характеристик была построена на основе экспертной оценки, проведенной на определенных массивах документов и запросов определенной группой экспертов. Если взять другие массивы и других экспертов, могут получиться, вообще говоря, результаты, отличные от данных. "Достоверные" результаты (обобщенные для всей АИС в целом) получатся на так называемой генеральной совокупности данных — всем массиве документов АИС, всем массиве возможных запросов, экспертной оценке, проведенной всеми потенциальными пользователями. Понятно, что такой эксперимент провести практически невозможно, да и не нужно по следующим причинам.

Во-первых, слишком точная оценка не требуется. Различия в несколько процентов особого значения не имеют. Во-вторых, полученная оценка является оценкой сверху, и, скорее всего, достигнуть ее практически не удастся. В-третьих, полученные показатели полноты и точности являются усредненными. При каждом же конкретном акте поиска они могут существенно отличаться от средних. А каждого пользователя, вообще говоря, интересуют результаты поиска по его запросам, им же оцененные.

И все же, для получения средних значений, близких к "достоверным", следует использовать так называемые представительные массивы документов и запросов и представительное жюри экспертов. Под представительностью понимается следующее. Характеристики выбранных документов, запросов и экспертов должны быть средне-статистическими, т. е. должны в среднем совпадать со средними характеристиками генеральной совокупности. На практике это означает, что документы должны охватывать всю тематику массива, разнообразие объема и детализации изложения вопросов. Запросы должны быть типичными для пользователей АИС, охватывать весь спектр тематики и степени детализации формулировок. Эксперты должны выбираться из числа будущих

пользователей и охватывать разнообразие подходов к оценке релевантности (жесткая, средняя, мягкая оценки). Представительности выборки обычно добиваются путем случайного выбора из генеральной совокупности достаточного числа элементов (в нашем случае — документов, запросов, экспертов).

Проверка представительности выборки может осуществляться путем разбиения ее случайным образом на две или более частей и сравнения полученных результатов. Если результаты окажутся близкими друг к другу, можно считать выборку представительной. В противном случае следует увеличить объем выборки.

Как уже отмечалось, понятие релевантности в случае тематического поиска является понятием нечетким. Примем гипотезу (условную), что в основе оценки релевантности пары (d, s) лежит некий объективный показатель — степень релевантности $r(d, s) \in [0; 1]$, а субъективность оценки заключена в отклонении от этого объективного значения. Если предположить, что состав экспертов является представительным, можно ожидать, что усредненная оценка экспертов нивелирует различия в частных оценках и будет близка к "объективному" значению $r(d, s)$. Экспериментальное значение $\eta(d, s)$ можно трактовать как случайную величину с математическим ожиданием $E\eta(d, s) = r(d, s)$ и дисперсией $D\eta(d, s) = r(d, s)(1 - r(d, s))$.

Если эксперты оценивают релевантность по шкале, содержащей $n + 1$ значение: $0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}, 1$, можно условно считать, что оценку проводит не один эксперт, а n независимых экспертов, каждый из которых подает свой "внутренний" голос: 0 или 1. В таком случае разброс мнений экспертов уменьшится по сравнению с оценкой по двухбалльной шкале. Так, при m натуральных экспертах дисперсия будет равна

$$D\eta(d, s) = \frac{r(d, s)(1 - r(d, s))}{nm} \leq \frac{1}{4nm}.$$

Это значит, что для достижения такой же точности оценивания потребуется значительно меньше экспертов.

Оценим необходимый объем массива. Пусть требуется, чтобы с уровнем достоверности α точность построения кривой $k_T = f(k_{\Pi})$ была не хуже δ , т. е. чтобы выполнялось условие

$$P(|k_T^{\text{н}} - f(k_{\Pi})| \leq \delta) = \alpha, \quad (8)$$

где $k_T^{\text{н}}$ — "истинное" значение коэффициента точности при заданном коэффициенте полноты; $f(k_{\Pi}) = k_T$ — значение коэффициента точности, полученное экспертным путем.

Пусть оценку релевантности осуществляют m экспертов по равномерной $(n + 1)$ -балльной шкале.

Положим $\lambda = \frac{k_T}{k_{\Pi}}$, $R = \sum_{(d,s) \in M \times S} \eta(d, s)$ — обобщенная мощность множества релевантных пар.

Тогда при некоторых допущениях, несущественно влияющих на конечный результат, можно показать, что для выполнения условия (8) достаточно, чтобы

$$R = \frac{\lambda}{nm} \left(\frac{1}{2\delta} \Phi^{-1}(\alpha/2) \right)^2,$$

где Φ^{-1} — функция, обратная к $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-z^2/2} dz$.

Для примера вычислим объем множества релевантных пар, достаточный для получения точности $\delta = 0,02$ с уровнем достоверности $\alpha = 0,95$ при наличии $m = 4$ экспертов, оценивающих степень релевантности по равномерной 11-балльной шкале ($n = 10$).

Учитывая, что $\Phi^{-1}(0,475) = 1,96$, получим $R \approx 60\lambda$.

Так, для оценки точности при $\lambda = 1$, т. е. при равенстве коэффициентов полноты и точности, достаточно иметь суммарный объем релевантных пар в количестве 60 единиц. При других соотношениях коэффициентов полноты и точности требуемый объем изменяется. Однако данной оценкой не следует пользоваться за пределами построенной экспериментальной кривой, поскольку в этом случае она оказывается искаженной. Так, в рассмотренном выше примере (см. рис. 1) целесообразно пользоваться приведенной оценкой для значений параметра λ в пределах примерно от 0,35 до 3,7, что соответствует правому и левому концам построенной кривой.

Методика оценки качества поискового механизма АИС

Как уже отмечалось, для различных предметных областей кривые зависимостей оптимальных показателей поиска текстовой информации могут различаться. Для предметной области с более четким выделением описываемых объектов кривая будет смещена вверх и вправо — ближе к точке (1; 1). Для предметных областей с нечеткими очертаниями объектов интереса кривая, наоборот, будет опускаться вниз и влево. Таким образом, можно определять степень нечеткости предметной области по положению кривой $k_T = f(k_{\Pi})$. В целях сравнения различных предметных областей удобно использовать скалярный (числовой) показатель. В качестве такого показателя в данном случае можно предложить площадь S , заклю-

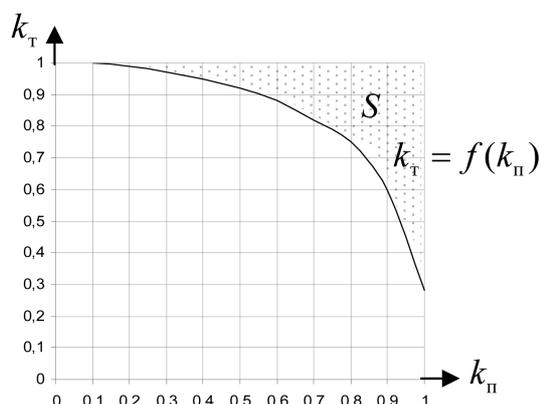


Рис. 2. Показатель S степени нечеткости предметной области

ченную между кривой $k_T = f(k_{\Pi})$ и граничными линиями $k_T = 1$ и $k_{\Pi} = 1$ (рис. 2):

$$S = \int_0^1 (1 - f(k_{\Pi})) dk_{\Pi}.$$

Чем меньше площадь S , тем четче предметная область (мера нечеткости ниже). Чем больше площадь S , тем предметная область менее четкая (мера нечеткости выше).

С помощью кривой $k_T = f(k_{\Pi})$, построенной для соответствующей предметной области, можно также оценивать качество поискового механизма АИС. Если АИС не содержит настроек, которые могут изменять соотношение полноты и точности поиска, можно предложить следующую процедуру оценки.

На тех же массивах документов и запросов и результатах экспертной оценки, на основе которых строилась кривая $k_T = f(k_{\Pi})$, по формулам (4) и (5) подсчитываются значения полноты и точности поиска анализируемой АИС. Получится некоторая точка (k_{Π}^0, k_T^0) , которая будет располагаться не выше кривой $k_T = f(k_{\Pi})$. Качество поискового

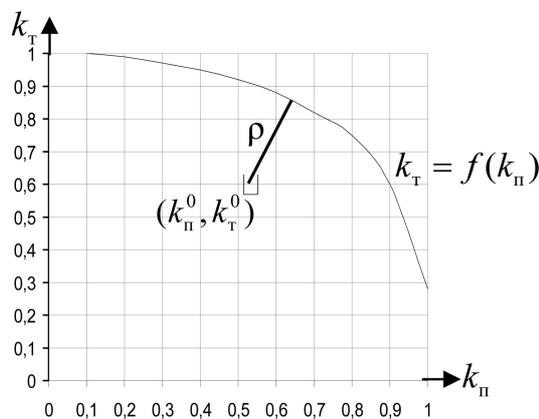


Рис. 3. Показатель ρ качества поискового механизма АИС в случае отсутствия его настройки

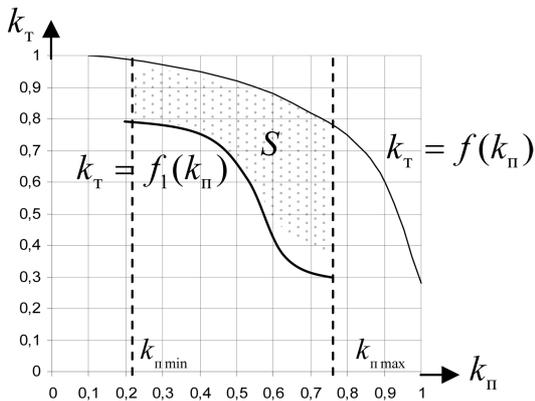


Рис. 4. Показатель S качества поискового механизма АИС в случае наличия его настройки

механизма может быть оценено как расстояние (например, евклидово) от точки (k_{π}^0, k_T^0) до ближайшей точки кривой $k_T = f(k_{\pi})$ (рис. 3):

$$\rho = \min_{\substack{(k_{\pi}, k_T) \\ k_T = f(k_{\pi})}} \sqrt{(k_{\pi} - k_{\pi}^0)^2 + (k_T - k_T^0)^2}.$$

Если АИС содержит инструментарий настройки, который позволяет изменять соотношение

полноты и точности поиска, следует найти значения (k_{π}^0, k_T^0) при различных параметрах этой настройки и построить для исследуемой АИС свою кривую $k_T = f_1(k_{\pi})$. Она будет располагаться ниже кривой $k_T = f(k_{\pi})$. Качество поискового механизма АИС может быть оценено как нормированная площадь S , заключенная между кривыми $k_T = f_1(k_{\pi})$ и $k_T = f(k_{\pi})$:

$$S = \frac{1}{k_{\pi \max} - k_{\pi \min}} \int_{k_{\pi \min}}^{k_{\pi \max}} (f(k_{\pi}) - f_1(k_{\pi})) dk_{\pi},$$

где $k_{\pi \min}$ и $k_{\pi \max}$ — соответственно наименьшее и наибольшее значения коэффициентов полноты, которые могут быть достигнуты в АИС (рис. 4).

Список литературы

1. Михайлов А. И., Черный А. И., Гиляревский Р. С. Основы информатики. М.: Наука, 1968.
2. Ланкастер Ф. У. Информационно-поисковые системы. М.: Мир, 1972.
3. Zadeh L. A. Fuzzy Sets // Information and control. 1965. N 8. P. 338–353.

ОПТИМИЗАЦИЯ

УДК 519.6

А. П. Карпенко, д-р физ.-мат. наук, проф.,
e-mail: karpenko@nog.ru,

В. Г. Федорук, канд. техн. наук, доц.,
МГТУ им. Н. Э. Баумана,
e-mail: fedoruk@comcor.ru

Один класс прямых адаптивных методов многокритериальной оптимизации

Рассматривается предложенный авторами класс прямых адаптивных методов непрерывной многокритериальной оптимизации, основанный на аппроксимации функции полезности.

Ключевые слова: задача многокритериальной оптимизации, функция предпочтений, прямой адаптивный метод.

Введение

Рассматривается один из важнейших частных случаев общей задачи принятия решений — задача непрерывной многокритериальной оптимизации

(МКО-задача), которую условно можно записать в виде

$$\min_{X \in D_X} \Phi(X) = \Phi(X^*). \quad (1)$$

Здесь D_X — множество допустимых значений вектора параметров X ; $\Phi(X) = (\phi_1(X), \phi_2(X), \dots, \phi_m(X))$ — векторный критерий оптимальности.

Существует несколько способов классификации методов решения МКО-задачи, например, классификация, основанная на содержании и форме использования дополнительной информации о предпочтениях лица, принимающего решения (ЛПР) [1]. В соответствии с этой классификацией выделяются следующие классы методов решения МКО-задачи:

- методы зондирования;
- априорные методы;
- апостериорные методы;
- адаптивные методы (человеко-машинные методы).

Каждая итерация адаптивных методов включает в себя фазу анализа, выполняемого ЛПР, и фазу расчетов, выполняемых программной системой многокритериальной оптимизации (МКО-системой). По характеру информации, получаемой МКО-систе-

мой от ЛПР на фазе анализа, можно выделить три класса адаптивных методов [2]:

- методы, в которых ЛПР непосредственно назначает веса частных критериев оптимальности;
- методы, в которых ЛПР накладывает ограничения на значения частных критериев оптимальности;
- методы, в которых ЛПР выполняет только оценку (например, в терминах "лучше", "хуже", "одинаково") предлагаемых МКО-системой альтернатив.

Последние методы называются прямыми адаптивными методами и основаны на предположении существования "функции полезности" $\psi(X)$, которая определена на множестве D_X и выполняет его отображение в пространство действительных чисел R , т. е. $\psi: D_X \rightarrow R$. При этом МКО-задача сводится к задаче выбора вектора $X^* \in D_X$ такого, что

$$\max_{X \in D_X} \psi(X) = \psi(X^*).$$

Обычно прямые адаптивные методы строятся на основе косвенной оценки функции $\psi(X)$, получаемой путем сравнения альтернатив [3, 4]. В данной работе рассматривается класс прямых адаптивных методов решения МКО-задачи, основанных на предположении, что при предъявлении ЛПР вектора параметров X , а также соответствующих значений всех критериев оптимальности это лицо может оценить соответствующее значение своей функции предпочтений $\psi(X)$ в терминах "очень плохо", "плохо", ..., "отлично" [5].

1. Постановка задачи

Пусть $X \in R^n$ — вектор варьируемых параметров, где R^n — n -мерное арифметическое пространство. Множеством допустимых значений вектора X является замкнутое множество $D_X = \Pi \cap D$, где "технологический" параллелепипед допустимых значений

$$\Pi = \{X | x_i^- \leq x_i \leq x_i^+, i \in [1:n]\},$$

а множество D формируют ограничивающие функции $g_j(X)$:

$$D = \{X | g_j(X) \geq 0, j \in [1:k]\}.$$

Векторный критерий $\Phi(X)$, $X \in \Pi$, имеет значения в пространстве R^m . ЛПР стремится минимизировать на множестве D_X каждый из частных критериев оптимальности $\phi_i(X)$, $i \in [1:m]$, что условно записывается в виде (1).

Критериальное множество задачи (множество достижимости) обозначим D_Φ ; множество Парето — $D_\Phi^* \subset D_\Phi$; множество эффективных по Парето векторов X — $D_X^* \subset D_X$. Положим, что частные критерии $\phi_i(X)$ тем или иным способом нормали-

зованы и сохраним за нормализованными критериями прежние обозначения.

Обозначим операцию скалярной свертки частных критериев $\varphi(X, \Lambda)$, где $\Lambda \in D_\Lambda$ — вектор весовых множителей; $D_\Lambda = \{\lambda_i | \lambda_i \geq 0, \sum \lambda_i = 1, i \in [1:m]\}$ — множество допустимых значений этого вектора. Способ свертки не фиксируется — это может быть аддитивная свертка, мультипликативная свертка, свертка Джоффриона и другие свертки [2]. При каждом фиксированном векторе Λ решение задачи (1) сводится к решению однокритериальной задачи глобальной условной оптимизации (ОКО-задачи)

$$\min_{X \in D_X} \varphi(X, \Lambda) = \varphi(X^*, \Lambda). \quad (2)$$

В силу ограниченности и замкнутости множества D_X решение этой задачи существует.

Хорошо известно, что в случае аддитивной свертки $\varphi(X, \Lambda) = \sum \lambda_i \phi_i(X)$, $i \in [1:m]$, решение X^* задачи (2) принадлежит множеству D_X^* . Для произвольной скалярной свертки точка X^* , полученная из условия (2), вообще говоря, не принадлежит этому множеству. В этой связи обратим внимание на то, что ЛПР не всегда выбирает решение, принадлежащее множеству D_X^* .

Если при каждом $\Lambda \in D_\Lambda$ решение задачи (2) единственно (а при использовании численных методов это всегда имеет место), то условие (2) ставит в соответствие каждому из векторов $\Lambda \in D_\Lambda$ единственный вектор X^* и соответствующие значения частных критериев $\phi_i(X^*)$, $i \in [1:m]$. Это обстоятельство позволяет полагать, что функция полезности определена не на множестве D_X , а на множестве D_Λ , т. е. $\psi: D_\Lambda \rightarrow R$. В результате МКО-задача сводится к задаче выбора вектора $\Lambda^* \in D_\Lambda$ такого, что

$$\max_{\Lambda \in D_\Lambda} \psi(\Lambda) = \psi(\Lambda^*). \quad (3)$$

Поскольку обычно $m \ll n$, переход от задачи (2) к задаче (3) важен с точки зрения уменьшения вычислительных затрат.

Если используется аддитивная свертка и множество D_Φ является выпуклым, то выражение (2) задает взаимно однозначное отображение множества D_Λ на множество D_Φ^* . В этих условиях для любого $\Lambda \in D_\Lambda$ вектор X^* , являющийся решением задачи (2), принадлежит множеству D_X^* . Если вместо аддитивной свертки используется свертка Джоффриона, то для получения того же результата не требуется выпуклость множества D_Φ [2].

Величина ψ считается лингвистической переменной со значениями от "очень плохо" до "отлично", которые представляют собой символы нормальных нечетких подмножеств универсального множества R . Полагается, что ядра этих множеств $\text{core}(\psi) = \psi$ имеют значения 1, 2, ..., 9 [6].

Наряду с задачей (3) в работе рассматривается задача

$$\max_{\Lambda \in D_{\Lambda}} \dot{\psi}(\Lambda) = \dot{\psi}(\Lambda^*) = \dot{\psi}^*. \quad (4)$$

Общая схема предлагаемых методов является итерационной и состоит из следующих основных этапов [5]:

- 1) задание ЛПР начальной точки $\Lambda_r^0 \in D_{\Lambda}$;
- 2) выполнение на основе планов первого порядка серии ρ экстремальных экспериментов по максимизации функции ψ на множестве D_{Λ} ;
- 3) построение по результатам указанных экспериментов первой аппроксимирующей функции $\bar{\psi}_{r, \rho}(\Lambda)$;
- 4) формирование области планирования для аппроксимации функции $\bar{\psi}_{r, \rho}(\Lambda)$;
- 5) построение в указанной области второй аппроксимирующей функции $\tilde{\psi}_{r, \rho}(\Lambda)$;
- 6) определение начальной точки Λ_{r+1}^0 для следующей итерации и переход к п. 2.

На этапе 2 используются симплекс-планы на основе правильных или неправильных симплексов, а также регрессионные планы первого порядка; на этапе 3 — аппроксимация кусочно-линейной функцией, персептронной нейронной сетью и нейронной сетью с радиальными базисными функциями, а также аппроксимация на основе нечетких множеств по методу Мамдани; на этапах 5 и 6 — линейная и квадратичная аппроксимации.

2. Экстремальный эксперимент на основе планов первого порядка (этап 2)

2.1. Симплекс-планы. Для правильных симплекс-планов МКО-система выполняет экстремальный эксперимент по следующей схеме [7]:

- 1) исходя из текущей начальной точки $\Lambda_r^0 = (\lambda_{r,1}^0, \lambda_{r,2}^0, \dots, \lambda_{r,m}^0)$ строит начальный правильный симплекс $S_r = \{\Lambda_{r,i}, i \in [1:m+1]\}$;
- 2) для каждой из вершин $\Lambda_{r,i}$ симплекса S_r выполняет следующие действия: решает задачу (2) с весами $\lambda_{r,i,j}, j \in [1:m]$; полученные значения компонентов векторов $X^*, \Phi(X^*)$ предъявляет ЛПР; запрашивает у ЛПР оценку предъявленного решения — требует ввести соответствующее значение лингвистической переменной $\psi_{r,i}$;
- 3) находит величину $\dot{\psi}_{r,L} = \min \dot{\psi}_{r,i}, i \in [1:m+1]$;
- 4) отражает вершину $\Lambda_{r,L}$ симплекса S_r относительно центра тяжести его оставшихся вершин Λ_r^C — находит вершину $\Lambda_{r+1,L}$ и новый симплекс S_{r+1} ;
- 5) для вершины $\Lambda_{r+1,L}$ выполняет действия, аналогичные действиям, указанным в п. 2, полагает $r = r + 1$ и переходит к п. 3.

Для неправильных симплекс-планов эксперимент выполняется по упрощенной схеме метода Нелдера—Мида [7]. Упрощение заключается в исключении из числа операций над симплексом операции редукции (поскольку эта операция требует от ЛПР оценки значений функции предпочтений в m точках — в отличие от одной точки для операций отражения, сжатия и растяжения симплекса).

2.2. Регрессионные планы. Экстремальный эксперимент строится на основе слабо не насыщенных регрессионных планов первого порядка [8], которые объединены в библиотеку планов $\{\Lambda_{i,j}, i = 1, 2, \dots, j \in [1:M_i]\}$, где M_i — число точек в i -м плане. В библиотеке каждой размерности m вектор-функции $\Phi(X)$ соответствует один или несколько планов. С точки зрения помехозащищенности наиболее эффективными являются планы полного факторного эксперимента (ПФЭ 2^m), требующие, однако, большого числа испытаний. Поэтому при разработке указанной библиотеки принято компромиссное решение — там, где это возможно, используются планы дробного факторного эксперимента 2^{m-p} (ДФЭ 2^{m-p}), $p = 1, 2, \dots$.

Рассматриваются градиентные методы с дроблением шага и крутого восхождения. При использовании градиентного метода с дроблением шага МКО-система выполняет эксперимент по следующей схеме:

- 1) исходя из текущей начальной точки Λ_r^0 строит область планирования в виде параллелепипеда $P_r = (\lambda_{r,i}^0 - \Delta\lambda_{r,i}, \lambda_{r,i}^0 \leq \lambda_{r,i} \leq \lambda_{r,i}^0 + \Delta\lambda_{r,i}, i \in [1:m])$. Здесь $\Delta\lambda_{r,i}$ — шаг варьирования величины $\lambda_{r,i}$ на r -й итерации;
- 2) выбирает из библиотеки планов соответствующий план первого порядка и для каждой точки плана решает задачу (2). Полученные значения компонентов векторов $X^*, \Phi(X^*)$ предъявляет ЛПР и получает значение лингвистической переменной $\psi_{r,i}$;
- 3) решает систему линейных алгебраических уравнений (СЛАУ) $F_r^T F_r B_r = F^T \dot{\Psi}_r$, где симметричная $(m+1) \times (m+1)$ матрица $F_r^T F_r$ есть информационная матрица (Фишера); $\dot{\Psi}_r = (\dot{\psi}_{r,1}, \dot{\psi}_{r,2}, \dots, \dot{\psi}_{r,M_i})^T$; $B_r = (b_{r,0}, b_{r,1}, \dots, b_{r,m})^T$ — вектор оценок коэффициентов регрессии [8];
- 4) вычисляет градиент построенной функции регрессии и делает шаг в его направлении — вычисляет координаты нового центра серии опытов Λ_{r+1}^0 , полагает $r = r + 1$ и переходит к п. 1.

При использовании градиентного метода крутого восхождения схема эксперимента близка к рассмотренной схеме.

3. Построение первой аппроксимирующей функции (этап 3)

3.1. *Кусочно-линейная аппроксимация на основе симплекс-планов.* Пусть последовательно построены правильные симплексы $S_r, S_{r+1}, \dots, S_{r+\rho}$. Тогда МКО-система строит функцию $\bar{\psi}_{r,\rho}(\Lambda)$ по приведенной ниже схеме (рис. 1):

1) путем решения $(\rho + 1)$ систем $(m + 1)$ линейных алгебраических уравнений определяет линейные функции $\psi_i(\Lambda)$, $i \in [r:r + \rho]$, имеющие в вершинах соответствующих симплексов значения $\psi_{i,j}, j \in [1:m + 1]$, а также плоскости $\Gamma_k(\Lambda) = 0$, $k \in [r:r + \rho - 1]$, проходящие через общие точки смежных симплексов;

2) строит кусочно-линейную аппроксимирующую функцию

$$\bar{\psi}_{r,\rho}(\Lambda) = \begin{cases} \psi_{r+\rho}(\Lambda), & \Lambda \in D_{r+\rho}; \\ \psi_{r+\rho-1}(\Lambda), & \Lambda \in D_{r+\rho-1}; \\ \dots & \dots \\ \psi_r(\Lambda), & \Lambda \in D_r \end{cases} \quad (5)$$

где множества D_i определяются выражениями

$$D_{r+\rho} = \{\Lambda | \text{sign}(\Gamma_{r+\rho-1}(\Lambda)) = \text{sign}(\Gamma_{r+\rho-1}(\Lambda_{r+\rho}^C))\};$$

$$D_{r+\rho-1} = \{\Lambda | \Lambda \notin D_{r+\rho}, \text{sign}(\Gamma_{r+\rho-2}(\Lambda)) = \text{sign}(\Gamma_{r+\rho-2}(\Lambda_{r+\rho-2}^C))\};$$

...

$$D_r = \{\Lambda | \Lambda \notin D_{r+\rho} \cup D_{r+\rho-1} \cup \dots \cup D_{r+1}, \text{sign}(\Gamma_r(\Lambda)) = \text{sign}(\Gamma_r(\Lambda_r^C))\}.$$

Здесь Λ_r^C — $(m + 1)$ -вектор координат центра тяжести симплекса S_r .

Построение аппроксимирующей функции $\bar{\psi}_{r,\rho}(\Lambda)$ на основе неправильных симплекс-планов выполняется по аналогичной схеме.

3.2. *Кусочно-линейная аппроксимация на основе регрессионных планов.* Пусть построены области планирования в виде параллелепипедов P_i , $i \in [r:r + \rho]$. Тогда МКО-система строит функцию $\bar{\psi}_{r,\rho}(\Lambda)$ по следующей схеме (рис. 2):

1) строит плоскости $\Gamma_i(\Lambda) = 0$, $i \in [r:r + \rho - 1]$, такие, что в плоскости $\Gamma_i(\Lambda) = 0$ лежит та из граней параллелепипеда P_{i-1} , которую пересекает отрезок прямой $[\Lambda_i^0, \Lambda_{i+1}^0]$;

2) в качестве искомой кусочно-линейной функции $\bar{\psi}_{r,\rho}(\Lambda)$ принимает функцию (5), где

$$D_{r+\rho} = \{\Lambda | \text{sign}(\Gamma_{r+\rho-1}(\Lambda)) = \text{sign}(\Gamma_{r+\rho-1}(\Lambda_{r+\rho}^0))\};$$

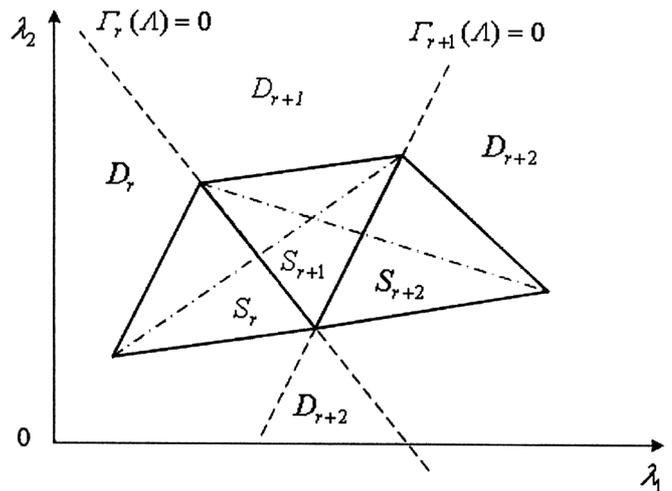


Рис. 1. К кусочно-линейной аппроксимации функции $\psi(\Lambda)$ на правильных симплекс-планах: $m = 2$; $\rho = 2$

$$D_{r+\rho-1} = \{\Lambda | \Lambda \notin D_{r+\rho}, \text{sign}(\Gamma_{r+\rho-2}(\Lambda)) = \text{sign}(\Gamma_{r+\rho-2}(\Lambda_{r+\rho-1}^0))\};$$

...

$$D_r = \{\Lambda | \Lambda \notin D_{r+\rho} \cup D_{r+\rho-1} \cup \dots \cup D_{r+1}, \text{sign}(\Gamma_r(\Lambda)) = \text{sign}(\Gamma_r(\Lambda_r^0))\}.$$

3.3. *Аппроксимация нейронными сетями.* Рассмотрим прежде двухслойную перцептронную нейронную сеть (MLP-сеть) с сигмоидальными передаточными функциями (рис. 3). При заданных весах $w_{i,j}, w_i$, $i \in [1:H], j \in [1:m]$ выход этой сети есть значение искомой аппроксимирующей функции $\bar{\psi} = F_3\left(\sum_{i=1}^H w_i u_i\right)$, где выходы нейронов скрытого слоя $u_i = F_2, i\left(\sum_{j=1}^m w_{i,j} \lambda_j\right)$; F_2, i — передаточная функция нейронов $N_{2,i}$; F_3 — нейрона N_3 [9].

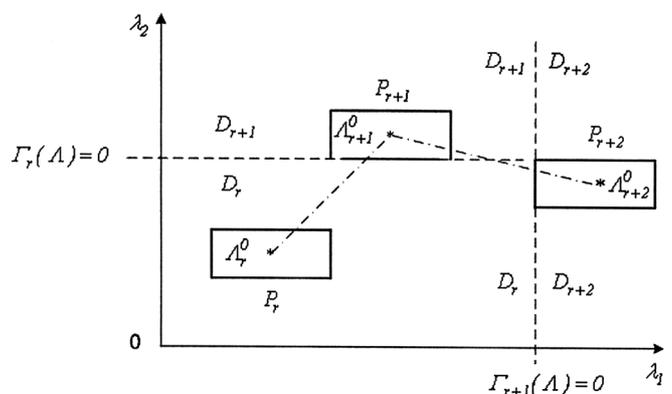


Рис. 2. К кусочно-линейной аппроксимации функции $\psi(\Lambda)$ на регрессионных планах первого порядка: $m = 2$; $\rho = 2$

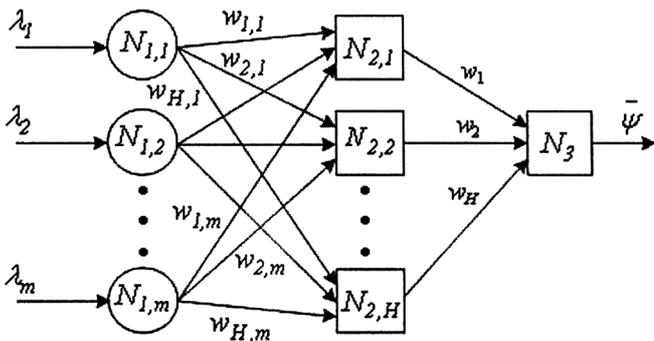


Рис. 3. Топология нейронной сети

В качестве обучающей выборки $((\Lambda_i, \dot{\psi}_i), i \in [1 : N])$ используется совокупность $N \geq 2(m + 1)$ различных точек множества D_Λ , в которых ЛПР определены значения функции полезности. Обучение сети осуществляется с помощью алгоритма обратного распространения.

Рассмотрим теперь нейронную сеть с радиальными базисными функциями (RBF-сеть) с гауссовыми передаточными функциями в скрытом слое и линейной передаточной функцией F_3 в выходном слое (см. рис. 3). Веса $w_{i,j}, i \in [1 : H], j \in [1 : m]$ положим равными единице, так что $u_i(\Lambda) = F_{2,i}(\Lambda)$. При некоторых весах w_i выход этой сети представляет собой значение искомой аппроксимирующей функции $\bar{\psi}_{r,\rho}(\Lambda) = \sum_{i=1}^H w_i u_i(\Lambda)$ [9].

При аппроксимации RBF-сетью МКО-система строит функцию $\bar{\psi}_{r,\rho}(\Lambda)$ по следующей схеме:

1) из выборки $((\Lambda_i, \dot{\psi}_i), i \in [1 : N])$ случайным образом формирует обучающие выборки $((\Lambda_i, \dot{\psi}_i)_1, i \in [1 : N_1]), ((\Lambda_j, \dot{\psi}_j)_2, j \in [1 : N_2])$, где $N_1 = \lceil N/2 \rceil, N_2 = N - N_1; \lceil \cdot \rceil$ — символ ближайшего большего целого;

2) строит двухслойную RBF-сеть, в которой каждому вектору $\Lambda_i, i \in [1 : H], H = N_1$ из первой обучающей выборки ставится в соответствие нейрон скрытого слоя с центром в точке Λ_i , так что $F_{2,i}(\Lambda) = \exp\left(-\frac{\|\Lambda - \Lambda_i\|^2}{2\sigma_i^2}\right)$. Здесь $\|\cdot\|$ — символ евклидовой нормы, σ_i — параметр, определяющий ширину колокола гауссовой передаточной функции $F_{2,i}(\Lambda)$;

3) повторяет некоторое фиксированное число раз следующие действия: присваивает величинам σ_i случайные начальные значения; путем решения СЛАУ, полученной на основе первой обучающей выборки, определяет веса w_i выходного нейрона;

с использованием второй обучающей выборки решает многомерную, вообще говоря, многоэкстремальную, задачу условной оптимизации $E(\Sigma^*) = \min_{\sigma_i > 0} E(\Sigma)$, где критерий оптимальности

$$E(\Sigma) = 0,5 \sqrt{\sum v_i \varepsilon_i^2}, i \in [1, N_2], \text{ а } \Sigma = (\sigma_1, \sigma_2, \dots, \sigma_H)^T,$$

$\varepsilon_i = \bar{\psi}_{r,\rho}(\Lambda_i) - \dot{\psi}(\Lambda_i); v_i$ — весовые множители;

4) из результатов, полученных в предыдущем пункте, выбирает решение, обеспечивающее минимальное значение ошибки E , а также соответствующие значения весов w_1, w_2, \dots, w_H .

3.4. Мамдани-аппроксимация на основе нечетких множеств. Пусть по схеме п. 3.3 построены обучающие выборки $((\Lambda_i, \dot{\psi}_i)_1, i \in [1 : N_1]), ((\Lambda_j, \dot{\psi}_j)_2, j \in [1 : N_2])$. Положим, что в рамках первой обучающей выборки переменная ψ приняла n_1 раз значение ψ_1, n_2 раз — значение ψ_2 и т. д. до n_9 и ψ_9 . Соответствующие входные векторы Λ обозначим $\Lambda_{i,j} = (\lambda_{i,j,1}, \lambda_{i,j,2}, \dots, \lambda_{i,j,m}), i \in [1, 9], j \in [1, n_i]$. Термы $\lambda_{i,j,k}, k \in [1 : m]$ и термы ψ_i примем нечеткими множествами, заданными на универсальном множестве $[0, 1]$:

$$\lambda_{i,j,k} = \int_0^1 \mu_{i,j,k}^\lambda(\lambda_k) / \lambda_k, \psi_i = \int_0^1 \mu_i^\psi(\psi) / \psi.$$

Здесь $\mu_{i,j,k}^\lambda, \mu_i^\psi$ — функции принадлежности величины λ_k терму $\lambda_{i,j,k}$ и значений лингвистической переменной ψ терму ψ_i соответственно.

Во введенных обозначениях многомерная функция принадлежности $\mu_i^\Lambda(\Lambda)$ имеет вид

$$\mu_i^\Lambda(\Lambda) = \bigvee_{j=1}^{n_i} w_{i,j} \left(\bigwedge_{k=1}^m (\mu_{i,j,k}^\lambda) \right), i \in [1, 9].$$

где $w_{i,j} = w_{i,j}(\delta)$ — весовые множители; $\delta \in [\delta^-, \delta^+] = [0^+, 1]$ — свободный параметр метода.

Обозначим a, b полуширины функций принадлежности $\mu_{i,j,k}^\lambda, \mu_i^\psi$ соответственно. Область допустимых значений параметров a, b, δ определим в виде параллелепипеда $D_p = \{(a, b, \delta) | \alpha^- < \alpha \leq \alpha^+, b^- < b \leq b^+, \delta^- < \delta \leq \delta^+\}$, где $\alpha^-, \alpha^+, b^-, b^+$ — заданные константы.

МКО-система строит аппроксимирующую функцию по следующей схеме:

1) формирует базу знаний в виде совокупности логических высказываний

$$\bigcup_{j=1}^{n_i} \left[\bigcap_{k=1}^m (\lambda_j = \lambda_{i,j,k}) \right] \rightarrow \psi_i, i \in [1, 9];$$

2) осуществляет параметрическую оптимизацию построенной нечеткой модели — повторяет

некоторое фиксированное число раз следующие действия: присваивает величинам a, b, δ допустимые случайные начальные значения; исходя из этих значений решает на второй обучающей выборке задачу многомерной, вообще говоря, многоэкстремальной условной оптимизации

$$\min_{\alpha, b, \delta \in D_p} E(\alpha, b, \delta) = E(\alpha^*, b^*, \delta^*),$$

где критерий оптимальности $E(\alpha, b, \delta) = \sqrt{\sum (\bar{\psi}_{r, \rho}(\Lambda_i) - \hat{\psi}(\Lambda_i))^2}$, $i \in [1 : N_2]$;

3) на основе результатов, полученных в предыдущем пункте, находит значения параметров a, b, δ , обеспечивающих минимальное значение ошибки E .

Для вычисления значения функции $\bar{\psi}_{r, \rho}(\Lambda)$ МКО-система выполняет следующие действия:

- определяет значение многомерной функции принадлежности $\mu_i^\Lambda(\Lambda)$;
- "срезает" функцию принадлежности $\mu_i^\Psi(\psi)$ на уровне $\mu_i^\Psi(\psi)$, $i \in [1, 9]$;
- агрегирует полученные нечеткие множества — строит нечеткое множество

$$\Psi = \max_{i \in [1, 9]} \left(\int_1^9 \min(\mu_i^\Lambda(\Lambda), \mu_i^\Psi(\psi)/\psi) \right)$$

с функцией принадлежности $\mu^\Psi(\Psi)$;

- определяет четкое значение функции $\bar{\psi}_{r, \rho}$, соответствующее входному вектору Λ , т. е. выполняет операцию дефазсификации нечеткого множества Ψ , например, по методу центра тяжести [6].

4. Формирование области планирования (этап 4)

Симплексы S_i и параллелепипеды P_i , $i \in [r : r + \rho]$ могут сформировать область, которая сильно вытянута и произвольно ориентирована в пространстве R^m (рис. 4). Поскольку качество аппроксимации существенно зависит от выбора области планирования $\Pi_{r, \rho}$, возникает задача определения этой области так, чтобы она была в некотором смысле оптимальной. В качестве критерия оптимальности области $\Pi_{r, \rho}$ используем ее объем $V_{r, \rho}$, т. е. будем искать эту область в виде параллелепипеда, включающего в себя все симплексы S_i или параллелепипеды P_i и имеющего минимальный объем.

В пространстве R^m наряду с системой координат $0\lambda_1\lambda_2\dots\lambda_m$ рассмотрим систему координат $0\lambda'_1\lambda'_2\dots\lambda'_m$, оси которой параллельны ребрам искомого параллелепипеда $\Pi_{r, \rho}$. Матрицу перехода от первой системы координат ко второй системе обозначим $A = (\alpha_{i, j}, i, j \in [1 : m])$, где $\alpha_{i, j} \in [0, 1]$ — направляющие косинусы. Таким образом, задача отыскания оптимального параллелепипеда $\Pi_{r, \rho}^*$

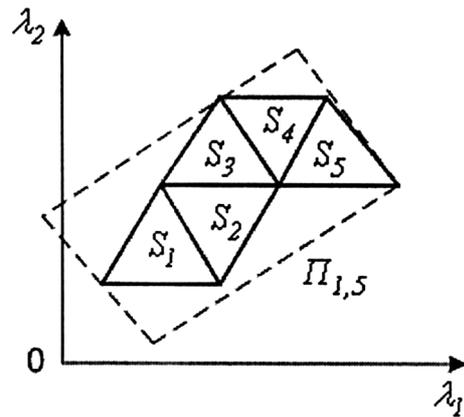


Рис. 4. К построению области планирования: $m = 2$; $r = 1$; $\rho = 4$; $\Pi_{1, 5}$ — прямоугольная область планирования

формализуется в виде многомерной, вообще говоря, многоэкстремальной задачи условной оптимизации

$$V_{r, \rho}^* = V_{r, \rho}(A^*) = \min_{A \in D_A} V_{r, \rho}(A), \quad (6)$$

где $D_A = (\alpha_{i, j} \in [0, 1], i, j \in [1 : m])$.

5. Построение второй аппроксимирующей функции (этап 5)

Для простоты записи сохраним за системой координат, в которой определен параллелепипед $\Pi_{r, \rho}^* = (\Lambda | \lambda_{r, i}^0 - \Delta\lambda_{r, i} \leq \lambda_{r, i} \leq \lambda_{r, i}^0 + \Delta\lambda_{r, i}, i \in [1 : m])$ прежние обозначения.

5.1. Аппроксимация линейной функцией. Построение линейной аппроксимирующей функции $\bar{\psi}_{r, \rho}(\Lambda) = b_{r, 0} + b_{r, 1}\lambda_1 + \dots + b_{r, m}\lambda_m$ МКО-система выполняет по следующей схеме:

- 1) исходя из центра области планирования $\Pi_{r, \rho}^*$ строит план ПФЭ 2^m ;
- 2) во всех точках построенного плана вычисляет значение первой аппроксимирующей функции $\bar{\psi}_{r, \rho}(\Lambda)$;
- 3) формирует СЛАУ для вектора оценок коэффициентов регрессии $B_r = (b_{r, 0}, b_{r, 1}, \dots, b_{r, m})^T$ и решает эту СЛАУ.

Аппроксимация линейной функцией на основе регрессионных планов первого порядка выполняется по аналогичной схеме, но используемые планы берутся из библиотеки планов, рассмотренной в п. 2.2.

5.2. Аппроксимация квадратичной функцией. В данном случае строится квадратичная регрессионная модель вида

$$\begin{aligned} \bar{\psi}_{r, \rho}(\Lambda) = & b_{r, 0} + \sum_{i=1}^m b_{r, i}\lambda_i + \\ & + \sum_{i=1}^{m-1} \sum_{j=i+1}^m b_{r, i, j}\lambda_i\lambda_j + \sum_{i=1}^m b_{r, i, i}\lambda_i^2, \end{aligned}$$

где общее число неизвестных коэффициентов равно $d = \frac{(m+1)(m+2)}{2}$.

Схема аппроксимации аналогична схеме эксперимента п. 5.1 с тем отличием, что в качестве планов используются планы ПФЭ 3^m второго порядка.

На базе ортогональных центрально-композиционных планов (ЦКП) второго порядка разработана библиотека регрессионных планов для задачи многокритериальной оптимизации, содержащей от 2 до 10 частных критериев оптимальности. В данном случае схема построения аппроксимирующей квадратичной функции аналогична схеме п. 5.1, но в качестве планов используются указанные библиотечные ЦКП.

Также на основе слабо насыщенных ЦКП и композиционных планов Хартли и Вестлейка второго порядка разработана библиотека регрессионных планов. Схема аппроксимации на основе этих планов аналогична схеме вычислений п. 5.1.

6. Определение начальной точки для следующей итерации (этап б)

6.1. *Линейная аппроксимирующая функция* $\tilde{\psi}_{r,\rho}(\Lambda)$. Здесь МКО-система находит новую начальную точку Λ_{r+1}^0 по следующей схеме:

1) определяет компоненты вектора градиента функции $\tilde{\psi}_{r,\rho}(\Lambda)$ в точке $\Lambda_{r,\rho}^0$;

2) вычисляет текущий шаг в направлении градиента $s_r = \varepsilon_1 \Delta \lambda_r^*$, где ε_1 — свободный параметр метода, а величина $\Delta \lambda_r^*$ выбирается из условия

$$\max_{i \in [1, m]} |b_{r,i} \Delta \lambda_{r,i}^*| = |b_{r,i}^* \Delta \lambda_{r,i}^*|;$$

3) вычисляет координаты нового центра серии опытов $\Lambda_{r+1}^0 = \Lambda_r^0 + s_r (B_{r,1} B_{r,2} \dots B_{r,m})^T$, где $B_{r,i}$ — нормированные компоненты вектора $b_{r,i}$, $i \in [1 : m]$;

4) проверяет принадлежность точки Λ_{r+1}^0 множеству допустимых значений D_Λ . Если $\Lambda_{r+1}^0 \in D_\Lambda$, то полагает $r = r + 1$ и переходит к следующей итерации. В противном случае уменьшает длину текущего шага s_r , т. е. полагает $s_r = \eta_1 s_r$, где η_1 — свободный параметр метода, и переходит к п. 3.

6.2. *Квадратичная аппроксимирующая функция* $\tilde{\psi}_{r,\rho}(\Lambda)$. Схема определения новой начальной точки аналогична схеме п. 6.1 с тем отличием, что МКО-система определяет координаты $(\lambda_{r+1,1}^0, \lambda_{r+1,2}^0, \dots, \lambda_{r+1,m}^0)^T = \Lambda_{r+1}^0$ точки максимума

аппроксимирующей квадратичной функции $\tilde{\psi}_r(\Lambda)$ как решение СЛАУ

$$\left\{ \begin{array}{l} b_{r,1} + \sum_{i=2}^m b_{r,1,i} \lambda_{r+1,i}^0 + 2b_{r,1,1} \lambda_{r+1,1}^0 = 0; \\ b_{r,2} + \sum_{i=3}^m b_{r,2,i} \lambda_{r+1,i}^0 + 2b_{r,2,2} \lambda_{r+1,2}^0 = 0; \\ \dots \\ b_{r,m-1} + b_{r,m-1,m} \lambda_{r+1,m}^0 + \\ + 2b_{r,m-1,m-1} \lambda_{r+1,m-1}^0 = 0; \\ b_{r,m} + 2b_{r,m,m} \lambda_{r+1,m}^0 = 0. \end{array} \right.$$

Заключение

На основе рассмотренных методов может быть построен широкий класс прямых адаптивных алгоритмов решения МКО-задачи. Один из таких алгоритмов построен, и его эффективность проверена при решении задачи многокритериальной оптимизации автономного электрогидравлического следящего привода [12]. Далее предполагается исследовать эффективность этих алгоритмов на других классах МКО-задач.

Список литературы

1. Карпенко А. П., Федорук В. Г. Обзор программных систем многокритериальной оптимизации. Отечественные системы // Информационные технологии. 2008. № 1. С. 15–22.
2. Лотов А. В. Введение в экономико-математическое моделирование. М.: Наука, 1984. 392 с.
3. Растринин Л. А., Эйдук Я. Ю. Адаптивные методы многокритериальной оптимизации // Автоматика и телемеханика. 1985. № 1. С. 5–26.
4. Miettinen K., Mäkelä M. M. Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet // Computers & Operations Research. 2000. № 27. P. 709–723.
5. Карпенко А. П., Федорук В. Г. Адаптивные методы решения задачи многокритериальной оптимизации, использующие аппроксимацию функции предпочтений лица, принимающего решения // Наука и образование: электронное научное издание. Инженерное образование, www.techomag.edu.ru (№ Гос. регистрации 0420700025, ЭЛ № ФС 77-30569). 2008. № 4.
6. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. Новосибирск: Изд-во Ин-та математики, 1999. 270 с.
7. Агтетков А. В., Галкин С. В., Зарубин В. С. Методы оптимизации. М.: Изд-во МГТУ им. Н. Э. Баумана, 2003. 440 с.
8. Грачев Ю. П., Плаксин Ю. М. Математические методы планирования эксперимента. М.: ДеЛи принт, 2005. 296 с.
9. Хайкин С. Нейронные сети: полный курс. М.: ООО "И. Д. Вильямс", 2006. 1104 с.
10. Карпенко А. П., Мальшев В. Н., Попов Д. Н., Федин В. А. Многокритериальная оптимизация автономного электрогидравлического следящего привода прямым адаптивным методом // Наука и образование: электронное научное издание. Инженерное образование, www.techomag.edu.ru (№ Гос. регистрации 0420700025, ЭЛ № ФС 77-30569). 2008. № 5.

В. Д. Аввакумов, канд. техн. наук, проф.,
Новоуральский государственный
технологический институт,
e-mail: awwam@mail.ru

Оптимальное размещение плоских объектов произвольной геометрической формы

Излагается численный способ решения задачи оптимального размещения плоских объектов произвольной геометрической формы, основанный на методе стыковки объектов. Приводятся решения задачи однорядного периодического размещения простых конгруэнтных объектов на полосе произвольной ширины методом Стояна и методом стыковки. Показано применение метода стыковки для сложного объекта.

Ключевые слова: оптимизация, размещение, плоские объекты сложной формы, область допустимых размещений, R -функция, плотное размещение, опорная функция, метод стыковки, функция плотного размещения.

Введение

В настоящее время существует широкий спектр методов решения задач оптимального размещения геометрических объектов, относящихся к задачам дискретной оптимизации и являющихся NP-трудными. Одним из направлений решения таких задач являются методы, основанные на построении области допустимых размещений.

К таким методам относится разработанный в 70-х годах метод Ю. Г. Стояна [1]. Данный метод использует введенные В. Л. Рвачевым R -функции для аналитического построения годографа функции плотного размещения и опорной функции, на основе которых находится искомое решение.

В конце 90-х годов был предложен метод, который строит область допустимых размещений с помощью операций Минковского [2].

Однако эти и другие известные методы решения задач размещения имеют ряд недостатков. Например, методы, использующие теорию R -функций, достаточно трудоемки и не являются универсальными по отношению к геометрии рассматриваемых объектов (для каждого объекта, отличающегося размерами и/или геометрией, требуется выводить свои соотношения, необходимые для решения задачи). Для метода, базирующегося на операциях Минковского, построение области допустимых решений для объектов сложной формы, как сказано в [2], требует значительных вычислительных ресурсов.

Таким образом, проблема разработки новых, более эффективных методов решения задач дискретной оптимизации на сегодняшний день остается актуальной.

В настоящей работе рассмотрено приведенное в [1] решение задачи периодического однорядного размещения конгруэнтных объектов в полосе и показано численное решение этой задачи с помощью метода стыковки, разработанного автором [3]. Также показано решение данным методом задачи размещения сложного геометрического объекта.

1. Постановка задачи

Рассмотрим задачу, приведенную в работе [1] (задача 4.1, пример 4.6): требуется найти такую ориентацию одинаковых объектов (угол поворота объектов θ^*) относительно полосы L и ее ширину, чтобы коэффициент заполнения полосы объектами был наибольшим. Объектом считается четырехугольник с координатами вершин $(3,6; -3,47)$, $(-4,47; -3,47)$, $(-2,6; 3,35)$, $(2,23; 1,74)$.

2. Решение задачи методом Стояна

В работе [1] решение задачи сводится к определению таких значений угла θ , при которых будет найдено минимальное значение выражения

$$\min_{\theta \in [0, \pi]} v(\theta) = \min_{\theta \in [0, \pi]} [h(\theta) + h(\theta + \pi)]f(\theta), \quad (1)$$

где $v(\theta)$ — площадь криволинейного параллелограмма, ограниченного линиями полосы L и идущими от одной линии полосы к другой одинаковыми (левыми или правыми) граничными линиями двух объектов; $h(\theta)$ — опорная функция объекта (расстояние от полюса до опорной прямой); $f(\theta)$ — функция плотного размещения объектов (расстояние между полюсами двух объектов при их плотном размещении); угол $\theta \in [0, \pi]$.

Последовательность решения задачи по методу, изложенному в [1], такова:

1) для данного вида размещаемого объекта выводится аналитическое выражение для опорной функции $h(\theta)$ (выражение (4.13) в [1]):

$$h(\theta) + h(\theta + \pi) = \begin{cases} 5\cos(\theta - 46^\circ 0' 18'') + 4,24\cos(\theta - 37^\circ 52' 30''), & \text{если } 0 \leq \theta \leq 74^\circ 44' 42''; \\ 5\cos(\theta - 46^\circ 0' 18'') + 5,657[\cos(\theta - 127^\circ 52' 30'')], & \text{если } 74^\circ 44' 42'' \leq \theta \leq 104^\circ 40' 35''; \\ 8,485[\cos(\theta - 127^\circ 52' 30'')], & \text{если } 104^\circ 40' 35'' \leq \theta \leq 161^\circ 33' 54''; \\ 4,24\cos(\theta - 217^\circ 52' 30'') + 5,657\cos(\theta - 127^\circ 52' 30''), & \text{если } 161^\circ 33' 54'' \leq \theta \leq 180^\circ; \end{cases} \quad (2)$$

2) записывается аналитическое выражение для функции плотного размещения $f(\theta)$ (выражение (3.43) в [1]):

$$f(\theta) = \begin{cases} \rho_{1,j} |\sec(\theta_{1,j} - \theta)|, & \text{если } \beta_{1,j} \leq \theta \leq \beta_{1,j+1}; \\ \rho_{2,j} |\sec(\theta_{2,j} - \theta)|, & \text{если } \beta_{1,j+1} \leq \theta \leq \beta_{1,j+2}, \end{cases} \quad (3)$$

где $\rho_{1,j}$ и $\rho_{2,j}$ — минимальные расстояния от полюса объекта до опорных прямых (составляющие годографа функции плотного размещения);

3) находятся корни уравнения

$$v'(\theta) = [h'(\theta) + h'(\theta + \pi)]f(\theta) + [h(\theta) + h(\theta + \pi)]f'(\theta) = 0, \quad (4)$$

среди которых выбирается тот, при котором достигается $\min v(\theta)$. Для поставленной задачи это следующие значения корней θ^* : 0° , $74^\circ 44' 42''$, $104^\circ 40' 35''$, $161^\circ 33' 54''$. Далее имеем:

$$\begin{aligned} \min_{\theta \in [0, \pi]} v(\theta) &= v(0^\circ) = v(74^\circ 44' 42'') = \\ &= v(104^\circ 40' 35'') = v(161^\circ 33' 54''). \end{aligned}$$

3. Решение задачи методом стыковки

Рассмотрим общую методику численного решения задач размещения с помощью *метода стыковки* [3], реализованного в программе СБОРКА [4]. Суть данного метода заключается в определении взаимного положения плоских объектов при движении одного из них к другому в заданном направлении до контакта объектов, если контакт возможен.

Пусть подвижный объект в первоначальном положении находится на некотором расстоянии от неподвижного. Сформулируем задачу следующим образом: требуется определить, на какое расстояние следует переместить подвижный объект в заданном направлении к неподвижному до момента их контакта. Эта задача сводится к вычислению минимального расстояния между подвижным и неподвижным объектами в заданном направлении при их начальном положении.

Сформулированная оптимизационная задача решается с использованием дискретно-непрерывного подхода: область возможного контакта объектов разделяется на множество пар линий их контуров, состоящих из отрезков прямых и дуг окружностей (отрезок—отрезок, отрезок—дуга, дуга—дуга). Между этими парами с помощью геометрических построений (с использованием соответствующих аналитических выражений) находятся локальные минимумы расстояний, среди которых выбирается глобальный минимум. Такой подход позволяет получить быстрый и устойчивый алгоритм поиска глобального минимума.

Алгоритмы, реализующие рассматриваемые движения, построены так, что движущийся объект может только касаться другого объекта, но не пересекать его. Таким образом, автоматически реализуется условие взаимного непересечения объектов.

Совокупность состыкованных объектов представляет собой виртуальную сборочную единицу (сборку). Математическое выражение для модели сборки может быть записано в виде

$$MS = \langle \{MD\}, \{MO\}, \{MR\} \rangle, \quad (5)$$

где $\{MD\}$ — множество геометрических моделей объектов, входящих в сборку; $\{MO\}$ — множество операций с объектами для формирования сборки (операторы задания *направления стыковки* и операторы *стыковки*); $\{MR\}$ — множество вычисляемых расстояний между объектами в сборке.

Геометрическая модель объекта строится с помощью языка ПЛОГ [4] и в общем виде записывается так:

$$MD = \langle \{R\}, \{T\}, \{L\} \rangle,$$

где $\{R\}$ — множество размеров, определяющих геометрию объекта; $\{T\}$ — множество характерных точек, определяющих геометрию объекта; $\{L\}$ — множество контуров объекта.

Контур объекта (объект может состоять из нескольких контуров) представляет собой замкнутую линию, состоящую из множества геометрических примитивов двух типов: отрезков прямых и дуг окружностей:

$$L = \langle \{L_p\}, \{L_d\} \rangle, \quad (6)$$

где $\{L_p\}$ — множество отрезков прямых; $\{L_d\}$ — множество дуг окружностей.

Для получения виртуальной модели сборки требуется описать сборку: указать, какие объекты в нее входят, где находятся описания этих объектов (либо в пакете исходных данных, либо в базах данных) и расположить объекты относительно друг друга, как того требует структура сборки, с помощью операций *стыковки*.

После получения виртуальной модели сборки программа СБОРКА проводит расчет необходимых расстояний между объектами.

Как было показано выше, поставленная задача решается путем нахождения таких значений угла θ , которые удовлетворяют выражению (1). Для этого надо иметь выражения для функций $h(\theta)$ и $f(\theta)$. *Метод стыковки* позволяет находить эти функции в численном (табличном) виде.

Рассмотрим алгоритмы вычисления функций $h(\theta)$ и $f(\theta)$, проведем их вычисления и найдем площадь криволинейного параллелограмма $v(\theta)$. Расчеты проводятся на PC Celeron M 1,5 GHz.

3.1. Нахождение опорной функции

Опорную функцию $h(\theta)$ будем вычислять следующим образом. К заданному объекту в направлениях $\theta_1 = 270^\circ + \theta$ и $\theta_2 = 90^\circ + \theta$, проходящих через его полюс (в качестве полюса объекта взят его центр масс — точка I), будем поступательно двигать опорные прямые (ОП) до контакта с объектом (рис. 1) (ОП представляют собой тонкие плиты длиной, превосходящие наибольшие габаритные размеры объекта).

Вычислим расстояние $q(\theta) = h(\theta) + h(\theta + \pi)$ между ОП в положении их контакта с объектом. Увеличивая угол θ на некоторую малую величину (на 1°) и поворачивая обе ОП против часовой стрелки на углы θ_1 и θ_2 , находим каждый раз положение их контакта с объектом и вычисляем расстояние q между ОП (до $\theta = 180^\circ$). Полученная таким образом опорная функция представлена на рис. 2.

Время расчета функции $q(\theta)$ по программе СБОРКА — 0,33 с.

Математическая модель сборки (5) в данном случае представляет собой выражение:

$$MS = \langle \{D_1, D_2\}, \{H_1, H_2, ST_1, ST_2\}, \{R\} \rangle,$$

где D_1 — описание геометрии четырехугольника (заданного объекта) (см. рис. 1); D_2 — описание плиты; H_1, H_2 — направления стыковки плит; ST_1 — операция стыковки плиты в направлении H_1 ; ST_2 — операция стыковки плиты в направлении H_2 ; R — вычисляемое расстояние между плитами.

Описание геометрии объекта будет выглядеть следующим образом:

$$D_1 = \langle T_1, L_1 \rangle,$$

где $T_1 = \{2, 3, 4, 5\}$ — множество характерных точек контура объекта; $L_1 = \{2-3, 3-4, 4-5, 5-2\}$ — множество отрезков прямых контура объекта.

Геометрическая модель плиты изображена на рис. 3.

Описание геометрии плиты будет выглядеть следующим образом:

$$D_2 = \langle R_2, T_2, L_2 \rangle,$$

где $R_2 = \{p_1, p_2\}$ — множество размеров плиты; $T_2 = \{3, 4, 5, 6\}$ — множество характерных точек контура плиты; $L_2 = \{3-4, 4-5, 5-6, 6-3\}$ — множество отрезков прямых контура плиты.

Алгоритм, реализующий нахождение опорной функции.

Шаг 1. Присвоение начальных значений параметрам: $\theta = 0^\circ, i = 1$.

Шаг 2. Стыковка D_2 с D_1 в направлении θ .

Шаг 3. Стыковка D_2 с D_1 в направлении $\theta + 180^\circ$.

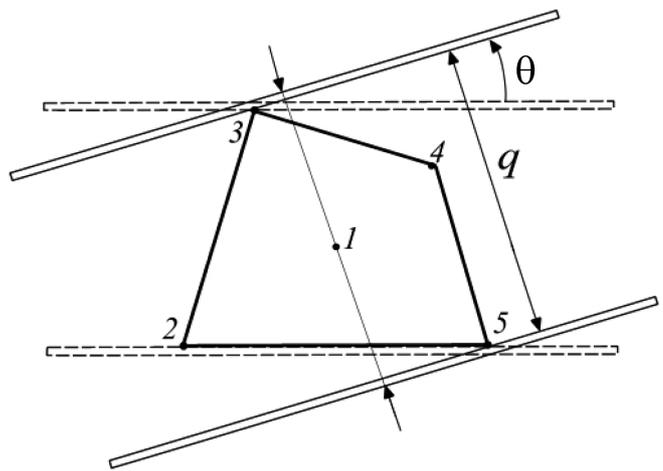


Рис. 1. Положение ОП после контакта с объектом

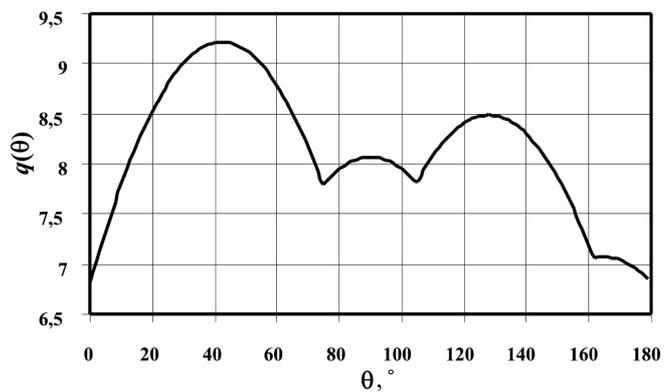


Рис. 2. Опорная функция $q(\theta)$

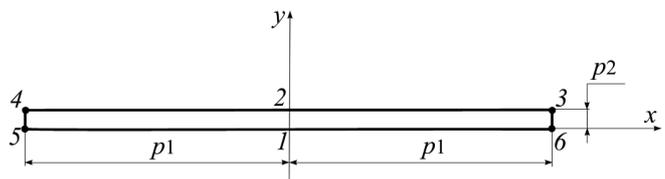


Рис. 3. Геометрическая модель плиты

Шаг 4. Вычисление расстояния $R[i]$ между плитами.

Шаг 5. Увеличение θ и i : $\theta = \theta + 1^\circ, i = i + 1$.

Шаг 6. Проверка: если $\theta \leq 180^\circ$, то переход к шагу 2.

Шаг 7. Останов.

3.2. Нахождение функции плотного размещения

Для вычисления функции $f(\theta)$ будем поступательно передвигать один объект к другому (неподвижному) в направлении, противоположном вектору, соединяющему полюсы O_1 и O_2 , до контакта объектов, находя в этом положении расстояние $O_1 O_2$ между полюсами (рис. 4).

Изменяя направление движения подвижного объекта к неподвижному на малую величину (1°)

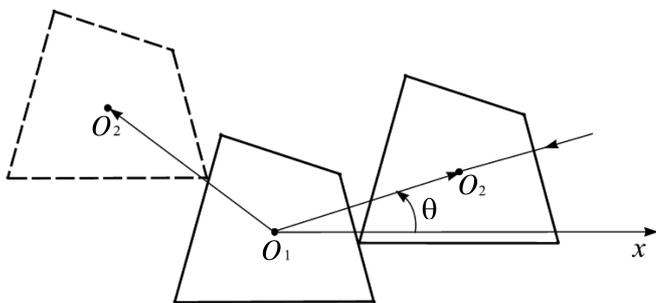


Рис. 4. Положение подвижного объекта

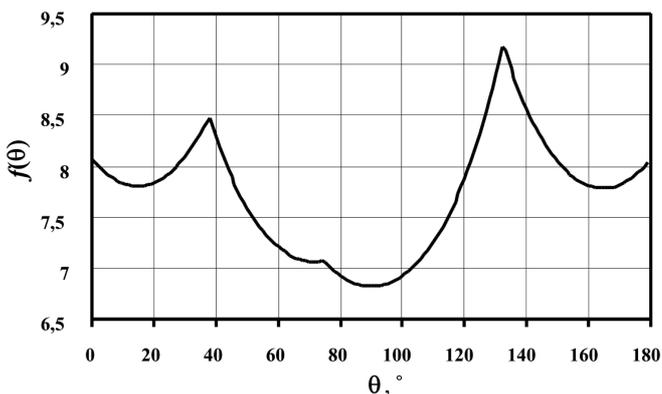


Рис. 5. Функция плотного размещения $f(\theta)$

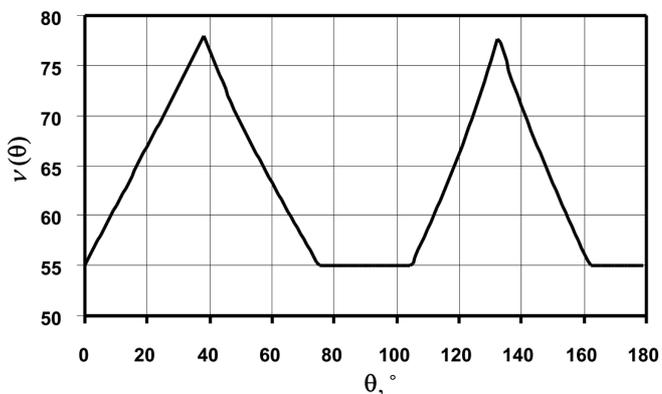


Рис. 6. Площадь параллелограмма

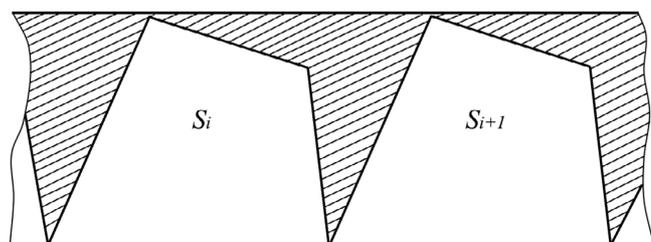


Рис. 7. Один из вариантов оптимального размещения заданного объекта

и находя расстояние между их полюсами, получим искомую функцию $f(\theta)$ (рис. 5).

Время расчета функции $f(\theta)$ по программе СБОРКА — 0,11 с.

Модель сборки (5) в данном случае имеет вид

$$MS = \langle \{D_1\}, \{H, ST\}, \{R\} \rangle,$$

где D_1 — описание заданного объекта (заданного четырехугольника); H — направление стыковки объекта к объекту; ST — операция стыковки; R — вычисляемое расстояние между полюсами объектов.

Алгоритм, реализующий нахождение функции плотного размещения.

Шаг 1. Присвоение начальных значений параметрам: $\theta = 0^\circ$, $i = 1$.

Шаг 2. Стыковка D_1 с D_1 в направлении θ .

Шаг 3. Вычисление расстояния $R[i]$ между полюсами объектов.

Шаг 4. Увеличение θ : $\theta + 1^\circ$

Шаг 5. Проверка: если $\theta \leq 180^\circ$, то переход к шагу 2.

Шаг 6. Останов.

3.3. Нахождение площади параллелограмма

Имея $h(\theta)$ и $f(\theta)$ по выражению (1) находим площадь криволинейного параллелограмма (рис. 6).

На рис. 6 видно, что минимальное значение 55.0 функция $v(\theta)$ принимает при $\theta^* = 0^\circ, 75^\circ, 104^\circ, 162^\circ$, что совпадает (с точностью, соответствующей выбранному по углу шагу) с результатами, полученными в [1].

На рис. 7 изображено оптимальное размещение заданного объекта при $\theta^* = 0^\circ$.

4. Применение метода стыковки для сложного объекта

Выше при решении задачи оптимального размещения рассматривался простой объект, представляющий собой выпуклый четырехугольник. Применим метод стыковки для более сложного объекта (рис. 8).

Форма объекта образмерена (параметризована) так, чтобы максимально соответствовала геометрическому объекту работы [2].

Геометрия объекта определяется девятью линейными размерами $P_1 - P_9$ (на рис. 8 это расстояния между близлежащими точками) и тремя угловыми размерами: $P_{10} - P_{12}$. На рис. 9—11 изображены зависимости $q(\theta)$, $f(\theta)$ и $v(\theta)$, полученные с помощью программы СБОРКА при значениях размеров: $P_1 = 5,2$; $P_2 = 2,2$; $P_3 = 2,6$; $P_4 = 0,5$; $P_5 = 1,7$; $P_6 = 2,4$; $P_7 = 0,75$; $P_8 = 0,5$; $P_9 = 0,8$; $P_{10} = 85^\circ$; $P_{11} = 70^\circ$; $P_{12} = 45^\circ$.

Площадь при заданных размерах объекта (вычисленная по программе МГН [4]) равна 15,097.

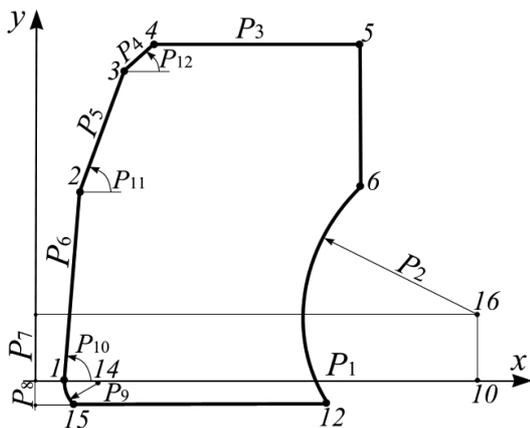


Рис. 8. Геометрическая модель сложного объекта

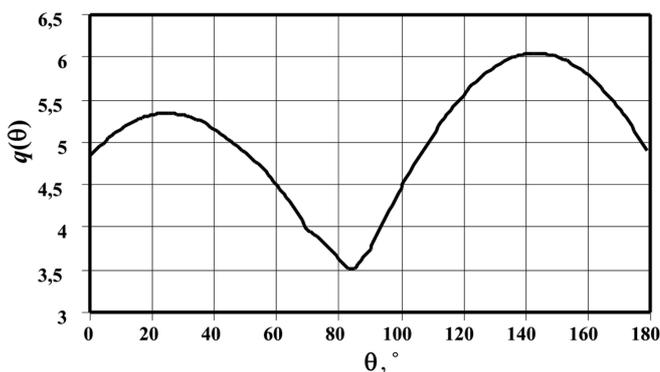


Рис. 9. Опорная функция $q(\theta)$ сложного объекта

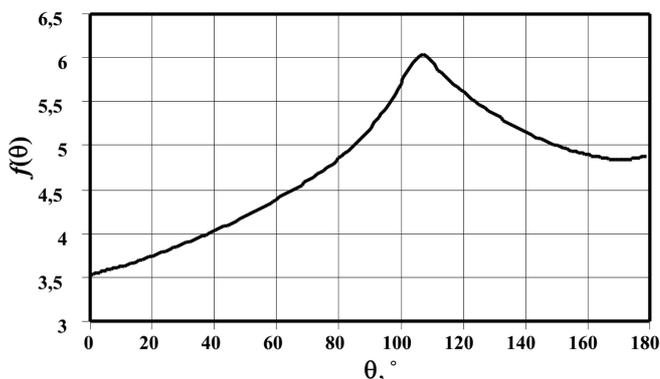


Рис. 10. Функция плотного размещения $f(\theta)$ сложного объекта

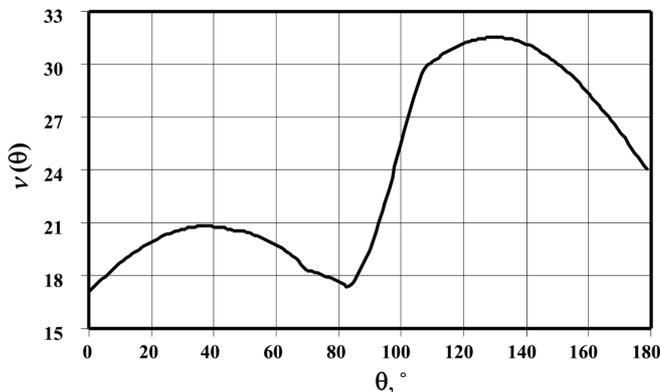


Рис. 11. Площадь параллелограмма сложного объекта

Минимальное значение площади параллелограмма (см. рис. 11) найдено при $\theta^* = 0^\circ$ и равно 17,120. Таким образом, коэффициент плотности заполнения $\chi = \frac{15,097}{17,120} 100 = 88,183 \%$ (в работе [2] $\chi = 84,935 \%$). Незначительное различие в коэффициентах плотности заполнения можно объяснить, по-видимому, тем, что объект, изображенный на рис. 8, не полностью соответствует геометрическому объекту работы [2].

На рис. 11 видно, что функция площади параллелограмма имеет еще один локальный минимум при $\theta^* = 83^\circ$: $\chi = 87,042 \%$.

Время расчета опорной функции и функции плотного размещения для данного сложного объекта практически совпадает с соответствующими временами для простого объекта (разделы 3.2 и 3.3).

Заключение

- ♦ Используемый для решения задачи оптимального однорядного размещения конгруэнтных объектов произвольной геометрической формы на полосе *метод стыковки*:
- в отличие от метода, изложенного в работе [1], не требует аналитического описания объектов, вывода условий взаимного непересечения объектов, математического выражения для функции цели, т. е. вывода выражений (1)–(4);
- может применяться для достаточно сложных невыпуклых объектов и блоков объектов;
- обладает высокой скоростью и точностью (алгоритм нахождения областей контакта объектов базируется на аналитических соотношениях);
- позволяет находить глобальный оптимум.
- ♦ Следует также отметить, что алгоритмы нахождения опорной функции и функции плотного размещения, являются инвариантным по отношению к:
- размерам объектов (в описании геометрии объекта меняются только числовые значения размеров);
- геометрии объектов (использование другого объекта влечет за собой замену только описания геометрии объекта).

Список литературы

1. Стоян Ю. Г. Размещение геометрических объектов. Киев: Наук. думка, 1975. 239 с.
2. Мартынов В. В., Валиуллин А. М. Регулярное размещение двумерных геометрических объектов сложной формы // Прикладная геометрия: электронный журнал. М.: МАИ. 2001. Вып. 3. № 4. С. 9–20. <http://www.mai.ru/~apg/Volume3/Number4/mvv34/martynov34.pdf>.
3. Аввакумов В. Д. Определение взаимного положения объектов при их перемещении в плоскости // Информационные технологии. 2006. № 10. С. 52–58.
4. Аввакумов В. Д. Функциональная размерная параметризация в САПР // Изв. Тульского гос. университета. Сер. "Машиностроение". Вып. 6 (специальный). Сборник избранных трудов конференции "Автоматизация и информатизация в машиностроении 2000" (АИМ 2000). Тула: ТулГУ, 2000. С. 22–28.

УДК 004.89

М. А. Князева, канд. техн. наук,
доц., ст. науч. сотр.,

С. А. Плохих, аспирант,
e-mail: mak@imcs.dvgu.ru

Институт автоматки и процессов
управления ДВО РАН, г. Владивосток

Концепция системы управления специализированного банка знаний о преобразованиях программ

Рассмотрены концепция системы управления специализированного банка знаний о преобразованиях программ и текущее состояние системы управления. Представлена архитектурно-контекстная диаграмма системы управления и ее формальная модель.

Ключевые слова: системы управления, трансформация программ, банк знаний, оптимизирующий компилятор, преобразователь программ.

Введение

Одним из подходов к разработке компьютерных программ является трансформационный подход, при котором разрабатываемая программа создается путем преобразования из других программ за счет применения набора трансформаций.

Преобразованием программ называется любая функция $h: P \rightarrow P$, где P — множество всех семантически корректных программ, а h — корректное (обобщающее, сужающее, эквивалентное) преобразование, такое, что для любой исходной программы $\pi \rightarrow P$ результирующая программа $h(\pi)$ вычисляет (обобщает, сужает, эквивалентно) π [1].

Обычно преобразование программ h рассматривается как суперпозиция элементарных преобразований — трансформаций. Трансформация состоит из формулы трансформации и формулы контекстного условия (условия применимости этой трансформации).

К преобразованиям программ относятся: трансляция программ, их оптимизация и распараллеливание, реинжиниринг программ при их переносе в новое операционное окружение, модификация программ при их сопровождении и др.

Наиболее часто трансформационный подход используется при трансляции с оптимизацией и распараллеливании программ. Основная цель использования таких преобразований — это улучшение качества программ.

Теоретические достижения в области трансляции, оптимизации и распараллеливании программ уже нашли свое практическое применение в современных оптимизирующих и распараллеливающих компиляторах, а также в системах построения трансляторов (СПТ) [2—5].

Предварительные исследования успешности применения одного фиксированного набора трансформаций проводились (с помощью оптимизирующего компилятора) неоднократно: измерялся эффект оптимизации (отношение времени работы/объема памяти неоптимизированной программы к тому же показателю оптимизированной программы) для одной трансформации, для комбинации двух трансформаций и даже для комбинации многих трансформаций.

Для исследования трансформаций программ в настоящее время разрабатывается класс систем трансформаций программ, в которых оптимизирующие и распараллеливающие преобразования программ задаются пользователем в непроцедурном виде и которые основаны на идее трансформационной машины, предложенной А. П. Ершовым [6]. Трансформационная машина — абстрактное вычислительное устройство, командами которого являются трансформации из некоторого фиксированного набора (базовые трансформации), а данными — преобразуемые программы. Примерами систем трансформаций программ могут служить система СОКРАТ (ИСИ СО РАН, г. Новосибирск) [7] и МСПП — многоцелевая система трансформаций программ (Южный федеральный университет, г. Ростов-на-Дону), основанная на непроцедурном описании выполняемых трансформаций в виде схемных правил и позволяющая описывать преобразования многосвязных программных структур [8], а также система СР — проект Мюнхенского технического университета [9]. Эти системы выполняют отдельные трансформации, задаваемые пользователем, но не обеспечивают выполнения последовательностей различных трансформаций, поэтому их использование для исследований весьма ограничено.

Кроме этого, для изучения трансформаций программ разрабатываются интегрированные среды. Примером такой среды является IRE —

среда, разрабатываемая в Институте системного программирования РАН [10].

Для того чтобы можно было проводить экспериментальное исследование переменных наборов трансформаций программ при их оптимизации и/или их распараллеливании, а также выполнять макетирование блоков оптимизации компиляторов, необходимо искать новые подходы к разработке методов и средств исследования преобразований программ. Один из них — декларативный.

В отделе интеллектуальных систем Института автоматизации и процессов управления ДВО РАН ведется разработка специализированного банка знаний о преобразованиях программ (СБкЗ_ПП) [11]. В основе банка лежит концепция о разделенном проектировании знаний и программных блоков. Знания о преобразованиях программ охватывают широкий спектр знаний в данной области, таких как: оптимизирующие преобразования; распараллеливающие преобразования; сведения о различных вычислительных архитектурах; знания о методах потокового анализа программ, о языках программирования высокого уровня. Программные блоки включают в себя подсистемы работы со знаниями и данными, находящимися в информационном наполнении банка знаний о преобразованиях программ. Все подсистемы, собранные в банке знаний, должны работать в определенном порядке и этот порядок зависит от входных данных и требований пользователя. Для управления всеми подсистемами требуется система управления специализированным банком знаний, способная отвечать всем требованиям к этому банку и в полной мере взаимодействовать с пользователем. Целью данной работы является описание требований, предъявляемых к системе управления и концепции системы управления специализированного банка знаний о преобразованиях программ (СУ СБкЗ_ПП).

1. Концептуальное описание СУ СБкЗ_ПП

Система управления является организатором работы всего банка СБкЗ_ПП. Система занимается управлением потоков данных, взаимодействием подсистем банка, планированием заданий пользователя, организацией работы пользователя.

На начальном этапе пользователю необходимо зарегистрироваться на сервере многоцелевого банка знаний (МБкЗ) [12]. МБкЗ является рабочей средой, содержащей несколько различных специализированных банков (СБкЗ), ориентированных на различные прикладные области. МБкЗ содержит в себе модуль регистрации пользователей и наделяет их полномочиями. После того как пользователь прошел регистрацию и получил свое персональное рабочее пространство, ему

предлагается скопировать на свою рабочую станцию клиентскую часть для взаимодействия с СБкЗ_ПП. Клиентская часть содержит в себе только интерфейс, чем обеспечивается небольшой объем программы.

После того как пользователь установил и запустил клиентскую часть системы, ему необходимо ввести данные о регистрации на сервере МБкЗ. Затем СУ СБкЗ_ПП начинает свое функционирование. Она авторизует пользователя и предлагает ему основные три режима работы с СБкЗ_ПП. Первый режим необходим для работы пользователя с базами знаний, содержащимися в СБкЗ_ПП. Второй режим дает пользователю возможность проводить эксперименты в области преобразования компьютерных программ. Третий режим является режимом построения макета оптимизирующего компилятора.

СБкЗ_ПП должен удовлетворять требованиям среды МБкЗ, в связи с этим все данные и подсистемы, кроме клиентской части, располагаются на стороне сервера. Для того чтобы главная база знаний не нарушалась, каждому пользователю дается возможность организовать свое рабочее пространство на стороне сервера.

Основным объектом, с которым работает СУ СБкЗ_ПП, является компьютерная программа. На начальном этапе при вводе ее в базу программ пользователем она представлена на языке высокого уровня, который поддерживает СБкЗ_ПП. После ее введения в базу программ в терминах онтологии языка программирования высокого уровня она преобразуется во внутреннюю структуру данных МСП (модель структурной программы). Далее на всех этапах работы СБкЗ_ПП происходит преобразование МСП [13]. На последнем этапе преобразования МСП переводится на язык ассемблера или на язык высокого уровня, выбранный пользователем.

Все правила преобразования программы описаны формально и содержатся в базе преобразований. Каждое преобразование содержит в себе следующие знания: неформальное описание преобразования; формальное описание преобразования; контекстное условие (КУ) — правило применения преобразования; формулу трансформации (ФТ) — правило преобразования; характеристическую функцию — параметр сортировки набора найденных участков, удовлетворяющих КУ.

СБкЗ_ПП является расширяемой системой как в плане увеличения знаний и данных, так и в возможности увеличения числа совместимых языков высокого уровня, расширения областей преобразований, введения дополнительных функций. Все эти расширения поддерживает и СУ на уровне дополнений и настроек. Каждая допол-

нительная подсистема должна удовлетворять всем тем же требованиям, что и СУ СБкЗ_ПП.

2. Функциональные требования к СУ СБкЗ_ПП

На основе проведенного анализа специализированных банков и систем управления были определены требования к системе управления СБкЗ_ПП.

Требование 1. Для того чтобы система была многопользовательской, в ней должна находиться подсистема администрирования пользователей СБкЗ_ПП. Система должна регистрировать новых пользователей, распределять права и теории, необходимые для их работы.

Требование 2. Чтобы СБкЗ_ПП имел возможность расширения, СУ должна позволять создание, удаление, изменение теорий, т. е. необходимо наличие редактора для работы с теориями.

Требование 3. Любой банк знаний должен позволять проводить операции над данными и знаниями, находящимися в нем, поэтому необходимо иметь возможность добавления новых подсистем (сервисов) в СБкЗ_ПП, содержащихся в программном наполнении.

Требование 4. Чтобы достичь взаимодействия между подсистемами программного наполнения в СБкЗ_ПП необходима подсистема управления ее программным наполнением.

Требование 5. Чтобы заносить, удалять и корректировать знания и данные в СБкЗ_ПП, необходимо наличие соответствующих редакторов знаний и данных для информационного наполнения СБкЗ_ПП.

Требование 6. Чтобы программное наполнение взаимодействовало с информационным наполнением, СУ должна предоставлять вовремя и корректно данные и знания из информационного наполнения для программных сервисов.

Требование 7. Так как все подсистемы в области преобразования программ не предоставляют средств визуального взаимодействия с пользователем, необходимо создание удобного способа обмена информацией с пользователем.

В связи с тем, что СБкЗ_ПП находится в среде МБкЗ, функции, связанные с администрированием пользователей, и работа с теориями осуществляется через сервисы МБкЗ. Добавление новых сервисов проходит через оболочку МБкЗ. Соблюдение первых трех требований перекладывается на среду МБкЗ.

3. Модель СУ СБкЗ_ПП

Рассмотрим формальную модель системы управления СБкЗ_ПП. В ней отражены ключевые подсистемы и связи между ними.

Формальная модель СБкЗ_ПП представляется парой:

$S_{\text{БкЗ_ПП}} = \langle \text{ПН}, \text{ИН} \rangle$, где ПН — это среда в банке, содержащая все программные единицы; ИН — онтологии, знания и данные.

ПН определяется следующим образом: $\text{ПН} = \langle \text{УС}, \text{А}, \text{КМ} \rangle$, где УС — управляющая система; А — набор функциональных модулей; КМ — клиентский модуль.

$\text{ИН} = \{B_j\}, j = 1, \dots, m$ — набор баз знаний, необходимый для работы в области преобразования программ.

УС — управляющая система организует процесс работы в среде СБкЗ_ПП. УС определяется тройкой, $\text{УС} = \langle \text{Р}, \text{ТН}, \text{И} \rangle$, где $\text{Р} = \{P_j\}, j = 1, \dots, k$ — режимы работы УС, которые можно показать как взаимодействие набора модулей $[A_1, \dots, A_k]$ через правила взаимодействия P_k ; ТН — текущая настройка для модели оптимизирующего компилятора; И — процедуры интерфейса.

$\text{А} = \{A_i\}, i = 1, \dots, n$ — набор функциональных модулей для текущей версии УС. Компонент А — список функциональных модулей в текущей версии УС. Каждый модуль характеризуется следующими параметрами:

ВхД — входные данные; ВыхД — выходные данные; ДопД — дополнительные данные (данные, использующиеся только для работы текущего блока и требующие отдельной теории в СБкЗ_ПП; КМ — клиентский модуль, необходимый для организации работы пользователя.

4. Описание работы СУ СБкЗ_ПП

Система управления позволяет пользователю выбрать один из следующих режимов работы.

1. Проведение экспериментов по преобразованию программ.

Данный режим предоставляет пользователю следующие возможности: введение исследуемой программы на языке программирования высокого уровня в базу знаний; загрузка сохраненной ранее программы на языке высокого уровня из СБкЗ_ПП; добавление новых преобразований в базу знаний о трансформациях программ; создание списка применяемых трансформаций; создание стратегии преобразования программ; выбор платформы для генерации исполняемого кода; анализ истории применения преобразований с возможностью исследования структуры программы.

На первом этапе происходит выбор программы для проведения эксперимента. Пользователь имеет следующие возможности: самостоятельно ввести программу на языке высокого уровня (ЯВУ); загрузить программу из списка программ в СБкЗ_ПП; загрузить программу, ранее сохраненную пользователем. Пользователь также может

исправить загруженную программу и сохранить ее. Далее система управления, используя подсистему генерации модели структурной программы, переводит программу на ЯВУ во внутреннее представление МСП. После этого программа измеряется по нескольким параметрам [14], с помощью подсистемы потокового анализа происходит расширение ее терминами потокового анализа. Подсистема потокового анализа программы использует знания о потоковом анализе, хранящиеся в базе знаний о методах потокового анализа программ СБкЗ_ПП. После измерения и анализа программа передается на вход преобразователю.

Перед началом процесса преобразования программы пользователь выбирает несколько параметров. Первое, что необходимо, это выбрать из базы знаний СБкЗ_ПП трансформации. Можно использовать преобразования, уже сохраненные в СБкЗ_ПП, а можно перейти в соответствующий редактор и ввести собственные преобразования. Далее выстраивается порядок применения набора преобразования и стратегия преобразования программы.

После того, как все параметры введены пользователем, система управления начинает процесс преобразования. На данном этапе работают следующие подсистемы: подсистема поиска и проверки контекстных условий, подсистема трансформации, подсистема потокового анализа программ, измерительная подсистема. Каждая из подсистем использует знания из соответствующих баз знаний СБкЗ_ПП [13, 15, 16].

Результатом работы преобразователя программ являются преобразованная МСП и отчет, построенный подсистемой построения отчета, содержащий данные измерительной подсистемы. Пользователь, проанализировав отчет, может изменить параметры работы преобразователя и запустить процесс преобразования программы заново.

На конечном этапе СУ СБкЗ_ПП, используя соответствующую подсистему, переводит преобразованную МСП на ЯВУ. Далее, с помощью подсистемы генерации кода [15], можно перевести программу на язык ассемблера. Генератор кода, используя знания из базы знаний о генерации, имеет возможность выбора платформы, для которой будет сгенерирован исполняемый код.

В результате работы пользователь получает исходный код программы на ЯВУ, код преобразованной программы на ЯВУ, исполняемый код преобразованной программы и подробный отчет со сравнительными характеристиками о преобразовании программы.

2. Построение макета оптимизирующего компилятора.

Режим позволяет построить конфигурацию оптимизирующего компилятора. Пользователь

выбирает входной язык, список трансформаций, стратегию применения трансформаций и платформу для генерации исполняемого кода. Эта информация сохраняется на стороне пользователя. Для ее применения достаточно указать только программу, которая будет компилироваться. После применения прототипа оптимизирующего компилятора пользователю будет предоставлена исполняемая программа с детальным отчетом по компиляции.

3. Работа с информационным наполнением СБкЗ_ПП.

При работе с информационным наполнением СБкЗ_ПП пользователю предоставляется возможность добавлять, изменять и удалять данные и знания о преобразованиях программ.

На момент разработки СУ в СБкЗ_ПП представлены следующие данные и знания: база программ в терминах онтологии языка программирования высокого уровня; база знаний о методах потокового анализа программ; база знаний о трансформациях программ; база знаний о целевых платформах.

Все данные и знания разделены на два типа: те, которые может корректировать пользователь, и данные и знания, которые корректировать может только администратор СБкЗ_ПП. Результат корректировки знаний отразится на работе первых двух режимов.

5. Архитектурно-контекстная диаграмма СУ СБкЗ_ПП

Рассмотрим архитектурно-контекстную диаграмму системы управления СБкЗ_ПП, изображенную на рисунке (см. четвертую сторону обложки).

Диаграмма отображает внешнее окружение СУ СБкЗ_ПП. На ней отражена работа пользователя с СБкЗ_ПП. Пользователь может решать три основные задачи: задача 1 — работа со знаниями и данными; задача 2 — проведение экспериментов над преобразованиями программ; задача 3 — построение макета оптимизирующего компилятора. Для решения задач в СБкЗ_ПП на компьютере пользователя устанавливается интерфейсная часть (тонкий клиент), которая взаимодействует с СУ на стороне сервера. Клиентская часть, которая находится на компьютере пользователя, включает в себя возможность трех режимов работы и соответственно три интерфейса, для решения каждой задачи. Интерфейс построения макетов компиляторов является графической клиентской программой для макетирования оптимизирующих компиляторов. Интерфейс преобразователя программ — это графическая интерфейсная часть для построения экспериментов по преобразованию про-

грамм. Интерфейс средств редактирования — это универсальный графический интерфейс для всех структурных редакторов знаний и данных по преобразованию программ.

Информационно-административная система МБкЗ (ИАС) необходима для регистрации прав и полномочий пользователя. После регистрации при начале работы с интерфейсной частью СБкЗ_ПП пользователю необходимо будет ввести *Логин* и *Пароль*, присвоенный ему при регистрации. СУ взаимодействует с ИАС и проверяет наличие данных пользователя и его права.

Интерфейсная часть взаимодействует с СУ через Интернет. После получения данных от пользователя СУ организует последовательность работы функциональных модулей (ФМ), находящихся в программном наполнении СБкЗ_ПП. Все знания и данные, необходимые для работы ФМ, СУ извлекает из информационного наполнения СБкЗ_ПП через оболочку МБкЗ.

Пользователей можно разделить на три категории, каждая из которых соответствует своей задаче.

Задача 1 — работа со знаниями и данными, в процессе которой приобретаются знания о преобразованиях программ. Пользователям необходимо знать, как представляются знания в данной области, по каждой теории в СБкЗ_ПП.

Задача 2 — построение экспериментов по преобразованиям программ [17]. Пользователи — студенты и экспериментаторы, получающие знания по преобразованиям программ и имеющие возможность проведения экспериментов с разными наборами преобразований и стратегией их применения.

Задача 3 — построение макетов оптимизирующих компиляторов для пользователей более высокого уровня квалификации в области преобразования программ.

Заключение

В работе описана концепция системы управления специализированного банка знаний о преобразовании программ. Система управления организует работу всех подсистем в СБкЗ_ПП, предоставляет данные и знания из баз знаний, организует работу с пользователем посредством визуальных компонентов интерфейса. Одной из приоритетных задач СУ является организация процесса преобразования программы. Формальная модель СУ отображает функциональные объекты и связи между ними. Архитектурно-контекстная диаграмма показывает основные системы

СБкЗ_ПП, их взаимосвязи с окружением и отображает общую концепцию построения систем в МБкЗ. Разработанная концепция является основой для детального проектирования системы управления СБкЗ_ПП.

Работа выполнена при финансовой поддержке ДВО РАН, инициативный научный проект "Интернет-система управления информацией о преобразованиях программ".

Список литературы

1. **Касьянов В. Н., Евстигнеев В. А.** Графы в программировании: обработка, визуализация и применение. — СПб.: БХВ — Петербург, 2003. — 1104 с.
2. **Воеводин В. В., Воеводин Вл. В.** Математические модели и методы в параллельных процессах. — М.: Наука, Гл. ред. физ.-мат. лит., 1986.
3. **Серебряков В. А., Галочкин М. П.** Основы конструирования компиляторов. — М.: Эдиториал УРСС, 2001. — 224 с.
4. **Терехов А. А., Вояковская Н., Бульчев Д., Москаль А.** Разработка компиляторов на платформе .NET. 2001. <http://se.math.spbu.ru>.
5. **Stallman R.** Using and Porting the GNU Compiler Collection. — New York: Addison-Vesley Publishing, 2000. — 556 p.
6. **Ершов А. П.** Трансформационная машина: тема и вариации // Проблемы теоретического и системного программирования. Новосибирск: Изд. НГУ, 1982. С. 5—24.
7. **Захаров Л. А., Касьянов В. Н., Кузьминов Т. В.** и др. СОКРАТ: Окружение надежного и эффективного программирования // Актуальные проблемы информатики, прикладной математики и механики. Ч. III. Информатика. Красноярск: Изд-во СО РАН, 1996. — С. 51—64.
8. **Букатов А. А., Коваль В. В.** Методы реализации трансформационной машины в многоцелевой системе трансформации программ // Информационные технологии. 2004. № 3. С. 24—30.
9. **Method of Programming. Selected Papers on the CIP — project /Ed. by M. Brody and M. Wirsing.** — Berlin a.o.: Springer-Verlag, 1991. — (Lect. Notes Comput. Sci.; Vol. 544).
10. **The IRE Home Page.** <http://www.ispras.ru/groups/ctt/ire.html>
11. **Клещев А. С., Князева М. А.** Управление информацией о преобразованиях программ. I. Анализ проблем и пути их решения на основе методов искусственного интеллекта // Изв. РАН. ТиСУ. 2005. № 5. С. 120—129.
12. **Орлов В. А., Клещев А. С.** Компьютерные банки знаний. Многоцелевой банк знаний // Информационные технологии. 2006. № 2. С. 2—8.
13. **Князева М. А., Волков Д. А.** Поточковый анализ программ, управляемый знаниями // Программные продукты и системы. 2007. № 1. С. 49—52.
14. **Князева М. А., Плохих С. А.** Измерительная подсистема для преобразователя программ // Научная сессия МИФИ-2005: сб. науч. тр. В 14 т. Т. 3. Интеллектуальные системы и технологии. М.: Изд-во МИФИ, 2005. С. 130—131.
15. **Князева М. А., Жеравин М. В.** Генерация низкоуровневого кода, управляемая знаниями // Информационные технологии. 2007. № 10. С. 7—12.
16. **Князева М. А., Маевский М. С.** Проверка контекстных условий и поиск участков экономии в системе преобразований программ // Информационные технологии. 2007. № 12. С. 70—73.
17. **Kovacheva Ts.** Virtual communication among teachers and students in mathematics computer based learning // Proc. of XXXIX International scientific conference on information, communication and energy systems and technologies ICES. — June 16—19, 2004. — Bitolj (Macedonia). — Vol. 2. — P. 691—693.

Е. Я. Волченков, канд. физ.-мат. наук,
независимый эксперт,
г. Москва, yeq-w@yandex.ru,
TS Soft Plus, <http://www.ts-splus.com>

Система комбинационной генерации тестов вход/выход: основные методы и алгоритмы

Описана система комбинационной генерации тестов вход/выход. Предлагается ряд новых методов или модификаций известных эвристических методов выделения эффективных подмножеств тестов. Система реализована в программном комплексе InputSpace TestGenerator.

Ключевые слова: автоматизированное тестирование программного обеспечения, комбинационная генерация тестов, входное пространство, многовыходовые программные гнезда, входные домены, комплект опорных тест-значений, совместимость тест-значений, граничные категории, профили использования, максимально различающаяся случайная (MPC) выборка, с-арные частичные комбинации, специальные с-MPC-покрытия.

Введение

Стандартные методы тестирования вход/выход предполагают "одномерное" описание входных доменов (методы эквивалентных классов, граничного анализа и др.). Такой подход, строго говоря, обоснован только для одновыходовых доменов. Однако в общем случае транзакции вход/выход многовыходовы и соответствующие входные домены необходимо рассматривать как многомерные. В данной статье делается попытка систематического многомерного обобщения соответствующих "одномерных" подходов, что позволяет ввести или развить ряд методов формирования эффективных подмножеств тестовых примеров для многомерного тестирования вход/выход. Результаты работы были использованы при разработке программного комплекса InputSpace TestGenerator [1].

1. Домены и представляющие тест-значения

Программные компоненты для ввода данных — *входы* — естественно группируются в пользовательском интерфейсе по функциональным задачам. *Программное гнездо* можно определить как группу программных входов, данные которых (возможно, совместно с некоторыми глобальными переменными) полностью определяют результат решения функциональной задачи, фиксируемый в конкретном *выходе* — программном ком-

поненте для вывода данных. Анализ сценариев решения пользовательских задач позволяет выделить в программном средстве конечную совокупность программных гнезд.

Элементарным тестовым примером при тестировании вход/выход будет совокупность вводимых во входы гнезда некоторых специально подобранных значений соответствующих переменных, которую мы будем называть *вектором тест-значений* (ВТЗ). Таким образом, полное описание теста (test case) должно включать маршрут прохождения до гнезда (сценарий задачи), конкретный ВТЗ и, возможно, указание последовательности посещения/пропуска входов при вводе (вектор трека). При систематическом тестировании для каждого программного гнезда испытывается целое подмножество ВТЗ — *тест-набор* (test suite) при сохранении общего сценария. Входное пространство гнезда из N входов будет представлять собой N -мерное пространство входных переменных. *Домен гнезда* — это подпространство входного пространства гнезда, для которого допустима обработка программной функцией, ассоциированной с этим гнездом (математически — область определения функции).

Домены (соответственно и гнезда) разбиваются на два существенно различающихся типа:

- домены с независимыми входами;
- домены с взаимозависимыми входами.

В первом случае, рассмотрением которого мы в основном ограничимся в данной статье, *описание домена* сводится к набору независимых граничных интервалов для каждого входа. Геометрически область определения представляется в этом случае гиперпараллелепипедом с ребрами, параллельными осям координат.

Во втором случае допустимые значения одного из входов гнезда зависят от выбранных значений других входов. Для переменных арифметического типа в этом случае можно говорить о функциональной зависимости входных переменных.

Входы арифметического типа. Основной их характеристикой является диапазон изменения или интервал допустимых значений. Интервал допустимых значений может быть разбит на отдельные субинтервалы, соответствующие отдельным классам эквивалентности входа.

Класс эквивалентности представляется своими границами — граничными точками. Для характеристики значений входа введем символические обозначения *граничных типов*. Для границ классов эквивалентности это будут соответственно Mn_* — граница минимума, Mx_* — граница максимума. Внутренней представляющей точке соответствует граничный тип, обозначаемый $_Md$. Для обеспечения возможности проведения отрицательного тестирования вводим также внедиапа-

зонные (недопустимые) значения (граничные типы — соответственно oMn и oMx).

В пределах интервала эквивалентности могут существовать особые [2, стр. 88] или чувствительные [3, стр. 199] значения (граничный тип sMd), в которых велик риск нарушения эквивалентности.

Таким образом, арифметический тип описывается конечной (и, вообще говоря, небольшой) совокупностью специальных или выделенных значений — *представляющих тест-значений*. Каждое тест-значение арифметического типа описывается идентификатором (именем), значением и граничным типом. Совокупность представляющих тест-значений мы будем называть *комплексом опорных тест-значений гнезда*.

Комплекс опорных тест-значений может быть расширен. Для тестирования внутренних областей эквивалентности можно использовать более одной или даже целый контрольный ряд представляющих точек (эквидистантные внутриинтервальные точки), а для тестирования границ — их ε -окрестности (граничные типы $Mn\pm$, $Mn\pm$, $sM\pm$), используемые в методе граничного анализа.

Перечислимый тип характеризует данные, представляющие конечный (перечислимый) набор дискретных единиц, природа которых не имеет значения в контексте тестирования. Каждая такая единица полностью описывается идентификатором (или именем). При малом числе единиц в качестве тест-набора может использоваться их полная совокупность. В противном случае целесообразна группировка единиц (если таковая возможна) в классы перечислимых и использование представителей классов в качестве тест-значений.

2. Комбинационная генерация

Рассмотрим пользовательскую подзадачу с N входами, для каждого из которых возможно m допустимых дискретных значений (имеются в виду входные переменные так называемого перечислимого типа). При каждом частном решении задачи пользователь вводит N конкретных значений, выбирая их из множества допустимых для каждого входа.

Если задача была бы разделима, то ошибка в любой отдельной однопараметрической функции вызывала бы общую ошибку выполнения задачи, и для тестирования в нашей задаче было бы достаточно однократного наличия в тест-наборе каждого допустимого значения, что можно обеспечить тест-набором из m элементарных тестов.

Поскольку переменные задачи не разделимы (за исключением очень специальных случаев), т. е. задача не разбивается на N однопараметрических подзадач, то при выполнении задачи результат зависит от "переплетения" конкретных

значений (маршрут выполнения функции зависит от конкретного сочетания значений по всем входам). Поэтому правильность реализации алгоритма в общем случае должна быть проверена при всех возможных комбинациях допустимых значений для каждого входа. Соответствующий тест-набор полного покрытия может быть сгенерирован для нашего примера путем создания всех возможных N -местных комбинаций допустимых значений (*комбинационная генерация*). Размер тест-набора увеличивается по сравнению с гипотетическим разделимым случаем с m до m^N .

В случае арифметических переменных подзадачи каждый вход мы вынуждены описывать некоторой выборкой значений (комплексом тест-значений). Однако даже в этом случае при попытке увеличения размеров выборки в целях детализации тестирования или даже при минимальном ее размере при увеличении числа переменных подзадачи¹ возникает проблема, связанная с ограниченностью ресурсов, фактически выделяемых на тестирование. Поэтому становится актуальной задача минимизации объемов тест-наборов без существенной потери их эффективности (основная проблема систематического тестирования). Она решается с помощью различных эвристических методов выделения регулярных подмножеств полного множества комбинаций тест-значений (фильтрации полного множества комбинаций).

Полное множество. Комбинирование тест-значений — генерация различных сочетаний тест-значений из комплектов тест-значений входов программного гнезда по одному значению для каждого входа в сочетании. Если имена (или идентификаторы входов) рассматривать как атрибуты, то результат такого комбинирования можно представить в форме реляционной таблицы — таблицы комбинаций тест-значений, в которой каждому сочетанию значений соответствует строка — кортеж, или *вектор комбинаций тест-значений* (ВКТЗ).

В полном множестве комбинаций тест-значений гнезда для каждого значения из комплекта каждого входа должны существовать ВКТЗ, в которых оно сочетается со всеми значениями комплекта любого другого входа.

Подмножества подпространств. Некоторые подмножества n входов N -входного гнезда могут иметь более высокий приоритет в пользовательской подзадаче (например, быть теснее связанными или оказывать большее влияние на результат). В этом случае выделить тестовые подмножества можно, введя ограничиваемое комбинирование. Для его реализации достаточно при комбиниро-

¹ Например, по данным статистического анализа для трех управляющих программ число входных параметров подпрограмм составляло в среднем 10, но могло достигать до 100 [2, с. 113].

вании "заморозить" не приоритетные входы, выбрав одно допустимое значение для каждого. Геометрически это означает сужение тестового пространства до $(N-n)$ -мерной гиперплоскости.

Подмножества подобластей. Если хотя бы у одного входа арифметического типа область определения содержит субинтервалы, то возможно выделение подмножеств, каждое из которых содержит ВКТЗ, содержащие в соответствующем входе тест-значения, принадлежащие только выбранному субинтервалу. Геометрически это означает выделение в N -мерном гиперпараллелепипеде входного пространства вложенного гиперпараллелепипеда меньшего размера.

Классы совместимости. В случае входов, описываемых иерархическими списками, возможны ситуации, когда элемент списка представляющих тест-значений одного входа не может одновременно присутствовать в векторе комбинации (ВКТЗ) с элементом списка другого входа (ср. [4]). Такая ситуация представляет собой по существу простейший тип взаимозависимости входов программного гнезда.

Обозначим T_I — множество тест-значений I -го входа перечислимого типа. Введем *отношение парной несовместимости* между определенными значениями I -го и J -го входов — $O_{IJ} = T_I \times T_J$. Отношение несовместимости может быть обобщено на произвольное подмножество входов. Наличие в ВКТЗ группы несовместимых тест-значений является признаком отнесения его к категории недопустимых. Отношение несовместимости позволяет разделить полное множество ВКТЗ два класса: *совместимых* и *несовместимых*.

3. Обобщенный метод граничного анализа. Граничные категории

Каждому опорному тест-значению в соответствии с интерпретацией его как точки интервала оси значений арифметического типа можно приписать *граничный тип* (символические обозначения граничных типов были приведены выше).

Для независимых входов полное пространство гнезда представляет собой многомерный параллелепипед — гиперпараллелепипед. Каждому вектору значений тест-примера можно сопоставить в этом пространстве точку, что позволяет комбинациям значений комплекта опорных тест-значений дать геометрическую интерпретацию. Граничные типы опорных тест-значений можно использовать как признаки для геометрической классификации векторов значений. Рассмотрим, например, следующий фрагмент тест-набора из трех ВКТЗ (табл. 1).

Интерпретируя отдельные комбинации как точки во входном пространстве, получим для рас-

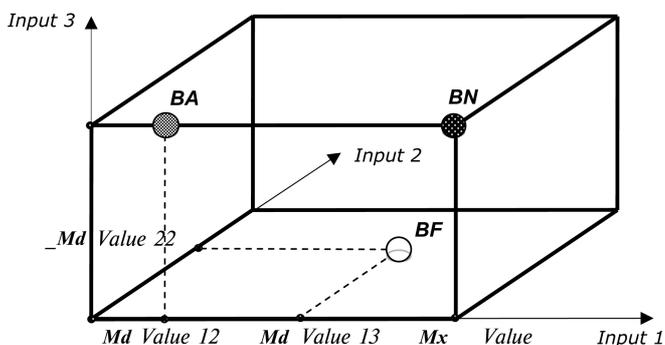


Рис. 1. Геометрическая иллюстрация граничных категорий

Таблица 1

Граничная категория	Input 1	Input 2	Input 3
BN	Mx_Value 12	Mn_Value 22	Mx_Value 34
BA	_Md Value 11	Mn_Value 22	Mx_Value 34
BF	_Md Value 15	_Md Value 27	Mn_Value 22

считываемого фрагмента картинку, иллюстрирующую переход от одномерного представления эквивалентных классов к их многомерному аналогу (рис. 1).

Имея в виду многомерное обобщение "одномерных" методов классов эквивалентности и граничных значений, по аналогии с граничными типами естественно ввести пространственные или *граничные категории* (рис. 1 и 2). Символические обозначения для граничных категорий составлены следующим способом. Первая буква обозначает общие типы: O — ВНЕ, B — ГРАНИЦА, I — ВНУТРИ. Для неграничных внутренних точек вводятся подклассы C — обыкновенные точки, S — чувствительные точки, CS — возможно оба типа. N — точка ϵ -окрестности. И, наконец, используются геометрические характеристики: B — узел, A — ребро, F — грань.

Таблица 2

Категории	Признаки						
	oMn oMx	$Mn-$ $Mx+$	$Mn-$ $Mx-$	$Mn+$ $Mx-$	$sM-$ $sM+$	sMd	$_Md$
OC	x	x	x	x	x	x	x
ONN		x					
ONF		x	x	x	x	x	x
BN			x				
BF			x	x	x	x	x
INBN				x			
INBF				x	x	x	x
INS					x		
INSC					x	x	x
IS						x	
ISC						x	x
IC							x

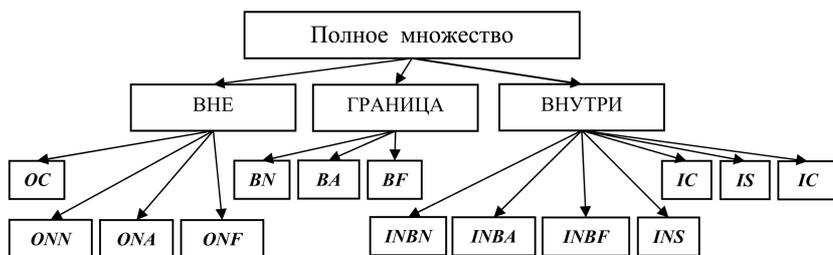


Рис. 2. Иерархическая группировка граничных категорий

Совокупность введенных граничных категорий полностью охватывает полное множество возможных комбинаций опорных тест-значений. Для определения граничной категории ВКТЗ используется табличный классификатор (табл. 2).

На приводимой схеме (рис. 2) показана группировка категорий по типам ВНЕ, ГРАНИЦА, ВНУТРИ, причем окрестностные категории (*IN...*, *ON...*) выделены в отдельный (нижний) ряд.

Введенная геометрическая интерпретация по существу является многомерным обобщением стандартных методов эквивалентных классов и граничного анализа. В многомерном случае наряду с прямыми аналогами $Mn_- / Mx_- \rightarrow BN$ и $Md \rightarrow IC$ появляются и "смешанные" категории. Здесь, как и в "одномерном" подходе, представляет интерес приоритизация тест-комбинаций по величине риска проявления дефекта программного продукта. Так, по степени "концентрации" экстремальных точек вида ГРАНИЦА граничные категории можно упорядочить следующим образом: $BN > BA > BF$.

Для дальнейшей детализации граничных категорий можно ввести дополнительные характеристики. Например, из *BF* можно выделить *BA* (ребро) или ввести показатель — число Mn_- / Mx в ВКТЗ типа ГРАНИЦА (степень экстремальности). Другой полезной характеристикой является число внеграничных окрестностей $Mn_- / Mx+$ для категории *ONF* [1].

4. Профили

Профиль использования программного гнезда позволяет упорядочить ВКТЗ по актуальности тестирования соответствующих областей входных доменов в связи с частотой использования конечным пользователем (или рисками).

Для получения профилей необходимо провести ранжирование [2, стр. 283] допустимых областей изменения значений соответствующих переменных. При отсутствии частотных распределений возможно применение экспертных оценок.

Для ранжирования квазинепрерывных переменных допустимую область изменения разбиваем на интервалы x_i (i — порядковый номер интервала). Каждому из интервалов приписываем ранг

приоритета r_i , т. е. величину, позволяющую упорядочить интервалы в порядке убывания частоты их использования пользователем или критичности — риска проявления сбоя (или, наконец, совместного учета частоты и критичности [5, стр. 98]). Для ранжирования дискретных переменных (входов перечислимого типа) каждому тест-значению приходится приписывать индивидуальное значение

ранга приоритета.

Профилем использования входа мы будем называть зависимость

$$r_i = f(x_i).$$

Число интервалов определяет детальность описания профиля использования. Для задания ранга приоритета r мы используем для наглядности интервал

$$100 \geq r_{100} > 0.$$

Входам в целом можно приписать также ранги или веса v_n

$$100 \geq v_n > 0.$$

Используя ранги веса входов и ранги значений, можно ввести ранги приоритета ВТЗ с помощью рейтинговой меры

$$R = 1/k \sum_n v_n (r_n / 100),$$

где n — номера входов; v_n — вес n -го входа; r_n — ранг тест-значения n -го входа; k — число входов в гнезде.

Из определения видно, что $100 \geq R > 0$.

Используя гистограмму рейтингов $H(R)$, можно подобрать пороговое значение рейтинга R^t , определяющее приоритетное подмножество желаемого размера. Кроме того, ранжирование переменных можно использовать для генерации случайного покрытия входного пространства гнезда по любому требуемому закону распределения [1].

5. Частичное c -комбинационное 100 %-е покрытие

Алгоритм. При увеличении числа входов размер полного множества ВКТЗ экспоненциально возрастает, и его использование становится нереализуемым практически. Так, для комплекта, включающего 13 входов с 3 опорными тест-значениями каждый размер полного множества составляет 3^{13} ВКТЗ. Компромиссный выход из положения — использование специальных случайных выборок.

Равномерная случайная выборка полного множества комбинаций тест-значений не является оптимальным тестовым подмножеством. Рас-

смотрим, например, следующий фрагмент таблицы ВКТЗ (табл. 3).

Перечисленные в примере ВКТЗ различаются только в одном входе. Можно ожидать, что при равномерной случайной выборке в полученном подмножестве не избежать накопления "избыточности" в этом смысле. Однако интуитивно ясно, что эффективность выборки должна возрастать, если в содержащихся в ней векторах будет участвовать как можно больше различающихся тест-значений.

Простейший способ "концентрации" различающихся тест-значений — использование выборки "максимально различимых" ВКТЗ. На каждом шаге формирования подмножества каждый вектор случайной вспомогательной выборки W_i сравнивается с уже отобранными ВКТЗ V_{ij} и для включения в подмножество выбирается максимально отличающийся кандидат. В качестве меры отличия проще всего использовать число несопадающих по каждому i -му входу значений:

$$D = \sum_J \sum_i \sim \delta(W_i - V_{ij}) \rightarrow \max, \quad (1)$$

где i — номер входа; J — индекс суммирования по подмножеству векторов ВКТЗ; V_{ij} — J -й вектор ВКТЗ отобранной выборки; W_i — вектор ВКТЗ — кандидата из вспомогательной выборки; δ — функция Кронекера ($\delta(W - V) = 1$ при $W = V$, $\delta(W - V) = 0$ при $W \neq V$); $\sim \delta = 1 - \delta$.

Этот простейший алгоритм можно пояснить в статистических аналогиях как формирование выборки с максимальной дисперсией.

Введем для ВКТЗ более глубокое различие, чем просто неравенство значений в одноименных входах.

ВКТЗ является N -арной (по числу входов N) комбинацией тест-значений. Ограничим "комбинационность", введя в рассмотрение парные (2-арные) комбинации входов гнезда (см. пример сочетаний пар для 5 входов в табл. 4). Обозначим парную комбинацию I -го и J -го входов через c_{IJ} , а множество всех возможных парных комбинаций для N входов через $C_2(N)$. Мощность этого множества $l_2 = |C_2(N)|$. В нашем примере $l_2 = 10$.

Каждый вход представляется некоторым числом опорных тест-значений (а гнездо — комплектом опорных тест-значений $\{T_j\}$, где T_j — множество тест-значений I -го входа). Наряду с комбинациями входов введем в рассмотрение и частичные парные комбинации тест-значений соответствующих входов [4]. Для получения полного множества парных комбинаций тест-значений для каждой возможной комбинации входов из $C_2(N)$ образуем множество пар тест-значений соответственно из T_I и T_J . Например, если I -й вход представлен m_I значениями, а J -й — m_J значениями, то для IJ -й комбинации входов c_{IJ} , всего возможны $m_I m_J$

Таблица 3

№	Input 1	Input 2	Input 3	Input 4	Input 5
1	Value 3	Value 2	Value 4	Value 7	Value 1
2	Value 3	Value 2	Value 4	Value 7	Value 2
3	Value 3	Value 2	Value 4	Value 7	Value 3

Таблица 4

Входы пар	Input 1	Input 2	Input 3	Input 4	Input 5	Число пар
Input 1	—	×	×	×	×	4
Input 2	—	—	×	×	×	3
Input 3	—	—	—	×	×	2
Input 4	—	—	—	—	×	1
Input 5	—	—	—	—	—	0
Итого						10

парных комбинаций значений. Для каждого N -входного ВКТЗ возможно l_2 парных комбинаций входов (в каждом входе — одно значение). Число различных парных комбинаций значений для двух ВКТЗ в общем случае не равно $2l_2$, так как они могут иметь одинаковые пары.

Случай парных частичных комбинаций естественно обобщается на c -арные частичные комбинации входов/значений (или c -комбинации).

Условимся в списке полного множества c -комбинаций использовать всегда один и тот же постоянный порядок следования отдельных c -комбинаций тест-значений.

Обобщение метрики (1) — c -расстояние между вектором V и подмножеством векторов S :

$$D_{VS} = \sum_s \sum_p \sim \delta(v_p, w_{sp}) \rightarrow \max,$$

где V — пробный вектор; W_s — вектор подмножества S ; v_p, w_{sp} — p -я пара соответственных частичных c -комбинаций значений для векторов V и W_s ; p — индекс суммирования по всем частичным c -комбинациям; s — индекс суммирования по всем векторам подмножества S .

В алгоритме максимально различающейся по c -комбинациям случайной выборки (c -МРС-выборки) в качестве критерия отбора кандидата использована метрика D_{VS} "дальности" вектора КТЗ от подмножества. Естественным признаком остановки накопления c -МРС-выборки для произвольного входного подмножества ВКТЗ является $D_{DS} = 0$.

Продолжая статистическую аналогию, получение c -МРС-выборки можно интерпретировать как переход к главным компонентам (так называемое преобразование Хотеллинга в статистике), при котором происходит концентрация дисперсии во вложенных подпространствах наименьшей размерности (см., например, [6]).

Размер вспомогательной выборки	10	50	100	200	300	700	1000
Размер 100%-го <i>c</i> -MPC-подмножества	27	25	24	22	21	21	21

Общее *c*-MPC-покрытие. Определим *c*-арное покрытие подмножества ВКТЗ как долю *c*-комбинаций тест-значений этого подмножества в полном множестве *c*-комбинаций комплекта тест-значений гнезда. При накоплении *c*-MPC-подмножеств важным порогом является достижение 100%-го *c*-комбинационного покрытия. 100%-покрывающая *c*-MPC-выборка обладает замечательными свойствами. При весьма небольшом размере она гарантирует совместное присутствие в тест-наборе всех, по-крайней мере, *c*-арных комбинаций тест-значений полного комплекта.

Процедура формирования *c*-MPC-подмножества осуществляется путем генерации на каждом шаге отбора новой вспомогательной случайной выборки фиксированного размера и поиска в ней вектора, максимально отличимого от уже накопленного подмножества. Процесс завершается, когда *c*-покрытие полного множества накопленным подмножеством достигает 100 %. ВКТЗ вспомогательной выборки формируются путем случайного выбора тест-значений входов из полного комплекта опорных тест-значений гнезда (общее *c*-MPC-покрытие).

Для минимизации размера *c*-MPC-подмножества процедуру повторяют с увеличивающимся размером вспомогательной выборки до тех пор, пока не наступит насыщение. В табл. 5 приведен соответствующий пример для комплекта, включающего 13 входов с тремя опорными тест-значениями каждый. Размер полного множества ВКТЗ составляет 1594323. Параметр комбинирования $c = 2$.

Размер достигнутой минимальной 100%-й парнопокрывающей выборки составляет 21.

Специальные *c*-MPC-покрытия. *c*-MPC-подмножество может формироваться из некоторого подмножества ВКТЗ, полученного, например, путем фильтрации полного множества одним из описанных выше методов (случай специального *c*-MPC-покрытия). В этом случае максимально достижимое *c*-MPC-покрытие может не достигать 100 %.

Минимальное (или близкое к минимальному) число векторов, обеспечивающих полное *c*-покрытие (21 из $1,6 \cdot 10^6$ в предыдущем примере), можно выбрать огромным множеством способов. Равноценна ли потенциальная тестовая эффективность получающихся всевозможных минимальных *c*-MPC-покрытий? Ответ будет отрицательным. Тест-эффективность соответствующих подмножеств определяется не только полным охватом всех

возможных *c*-арных частичных комбинаций, но и индивидуальной эффективностью самих векторов членов комбинационно покрывающей выборки, зависящей от их конкретного состава. Отсюда вытекают следующие пути повышения эффективности общего *c*-MPC-покрытия:

- найти неполное *c*-MPC-покрытие заданного специального подмножества тест-векторов. Последовательно расширять покрытие за счет добавления других специальных подмножеств;
- если в предыдущем случае 100 %-го покрытия не достигается, то следует дополнить его до полного общепокрывающим способом (беря кандидаты из полного множества).

Заключение

Система комбинационной генерации тест-наборов, реализованная в программном комплексе InputSpace TestGenerator [1], предоставляет широкий набор инструментов, позволяющих гибко формировать оптимальные тест-подмножества исходя из особенностей тестируемого приложения и объема имеющихся ресурсов. Программный комплекс может быть использован в качестве генератора тестов при системном и модульном систематическом функциональном тестировании.

Список литературы

1. **TS Soft Plus:** Методология тестирования и программный продукт. <http://www.ts-splus.com>. Сайт, посвященный программному комплексу InputSpace TestGenerator.
2. **Липаев В. В.** Проектирование математического обеспечения АСУ (системотехника, архитектура, технология). М.: Сов. радио, 1977. 400 с.
3. **Канер С., Фолк Д., Нгуен Е. К.** Тестирование программного обеспечения: Пер. с англ. К.: ДиаСофт, 2000. 544 с.
4. **Cohen D. M., Dalai S. R., Fredman M. L., Patton G. C.** The AETG System: An Approach to Testing Based on Combinatorial Design // IEEE Transactions on Software Engineering. 1997. № 7. P. 437–444.
5. **Макгрегор Д., Сайкс Д.** Тестирование объективно-ориентированного программного обеспечения. Практическое пособие. Пер. с англ. К.: ООО "ТИД "ДС", 2002. 432 с.
6. **Волченков Е. Я.** Преобразование главных компонентов и методы его применения в распознавании образов и обработке изображений // Зарубежная электроника. 1991. № 3. С. 59–72.

И. И. Губайдуллин, аспирант,
А. И. Фрид, д-р техн. наук, проф.,
 Уфимский государственный авиационный
 технический университет (УГАТУ),
 e-mail: ildargubaydullin@mail.ru

Метод экспериментальной оценки надежности программного обеспечения*

Рассмотрен метод оценки надежности программного обеспечения, основанный на моделировании работы программы в зависимости от законов распределения исходных данных. Показано применение метода для оценки надежности программы, реализующей задачу поиска скоростных коэффициентов некоторых химических реакций.

Ключевые слова: программное обеспечение, программа, надежность программы, оценка программы, оценка надежности, моделирование программы.

Введение

Надежность программного обеспечения (ПО) влияет на надежность системы в целом. Нарушение работоспособности ПО часто приводит к не менее тяжелым последствиям, чем отказы техники. Неправильная работа программ может спровоцировать отказы технических устройств, установив для них более тяжелые условия функционирования, поэтому вопросам обеспечения и поддержки надежности ПО всегда уделяется большое внимание.

Имеются различные модели и методы оценки надежности ПО [1, 2]. Все имеющиеся методы можно разделить на два типа:

- априорные, основанные на исходных данных, которые можно получить до начала отладки и эксплуатации программ;
- апостериорные (эмпирические), основанные на результатах отладки и опытной или нормальной эксплуатации программы.

Несмотря на наличие различных методов единого подхода к оценке надежности пока нет. К сожалению, в имеющейся литературе практически не рассматривается применение методов к конкретным программам. Во-первых, это связано с тем, что программа сама по себе является доста-

точно сложным объектом. Во-вторых, многие методы оценки учитывают связку "человек—программа" и при этом характеризуют не надежность программы, а ее изготовителей и инструментарий. Вместе с тем, проблема обеспечения и оценки надежности является общепризнанной.

Основным международным стандартом качества в области инженерии программного обеспечения в настоящее время является стандарт ISO/IEC 9126. Он определяет номенклатуру, атрибуты и метрики требований качества ПО. Согласно этому стандарту, оценка качества ПО основана на трехуровневом рассмотрении [3, 4]:

1) цели (*goals*) или характеристики (*characteristics*) качества ПО;

2) атрибуты (*attributes*) или субхарактеристики (*subcharacteristics*), показывающие приближение к целям;

3) метрики (*metrics*) — количественные характеристики степени наличия атрибутов.

Надежность в такой структуре является характеристикой качества ПО. В свою очередь, надежность описывается субхарактеристиками:

- зрелостью или завершенностью (*maturity*);
- устойчивостью к отказам (*fault tolerance*);
- способностью к восстановлению (*recoverability*);
- соответствием стандартам надежности (*reliability compliance*).

В российской системе стандартизации показатели качества ПО устанавливают ГОСТ 28195-89 и ГОСТ Р ИСО/МЭК 9126-93 [5, 6]. В отличие от последнего, ГОСТ 28195-89 больше ориентирован на разработчиков и содержит оценочные элементы для измерения фактора "Надежность ПС". Среди этих оценочных элементов, являющихся аналогами понятия "метрики" стандарта ISO 9126, отметим два наиболее существенных для данной работы:

- показатель устойчивости к искажающим воздействиям. Рассчитывается по формуле $P(Y) = 1 - D/K$, где Y — устойчивость программы к искажающим воздействиям; D — число экспериментов, в которых искажающие воздействия приводили к отказу; K — число экспериментов, в которых имитировались искажающие воздействия.
- вероятность безотказной работы. Рассчитывается по формуле $P = 1 - N_Q/N$, где N_Q — число зарегистрированных отказов; N — число экспериментов.

В работе предлагается метод второго типа оценки надежности ПО, который позволяет экспериментально путем моделирования работы программы оценить надежность ПО в зависимости от законов распределения исходных данных. В качестве показателя надежности ПО выбран оценочный элемент "вероятность безотказной работы".

* Работа поддержана грантом РФФИ № 08-08-00357 "Методы проектирования отказоустойчивых программно-аппаратных вычислительных комплексов для систем управления сложными техническими объектами с использованием средств искусственного интеллекта".

Экспериментальный метод оценки надежности ПО

Очевидно, что в природе нет ни одного физического явления, в котором не присутствовали бы в той или иной мере элементы случайности. Случайные отклонения сопутствуют любому закономерному явлению [7]. Во многих задачах промышленности программисты неизбежно связаны с такими явлениями. Примером случайных данных являются данные, полученные от различных измерительных устройств. Идея предлагаемого метода заключается в моделировании работы программы, когда на ее вход поступают данные, поведение которых можно описать определенными законами распределения.

Охарактеризуем надежность ПО как вероятность отказа при определенных распределениях исходных данных. Под отказом понимается отклонение поведения системы от предписанного, которое заключается в том, что система перестает выполнять предписанные ей функции [8]. Все отказы ПО разделим на две группы:

- отказы, возникшие из-за наличия дефектов программы;
- отказы, связанные с недостижимостью целей, возложенных на программу.

Отказы второй группы являются алгоритмическими в том случае, если заложенный алгоритм решения задачи не позволяет получить приемлемые результаты за заданное время.

Примерами отказов первой группы являются такие события, как деление на ноль, переполнение разрядной сетки, обращение к несуществующему объекту и т. п. Отказы второй группы — заикливание программы, получение результатов по неточной математической модели объекта, отсутствие результата за заданный интервал времени и т. п.

Предлагаемый метод направлен на определение вероятности алгоритмических отказов. При этом принимается допущение, что для конкретной программы заранее известны типы возможных отказов. Вероятность появления таких отказов рассчитывается путем моделирования работы программы в зависимости от законов распределения исходных данных.

Описание методики оценки надежности программы на примере программы решения квадратного уравнения

Для демонстрации метода оценим надежность программы, реализующей решение квадратного уравнения. Пусть уравнение имеет вид

$$Ax^2 + Bx + C = 0,$$

где A , B , C — случайные входные величины, подчиняющиеся нормальному закону распределения.

Для нахождения корней уравнения воспользуемся формулой

$$x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}.$$

Рассмотрим случай, когда решение квадратного уравнения не должно содержать комплексные корни. Тогда попытка извлечения квадратного корня из отрицательного числа приведет к отказу программы.

В данном случае возможны три типа событий, приводящие к отказу программы:

1) результат математической операции окажется за пределами разрядной сетки;

2) $B^2 - 4AC < 0$; (1)

3) результат $2A$ окажется в области машинного нуля.

К отказу первого типа относится выход результата за разрядную сетку таких операций, как

$$AC; 4AC; B^2 - 4AC; -B \pm \sqrt{B^2 - 4AC}.$$

В данном примере будем считать, что результаты этих операций укладываются в разрядную сетку и поэтому вероятность события 1 равна нулю. Вероятность события 3 примем равной нулю, так как в противном случае теряется смысл квадратного уравнения. Пусть Q — вероятность отказа программы. Тогда значение Q равно вероятности второго события, т. е.

$$Q = P(B^2 - 4AC < 0).$$

Расчет величины Q для программы решения квадратного уравнения проводился в среде Matlab для различных параметров законов распределения входных величин. Изучалось изменение величины Q при изменении математического ожидания (МО) входных величин в диапазоне $[-100, 100]$. МО величины B фиксировалось, а МО величин A и C изменялось с интервалом, равным 5. При этом дисперсия у всех входных величин была равна 100 и оставалась неизменной.

На рис. 1 представлены графики функции Q в зависимости от МО величин A и C , где $МО_B = +40; 40$.

Графики функций мало отличаются друг от друга, так как в формуле (1) входная величина B имеет четную степень. В дальнейшем будем рассматривать графики функции, когда $МО_B$ имеет неотрицательное значение.

Рассмотрим случаи, когда $МО_B = 0$ и $МО_B = 80$ (рис. 2, 3).

Видно, что при приближении $МО_B$ к нулю увеличивается площадь областей, в которых вероятность отказа близка к единице. На рис. 3 представлен вид графиков рис. 2 сверху.

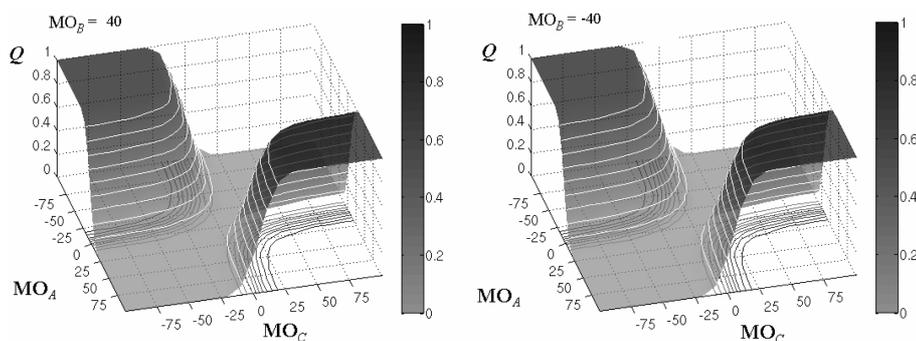


Рис. 1. Зависимость изменения величины Q от MO_A и MO_C при $MO_B = -40$ и $MO_B = 40$

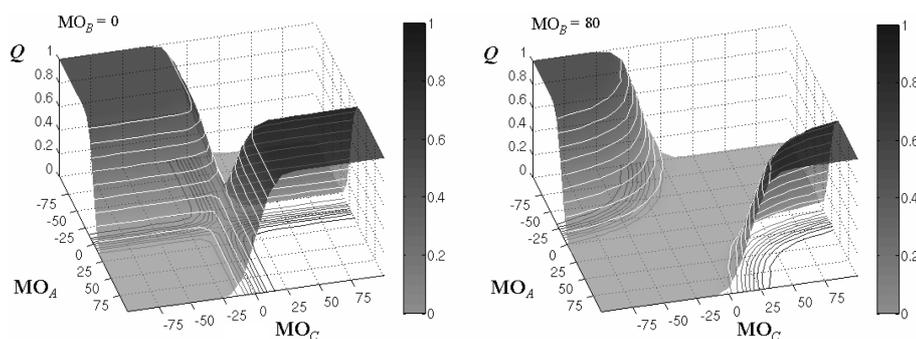


Рис. 2. Зависимость изменения величины Q от MO_A и MO_C при $MO_B = 0$ и $MO_B = 80$

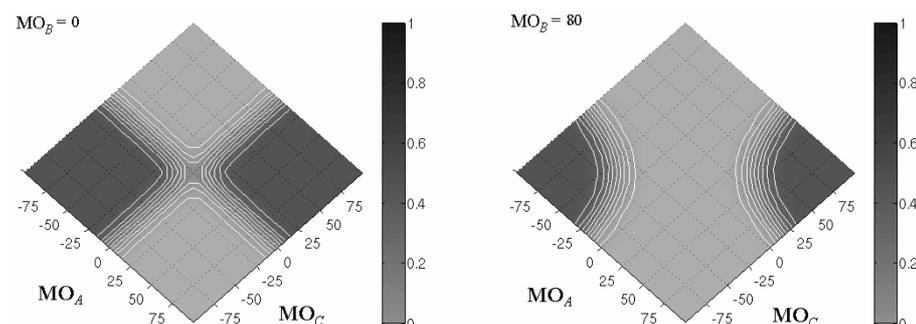


Рис. 3. Зависимость изменения величины Q от MO_A и MO_C при $MO_B = 0$ и $MO_B = 80$ (вид сверху)

Из графиков, представленных на рис. 1–3, можно определить MO_A , B и C , при которых программа будет обладать заданным уровнем надежности. Например, если требуемый уровень надежности составляет 0,9, то MO_A должно быть больше +10, при этом $MO_B = \pm 40$ и $MO_C < -10$.

Применение метода оценки надежности к программе, реализующей формулу $(1 - \log(A + B))^{-1}$, рассматривается в [9].

Таким образом, предлагаемая методика оценки надежности состоит из четырех этапов:

- 1) определение законов распределения исходных данных;
- 2) выявление возможных ошибок в программе;
- 3) моделирование программы с определенными законами распределения исходных данных, оценка надежности;

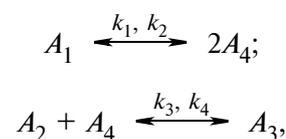
4) определение допустимых параметров законов распределения исходных данных, при которых программа будет обладать заданной надежностью.

Применение метода к оценке надежности программы, реализующей задачу химической промышленности

Применим разработанный метод для оценки надежности программы, реализующей решение одной задачи для химической промышленности.

Программа, решающая химическую задачу, включает в себя оптимизационные алгоритмы и алгоритмы решения системы обыкновенных нелинейных дифференциальных уравнений (СОНДУ). Необходимость проведения исследования такого типа алгоритмов явилась следствием частых остановов при выполнении программы, с помощью которой решается задача поиска скоростных констант химической реакции.

Цель решения задачи — определить скоростные коэффициенты каждой стадии многостадийной химической реакции. Проводится исследование определенной реакции, в процессе которой через заданные интервалы времени фиксируют выход некоторых веществ, причем выход остальных веществ не наблюдается [10]. В качестве примера рассмотрим модель реакции, имеющей следующий вид:



где A_j — определенные вещества (конкретный химический состав каждого вещества в данной работе не приводится), выход веществ A_1 , A_3 — наблюдаем, остальных — нет; k_i (c^{-1}) — скоростной коэффициент i -й стадии.

Математическое описание этой модели, согласно теории химической кинетики, представляется следующей СОНДУ:

$$\begin{cases} dx_1/dt = -w_1; \\ dx_2/dt = -w_2; \\ dx_3/dt = w_2; \\ dx_4/dt = 2w_1 - w_2; \\ dx_5/dt = w_1 - w_2, \end{cases}$$

где w_i (c^{-1}) — скорость i -й стадии.

Для расчета скоростей используются следующие формулы:

$$\begin{aligned} w_1 &= k_1 x_1 - k_3 x_4^2; \\ w_2 &= k_2 x_2 x_4 - k_4 x_3, \end{aligned}$$

где x_i — мольные доли i -го вещества в конкретный момент времени.

Кроме того, имеются физические ограничения в виде

$$\begin{cases} k_1 < k_3; \\ k_2 > k_4. \end{cases} \quad (2)$$

Исходными данными для программы являются: схема превращений, т. е. математическая модель реакции, первоначальные мольные доли веществ и экспериментальные данные, полученные в процессе проведения реакции через фиксированные интервалы времени. После ввода исходных данных начинается основная работа программы — поиск скоростных коэффициентов. При этом используются три основных блока:

1) блок, реализующий алгоритм оптимизации прямого поиска; в качестве метода оптимизации использовался метод доверительных интервалов [11, 12];

2) блок решения СОНДУ методом Рунге—Кутта 4-го порядка точности [13];

3) блок сравнения теоретических и экспериментальных данных.

В начале работы оптимизационный алгоритм (ОА) выбирает первоначальный вектор скоростных коэффициентов. Подставив этот вектор в СОНДУ, находится решение, из которого извлекается информация о количественном соотношении выхода веществ. Найденные процентные доли веществ, которые являются теоретическим решением, сравниваются с данными, полученными в процессе эксперимента. Цель ОА заключается в подборе таких скоростных коэффициентов, при которых разность между теоретическими и экспериментальными данными будет удовлетворять некоторой наперед заданной величине. Разница между этими данными оценивается методом наименьших квадратов. Суть этого метода заключается в том, что через ряд экспериментальных точек проводят такую зависимость $F(x, k_0, k_1, \dots, k_n)$, сумма квадратов отклонений которой от эксперимен-

тальных значений при соответствующих значениях k_0, k_1, \dots, k_n минимальна:

$$\zeta = \min \left(\sum_i^m (y_i - F(x_i, [k]))^2 \right),$$

где y_i — ряд экспериментальных точек, определенных для соответствующих интервалов времени x_i ; m — число экспериментальных точек; $[k]$ — вектор скоростных коэффициентов; $F(x_i, [k])$ — функция теоретических значений при определенном наборе k .

Упрощенная блок-схема программы вычисления скоростных коэффициентов представлена на рис. 4.

Функционирование рассматриваемого алгоритма зависит от выбора первоначального вектора k , т. е. от выбора начальной точки. В связи с этим вероятность отказа программы будет рассчитываться в зависимости от выбора начальной точки.

В качестве отказа для данной программы примем событие, при котором метод оптимизации завершил работу, но $\xi > \xi_0$, т. е. когда сумма квадратов отклонений между экспериментальными и теоретическими данными превышает заданную величину ξ_0 . Пусть Q — вероятность отказа, а P — вероятность безотказной работы. Функционально Q зависит от первоначальной точки, т. е. $Q(k_1, k_2, k_3, k_4)$. Метод оценки надежности основывается на моделировании работы программы, когда на вход поступают данные, распределенные по определенному закону. Примем, что случайные величины k_1, k_2, k_3 и k_4 распределены по нормальному закону с различными математическими ожида-

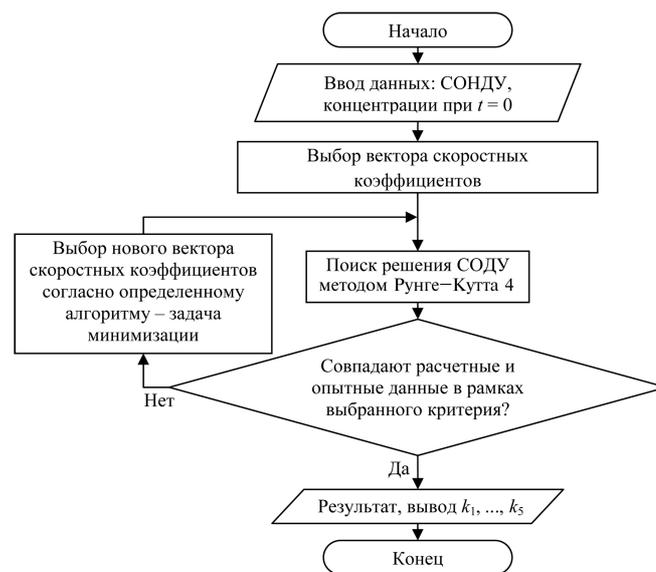


Рис. 4. Блок-схема программы расчета скоростных коэффициентов

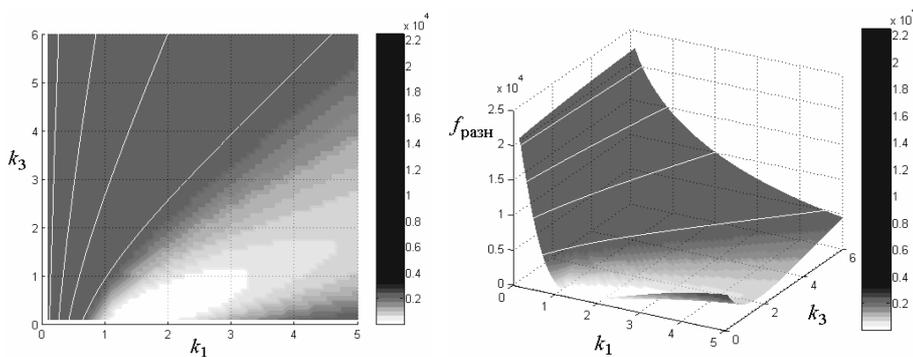


Рис. 6. Функция разницы $f_{\text{разн}}(k_1, k_3)$

ниями, но с постоянным средним квадратическим отклонением (СКО), равным 1.

В результате моделирования для конкретного вектора k , состоящего из математических ожиданий, было определено значение вероятности отказа. Для наглядности зафиксируем математические ожидания k_3 и k_4 , равными 11 и 1 соответственно. В этом случае изменение вероятности безотказной работы программы P в зависимости от МО k_1 и k_2 можно представить графически (рис. 5, см. вторую сторону обложки).

Области, в которых алгоритмический отказ наименее вероятен, обозначены темным цветом. Для обеспечения высокой надежности программы необходимо выбирать МО k_1 и k_2 из интервалов $[1; 6]$ и $[2; 5]$ соответственно.

Теперь предположим, что МО $k_2 = 3$, а МО $k_4 = 1$. Функцию разницы между теоретическими и экспериментальными данными в зависимости от математических ожиданий вектора k обозначим как $f_{\text{разн}}$. Пусть $f_{\text{разн}}(k_1, k_3)$ — функция разницы $f_{\text{разн}}$ от величин k_1 и k_3 ; при этом все остальные элементы вектора k , кроме k_1 и k_3 , фиксированы и не влияют на изменение функции $f_{\text{разн}}$. На рис. 6 представлена функция $f_{\text{разн}}(k_1, k_3)$, в которой не учитывались ограничения (2).

На рис. 7 (см. вторую сторону обложки) показана вероятность безотказной работы в зависимости от МО k_1 и k_3 . При отсутствии ограничений программа решения скоростных коэффициентов находит такой вектор k , при которых $\xi < \xi_0 = 120$. При наложении ограничений значение критерия ξ оказывается больше допустимого, что считается отказом. Программа моделирования находит области, в которых вероятность отказа минимальна. Наиболее допустимые соотношения изображены на графиках рис. 7. При этом с учетом условия $k_1 < k_3$ МО $k_1 \in (1; 5)$ и МО $k_3 > 5$. В этом случае вероятность безотказной работы составит более 0,7.

Заключение

Одним из достоинств метода является его простота, недостатком — то, что полученная оценка надежности ПО учитывает лишь вероятности ожидаемых отказов, оставляя за бортом возможные скрытые дефекты в программе.

При моделировании работы программы в зависимости от распределения исходных данных, были получены области, в которых вероятность безотказ-

ной работы программы для расчета скоростных коэффициентов составляет более 0,7. При отсутствии ограничений (2) можно определить такие области, в которых вероятность безотказной работы программы близка к 1.

Предлагаемый метод позволяет определить как вероятности ожидаемых отказов, так и возможные параметры законов распределения исходных данных, при которых программа будет обладать заданным уровнем надежности.

Список литературы

1. Чересов Г. Н. Надежность аппаратно-программных комплексов: Учеб. пособие. СПб.: Питер, 2005. 479 с.
2. Пальчун Б. П., Юсупов Р. М. Оценка надежности программного обеспечения. СПб.: Наука, 1994. 84 с.
3. Кулямин В. В., Петренко О. Л. Место тестирования среди методов оценки качества ПО // Тр. ИСП РАН. 2003. № 4. С. 163–176.
4. Zeiss B., Vega P., Schieferdecker I., Neukirchen H., Grabowski J. Applying the ISO 9126 Quality Model to Test Specifications — Exemplified for TTCN-3 Test Specifications // Software Engineering 2007. Lecture Notes in Informatics. Bonn. 2007. P. 231–242.
5. ГОСТ 28195—89. Оценка качества программных средств. Общие положения. Государственный комитет СССР по стандартам. М.: Изд-во стандартов, 1989. 38 с.
6. ГОСТ Р ИСО/МЭК 9126—93. Оценка программной продукции. Характеристики качества и руководства по их применению. Госстандарт России. М.: Изд-во стандартов, 1994. 7 с.
7. Вентцель Е. С. Теория вероятностей: Учеб. для вузов. 6-е изд. стер. М.: Высш. шк., 1999. 576 с.
8. Лаврищева Е. М., Рожнов А. М. Концепция аналитической оценки характеристик качества программных компонентов // Пятая междунар. научно-практ. конф. по программированию УкрПрог, 1–3 июня 2004 г., г. Киев, Украина. 2004.
9. Губайдуллин И. И., Фрид А. И. Методика оценки надежности программного обеспечения в зависимости от распределения исходных данных // Естественные и технические науки. 2007. № 3. С. 194–197.
10. Спивак С. И., Губайдуллин И. М., Вайман Е. В. Обратные задачи химической кинетики: Учеб. пособие. Уфа: Изд. БашГУ, 2003. 109 с.
11. Аттетков А. В. и др. Методы оптимизации. М.: Изд. МГУ им. Н. Э. Баумана. 2001.
12. Марчук Г. И. Методы вычислительной математики. М.: Наука, 1989. 608 с.
13. Джонсон К. Численные методы в химии: Пер. с англ. М.: Мир, 1983. 504 с.

УДК 004.932:004.451.353

А. О. Левашкина аспирант,
С. В. Поршнева, д-р техн. наук, зав. каф.,
Уральский государственный
технический университет

Сравнительный анализ супервизорных критериев оценки качества сегментации изображений

Описаны методики сравнительного анализа супервизорных критериев оценки качества сегментации изображений. Приведены результаты сравнительного анализа ряда супервизорных критериев качества сегментации изображений (*FOM*, Хаусдорфа (*Hausdorff distance*), Баддели (*Baddeley's distance*), *Dku*, *RMS*, *RI*, *GCE*, *VI*), полученные с использованием изображений из базы университета Беркли. На их основе сделан обоснованный вывод о том, что для оценки качества алгоритмов сегментации изображений целесообразно использовать критерии *Dku*, *GCE*, *RI*, *RMS*.

Приведена методика сравнения алгоритмов сегментации изображений и результаты сравнения ряда алгоритмов сегментации изображений (*Кенни (Canny)*, *log*, *zerrocross*, *Превита (prewitt)*, *Робертса (roberts)*, *JSEG-алгоритм*).

Ключевые слова: сегментация изображения, оценка качества сегментации, супервизорные критерии.

Введение

Сегментация изображений является фундаментальным процессом в системах компьютерного зрения и приложениях, выполняющих обработку и анализ графической и видеoinформации. Данная процедура используется для разделения изображения на отдельные области, которые соответствуют различным объектам реального мира. Сегментацию следует рассматривать как начальный этап построения формального описания сцены, качество выполнения которого во многом определяет успех решения задачи распознавания изображений, интерпретации и идентификации визуально наблюдаемых объектов.

На сегодняшний день разработано большое число алгоритмов сегментации изображений (АСИ), использующих различные признаки и характеристики изображений. Необходимо отме-

тить, что при исследовании АСИ неизменно возникают следующие проблемы:

1. Проблема выбора наилучшего АСИ, соответствующего классу анализируемого изображения.

2. Проблема нахождения критерия, позволяющего оценивать обоснованность выбора АСИ.

Отметим, что ранее в работах [1–5] уже проводились исследования следующих супервизорных критериев оценки качества сегментации изображений: *FOM* [2], *Hausdorff distance (H)* [2], *Baddeley's distance (B)* [2], *Dku* [2], *M* [2], *V* [2], *RMS* [2], *Rand Index (RI)* [6], *Global Consistency Error (GCE)* [7], *Variation of Information (VI)* [8]. Однако полученные при этом результаты оказываются противоречивыми друг другу. Например, в работах [4, 5] было обнаружено, что у значений критериев *M* и *V*, вычисленных для набора из 100 изображений, наблюдались аномальные выбросы, которые согласно критерию Романовского не являются статистически значимыми. А в работах [2, 9], наоборот, данные критерии рекомендуются к использованию на практике. В работе [1] сделан вывод, что для оценки качества сегментации изображений целесообразно использовать супервизорный критерий *FOM*, а в работе [5] критерий *FOM* оказался отбракованным.

Таким образом, в связи с имеющимися в научной литературе противоречиями разработка методов сравнения качества работы АСИ является актуальной задачей.

В частности, представляет интерес решение следующих задач.

Задача 1. Оценка чувствительности критериев к результатам сегментации, выполняемым различными экспертами.

Задача 2. Оценка соответствия значения супервизорных критериев визуальной оценке результатов сегментации, выставяемой экспертом (при визуальной классификации результатов сегментации).

Задача 3. Оценка соответствия значения супервизорных критериев визуальной оценке, выставяемой экспертами (при формальном оценивании результатов сегментации по числу выделенных объектов).

Цель статьи — описание методики сравнения известных супервизорных критериев и выбор тех из них, которые наиболее целесообразно использовать в дальнейшем для оценки качества сегментации изображений.

1. Сравнение супервизорных критериев оценки качества сегментации

Оценка чувствительности критериев к результатам сегментации, выполняемым различными экспертами. Решение данной задачи необходимо в связи с тем, что сегментация изображений, выполняемая разными экспертами, достаточно субъективна, что иллюстрирует рис. 1.

Для сравнения критериев использовалась методика, реализующаяся следующей последовательностью действий.

1. Выбрать базу данных, содержащую изображения и результаты их сегментации, выполненные визуально различными экспертами (объекты идеальной формы (ОИФ)).

2. Сформировать набор изображений

$$U = \{U^1, U^2, \dots, U^K\}, k = \overline{1, K},$$

где K — число выбранных изображений, и соответствующие им множества ОИФ $\{ОИФ\}_j^{(k)}$, выделенные экспертами, $\{ОИФ\}_j^{(k)} = \{ОИФ_1^k, ОИФ_2^k, \dots, ОИФ\}_j^{(k)}, j = \overline{1, J}, J$ — число множеств ОИФ.

3. Сравнить попарно ОИФ, входящие в набор $\{ОИФ\}_j^{(k)}$ для каждого изображения; вычислить значения выбранных супервизорных критериев (результат выполнения этого шага — набор $same_true_d^{(k)}, d = \overline{1, D}, D$ — число определяемых критериев).

4. Сравнить попарно ОИФ, входящие в наборы $\{ОИФ\}_j^{(k)}$ разных изображений; вычислить значения выбранных супервизорных критериев (результат выполнения этого шага — набор $diff_true_d^{(k)}$).

5. Выполнить сегментацию выбранных изображений ($U^{(k)}$) с помощью одного из АСИ:

$$S^{(1)}, S^{(2)}, \dots, S^{(i)}, \dots, S^{(k)}.$$

6. Сравнить попарно результаты сегментации ($S^{(k)}$) (результат выполнения этого шага — набор $diff_seg_d^{(k)}$).

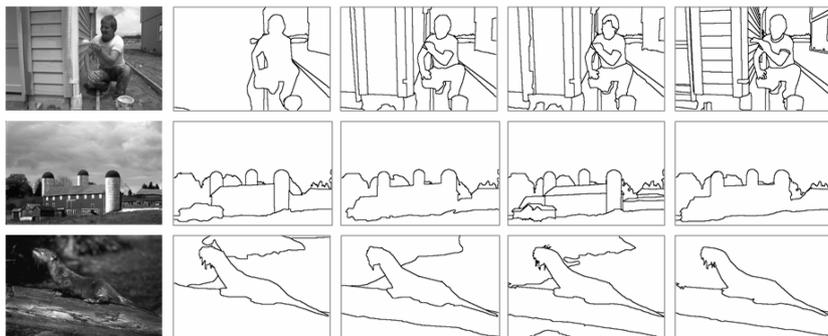


Рис. 1. Примеры сегментаций изображений из базы университета Беркли, выполненных человеком (ОИФ)

7. Оценить статистическую значимость различий между $same_true_d^{(k)}, diff_true_d^{(k)}, diff_seg_d^{(k)}$ с помощью непараметрических критериев.

Для проверки статистической значимости различий при сравнении двух не связанных выборок были использованы непараметрические критерии, которые не предполагают предварительного вычисления параметров распределения. Эти критерии сопоставляют не сами по себе полученные величины, а порядок их расположения, их соотношение по типу "больше — меньше". В качестве таковых были использованы следующие непараметрические критерии:

- Н-критерий Крускала—Уоллиса (*Kruskal—Wallis test*) [14, 15];
- медианный критерий (*Median test*) [15];
- тест Колмогорова—Смирнова [16].

Очевидно, что критерий оценки качества сегментации изображений оказывается чувствительным к результатам сегментации изображения, выполненной различными экспертами, когда различия между случайными выборками $same_true$ и $diff_true$, $same_true$ и $diff_JSEG$ оказываются статистически значимыми.

Оказалось, что для большинства рассматриваемых критериев оценки качества сегментации изображений отличия являются статистически значимыми (за исключением критерия VI). Следовательно, данные критерии оказываются достаточно чувствительными к результатам сегментации, выполненной разными экспертами.

Оценка соответствия значения супервизорных критериев визуальной оценке результатов сегментации, выставяемой экспертом. Здесь использовалась такая последовательность действий.

1. Выбрать базу данных, содержащую изображения и результаты их сегментации, выполненные визуально различными экспертами ОИФ.

2. Сформировать набор изображений

$$U = \{U^1, U^2, \dots, U^K\}, k = \overline{1, K},$$

где K — число выбранных изображений, и соответствующие им множества ОИФ $\{ОИФ\}_j^{(k)}$, выделенные экспертами

$$\{ОИФ\}_j^{(k)} = [ОИФ_1^k, ОИФ_2^k, \dots, ОИФ_p^{(k)}], j = \overline{1, J},$$

J — число множеств ОИФ.

3. Выполнить сегментацию выбранных изображений $U^{(k)}$ с помощью используемых АСИ

$$S_1^{(1)}, S_1^{(2)}, \dots, S_p^{(k)},$$



Рис. 2. Примеры трех выделенных классов результатов сегментации

где S — результат сегментации (набор объектов неидеальной формы (ОИФ)); p — номер используемого алгоритма ($p = \overline{1, P}$); P — число выбранных алгоритмов.

4. Выделить по результатам визуального анализа результатов сегментации $S_p^{(k)}$ человеком три класса результатов: недосегментация (*UNDER*); пересегментация (*OVER*); компромиссная сегментация (*COMPROMISE*) (соответствующие примеры представлены на рис. 2):

- $UNDER_f^{(k)}$ — результат сегментации, на котором визуально, по мнению человека, выделено меньше сегментов, чем на ОИФ;
- $OVER_f^{(k)}$ — результат сегментации, на котором визуально, по мнению человека, выделено больше сегментов, чем на ОИФ;
- $COMPROMISE_f^{(k)}$ — результат сегментации, который визуально, по мнению человека, похож на ОИФ;

где $f = \overline{1, F}$ — число результатов сегментации в каждом классе.

5. Сравнить $UNDER_f^{(k)}$ и $ОИФ_j^{(k)}$ (j — выбирается случайным образом). Вычислить значения критериев (результатом выполнения действий, описанных в данном пункте, является таблица с элементами $C_UNDER_ОИФ_f^{(k)}$).

6. Сравнить $OVER_f^{(k)}$ и $ОИФ_j^{(k)}$ (j — выбирается случайным образом). Вычислить значения критериев (результатом выполнения действий, описанных в данном пункте, является таблица с элементами $C_OVER_ОИФ_f^{(k)}$).

7. Сравнить $COMPROMISE_f^{(k)}$ и $ОИФ_j^{(k)}$ (j — выбирается случайным образом). Вычислить значения критериев (результатом выполнения действий, описанных в данном пункте, является таблица с элементами $C_COMPROMISE_ОИФ_f^{(k)}$);

8. Проверить значимость отличий распределений случайных величин

$$C_UNDER_ОИФ_f^{(k)},$$

$$C_OVER_ОИФ_f^{(k)},$$

$C_COMPROMISE_ОИФ_f^{(k)}$ с помощью непараметрических критериев.

Для критерия, значения которого соответствуют визуальной оценке, данной человеком, различия между группами *UNDER*, *OVER*, *COMPROMISE* должны

быть статистически значимыми.

Оказалось, что для большинства рассматриваемых критериев оценки качества результатов сегментации изображений (за исключением критерия *FOM*) отличия являются статистически значимыми. Следовательно, данные критерии соответствуют визуальной оценке результатов сегментации, выставяемой экспертом, при условии, что результаты сегментации классифицируются визуально.

Оценка соответствия значения супервизорных критериев визуальной оценке, выставяемой экспертами.

1. Сформировать базу данных, содержащую $ОИФ^1$ (одно изображение) и искусственно сгенерированные результаты сегментации, соответствующие трем классам результатов сегментации (пример на рис. 3):

- $UNDER_f$ — результат сегментации, на котором число сегментов меньше, чем на ОИФ;
- $OVER_f$ — результат сегментации, на котором число сегментов больше, чем на ОИФ;
- $COMPROMISE_f$ — результат сегментации, на котором число сегментов равно числу сегментов на ОИФ.

Здесь $f = \overline{1, F}$, F — число результатов сегментации в каждом классе.

2. Сравнить $UNDER_f$ и $ОИФ^1$. Вычислить значения критериев (результатом выполнения действий



Рис. 3. Результаты сегментации

вий, описанных в данном пункте, является таблица с элементами $C_UNDER_ОИФ_f^1$.

3. Сравнить $OVER_f$ и $ОИФ^1$. Вычислить значения критериев (результатом выполнения действий, описанных в данном пункте, является таблица с элементами $C_OVER_ОИФ_f^1$).

4. Сравнить $COMPROMISE_f$ и $ОИФ^1$. Вычислить значения критериев (результатом выполнения действий, описанных в данном пункте, является таблица с элементами $C_COMPROMISE_ОИФ_f^1$).

5. Проверить значимость отличий распределений случайных величин $C_UNDER_ОИФ_f^1$, $C_OVER_ОИФ_f^1$, $C_COMPROMISE_ОИФ_f^1$ с помощью непараметрических критериев.

Для критерия, значения которого соответствуют визуальной оценке, данной человеком, различия между группами $UNDER$, $OVER$, $COMPROMISE$ должны быть статистически значимыми.

Оказалось, что для большинства критериев (за исключением критерия $B(1)$) отличия являются статистически значимыми. Следовательно, данные критерии соответствуют визуальной оценке, выставляемой экспертами, при условии, что результаты сегментации оцениваются формально по числу выделенных объектов.

Анализ результатов сравнения критериев качества сегментации изображения. Из всех рассмотренных в работе критериев можно выделить следующие четыре критерия (Dku , GCE , RI , RMS), обнаруживающих следующие свойства:

- чувствительность к результатам сегментации, выполняемым различными экспертами;
- соответствие значений данных критериев визуальной оценке результатов сегментации, выставляемой экспертом при визуальной классификации результатов сегментации;
- соответствие значений данных критериев визуальной оценке, выставляемой экспертами при формальном оценивании результатов сегментации по числу выделенных объектов.

Таким образом, критерии качества сегментации изображения Dku , GCE , RI , RMS целесообразно использовать в дальнейшем для сравнительного анализа АСИ.

2. Сравнение алгоритмов сегментации изображений

Обоснованный выбор супервизорных критериев качества сегментации изображений позволяет провести количественные сравнения АСИ. В качестве таковых нами выбраны наиболее популярные на сегодняшний день алгоритмы сегментации: Кенни (*Canny*), *log*, *zerrocross*, Превитта (*pre-*

witt), Робертса (*roberts*) и *JSEG*-алгоритм. Напомним, что данные алгоритмы (за исключением *JSEG* [10, 11]) осуществляют сегментирование изображений на основе выделения краев. При этом границы изображений, выделяемые операторами Кенни, *log*, Превитта, Робертса, *zerrocross* [12], в большинстве случаев оказываются разрывными. Данное обстоятельство определяет необходимость их дополнительной обработки, позволяющей связать найденные фрагменты границ. В нашей работе для замыкания границ мы использовали морфологические операции замыкания областей [12], которые реализованы в виде соответствующих функций в пакете MATLAB (далее используется обозначение МСК — морфологическое связывание контуров). Параметры данных функций выбирали эмпирически на основе визуального анализа результатов их применения.

Для сравнения результатов работы АСИ была использована методика, реализующаяся следующей последовательностью действий.

1. Выбрать базу данных, содержащую изображения и результаты их сегментации, выполненные визуальными экспертами (ОИФ).

2. Сформировать случайным образом множество тестовых изображений

$$U = \{U^1, U^2, \dots, U^i, \dots, U^k\},$$

где $k = \overline{1, K}$, K — число выбранных изображений, и соответствующих им множеств ОИФ $\{ОИФ_j^{(k)}\}$, выделенных экспертами:

$$\{ОИФ_j^{(k)}\} = \{ОИФ_1^k, ОИФ_2^k, \dots, ОИФ_J^k\},$$

$$j = \overline{1, J},$$

J — число множеств ОИФ.

3. Сегментировать изображения, входящие в множество U , используя каждый из выбранных АСИ, получив набор множеств

$$S_1^{(1)}, S_1^{(2)}, \dots, S_p^{(i)}, \dots, S_m^{(k)}, p = \overline{1, P},$$

где $S_p^{(i)}$ — результат сегментации (набор объектов неидеальной формы (ОНФ)) i -го изображения; P — число выбранных АСИ.

4. Сформировать случайным образом из набора множеств $\{ОИФ_j^{(k)}\}$ базовый набор множеств $\{ОИФ_m^{(i)}\}$, $m = \overline{1, M}$, M — число множеств, включенных в базовый набор.

5. Вычислить для каждого i , используя наборы множеств $S_p^{(i)}$, $\{ОИФ_m^{(i)}\}$, множество значений выбранных супервизорных критериев

$$\{C_{l,p,m}^{(i)}\}, l = \overline{1, L},$$

Средние значения критериев *RMS* и *Dku* для сравниваемых алгоритмов

Название алгоритма		<i>RMS</i> ↓			<i>Dku</i> ↓		
1	<i>canny</i> + МСК	0,35025	±	0,00002	0,03757080	±	0,0000008
2	<i>JSEG</i>	0,26465	±	0,00002	0,0217450	±	0,0000005
3	<i>log</i> + МСК	0,30140	±	0,00001	0,0278376	±	0,0000004
4	<i>prewitt</i> + МСК	0,233283	±	0,000009	0,0166160	±	0,0000002
5	<i>roberts</i> + МСК	0,208712	±	0,000007	0,0133232	±	0,0000001
6	<i>sobel</i> + МСК	0,23274	±	0,00001	0,0166279	±	0,0000002
7	<i>zerrocross</i> + МСК	0,30185	±	0,00001	0,0279201	±	0,0000004

Таблица 2

Средние значения критериев *RI* и *GCE* для сравниваемых алгоритмов

Название алгоритма		<i>RI</i> ↑			<i>GCE</i> ↓		
1	<i>canny</i> + МСК	0,78369	±	0,00002	0,045679	±	0,000005
2	<i>JSEG</i>	0,86696	±	0,00002	0,044182	±	0,000003
3	<i>log</i> + МСК	0,83316	±	0,00001	0,045999	±	0,000005
4	<i>prewitt</i> + МСК	0,896169	±	0,000005	0,043885	±	0,000002
5	<i>roberts</i> + МСК	0,915137	±	0,000003	0,039929	±	0,000001
6	<i>sobel</i> + МСК	0,895874	±	0,000006	0,043286	±	0,000003
7	<i>zerrocross</i> + МСК	0,83309	±	0,00001	0,045512	±	0,000004

где L — число выбранных супервизорных критериев.

6. Вычислить для каждого p -го АСИ и l -го критерия качества сегментации изображения средние значения и дисперсию супервизорных критериев

$$\bar{C}_{p,d} = \frac{1}{M \cdot I} \sum_{i=1}^I \sum_{m=1}^M C_{l,p,m}^{(i)}$$

Сравнение АСИ выполнялось на основе изображений и ОИФ базы университета Беркли [13], которая характеризуется следующими параметрами:

1. Общее число изображений, содержащихся в базе, равняется 300. При этом создатели базы заявляют, что 100 изображений из 300 предназначено для настройки супервизорных алгоритмов сегментации, 200 — для сравнения АСИ. Поскольку мы сравниваем несупервизорные АСИ, то сравнение выполнялось с использованием всего набора изображений.

2. Общее число результатов сегментации, выполненных экспертами (ОИФ), для каждого изображения составляет от 5 до 9.

Для получения статистически значимых результатов сравнения АСИ, в соответствии с описанной выше методикой, мы провели 50 независимых испытаний, в которых использовали 100 изображений.

Средние значения критериев *Dku*, *GCE*, *RI*, *RMS* представлены в табл. 1, 2.

Из табл. 1, 2 видно, что согласно четырем выбранным критериям лучшие результаты у оператора Робертса (*roberts*) с применением дополнительной обработки по связыванию контуров.

Заключение

В статье описаны методики сравнительного анализа супервизорных критериев оценки качества сегментации изображений и результаты количественного сравнения критериев *FOM*, Хаусдорфа (*Hausdorff distance*), Баддели (*Baddeley's distance*), *Dku*, *RMS*, *RI*, *GCE*, *VI* и результаты с использованием 300 изображений из базы университета Беркли. При этом установлено, что критерии качества АИС *Dku*, *GCE*, *RI*, *RMS*:

- позволяют различать результаты сегментации, выполненной различными экспертами;
- значения данных критериев соответствуют визуальной оценке результатов сегментации, выставляемой экспертом;
- значения данных критериев соответствуют визуальной оценке, выставляемой экспертами, при формальном оценивании результатов сегментации по числу выделенных объектов.

Сделан обоснованный вывод о целесообразности использования критериев *Dku*, *GCE*, *RI*, *RMS* для оценки качества алгоритмов сегментации изображений.

Предложена методика количественного сравнения алгоритмов сегментации изображений. В соответствии с предложенной методикой проведено сравнение ряда АСИ (Кенни (*Canny*), *log*, *zerrocross*, Превитта (*prewitt*), Робертса (*roberts*) и *JSEG*-алгоритм) и выбран среди них наилучший (*roberts*).

Список литературы

1. Chabrier S., Laurent H., Emile B., Rosenberger C., Marché P. A comparative study of supervised evaluation criteria for image segmentation // EUSIPCO, Vienne, 2004. P. 1143–1146.

2. **Zhang Y.-J.** Advances in Image and Video Segmentation. Idea Group Inc. (IGI), 2006. 457 p.

3. **Chabrier S.** Contribution à l'évaluation de performances en segmentation d'images // Thèse ou HDR soutenue, ENSI de Bourges, 2005. [электронный ресурс: режим доступа 03.03.2009] http://www.ensi-bourges.fr/LVR/SIV/These_Sebastien_Chabrier.pdf

4. **Левашкина А. О.** Методика сравнения супервизорных критериев оценки качества сегментации изображений // Научные труды Международной научно-практической конференции "СВЯЗЬ-ПРОМ 2008", Екатеринбург: Изд-во УГТУ-УПИ, 2008. С. 63—65.

5. **Левашкина А. О.** Сравнительный анализ супервизорных критериев оценки качества сегментации изображений // Научные труды международной научно-практической конференции "СВЯЗЬ-ПРОМ 2008". Екатеринбург: Изд-во УГТУ-УПИ, 2008. С. 66—68.

6. **Rand Index**, Wikipedia, free encyclopedia. [электронный ресурс: режим доступа 19.03.2008] http://en.wikipedia.org/wiki/Rand_index.

7. **Unnikrishnan R., Pantofaru C., Hebert M.** Toward objective evaluation of image segmentation algorithms // IEEE Transactions on Pattern Analysis and Machine Intelligence. June 2007. Vol. 29. N 6.

8. **Rosenberg A., Hirschberg J.** V-measure: A conditional entropy-based external cluster evaluation measure // Proc. of the 2007 Joint Conference on Empirical Methods in Natural Languages Processing and Computational Natural Language Learning. Prague, June 2007. P. 410—420.

9. **Peli T., Malah D.** A study of edge detection algorithms // Computer Graphics and Image Processing. 1982. N 20. P. 1—21.

10. **Deng Y., Manjunath B. S., Shin H.** Color image segmentation // Proc. IEEE Computer society conference on computer vision and pattern recognition, CVPR '99, Fort Collins, CO. June 1999. Vol. 2. P. 446—451.

11. **JPEG** free source code [электронный ресурс: режим доступа 17.03.2008] <http://vision.ece.ucsb.edu/segmentation/iseg/software/>

12. **Гонсалес Р., Вудс Р.** Цифровая обработка изображений. М.: Техносфера, 2006. 1072 с.

13. **Berkeley Segmentation Dataset** — база изображений университета Беркли [электронный ресурс: режим доступа 17.03.2008] <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench>

14. **Будрейка Н. Н.** Использование непараметрических критериев проверки статистических гипотез [электронный ресурс: режим доступа 16.04.2008] <http://www.matlab.mgppu.ru/work/0014.htm>

15. **Непараметрические** статистические критерии (nonparametric statistical tests) [электронный ресурс: режим доступа 16.04.2008] <http://www.sexualdysfunction.ru/583.html>

16. **Техническая** библиотека lib.qrz.ru. Тест Колмогорова—Смирнова [электронный ресурс: режим доступа 16.04.2008] <http://lib.qrz.ru/node/11253>

УДК 004.932.4

А. А. Масалович, аспирант,
МГУ им. М. В. Ломоносова,
e-mail: anton_m@abbyu.com

Численные методы детектирования и удаления геометрических искажений в изображениях текстовых документов

Описан метод аппроксимации деформации текстового документа кубическим патчем Безье, основанный на непрерывном гранично-скелетном представлении текста. Рассмотрены основные шаги алгоритма: построение непрерывного гранично-скелетного представления изображения, преобразование скелета, выделение в скелете ветвей, лежащих между строками текста, итерационный процесс построения патча Безье на основе аппроксимаций межстрочных ветвей, распрямление искривленного изображения.

Ключевые слова: распознавание текста, преобразование изображений, распрямление строк, непрерывное гранично-скелетное представление, внешний скелет, патч Безье.

Введение

Большинство современных профессиональных систем распознавания текста рассчитаны на то, что строки текста на изображении будут прямыми

и горизонтальными. Поэтому в последнее время очень большое внимание уделяется методам, позволяющим удалять геометрические искажения с изображений документов. Однако, несмотря на возросший интерес к этой области и наличие большого числа новых методов устранения геометрических искажений, универсального решения этой проблемы так до сих пор не было найдено.

В данной статье будет предложен новый метод для детектирования удаления геометрических искажений с изображения текста, основанный на аппроксимации межстрочных просветов в тексте, полученной в результате анализа внешнего гранично-скелетного представления изображения. Представлены обобщение и развитие идей метода, описанных ранее в ряде докладов [1—5].

Постановка задачи и методы ее решения

Математическая постановка задачи

Дано: Изображение I в виде цветовой функции на координатной плоскости и область D исходного изображения:

$$I = \{C : R^2 \rightarrow \{0; 1\}; \\ D = [0; Width] \times [0; Height] \subseteq R^2\}.$$

Некая функция оценки горизонтальности строк на изображении $K(I): I \rightarrow R$.

Задача. Построить двумерную непрерывную функцию преобразования изображения $\Phi(x, y)$, такую, что на преобразованном изображении I

функция оценки была бы существенно выше, чем на исходном изображении:

$$\Phi(x, y) : R^2 \rightarrow R^2;$$

$$\bar{I} = \{ \bar{C}(x, y) = C(\Phi_x(x, y), \Phi_y(x, y)); \bar{D} = \Phi(D) \};$$

$$K(\bar{I}) \gg K(I).$$

Оценка результата. К сожалению, невозможно точно оценить степень распрямления изображения текстового документа. Происходит это вследствие того, что нет точного способа определить базовую линию строки текста. Поэтому для оценки работы алгоритма мы будем использовать отношение правильно распознанных символов к общему числу всех символов на странице до и после распрямления строк текста. Данный критерий весьма эффективен, так как кроме интегральной оценки самого алгоритма позволяет также сравнивать различные системы удаления геометрических искажений между собой.

Предположения о формате исходного изображения. На вход разрабатываемого алгоритма подается изображение текстового документа. Логично сделать несколько предположений о формате обрабатываемого изображения: изображение должно быть черно-белым (черный текст на белом фоне); на изображении не должно быть ненужных черных объектов (разметки, разделителей, мусора, иллюстраций); текст на изображении должен быть расположен в одну колонку; в тексте на изображении должно быть как минимум несколько строчек.

Все современные системы распознавания документов могут привести изображение документа к требуемому виду. На рис. 1 приведен типичный пример искривленного изображения, предназначенного для распрямления.

Существующие решения. Существует достаточно много подходов [9—18] к решению задачи удаления геометрических искажений с изображения текста. Большинство существующих методов исправления геометрических искажений страницы проводят работу в два этапа — нахождение искривления базовых линий строк и построение искривляющей функции на основе данных искажений. У такого подхода есть несколько недостатков.

Во-первых, определение искажения по базовым линиям строк не очень стабильно, так как для этого требуется уметь объединять соседние слова в строки. В данной работе предлагается работать не со строками, а с межстрочными просветами, построенными на ребрах внешнего скелета изображения.

Во-вторых, в существующих методах предлагаются или очень жесткие модели на искажение страницы или, наоборот, методы построения аппроксимации искажения страницы, чувствительные к локальным неточностям. В данной работе предлагается аппроксимация искажений патчем Безье, который, с одной стороны, не накладывает жестких ограничений на модель искажения стра-

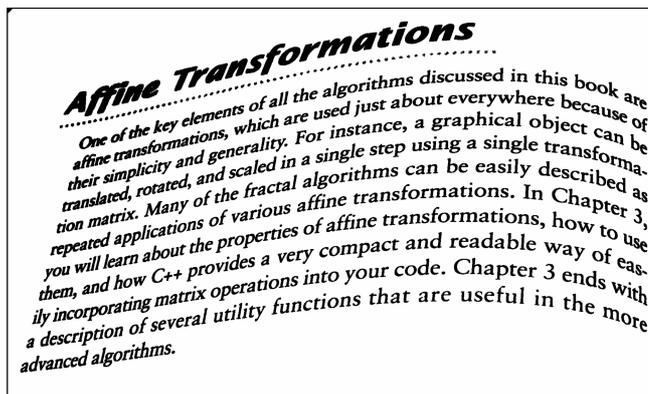


Рис. 1. Пример обрабатываемого изображения

ницы, а с другой стороны, исправляет локальные ошибки в определении искривления отдельных межстрочных просветов.

Определение межстрочных просветов на изображении

Непрерывное гранично-скелетное представление изображения

Для решения поставленной задачи воспользуемся непрерывным гранично-скелетным представлением изображения [7, 8].

Скелетом многоугольной фигуры на плоскости называется такое множество точек, что для каждой точки имеется не менее двух ближайших точек на границе фигуры.

Непрерывное гранично-скелетное представление изображения — это совокупность многоугольников минимального периметра, разделяющих все черные и белые точки изображения и скелета, построенного на этой совокупности многоугольников (рис. 2, см. третью сторону обложки).

Для решения задачи распрямления строчек нас будет интересовать часть скелета изображения, лежащая в белой фоновой области изображения. Эту часть скелета будем называть *внешним скелетом изображения*.

Скелет каждой многоугольной фигуры состоит из линий, которые представляют собой сегменты прямых и парабол. Скелет можно рассматривать как плоский граф, у которого ребрами являются сами фрагменты прямых и парабол, а вершинами — концевые точки этих сегментов, т. е. точки соединения сегментов друг с другом. В дальнейшем при работе со скелетом будет всегда использоваться его представление в виде графа.

Предобработка изображения и скелета текста

Для алгоритма аппроксимации геометрических искажений на изображении нужны только ребра скелета, лежащие между объектами в различных

строках текста. Для ускорения фильтрации ненужных элементов скелета применяется предобработка изображения и скелета.

Предобработка изображения. Каждое слово текста на изображении лучше обрабатывать как единый объект, чтобы не работать с ребрами скелета изображения, лежащими между двумя соседними буквами в слове. Для объединения соседних символов в слова мы заливаем короткие белые горизонтальные пропуски на изображении. Иными словами, если расстояние между двумя черными точками в одной строке пикселей меньше предопределенного порога, то заливаем черным цветом этот белый промежуток. Максимальное расстояние между пикселями для заливки определяется по разрешению исходного изображения. Данная обработка позволяет избавиться от множества ненужных для разрабатываемого алгоритма ветвей скелета (рис. 3).

Стоит отметить, что предобработка делается на изображении только для построения скелета. Впоследствии во время удаления геометрических искажений используется оригинальное изображение, а не изображение с залитыми строками.

Предобработка скелета. Стоит отметить, что для потребностей алгоритма аппроксимации геометрических искажений изображения нужен только внешний скелет изображения, поэтому внутренний скелет изображения просто не строит.

После того как скелет построен, часть ребер в нем можно удалить еще до работы основного алгоритма аппроксимации межстрочных искажений. Ненужные для разрабатываемого алгоритма ребра скелета — это ребра, лежащие между различными частями одного объекта. Стандартный процесс построения скелета позволяет запоминать в процессе работы, с какими объектами связано каждое ребро скелета. Поэтому удаление ребер, относящихся только к одному объекту, является тривиальной задачей.

В результате фильтрации получается скелет, содержащий только существенные для алгоритма детекции искажений ребра (рис. 4).

Выделение и аппроксимация искажения межстрочных пролетов в тексте

Кластеризация ребер скелета. Для метода аппроксимации геометрических искажений изображения нужно кластеризовать ребра скелета на ребра, лежащие между объектами в одной строке, и ребра, лежащие между объектами в разных строках.

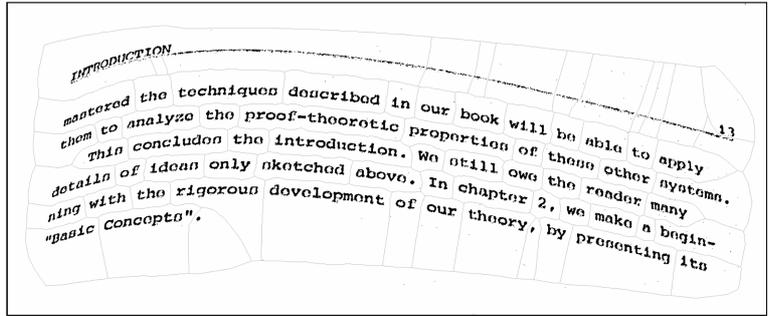
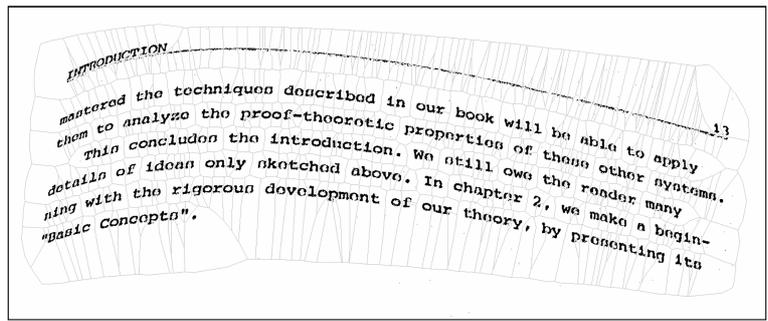


Рис. 3. Скелет, построенный на обычном и обработанном изображении

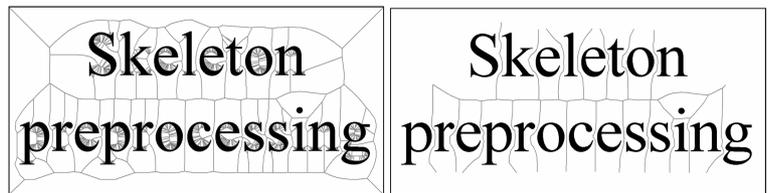


Рис. 4. Скелет изображения до и после предобработки

Для кластеризации ветвей сначала исключим из рассмотрения ветви скелета, которые короче некоего предопределенного порога. В качестве порога длины ветвей возьмем величину, зависящую от разрешения изображения. После исключения коротких ветвей скелета построим линейную аппроксимацию оставшихся ветвей. Отметим на гистограмме углы наклона всех линейных аппроксимаций (рис. 5).

После построения гистограммы определяется порог, разделяющий два кластера. Для каждого возможного значения порогового угла (с шагом в один градус) определяется слева и справа среднее значение гистограммы (M_R и M_L) и дисперсия значений гистограммы (D_R и D_L). Степень дисперсии кластеров для данного порога определяется как отношение суммы дисперсий к расстоянию между средними значениями:

$$Disp = (D_R + D_L)/(M_R - M_L).$$

Порог, разделяющий кластеры, выбирается как порог с наименьшим значением степени дисперсии кластеров.

После того как порог выбран, из скелета удаляются все длинные ветви, угол наклона которых больше порогового значения. Эксперименты по-

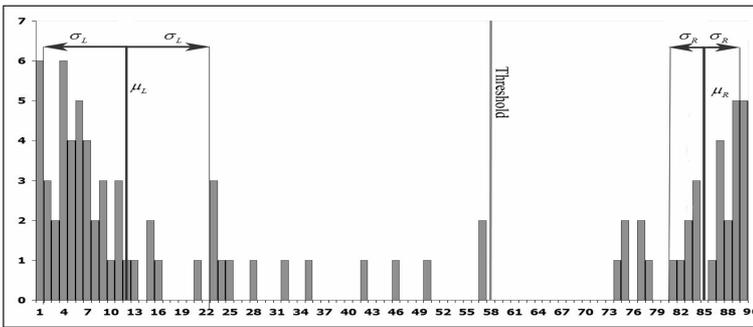


Рис. 5. Пример гистограммы углов наклона ветвей скелета с определенным порогом

казали, что описанный выше выбор порога позволяет удалить из скелета большинство ребер, которые лежат между объектами в одной строке (рис. 6, см. третью сторону обложки).

Разрешение неоднозначностей в скелете. После того как из скелета были удалены все вертикальные ребра, все ребра оставшиеся в скелете — это ребра, лежащие между различными строками текста, и некоторое число мелких мусорных ребер. В идеальном случае хочется, чтобы между соседними строками текста находилась одна длинная ветвь скелета. Однако сразу после фильтрации длинных ребер это не так, потому что в скелете присутствуют мелкие ненужные ребра. Существует набор простых эмпирических правил, которые после применения к скелету позволяют разрешить в нем все возможные неоднозначности (подробнее см. [1, 2]).

После того как в скелете останутся только длинные ветви, лежащие в межстрочных промежутках, мы аппроксимируем каждую ветвь кривой Безье третьего порядка.

Аппроксимация геометрических искажений на изображении

Построение двумерного патча Безье

Постановка задачи. После выделения и аппроксимации межстрочных ветвей в скелете бинарного изображения текста нужно построить аппроксимацию искажения всей страницы. Для этого используется аппроксимация искажения двумерным патчем Безье.

Двумерный патч Безье — это функция $V: [0, 1] \times [0, 1] \rightarrow R^2$, задаваемая выражением

$$V_{nm}(t, u) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} b_{i,n}(t) b_{j,m}(u),$$

где P_{ij} — функция компонент векторов опорных вершин; $b_{k,l}(x)$ — полиномы Бернштейна,

$$b_{k,l}(x) = \binom{l}{k} x^k (1-x)^{l-k}, \quad \binom{l}{k} = \frac{l!}{k!(l-k)!}.$$

Размерность патча по двум координатам может различаться. Если в патче зафиксировать один из параметров, то функция по второму параметру будет кривой Безье с

контрольными точками $\left(\sum_{j=0}^m P_{ij} b_{j,m}(u_0) \right)$:

$$B_n(t) = B_{nm}(t, u_0) = \sum_{i=0}^n \left(\sum_{j=0}^m P_{ij} b_{j,m}(u_0) \right) b_{i,n}(t)$$

Метод аппроксимации набора кривых патчем Безье. Идея метода аппроксимации заключается в том, что можно минимизировать не разность между исходными кривыми и соответствующими кривыми из патча, а разность между контрольными точками этих кривых.

Задачу аппроксимации контрольных точек набора кривых патчем Безье будем решать следующим образом.

Есть набор кривых $\left\{ B^k(t) = \sum_{i=0}^n Q_i^k b_{i,n}(t) \right\}, k = 1 \dots K$.

Рассмотрим набор контрольных точек кривых с одинаковым порядковым номером — $\{Q_i^k\}$. Поставим в соответствии каждой точке из набора то же значение параметра u , что соответствует исходной кривой. Тогда мы можем построить кривую Безье, аппроксимирующую данный набор точек:

$$\bar{B}_i(u) = \sum_{j=0}^m P_{ij} b_{j,m}(u); \quad \sum_{k=1}^K (\rho(\bar{B}_i(u_k), Q_i^k))^2 \rightarrow \min.$$

Для каждого порядкового номера контрольных точек исходных кривых (от 0 до n) получается набор из m контрольных точек аппроксимирующей кривой. Таким образом, всего получается набор из $m \cdot n$ контрольных точек. Эти точки и используются для построения патча Безье (рис. 7, см. третью сторону обложки).

Метод итерационной подгонки патча Безье.

При построении патча Безье используется итерационный метод подгонки. После построения патча для каждой исходной кривой определяется расстояние от данной кривой до соответствующей ей кривой из патча. Если это расстояние больше некоего предопределенного порога, то данная кривая удаляется из набора кривых для построения патча.

После удаления всех неподходящих кривых для оставшихся кривых Безье уточняется их параметризация. После этого по отфильтрованному набору кривых с новыми определенными параметрами строится новая аппроксимация патчем Безье.

Процесс итерационной подгонки продолжается до тех пор, пока максимальное отклонение между исходными кривыми и построенным патчем

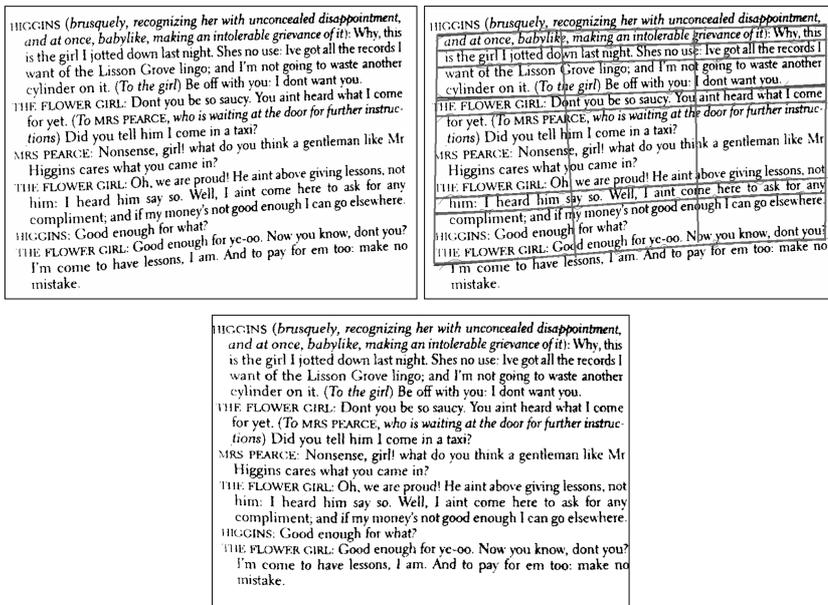


Рис. 9. Распрямленное изображение

Безье не станет меньше предопределенного порога (рис. 8, см. третью сторону обложки).

Результаты экспериментов

В 2007 году описанный алгоритм принимал участие в соревновании алгоритмов по удалению геометрических искажений с изображений документов (*Document Image Dewarping Contest*), проходящем в рамках Второго Международного симпозиума по анализу и распознаванию документов с фотокамер (*Second International Workshop on Camera-Based Document Analysis and Recognition — CBDAR-2007*) [6].

Для проведения соревнования организаторами была создана тестовая база из 100 искривленных черно-белых изображений. В качестве оценки качества тестируемых алгоритмов было использовано качество распознавания документов после распрямления. В соревновании участвовало три различных метода. Результаты соревнования показали, что все три метода работают с хорошими и статистически равными результатами.

На рис. 9 показаны результаты применения описанного алгоритма к конкретному изображению.

Заключение

В данной работе описан метод удаления геометрических искажений с изображений текстовых документов.

Для аппроксимации искажений документа использована аппроксимация межстрочных просветов текста. В качестве основного инструмента для аппроксимации межстрочных просветов используется непрерывное гранично-скелетное представле-

ние изображения. В работе описан алгоритм построения аппроксимирующего патча на основе набора кривых Безье, аппроксимирующих межстрочные искажения. Также описан процесс итерационной подгонки построенного патча.

Был проведен публичный независимый эксперимент для оценки качества предлагаемого метода. Эксперименты показали высокую надежность предлагаемого метода.

Список литературы

1. Masalovitch A., Mestetskiy L. Usage of continuous skeletal representation for document images de-warping // Proc. of the Second International Workshop on Camera-Based Document Analysis and Recognition (CBDAR-2007). Curitiba, Brazil, 2007. P. 45—52.
2. Masalovitch A., Mesteckiy L. Document image deformation approximated by the means of continuous skeletal representation of the image // Proc. of the Ninth International conference on pattern recognition and information processing (PRIP-2007). Minsk, Belarus, 2007. P. 279—284.
3. Масалович А. А., Местецкий Л. М. Использование патча Безье для аппроксимации искажения текстовых документов // Труды 17-й Международной конференции по компьютерной графике и зрению (Графикон-2007), Москва, 2007. С. 239—243.
4. Masalovitch A., Mesteckiy L. Usage of 2-dimensional Bezier Patch for document images deformation approximation // Proc. of the Eighth International conference on pattern recognition and images analysis (PRIA-8-2007). Vol. 3. Yoshkar-Ola, 2007. P. 21—26.
5. Масалович А. А., Местецкий Л. М. Распрямление текстовых строк на основе непрерывного гранично-скелетного представления изображений // Труды 16-й Международной конференции по компьютерной графике и зрению (Графикон-2006), Новосибирск, 2006.
6. Shafait F., Breuel T. Document image dewarping contest // Proc. of the Second International Workshop on Camera-Based Document Analysis and Recognition (CBDAR-2007), Curitiba, Brazil, 2007. P. 181—189.
7. Местецкий Л. М. Скелет многосвязной многоугольной фигуры // Труды 15-й Международной конференции по компьютерной графике и зрению (Графикон-2005), Новосибирск, 2005.
8. Местецкий Л. М. Непрерывный скелет бинарного растрового изображения // Труды Международной конференции по компьютерной графике и зрению (Графикон-98). Новосибирск, 1998.
9. Xu-Cheng Yin, Jun Sun, Satoshi Naoi. Perspective rectification for mobile phone camera-based documents using a hybrid approach to vanishing point detection // Proc. of the Second International workshop on camera-based document analysis and recognition (CBDAR-2007). Curitiba, Brazil, 2007. P. 37—44.
10. Schneider P., Block M., Rojas R. Robust document warping with interpolated vector fields // Proc. of the Ninth International conference on document analysis and recognition (ICDAR-2007). Curitiba, Brazil, 2007. P. 113—117.
11. Gatos B., Pratikakis I., Ntirogiannis K. Segmentation based recovery of arbitrarily warped document images // Proc. of the Ninth International conference on document analysis and recognition (ICDAR-2007). Curitiba, Brazil, 2007. P. 989—993.
12. Bin Fu, Minghui Wu, Rongfeng Li, Wenxin Li, Zhuoqun Xu, Chunxu Yang. A model-based book dewarping method using text line detection // Proceedings of the Second International Workshop on

Camera-Based Document Analysis and Recognition (CBDAR-2007). Curitiba, Brazil, 2007. P. 63–70.

13. **Minghui Wu, Rongfeng Li, Bin Fu, Wenxin Li, Zhuoqun Xu.** A model-based book dewarping method to handle 2D images captured by a digital camera // Proc. of the Ninth International Conference on Document Analysis and Recognition (ICDAR-2007). Curitiba, Brazil, 2007. P. 158–162.

14. **Ezaki H., Uchida S., Asano A., Sakoe H.** Dewarping of document images by global optimization // Proc. of the Eighth International conference on document analysis and recognition (ICDAR-2005). Seoul, South Korea, 2005. P. 302–306.

15. **Ulges U., Lampert C., Breuel T.** A Fast and stable approach for restoration of warped document images // Proc. of the Eighth In-

ternational conference on document analysis and recognition (ICDAR-2005). Seoul, South Korea, 2005. P. 384–388.

16. **Li Zhang, Chew Lim Tan.** Warped image restoration with application to digital libraries // Proc. of the Eighth International conference on document analysis and recognition (ICDAR-2005). Seoul, South Korea, 2005. P. 192–196.

17. **Yamashita A., Kawarago A., Kaneko T., Muira K. T.** Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system // Proc. of international conference ICPR, 2004. P. 482–485.

18. **Brown M. S., Seales W. B.** Image restoration of arbitrarily warped documents // IEEE Transactions on pattern analysis and machine intelligence, Vol. 26. Issue 10. 2004. P. 1295–1306.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 550.84:551.49

О. Е. Архипова, канд. техн. наук, ст. науч. сотр.,
Южный научный центр РАН,
Ростов-на-Дону

Концепция региональной эколого-информационной системы мониторинга

Описаны основные принципы создания системы многоуровневого регионально-адаптированного комплексного мониторинга. В качестве модельного объекта рассматриваются морские и наземные экосистемы, где представлены разнообразные водные и земельные комплексы и виды водопользования, а водные ресурсы выступают одним из лимитирующих факторов социально-экономического развития. Результат исследования, проводимого в рамках Программы фундаментальных исследований Отделения наук о Земле РАН, позволит определить пути стабилизации экологической ситуации, выработать подходы и дать рекомендации по устойчивой эксплуатации ресурсов в зонах с экстремальными природными условиями.

Ключевые слова: геоинформационные технологии, геоинформационные системы, базы данных, базы знаний, база геоданных, система экологического мониторинга.

Введение

Экономическое развитие России в XXI веке и обеспечение высокого качества жизни на всей ее территории во многом зависят от естественных и антропогенных факторов жизнедеятельности. На территориях и в морских бассейнах с длительной историей хозяйственного освоения отмечается сильная деградация естественных экосистем. Для каждой природной зоны характерен свой комплекс проблем, поэтому необходима разработка соответствующей стратегии региона.

При экспериментальном исследовании экосистем основная тенденция в мировой практике заключается в их комплексном анализе и исследовании иерархической организации экосистем. Появление эффективных методов экспериментального изучения экосистем привело к стремительному накоплению экспериментальных данных, что делает невозможным комплексную интеграцию в рамках частных подходов. В настоящее время в мировом сообществе информационные технологии играют все более значимую и критическую роль. Они формируют базовую инфраструктуру исследований, обеспечивают эффективный междисциплинарный диалог представителей различных научных областей, изучающих состояние и динамику экосистем. Особо важны подобные разработки на региональных уровнях, когда группы исследователей должны осуществлять совместную работу, обмен данными и знаниями. Создание единого информационного пространства предоставит качественно новые возможности анализа состояния экосистем. В мире существуют технологии обработки пространственно-распределенных данных, решающие такую задачу, что служит важнейшей мотивацией проведения исследований по этому направлению. Наконец, развитие технологий для пространственного моделирования взаимодействия человека и биосферы позволит решать задачи прогноза развития деструктивных процессов в экосистемах [1].

Геоинформационные технологии полезны там, где данные привязаны к точке с пространственными координатами. Данные анализа состояния водного объекта всегда связаны с точкой отбора проб. В продвинутых пакетах программного обеспечения геоинформационных технологий имеются даже специальные приложения для поддержки обработки первичных данных отбора проб. Дополнительным доводом в пользу геоинформационных технологий является получившая совсем недавно распространение возможность использо-

вания сети Интернет для представления результатов работы с данными.

Результат исследования, проводимого в рамках Программы фундаментальных исследований Отделения наук о Земле РАН "Развитие технологий мониторинга, экосистемное моделирование и прогнозирование при изучении природных ресурсов в условиях аридного климата", позволит определить пути стабилизации экологической ситуации, выработать подходы и дать рекомендации по устойчивой эксплуатации ресурсов в зонах с экстремальными природными условиями.

Разрабатываемая система должна обеспечивать выполнение широкого круга запросов о текущем состоянии окружающей среды, решение оперативных задач информационного обеспечения при анализе временной динамики состояния морских и наземных экосистем.

Основные требования к созданию региональной экологической информационной системы

Общие требования к созданию крупных экологических систем мониторинга сводятся к необходимости использования распределенных баз данных, охватывающих информацию по различным отраслям знаний, геоинформационных технологий для отображения пространственно-распределенных объектов и использования методов пространственного моделирования, методов математического моделирования, основанных на анализе предметных областей, и Интернет-технологий для организации доступа к распределенным ресурсам.

В сложившейся практике мониторинга гарантированно присутствуют только этапы получения, обработки и анализа данных. Как правило, эти этапы выполняются по стандартным технологиям. Формирование баз данных при проведении мониторинга не всегда обязательно, но проведение наблюдений практически всегда сопровождается их архивацией. Обычно разработчики ориентируются на усвоение всей доступной информации и оценивают результаты своей работы по количеству накопленных данных. Современные концепции мониторинга отдают приоритет реально действующим системам наблюдений. Центральным звеном системы мониторинга является автоматизированная экспертно-информационная система, в которой обобщаются все материалы наблюдений прошлых лет, непрерывно пополняются и анализируются данные о текущем состоянии экосистем, в том числе с использованием телеметрических каналов сбора информации, отрабатываются результаты математического моделирования различных гидрофизических и гидрометеорологических процессов [5].

Существенными отличительными чертами, определяющими архитектуру и реализацию подобной системы, являются:

- системный анализ;
- наличие моделей различного уровня;
- зависимость выбора подходов и моделей от целей и задач исследования;
- необходимость работы с динамическими базами пространственно-распределенных данных, в том числе данных мониторинга с учетом их при моделировании.

Из сказанного следуют определенные требования к реализации подобной интегрированной системы. Во-первых, это наличие нескольких уровней выполнения проекта (от прототипа до действующей системы). Во-вторых, это широкое использование новых информационных технологий. При разработке системы необходимо учитывать следующие требования:

- система должна обладать модульностью и гибкостью для развития и совершенствования;
- она должна разрабатываться как открытая и наращиваемая система с максимальным использованием стандартных и общепринятых подходов и средств реализации.

Для реализации целей мониторинга предложен проект стандарта системы регионально-адаптированного экологического информационного мониторинга (РЭИС), которая включает системы наблюдений по различным тематическим областям. Целью формирования РЭИС является использование фактической информации в сфере изучения, воспроизводства, рационального использования и охраны природных ресурсов, охраны окружающей среды для решения практических задач.

В ходе разработки должны быть использованы:

- базы данных (БД);
- географические информационные системы;
- современные компьютерные модели;
- экспертные системы;
- системы дешифрования и распознавания данных дистанционного зондирования (ДДЗ).

РЭИС должна обеспечивать решение множества задач:

- подготовки интегрированной информации о состоянии окружающей среды, прогнозов вероятных последствий хозяйственной деятельности и рекомендаций по выбору вариантов безопасного развития региона;
- имитационного моделирования процессов, происходящих в окружающей среде, накопление информации по временным трендам параметров этой среды;
- подготовки электронных карт, отражающих состояние окружающей среды региона;
- обработки и накопления в базах данных результатов локального и дистанционного мони-

торинга, выявления параметров окружающей среды, наиболее чувствительных к антропогенным воздействиям;

- обмена информацией о состоянии окружающей среды (импорта и экспорта данных) с другими экоинформационными системами.

Структура РЭИС

На основе исследования отечественного и зарубежного опыта разработки экологических систем мониторинга была предложена общая схема построения системы, представленная на рисунке. В состав РЭИС входят следующие основные модули: геоинформационная система (ГИС), банк

моделей, модуль обработки данных дистанционного зондирования (ДДЗ), автоматизированные рабочие места (АРМ) для ведения баз данных водных и наземных экосистем, база знаний.

Основа интегрированной системы мониторинга — база данных, в состав которой входят база данных предметных областей и база геоинформационных данных, в том числе банк региональных цифровых карт.

ГИС выполняет следующие функции:

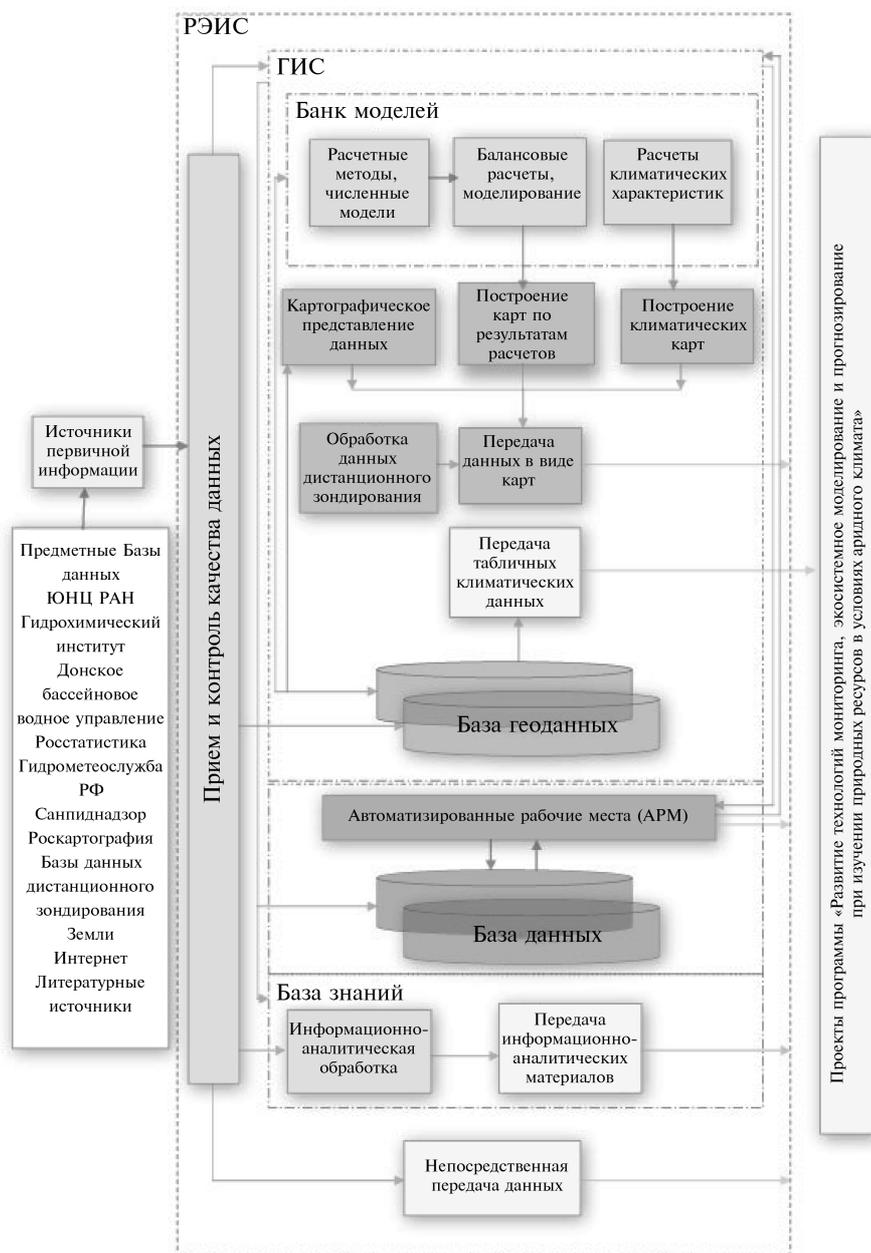
- определяет общую структуру справочников для различных инструментальных средств и моделей;
- поставляет распределенные данные к моделям;
- помогает в визуализации пространственной модели в виде тематических карт;
- подготавливает картографический материал для дальнейшего анализа в блоке экспертных оценок и передает информацию пользователю системы.

Средствами ГИС обеспечивается интерфейс пользователя.

В состав банка моделей входят расчетные методы и численные модели, балансовые модели, расчеты климатических характеристик.

Большинство ГИС-приложений в моделировании окружающей среды используют ГИС для предварительной обработки данных, для создания карты из входных данных или для визуализации результата моделирования. Возможен другой подход, когда модель формируется в пределах ГИС, в этом случае числовые возможности модели ограничены рамками географической информационной системы. В концепции интегрированной среды использован синтез обоих подходов. Модули, непосредственно работающие с пространственным представлением объектов, включены в состав ГИС.

В целях ведения мониторинга влияния антропогенного воздействия на экосистему региона и оценки комплексных показателей экосистем и ряда физических параметров среды в РЭИС введен блок обработки данных дистанционного зондирования. С помощью этого блока проводится дешифрова-



Структура региональной экологической информационной системы (РЭИС)

ние космических снимков, что позволяет выделить на исследуемой территории отдельные слои, такие как водная поверхность, антропогенная поверхность, открытая почва и классифицированная растительность.

В состав автоматизированных рабочих мест входят АРМ наземных экосистем и АРМ водных экосистем. Это управляющие программы, выполняющие функции получения, визуализации и анализа данных. Основные требования к интерфейсу АРМ — обеспечение быстрой навигации и доступа к интересующим данным, сортировка отображаемых данных, поиск данных, наличие вспомогательной информации.

База знаний представлена оценками экспертов по различным отраслям знаний, которые используются для информационно-аналитических отчетов РЭИС.

Структура и принципы организации основной базы данных РЭИС. Одно из необходимых условий функционирования системы — информационное наполнение базы данных. В ходе выполнения проекта разработана схема основной базы данных, т. е. структура содержащихся в ней данных и отношений между ними. Основная база данных по мониторингу строится на основе данных, поступающих из различных источников, — это базы данных сторонних организаций, занимающихся регулярным или эпизодическим контролем содержания загрязняющих веществ и других показателей, данные сети Интернет, телеметрические данные, а также собственная база данных Южного научного центра РАН, полученная в результате проведения экспедиционных исследований. Информационная среда системы разработана на основе однопользовательской СУБД Microsoft Access. Она обеспечивает объединение и использование распределенной информации, а ГИС-технология — ее обработку в соответствии с географической или административной привязкой.

Информационная среда состоит из базы моделей природных и техногенных объектов, баз данных результатов контроля и анализа, а также нормативной базы.

Структура и принципы организации базы геоданных. База геоданных (БГД) системы использует модель данных на основе топологически связанных классов пространственных объектов. База геоданных определяет обобщенную модель данных для географической информации. Через сеть Интернет осуществляется связь БГД с мировыми центрами информации (сайты meteo.ru, meteo.infospace.ru, ocean.info).

Полученные данные заносятся в БГД, выполняются обработка и анализ полученной информации. В основе БГД лежит привязка к карте пространственно-распределенных объектов инфор-

мации из объектно-ориентированных баз данных различного уровня и основной базы данных системы мониторинга.

Примером таких данных может служить информация:

- о водопользователях разного уровня, таких как промышленные и сельскохозяйственные предприятия, городские и сельские предприятия коммунального водоснабжения, жители определенных территорий и др.;
- о результатах мониторинговых и экспедиционных наблюдений за погодой и климатом, водным и гидрохимическим режимом рек и водоемов;
- о статистических показателях развития отдельных административно-хозяйственных единиц (предприятия, территории).

БГД объединяет возможность легкого доступа к информации с возможностью выбора (с помощью специальных запросов) и обобщения всей имеющейся информации по временным интервалам, водосборным территориям и участкам рек.

Формирование банка цифровых карт. Для современного программного обеспечения ГИС характерен отказ от хранения данных в файловых системах в пользу полной интеграции с традиционными хранилищами баз данных. Поскольку базовое программное обеспечение ГИС ведущих разработчиков предоставляет практически равные функциональные возможности, выбор конкретной платформы ГИС становится вторичным моментом при реализации проекта и определяется в первую очередь наличием комплексных решений и опытом разработок в данной области. Наличие лицензионного программного обеспечения ArcGis Desktop у исполнителя и большой опыт работы в создании картографических баз данных с использованием модулей семейства ArcGis обусловили выбор в качестве основного программного продукта ArcGis Desktop.

Объединение электронно-картографических материалов и тематических баз данных позволяет делать качественные оценки экологического риска любой хозяйственной деятельности, а в применении к наземным территориям и морским акваториям является незаменимым инструментом повышения качества экологического мониторинга и контроля состояния окружающей среды.

Формируемый банк региональных цифровых карт обеспечивает решение как задач отображения плана местности, так и задач привязки атрибутивной информации именно к пространственным объектам.

Послойный принцип организации информации, принятый в ГИС, очень нагляден и хорошо соотносится с приемами традиционной картографии. Он заключается в том, что выбирается некоторое разбиение объектов на тематические слои, и

объекты, отнесенные к одному слою, образуют некоторую логически (а часто и физически) отдельную единицу данных. Выбор способа организации данных в ГИС, в первую очередь модели данных, очень важен, поскольку напрямую определяет многие функциональные возможности системы. От выбранной модели зависит пространственная точность представления графической части информации, возможность получения качественного картографического материала и организации контроля качества карт.

В значительной степени способом организации данных в ГИС определяется также быстродействие системы. Картографическая информация, вводимая в систему, проходит качественную обработку.

К настоящему времени разработаны оригинальная технология и алгоритмы интеграции картографической информации в ГИС, созданы программы, обеспечивающие наполнение ГИС картографической информацией, решена задача переноса картографических данных в форматы наиболее распространенных ГИС.

Принципы взаимодействия основных модулей системы. После контроля и проверки первичных данных информация поступает в базу данных интегрированной системы. Интегрированное приложение рассматривается как отдельная среда, предлагающая расширенный набор функций. В этом приложении ГИС используется как начальная и конечная система и включает в себя дополнительные характеристики моделирования. Этот выбор также обусловлен тем фактом, что большинство моделей имеют довольно слабый пользовательский интерфейс, хотя и обладают мощными вычислительными возможностями. Кроме того, они имеют мало встроенных возможностей для пространственного размещения данных. Интегрировать эти две среды позволяет архитектура клиент—сервер, в которой ГИС выступает как клиент, а модель — как сервер [2, 3].

Средствами ГИС обеспечиваются: дружелюбный интерфейс пользователя, предварительная обработка входной информации для модели на основе пространственного представления об объектах, выбор численной модели и передача расчетной функции банку моделей. Взаимодействие между ГИС и АРМ также осуществляется на уровне передачи данных и клиент—серверной технологии.

Специальные модули АРМ готовят базу исходных данных в форме, необходимой для связи с пространственными объектами ГИС, ГИС служит средством визуализации, предоставляющим функции пространственного анализа информации, а также средством обработки и представления результатов анализа в форме тематических карт.

Таким образом, интегрированная среда — это объединение основных компонентов — банка моделей, СУБД, ГИС и АРМ.

Прототип РЭИС. Создание интегрированной системы предполагает несколько уровней выполнения проекта (от прототипа до действующей системы). Принципы согласования отдельных модулей были определены в ходе построения прототипа системы. Разработка прототипа предполагает отработку основных принципов функционирования системы на ограниченной области исследования. В качестве такой области были выбраны часть бассейна Азовского моря, включающая регион Ростовской области, как пример наземной экосистемы и экосистемы водных ресурсов и морская экосистема Азовского моря [3].

В качестве прототипов рабочих мест были рассмотрены АРМ "Суша" на примере бассейна реки Маныч и Нижнего Дона и АРМ "Морские системы" на примере Азовского моря. Эти рабочие места опираются на общие принципы функционирования и решают общие задачи, однако существенно различаются составом основной базы данных.

Была осуществлена пространственная привязка информации основной базы данных РЭИС по наблюдениям, проводимым во время рейсов, к объектам мониторинга на электронной карте. Данные мониторинга распределения солености и температуры в Азовском море были использованы для построения полей распределения температур и солености Азовского моря методами пространственного анализа ГИС, а также как исходные данные для построения модельного распределения полей солености и температур.

ГИС "Азовское море" является рабочим местом исследователя, позволяет интерактивно взаимодействовать с АРМ "Морские системы", получая необходимую информацию по станциям. Методы ГИС позволяют оперативно проводить необходимую выборку из базы данных, тем самым существенно сокращая время транзакций с базой данных. Использование методов пространственного и геостатистического анализа натуральных измерений позволяет оперативно обрабатывать данные и предоставлять их для дальнейшего анализа состояния экосистемы [4].

Рабочее место пользователя РЭИС предусматривает доступ к определенным модулям системы с помощью АРМ и ГИС. Организация РЭИС предполагает наличие нескольких рабочих мест в рамках различных групп исследователей, работающих с общей базой данных и решающих определенный круг задач.

Был сделан выбор централизованной базы данных, располагаемой на сервере. Основным средством управления основной базой данных систе-

мы определен MySQL. Общая база данных поддерживается на уровне СУБД MySQL с организацией связи с рабочими местами посредством Microsoft Access. Основная база данных для работы ГИС также располагается на сервере. При разработке прототипа был определен принцип взаимодействия основной базы данных, созданной на основе мониторинга и базы геоданных. На сервере баз данных хранятся базы данных предметных областей РЭИС и атрибутивные базы данных ГИС.

База данных ГИС и предметные базы данных на сервере доступны только для просмотра информации, отображения пространственно-распределенных данных на карте. Для решения пользовательских задач каждый пользователь имеет собственную базу геоданных на локальном компьютере, где хранит результаты анализа и собственные данные. Доступ к предметной базе данных возможен с локальных рабочих мест через соответствующие АРМ.

Определен состав основных функций, доступных в ГИС, для обработки данных мониторинговых наблюдений. Это, в первую очередь, функции пространственного анализа (построение поверхностей, изолиний, статистический анализ).

Через АРМ доступны функции просмотра информации по рейсам, статистический анализ данных и экспорт данных в пользовательский формат. В настоящее время в рамках прототипа РЭИС "Азовское море" созданы следующие АРМ:

- оператора БД;
- оператора БГД;
- специалиста по обработке телеметрических данных;
- администраторов БД и ГИС;
- исследователей.

Каждое из этих рабочих мест имеет свои функциональные особенности [1].

Заключение

В целом концептуальные положения создания региональной экологической информационной системы в соответствии с общей методологией сводятся к следующему.

- ◆ РЭИС представляет собой интегрированную среду, в состав которой входит геоинформационная система, модуль дешифровки ДДЗ, система моделей, автоматизированные рабочие места и база знаний.
- ◆ Основу базы данных РЭИС составляют тематические базы данных мониторинга водных и наземных экосистем региона.
- ◆ Одним из критериев правильности функционирования РЭИС является достоверность базы данных. Как следствие, одна из основных подсистем РЭИС — система контроля и проверки достоверности данных.
- ◆ Для целей наблюдения за влиянием антропогенного воздействия на окружающую среду введен модуль дешифрирования и обработки космических снимков и данных аэрофотосъемки.
- ◆ В качестве прототипа РЭИС организованы два рабочих места АРМ "Морские системы" и "Системы суши".

Список литературы

1. **Архипова О. Е., Бердников С. В., Дашкевич Л. В., Дженюк С. Л., Кулыгин В. В.** Новые информационные технологии в задачах экосистемного мониторинга Азовского моря // Экосистемные исследования Азовского, Черного, Каспийского морей и их побережий. Т. IX. Апатиты: КНЦ РАН, 2007. С. 14—22.
2. **Архипова О. Е.** Информационно-аналитическая среда оценки качества поверхностных вод речного бассейна // Известия высших учебных заведений. Северо-Кавказский регион. Естественные науки. Приложение. 2006. № 6. С. 3—12.
3. **Архипова О. Е.** Информационно-аналитическая среда оценки качества поверхностных вод речного бассейна // Автореф. дис., 2006. С. 1—24.
4. **Архипова О. Е., Дашкевич Л. В., Кулыгин В. В.** Принципы организации океанографической базы данных // Экология. Экономика. Информатика. XXXIV школа-семинар "Математическое моделирование в проблемах рационального природопользования" (11—16 сентября 2006 г.) Ростов-на-Дону: СКНЦ ВШ, 2006. С. 129—133.
5. **Дженюк С. Л.** Преобразованные экосистемы и требования к их комплексному мониторингу // Современные климатические и экосистемные процессы в уязвимых природных зонах (арктических, аридных, горных): Тез. докл. междунар. науч. конф. (г. Азов, 5—8 сентября 2006 г.). Ростов-на-Дону: ЮНЦ РАН, 2006. С. 66—68.

С 27 по 30 октября 2009 г. в г. Москва (ВВЦ) пройдет
20-я юбилейная выставка информационных технологий и компьютеров,
смотр достижений российских ученых-разработчиков программного обеспечения

ИНФОРМАЦИЯ

Softool-2009

Выставка Softool является самым крупным и представительным в России форумом, на котором ведущие производители программных и аппаратных средств вычислительной техники демонстрируют новейшие разработки в области информационных технологий и их применения в экономике страны.

В рамках выставки пройдет Всероссийская конференция "*Информационные технологии в России*". На ней вы сможете услышать ведущих специалистов, узнать о новейших разработках и последних тенденциях в разных областях информационных технологий.

Новые разделы выставки:

- Технологии электронного государства
 - Региональная информатизация
 - Свободное ПО
- Тел./факс: (495) 624-7070, 624-4556, 621-0659
e-mail: softool@softool.ru

К. И. Бушмелева, канд. физ.-мат. наук, доц.,
И. И. Плюснин, директор,
 Центр лазерных технологий,
 Сургутский государственный университет

Автоматизированная геоинформационная система мониторинга технического состояния магистральных газопроводов

Рассмотрены основные особенности геоинформационной системы мониторинга состояния магистральных газопроводов, разработанной на базе авиационного программно-аппаратного диагностического комплекса и используемой для диагностирования дефектов и обнаружения утечки газа из магистральных газопроводов.

Ключевые слова: мониторинг, магистральный газопровод, лазерный локатор утечки газа, программно-аппаратный диагностический комплекс, геоинформационная система, программное обеспечение.

Введение

Одним из направлений работ Центра лазерных технологий Сургутского государственного университета является создание авиационных лазерных программно-аппаратных комплексов и систем для мониторинга технического состояния и обнаружения утечки транспортируемого газа (метана) по магистральным газопроводам (МГ).

Длительная эксплуатация газопроводных систем предъявляет повышенные требования к контролю их технического состояния, поэтому проблема обнаружения утечки газа из МГ является весьма актуальной. В силу различных причин утечки метана происходят в разных местах газовых магистралей, и чем раньше удастся их обнаружить и ликвидировать, тем к меньшим потерям и последствиям это приведет. Периодический контроль и освидетельствование состояния газопроводов дают возможность продлевать ресурс их эксплуатации. При этом важной задачей при проведении обследования и обнаружении утечек газа из газопроводов является выбор методов исследования, позволяющих комплексно решать поставленные задачи.

В настоящее время существуют различные устройства для обнаружения утечек газа из газопроводов, которые отличаются принципом действия, чувствительностью, областью применения, размерами и т. д. [1–5]. Однако, как показывает

практика их эксплуатации, для работы на газопроводах наилучшими по своим характеристикам являются лазерные локаторы, размещаемые на вертолетах, самолетах или других летательных аппаратах.

Данные средства с помощью дополнительных приборов обеспечивают дистанционный поиск, точную локализацию места утечки метана и идентификацию технологических дефектов на МГ.

За истекшие пять лет в России было создано большое число технических средств, которые обеспечивают поиск и идентификацию дефектов на МГ с бортов летательных аппаратов. Тем не менее, эффективность их использования невысока из-за отсутствия удобных и универсальных средств постобработки результатов обследования.

В Сургутском государственном университете в последние годы ведутся работы по созданию и усовершенствованию комплексной системы дистанционного лазерного зондирования МГ с борта летательного аппарата. Изготовлено несколько экземпляров лазерного локатора утечек газа (ЛУГ), имеется патент на данную разработку [2]. ЛУГ является одним из основных элементов программно-аппаратного диагностического комплекса (ПАДК) (рис. 1), предназначенного для оперативного дистанционного обнаружения утечек метана из магистральных газопроводов ОАО "Газпром" с определением координат утечки и состояния трассы МГ [3].

Предлагаемая в данной работе геоинформационная система (ГИС) мониторинга технического состояния МГ также разработана на базе ЛУГ [1] и способна решать следующий круг задач: ввод, накопление, хранение, редактирование и оперативная обработка данных по обследованию газопроводов; формирование отчетов по результатам летных испытаний для последующего детального анализа полученной информации; поиск по базе данных результатов обследований за определенный период; привязка полученных данных к соответствующей электронной карте ГИС.



Рис. 1. Внешний вид ПАДК в рабочем состоянии

Программно-аппаратный диагностический комплекс

ПАДК планируется включить в штатный состав оборудования вертолета (Ми-8, Ка-226АГ или любого другого летательного аппарата) и использовать его для патрулирования, в целях осуществления мониторинга магистральных газопроводов.

Основными компонентами ПАДК являются:

- локатор утечек газа;
- система обработки и регистрации данных;
- система видеонаблюдения;
- набор специального программного обеспечения;
- электронная карта полетов;
- база данных ГИС;
- автоматизированное рабочее место оператора ЛУГ.

Локатор предназначен для обнаружения утечек метана из МГ и других объектов газотранспортной системы [4].

Функционально и конструктивно в ЛУГ объединены следующие блоки.

1. Передающая оптическая система (ПОС), состоящая из двух He—Ne-лазерных излучателей, генерирующих излучение на длинах волн $\lambda_1 = 3,3922$ мкм и $\lambda_2 = 3,3912$ мкм с сильным и слабым поглощением в метане соответственно. Непрерывное излучение лазеров пропускается поочередно магнитооптическими модуляторами (МОМ 1 и МОМ 2), установленными на выходах лазеров, которыми, в свою очередь, управляют устройства управления (УУ). С помощью системы зеркал ПОС излучение лазеров смешивается и в виде одного луча направляется в область предполагаемой утечки метана.

2. Приемная система (ПС), состоящая из телескопа, в котором смонтированы оптические узлы — главное параболическое зеркало и защитное кварцевое стекло, размещенное в фокусе телескопа. На стекле закреплен охлаждаемый фотоприемник с предварительным усилителем и системой отвода теплоты.

Смешанные поочередно лучи лазеров, отраженные от подстилающей поверхности вблизи МГ, попадают на фотодетектор (ФД) телескопа и преобразуются в электрический сигнал. После усиления электрические сигналы поступают на УУ.

Блоки ПОС, ПС и УУ составляют замкнутую систему автоматического управления по отклонению, где объектом управления является совокупность лазеров, МОМ и системы зеркал, формирующая регулируемый по значению совмещенный лазерный луч.

При пересечении смешанным лазерным лучом облака газа на выходе ФД формируется сигнал рассогласования, пропорциональный разности

поглощения излучений лазеров 1 и 2 в газе. Сигнал рассогласования детектируется и поступает для измерения и обработки в ЭВМ.

Система обработки и регистрации данных (СОРД) предназначена для обработки электрических сигналов, поступающих на аналого-цифровой преобразователь (АЦП), по определенной программе, отображения ее на экране и записи в память ЭВМ.

Также в СОРД входит спутниковая навигационная система (СНС) в виде приемника GPS, подключенного к ЭВМ. Эта система обеспечивает определение координат мест утечки газа в ходе проведения работ по заданному маршруту (треку). Трек отображается на электронной карте, выводимой на экран ЭВМ, и запоминается.

Таким образом, СОРД обеспечивает привязку сигналов локатора к конкретным координатам на земной поверхности, регистрирует поступающую информацию и осуществляет ее предварительную обработку.

ЭВМ, входящая в ПАДК, обрабатывает результаты исследований, выдавая их на экран дисплея (в режиме "Самописец") в виде непрерывной осциллограммы. При этом оператор, работающий с локатором, имеет возможность в режиме реального времени судить о результатах обследования. Обработанная информация параллельно записывается в память ЭВМ и служит бортовым протоколом проводимого обследования.

В систему видеонаблюдения (СВ) ПАДК входит цифровой фотоаппарат *Canon EOS 300*, который по командам из ЭВМ фотографирует изображение мест утечки.

В составе ПАДК работает специальное программное обеспечение (ПО) (см. таблицу), пред-

ПО, используемое в ПАДК

Наименование ПО	Назначение ПО
Kashmir 3D, ArcView 3.2, MapInfo Professional 8.5; HandyScope 3; Cannon RemoteCapture	Ввод информации с оборудования локатора: от GPS-приемника; с АЦП; с цифровой фотокамеры
Adobe Photoshop CS, Adobe ImageReady CS	Обработка снимков с цифровой фотокамеры и растровых изображений
L-Graph, HandyScope 3, ACTest, MathCAD 2000, MatLAB 6.5	Обработка данных с АЦП
ArcView 3.2, MapInfo Professional 8.5, Garmin MapSource Kashmir 3D	Обработка картографической информации
Crystal Reports и Microsoft Power Point	Обработка данных, записанных GPS-приемником
ПО, входящее в комплект поставки отдельных приборов локатора	Создание полуавтоматических отчетов и презентаций
	Преобразование данных, полученных с локатора

ставляющее собой комплекс программных продуктов, используемых для решения задач, связанных с работой локатора.

Электронная карта полетов представляет собой интерактивную ГИС, в состав которой входит банк графических данных, содержащих цифровые карты, фотоизображения и GPS-треки маршрута передвижения летательного аппарата.

База данных (БД) ГИС содержит хронологическую информацию о состоянии исследуемых объектов, привязанных к электронной карте.

Автоматизированное рабочее место (АРМ) оператора ЛУГ с поддержкой геоинформационных технологий [5] предназначено для получения оперативной информации об обстановке на МГ и включает в себя созданные элементы АРМ, разработанное программное обеспечение обработки данных из локатора совместно с фотоизображениями участков МГ и созданную компьютерную карту коридора трассы.

"АРМ оператора ЛУГ"

"АРМ оператора ЛУГ" представляет собой автоматизированную ГИС мониторинга технического состояния МГ, способную решать следующий ряд задач:

- ввод, накопление, хранение и редактирование информации о полученных результатах обследований МГ;
- экспорт данных, собранных с оборудования ПАДК;
- контроль корректности полученных результатов;
- формирование промежуточных отчетов по результатам летных обследований для последующей их обработки;
- ведение необходимых справочников;
- поиск по базе данных результатов испытаний за определенный период;
- привязка полученных данных к соответствующей электронной карте ГИС.

Разработка программного обеспечения данной ГИС велась методом расширения ядра *MapInfo Professional*, поддерживающего механизмы связи с внешними, вновь создаваемыми модулями, и предоставления им определенных сервисов. Внешние модули, в свою очередь, взаимодействуют с ядром посредством предложенных интерфейсов. Процессы обработки информации ведутся средствами комплекса специализированного программного обеспечения ЛУГ и компонентами ПАДК.

Общий вид организационно-функциональной структуры ГИС представлен на рис. 2.

Входная информация представлена массивом данных различного формата, собранных при облете с помощью оборудования ПАДК и являю-



Рис. 2. Организационно-функциональная структура "АРМ оператора ЛУГ"

щихся основными источниками для разработки информационного обеспечения системы.

Данные GPS — массив данных, собранных с GPS-приемника: время (в миллисекундах), координаты (широта — долгота, в градусах, минутах, секундах или долях градусов), высота (в метрах), скорость полета (в метрах в секунду).

Данные АЦП — массив данных, собранных АЦП локатора: время (в миллисекундах), амплитуда сигнала (в милливольтках).

Фотоданные — массив фотографий, созданных фотоаппаратом, подключенным к локатору. При этом обеспечивается точность по времени не более 1 мс, размер изображения не менее 40×60 м, разрешение 8...10 Мпикселей.

Для взаимодействия прикладных программ создана БД, которая разработана с помощью программного средства *Microsoft SQL Server 2000*. Для доступа к БД использовалась технология ADO. Схема базы данных содержит пять взаимосвязанных таблиц:

- таблица "Измерения" содержит информацию о полученных результатах измерений после облетов участков местности. Сбор информации ведется при получении соответствующих данных с оборудования ПАДК;
- таблица "Карты местности" содержит информацию о полученных результатах в виде карт местности по конкретному номеру маршрута;
- таблица "Категории" содержит описания всех категорий утечек, в том числе по мощности утечки;
- таблица "Участки" содержит информацию об участках, по которым были совершены облеты, с сохранением координат GPS, данных о населенном пункте;
- таблица "Утечки" содержит информацию о зафиксированных утечках, включая дату опреде-

ления утечки, присвоенную категорию, подтверждающие фото.

Анализ собранных данных проводится персоналом, после чего данные привязывают к карте местности, над которой проводился облет.

Выходная информация представлена отчетами по результатам работы, сохраняемыми в БД. Внешние формы выходных документов определяются проводимыми исследованиями и необходимостью представления результатов и имеют следующий вид: электронная карта местности проводимого облета, на карту нанесен участок магистрального газопровода; GPS-трек полета вертолета относительно трассы газопровода; фотоинформация, синхронизированная с обнаруженной утечкой и АЦП данными, характеризующими наличие и относительную мощность эмиссии газа из МГ.

Программный комплекс разработан в среде визуального программирования *Borland Delphi 7 Enterprise* с использованием *MapInfo Professional 8.5*. При проектировании системы использовался язык структурированных запросов SQL. Программный продукт работает под управлением операционной системы *Windows 2000/XP/Vista*.

Разработанное программное обеспечение системы представляет собой совокупность следующих модулей:

- About — модуль, предоставляющий информацию о разработчике;
- Help — модуль справочной системы пользователя;
- Main — модуль главной формы;
- Kartpr — модуль, позволяющий вести учет существующих карт местности, фотоснимков;
- Utechki — модуль, позволяющий работать с данными, полученными по утечкам. В этом модуле происходит выгрузка всех зафиксированных утечек;
- FlightDat — модуль, позволяющий работать с данными, полученными с оборудования локаатора, включая построенные GPS-треки, графики с показаниями АЦП локаатора;
- Marsh — модуль, позволяющий работать с маршрутами облетов;
- Reports — модуль, позволяющий получать различные отчеты по задаваемым критериям.

Автоматизированная геоинформационная система "АРМ оператора ЛУГ" имеет интуитивно понятный, удобный и универсальный интерфейс для эффективного управления программой. Структурная схема интерфейса представлена на рис. 3.

Главное окно программы имеет два главных пункта меню "Файл" и "Справка". Меню "Файл" дублирует следующие кнопки быстрого доступа, используемые для удобства работы пользователя: "Карты облетов", "Утечки", "Маршруты" и "Отчет



Рис. 3. Структурная схема интерфейса "АРМ оператора ЛУГ"

результатов". Пункт меню "Справка" позволяет получить доступ к информации "Руководство пользователя", "Описание ЛУГ" и "О программе".

При выборе пункта меню "Карты облетов" открывается окно, позволяющее загрузить необходимую карту местности с участками, по которым проводился облет, с сохранением координат GPS и данных о населенном пункте; найти данную карту из БД можно по конкретному номеру карты.

При выборе пункта меню "Утечки" открывается окно, содержащее информацию об участке местности, где произошла утечка газа из МГ, и фотографию, соответствующую данному участку. Поиск утечки осуществляется по номеру, присвоенному участку, и дате определения утечки. Для редактирования информации в БД используется навигатор таблиц.

При выборе пункта меню "Маршруты" существует возможность просмотра информации о маршрутах облета с указанием номера, наименования населенного пункта и отображения координат GPS.

При выборе пункта меню "Отчет результатов" открывается окно, позволяющее выбрать раздел "Загрузка MapInfo Professional". После чего необходимо открыть карту местности, с нанесенным GPS-треком и точечными объектами. При выборе точечных объектов на карте можно получить информацию о широте и долготе. Далее к точечным объектам необходимо привязать АЦП и фотоданные, в результате будет сформирован результирующий отчет.

Результаты, полученные в ходе летных испытаний на участке МГ с использованием ПАДК и обработанные на "АРМ оператора ЛУГ", представлены на рис. 4 (см. вторую сторону обложки). Рисунок содержит:

- электронную карту местности, выполненную с использованием ГИС-технологии;

- траекторию полета вертолета относительно трассы газопровода, позволяющую контролировать и анализировать ее, в целях выявления во время и после полета необследованных участков на трассе МГ;
- координаты утечек газа, "привязанные" к местности и к МГ;
- фотоинформацию, фотография синхронизирована с обнаруженной утечкой, центр ее указывает на точное местонахождение утечки газа (утечка подтверждена визуально "бурлением" воды). Фотокадр, отображаемый в реальном масштабе времени на экране ЭВМ, позволяет контролировать и корректировать точное местоположение "падения" лазерного излучения локатора относительно газопровода;
- цифровую осциллограмму, которая характеризует наличие и относительную мощность утечки газа.

По результатам анализа формируется предложение о проведении дополнительных работ по диагностированию зарегистрированных утечек газа из МГ для повышения достоверности информации и вероятности обнаружения утечки.

Полученные результаты могут быть использованы для прогнозирования технического состояния газопровода и проведения комплекса мероприятий по повышению устойчивости к системным авариям.

Заключение

Совместное использование ЛУГ и ГИС обеспечивает наглядное представление информации

об утечках пользователю, делая ее удобной для анализа, последующей обработки и накопления статистики. Автоматизированная система может быть использована для обработки информации с других типов локаторов.

Использование автоматизированной ГИС мониторинга магистральных газопроводов повышает эффективность анализа состояний МГ за счет сокращения времени обработки поступающей информации специалистам.

Своевременное обнаружение утечек с последующим использованием в процессе обработки и анализа автоматизированной геоинформационной системы мониторинга магистральных газопроводов позволит обнаружить и предотвратить серьезные аварии на дефектных участках магистральных газопроводов.

Список литературы

1. Попов А. И. Лазерный адсорбционный анализ и его применение. М.: МИФИ, 1982. — 180 с.
2. Пат. RU 51745 U1 РФ. Локатор утечек газа "ЛУГ" / И. И. Плюснин, К. И. Бушмелева и др. // Бюл. 2006. № 6.
3. Плюснин И. И., Бушмелева К. И., Бушмелев П. Е. Мобильная система диагностического обслуживания и мониторинга газопроводных систем // Фундаментальные исследования. 2006. № 1. С. 61—63.
4. Плюснин И. И., Бушмелева К. И. и др. Лазерный детектор метана // Материалы Межд. науч.-тех. конф. "Датчики и преобразователи информации систем измерения, контроля и управления". Судак, 2002. С. 125—126.
5. Плюснин И. И., Бушмелева К. И., Майер И. В., Бушмелев П. Е. Система диагностирования дефектов магистральных газопроводов с использованием ГИС-технологий // Современные наукоемкие технологии. 2005. № 8. С. 46—48.

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 510.22:66.01

И. В. Гермашев, канд. техн. наук, доц.,

Волгоградский государственный педагогический университет,

В. Д. Васильева, канд. техн. наук, доц., **В. Е. Дербишер**, д-р хим. наук, проф.,

Е. В. Дербишер, канд. техн. наук, ст. препод., **А. Ю. Силина**, аспирант,

Волгоградский государственный технический университет

Применение теории нечетких множеств для решения задач химической технологии

Представлен метод формализации и анализа информации о химических веществах и структурах с использованием аппарата теории нечетких множеств, примененный при решении задач химической технологии для ранжирования веществ в зависимости от выбранного признака, а также прогнозирования их свойств.

Ключевые слова: нечеткая информация, теория нечетких множеств, химические вещества, функциональная активность, формализация, прогнозирование свойств, ранжирование, тестирование.

Введение

При решении многих современных прикладных задач химии и химической технологии большое место занимает обработка и анализ информации о веществах и химических структурах, исходных и конечных продуктах химической технологии и т. д. При этом обширная информация о химических веществах, представленная в виде параметров их структуры и свойств в многочисленных монографиях, учебниках, справочниках, научных статьях и других профильных и смежных изданиях, в силу разных особенностей во многих случаях носит неоднородный, неполный, неточный, порою противоречивый характер. Можно выделить несколько типов такого рода информации: качественная информация в виде субъективных суждений специалистов, отсутствие количественных данных о параметрах, погрешность в их измерениях и расчетах, их интервальная оценка, ошибки, несовершенство методик и т. д. При анализе и систематизации такого рода нечеткой информации возникают существенные проблемы, связанные с ее математической обработкой, ввиду ограниченной применимости большинства математических методов и больших потерь информации при попытке ликвидировать ее нечеткость.

Уменьшить потери информации, "сгладить" нечеткость и за счет этого повысить объективность принятия решений, в большой мере позволяет использование разрабатываемых в последние годы специальных математических методов, ориентированных на анализ нечетких данных, например аппарата теории нечетких множеств, который является предметом настоящей работы. Этот относительно новый раздел математики сегодня активно развивается и находит прикладное применение, в основном в робототехнике, машиностроении, генетике, областях, связанных с проблемами управления техническими системами, функционированием вычислительной техники, принятием компромиссных решений [1]. В области химической технологии отдельные разработки появились в 80-е годы прошлого века [2] и в настоящее время их развитие, можно сказать, только начинается [3–5].

Настоящая работа посвящена анализу нечеткой информации о химических веществах и структурах с помощью аппарата теории нечетких множеств и применению результатов анализа для решения ряда прикладных задач, связанных с выбором веществ (материалов) из набора известных в зависимости от выбранного признака, а также прогнозирование свойств новых или малоизученных химических веществ (структур) на основе моделирования зависимости "структура — свойство". Данные задачи часто возникают при оценке в

широком смысле качества химической продукции, на предпроектной стадии создания химической продукции в виде сложных химических систем, например многокомпонентных полимерных композиций, при решении задач экологического нормирования веществ и т. д.

Рассмотрим обобщенную задачу обработки нечеткой информации о веществах и структурах.

Формализация нечеткой информации

Методика формализации нечеткой информации о структуре и свойствах веществ путем создания нечетких множеств заключается в следующем.

Пусть $S = \{s_i | i = \overline{1, n}\}$ — множество, состоящее из n объектов (применительно к данной работе, например, химических веществ). Для s_i определено m характеристик Q_{ij} , $j = \overline{1, m}$. В зависимости от условий применения набор рассматриваемых характеристик может расширяться и сужаться.

Построим для каждой характеристики (критерия) нечеткие множества вида

$$\hat{Q}_{ij} = \{(x_j, \mu_{ij}(x_j))\}, \quad i = \overline{1, n}, \quad j = \overline{1, m}, \quad (1)$$

где $\mu_{ij}(x_j) \in [0; 1]$ — функция принадлежности нечеткого множества.

В процессе построения нечетких множеств определяем область значений переменных x_j (отрезок X_{ij}):

$$x_j \in X_{ij} = [q_{ij} - \delta_{ij}, q_{ij} + \delta_{ij}], \\ i = \overline{1, n}, \quad j = \overline{1, m}, \quad (2)$$

где q_{ij} — значение переменной x_j , которое в наибольшей степени удовлетворяет характеристике Q_{ij} , δ_{ij} — радиус ($\delta_{ij} > 0$), в пределах которого значения x_j удовлетворяют критерию Q_{ij} .

Далее для критерия Q_{ij} подбираем функцию принадлежности μ_{ij} . Проведенные нами отдельные исследования по выбору функций принадлежности позволили подобрать ряд функций, хорошо зарекомендовавших себя при анализе информации о химических соединениях [6]. Приведем для примера одну из этих функций:

$$\mu_{ij}(x_j) = \left(\frac{\delta_{ij}^2}{255(x_j - q_{ij})^2 + \delta_{ij}^2} \right)^{\frac{1}{8}}, \\ i = \overline{1, n}, \quad j = \overline{1, m}. \quad (3)$$

Исходя из построения X_{ij} получаем, что в q_{ij} функция принадлежности имеет точку максимума, в пределах X_{ij} функция принимает значения не менее 0,5, а вне X_{ij} — равными 0:

$$\mu_{ij}(q_{ij}) = 1, \quad \mu_{ij}(x_j) \geq 0,5.$$

Таблица 1

Общее представление данных, характеризующих выборочные свойства химических соединений

i	Характеристики Q_{ij}							
	Нечисловые				Числовые			
	j							
	1	2	3	4	5	6	7	8
1	[1; 5]	Активное	61	41	27	42	50	58
2	[3; 10]	Высокоактивное	49	54	34	44	38	49
3	—	Высокоактивное	37	48	53	60	31	47
4	[5; 6]	—	49	46	51	57	50	40

Приведем пример подобной обработки информации о химических соединениях. Пусть ряд химических соединений (объектов) характеризуется данными, приведенными в табл. 1.

Определим области значений переменных x_j . Для характеристик Q_{i1} , представляющих собой интервальную оценку, в качестве q_{i1} берем середину интервала, в качестве δ_{i1} — половину интервала. Для характеристик, значение которых неизвестно (Q_{31}), в качестве отрезка X_{ij} можно взять, например, отрезок, покрывающий отрезки для всех объектов, т. е. $X_{31} = [1; 10]$. Характеристики Q_{i2} представляют собой словесное описание. Представим всю область, описываемую данными терминами, в виде отрезка $X_{i2} = [0; 1]$. Поскольку четкой границы между значениями, представляемыми в табл. 1, нет, проведем разграничение этих значений следующим образом: неактивные — $[0; 0,3]$, умеренно активные — $[0,3; 0,5]$, активные — $[0,5; 0,7]$, высокоактивные — $[0,7; 1]$. Для неизвестного значения объекта № 4 поступим аналогично вышесказанному, примем $X_{42} = [0; 1]$.

Остальные характеристики представляют собой конкретные числа и их можно взять в качестве значений q_{ij} , а в качестве δ_{ij} — абсолютную погрешность. Пусть, например, все $\delta_{ij} = 1$ ($i = \overline{1, n}, j = \overline{1, m}$).

Значения для q_{ij} и δ_{ij} приведены в табл. 2.

Подготовленная таким образом информация используется в дальнейшем для расчета функции принадлежности $\mu_{ij}(x_j)$ (2) и построения нечетких

множеств \hat{Q}_{ij} (1). Последние применяются далее при решении конкретных задач ранжирования химических веществ и прогнозирования их свойств.

Применение метода для ранжирования химических веществ (материалов)

Ранжирование продукции в однородном наборе по какому-либо критерию, который может представлять собой как отдельный параметр (характеристику), так и комплекс параметров, определяющих в широком понимании качество продукции, является распространенной задачей химии и химической технологии. Так, при проектировании, например, различных полимерных материалов (композиций), как правило, представляющих собой сложные многокомпонентные системы, возникает задача выбора оптимальных (качественно и количественно) активных компонентов (добавок, ингредиентов), улучшающих свойства полимерной композиции (например, стабилизаторы, пластификаторы, антиоксиданты, антипирены, олигомеры, наполнители и др.), из набора известных или вновь создаваемых на основе их ранжирования, и формирование составов.

В качестве примера применения рассматриваемого метода приведем ранжирование термостабилизаторов (ТС), добавляемых в композиции на основе полиэтилена (ПЭ), в их однородном наборе.

Пусть имеется некоторое множество (набор) активных добавок (термостабилизаторов) $S = \{s_i | i = \overline{1, n}\}$. Расположим соединения в порядке убывания технологической активности (функции термостабилизации).

При добавлении ингредиента в полимерную композицию возникает некоторый технический эффект, описываемый в той или иной мере комплексом характеристик $Q_{ij}, j = \overline{1, m}$. В нашем случае для упрощения вычислений выбраны две характеристики ($m = 2$): температура плавления добавки Q_{i1} ($^{\circ}\text{C}$) и концентрация добавки в ПЭ Q_{i2} (%), и определены значения Q_{ij} для каждого вещества. При этом области значений $G_j, j = \overline{1, m}$ для каж-

Таблица 2

Параметры функции принадлежности $\mu_{ij}(x_j)$

i	j															
	1		2		3		4		5		6		7		8	
	q_{ij}	δ_{ij}														
1	3	2	0,6	0,1	61	1	41	1	27	1	42	1	50	1	58	1
1	6,5	3,5	0,8	0,1	49	1	54	1	34	1	44	1	38	1	49	1
3	5,5	4,5	0,8	0,1	37	1	48	1	53	1	60	1	31	1	47	1
4	5,5	0,5	0,5	0,5	49	1	46	1	51	1	57	1	50	1	40	1

дой характеристики располагаются в следующих пределах:

$$G_1 = [-273; +\infty), \quad G_2 = [0; 100].$$

Далее выберем эталон s_0 — гипотетическую добавку ("идеальное" вещество), имеющую наилучшие (предельные) характеристики, как с точки зрения предъявляемых к полимерным системам на основе ПЭ требований, так и возможностей представленных на экспертизу элементов множества S . Это добавка, имеющая $Q_{01} = 304$ и $Q_{02} = [0,1; 0,3]$.

В соответствии с приведенной выше методикой определяем области значений переменных x_j (отрезок X_{ij}) (2) и рассчитываем значение функции принадлежности $\mu_{ij}(x_j)$ вида (3). Таким образом, для каждой характеристики Q_{ij} строим нечеткие множества (1). При этом следует учитывать, что $i = \overline{0, n}, j = \overline{1, m}$.

Чтобы определить, в какой мере характеристика добавки соответствует эталону, вычислим индекс схожести v_{ij} двух соответствующих нечетких множеств \hat{Q}_{0j} и \hat{Q}_{ij} :

$$v_{ij} = \max_{x_j \in G_j} \min(\mu_{0j}(x_j), \mu_{ij}(x_j)), \quad i = \overline{1, n}, j = \overline{1, m}.$$

Для функций принадлежности вида (3) используем следующее решение:

$$v_{ij} = \mu_{0j} \left(\frac{q_{0j}\delta_{ij} + q_{ij}\delta_{0j}}{\delta_{ij} + \delta_{0j}} \right), \quad i = \overline{1, n}, j = \overline{1, m}.$$

Далее, чтобы вычислить интегральную оценку (итоговый ранг) меры эффективности v_i добавки s_i по предъявленным требованиям, используем процедуру взвешенного голосования:

$$v_i = \sum_{j=1}^m \alpha_j v_{ij}, \quad \sum_{j=1}^m \alpha_j = 1, \quad \alpha_j \geq 0.$$

Коэффициент α_j показывает значимость j -го критерия. В данном примере с целью упрощения демонстрации метода коэффициенты α_j приняты равными друг другу.

В табл. 3 приведены выборочные результаты расчета v_i и ранжирования ТС ПЭ.

Таким образом, использование данного метода позволяет поставить в соответствие каждому ранжируемому веществу (материалу) число, характеризующее его меру близости к набору предъявляемых требований, и расположить вещества в порядке, соответствующем их качественной градации: Понятно, что в качестве эталона можно выбрать и "идеальное вещество", не противоречащее научно-техническим знаниям, принципиально результаты от этого не изменятся.

Применение метода для прогнозирования свойств малоизученных химических веществ (соединений)

Экспертиза малоизученных и вновь созданных химических веществ, поиск эффективных областей их применения приводит к необходимости прогнозирования их свойств.

В основном прогнозирование свойств химических веществ в работах последних лет базируется на моделировании (математическом или информационном) зависимости "структура — свойство". В этом направлении, особенно в области прогнозирования биологической активности веществ, имеются серьезные достижения [7—10]. Математический аппарат теории нечетких множеств, как это показано ниже, может выступать здесь в роли дополнительного инструмента, с помощью которого осуществляется моделирование и в ряде случаев упрощается сама "технология" моделирования.

Рассмотрим пример применения нечетких множеств для решения задачи прогнозирования функциональной активности химических веществ-добавок к полимерным композициям (технического эффекта) путем моделирования зависимости "структура — свойство".

В качестве подготовительной процедуры для прогнозирования была сформирована профилированная база данных однокомпонентных органических добавок к полимерным композициям с одновременной их сортировкой по функциональному назначению (термо- и светостабилизаторы (СС), антипирены (АП), антиоксиданты и т. д.) и уровню активности в соответствии с выбранным критерием. При этом наполнение базы данных осуществлялось введением описания структуры химических соединений с помощью дескрипторов разного уровня информативности, иерархический комплекс которых включает дескрипторы элементного уровня, структурной формулы, электронного уровня, межмолекулярных взаимодействий. Здесь подробное рассмотрение этого вопроса опущено, так как к предмету статьи имеет косвенное отношение.

Важно одно, что таким образом были созданы множества дескрипторов классов веществ с определенными технологическими функциями. Такое построение можно назвать статистическим образом (информационной моделью) активной добавки.

В основу рассматриваемого метода прогнозирования свойств с использованием построения нечетких множеств положен тезис, заключающийся в том, что если при добавлении к некоторому конечному (!) множеству нового элемента (в нашем случае дескриптора) это множество никак не изменяется, то новый элемент классифицируется, как принадлежащий данному множеству. То есть для прогнозирования свойств необхо-

Результаты ранжирования термостабилизаторов ПЭ

i	Химическое название	Структурная формула	Технические характеристики		Эффективность v_i
			Температура плавления Q_{i1} , °C	Концентрация в ПЭ Q_{i2} , %	
1	Трифенилфосфит		25	0,1—3	0,60
2	Фенил-β-нафтиламин		108	0,2—1,5	0,68
3	2,6-Ди-(α-метилбензил)-4-метилфенол		~30	0,5	0,70
4	4-(нафталин-1-илимино)бутан-2-ол		190	0,2—1,5	0,71
5	N-фенилнафталин-1-амин		62	0,1—0,5	0,74
6	N ¹ ,N ⁴ -дифенилбензин-1,4-диамин		152	0,1—0,3	0,78
7	1Н-бензо[d]имидазол-2-тиол		304	до 0,5	1,00

димо дескрипторы диагностируемого соединения добавить в статистический образ. Получившийся новый статистический образ должен несущественно (в математическом смысле) отличаться от предыдущего. В этом случае с определенной степенью вероятности предполагается, что соединение имеет определенную функциональную активность, и, наоборот, отличия будут значительными, если соединение неактивно.

Итак, имеется множество соединений (добавок) $S = \{s_i | i = \overline{1, n}\}$, имеющих характерную техническую (технологическую) функцию. Проранжируем все s_i , как это было показано выше, и построим, например, четыре множества:

$S^0 = \{s_i^0, i = \overline{1, n^0}\}$ — множество неактивных добавок;

$S^1 = \{s_i^1, i = \overline{1, n^1}\}$ — множество умеренно активных добавок;

$S^2 = \{s_i^2, i = \overline{1, n^2}\}$ — множество активных добавок;

$S^3 = \{s_i^3, i = \overline{1, n^3}\}$ — множество высоко активных добавок.

$$\text{Причем } S = \bigcup_{k=0}^3 S^k, n = \sum_{k=0}^3 n^k.$$

Определим множество дескрипторов $D = \{d_j | j = \overline{1, m}\}$, характеризующих структуру химических соединений. Частоту появления дескриптора d_j в соединении s_i обозначим через v_{ij} .

Вычислим частоту появления v_j^k дескриптора d_j среди добавок множества S^k :

$$v_j^k = \sum_{i=1}^{n^k} v_{ij}^k,$$

где v_{ij}^k — частота появления дескриптора d_j в структуре s_i^k .

Для каждого класса добавок S^k построим нечеткое множество \hat{D}^k дескрипторов, описывающих соединения данного класса:

$$\hat{D}^k = \{d_j | \mu^k(d_j) | j = \overline{1, m}\}.$$

Множество \hat{D}^k является статистическим образом добавки класса S^k .

Определим функцию принадлежности $\mu^k(d_j)$ как оценку вероятности появления дескриптора d_j среди дескрипторов класса S^k :

$$\mu^k(d_j) = \begin{cases} \frac{v_j^k}{v_j}, & v_j \neq 0 \\ \frac{1}{4}, & v_j = 0 \end{cases}, \text{ где } v_j = \sum_{k=0}^3 v_j^k.$$

Для того чтобы сделать прогноз о принадлежности некоторого химического соединения s одному из классов добавок S^k , следует выполнить следующие действия.

Добавим соединение s в класс S^k , получим новый класс $S_s^k = S^k \cup \{s\}$ и построим для S_s^k статистический образ \hat{D}_s^k так, как это показано выше.

Вычислим Δ^k — индекс изменений, произошедших в классе S^k после добавления соединения s . В качестве оценки величины Δ^k можно взять, например, индекс несходства нечетких множеств \hat{D}^k и \hat{D}_s^k :

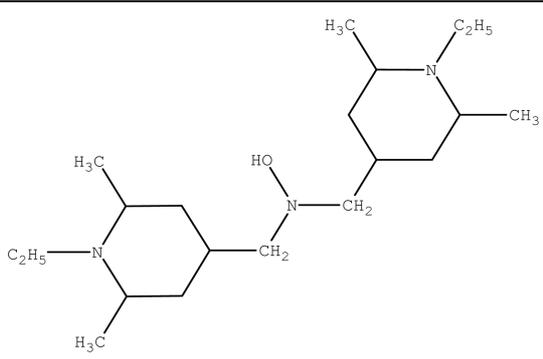
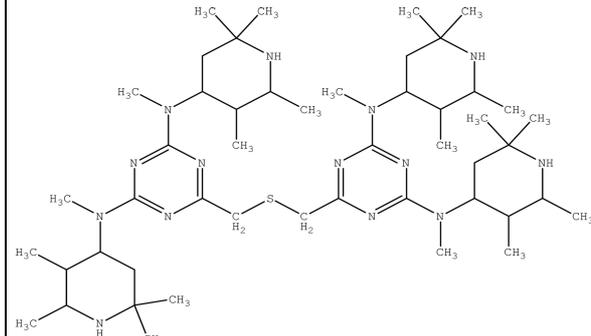
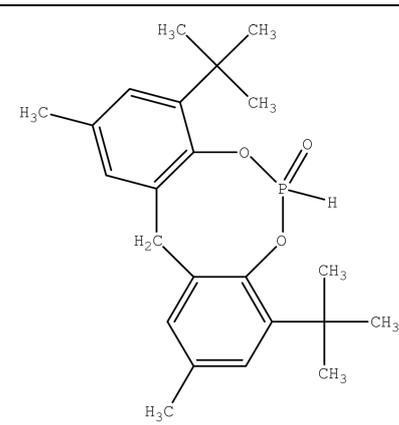
$$\Delta^k = 1 - S((\hat{D}^k)^C, (\hat{D}_s^k)^C) = 1 - \frac{|(\hat{D}^k)^C \cap (\hat{D}_s^k)^C|}{|(\hat{D}^k)^C \cup (C\hat{D}_s^k)^C|}.$$

Очевидно $\Delta^k \in [0; 1]$. Индекс Δ^k принимает значение 0, если нечеткое множество никак не изменилось, и значение 1, если нечеткое множество настолько изменилось, что новое нечеткое мно-

Таблица 4

Результаты тестирования модифицированных структур химических соединений

№№	Химическое название	Структурная формула	Техническая функция	Прогнозируемая группа активности (Δ^k)
1	3,9-бис-(1,2,2,6,6-пентаметил-4-пиперидилокси)-2,4,8,10-тетраокса-3,9-фосфа-спиро-(5,5)-ундекан		ТС СС	Активные (0,68) Умеренно активные (0,71)
2	5-бутил-2-(2,4-дигетрабутил-6-метилбензил)-5-этил-[1,3,2]диоксафосфонат		ТС СС АП	Высокоактивные (0,73) Неактивные (0,47) Умеренно активные (0,72)

№№	Химическое название	Структурная формула	Техническая функция	Прогнозируемая группа активности (Δ^k)
3	N,N'-бис-(1-этил-2,6-диметил-пиперидин-4-илметил)гидроксиламин		ТС СС	Активные (0,78) Умеренно активные (0,76)
4	N,N'-диметил-N,N'-бис-(2,2,5,6-тетраметилпиперидин-4-ил)-илметилсульфанилметил-[1,3,5]триазин-2,4-диамин		ТС СС	Умеренно активные (0,70) Активные (0,72)
5	4,8-ди-трет-бутил-2,10-диметил-12п-5,7-диокса-6-фосфа-добензо[a,b]циклооктан-кислота		ТС СС АП	Неактивные (0,73) Высокоактивные (0,75) Умеренно активные (0,79)

жество не имеет ни одного общего элемента со старым. То есть чем больше значение Δ^k , тем более радикальные изменения произошли в множестве S^k . При этом следует указать, что объем нечеткого множества в целях повышения достоверности результатов должен составлять не более 50 элементов.

Далее осуществляем классификацию s , т. е. относим соединение к тому классу, где значение Δ^k окажется наименьшим.

Таким образом, $s \in S^r$, где $r = \arg \min_{k=1,4} \Delta^k$.

На основе классификации принимается решение о возможных свойствах тестируемых соедине-

ний и далее проводится их сортировка для принятия решения, например о синтезе какого-либо из них.

Пример выборочных результатов тестирования набора малоизученных органических соединений на предмет наличия у них технологической активности при использовании в качестве добавок к композициям на основе полипропилена приведен в табл. 4.

Проверка прогностической способности данного метода на тестовой выборке показала надежность прогноза свыше 70 %, что, учитывая сложность задачи, является неплохим результатом.

Заключение

Таким образом, рассмотренный метод формализации и анализа информации о химических веществах и структурах позволяет обрабатывать нечеткую информацию и имеет выраженную практическую направленность.

Применение данного метода для решения указанных выше задач предполагает обработку информации, как обычными способами, так и использование компьютерных технологий, создание проблемно ориентированных компьютерных систем.

Список литературы

1. **Juang C.-F.** Temporal problems solved by dynamic fuzzy network based on genetic algorithm with variable-length chromosomes // *Fuzzy Sets and Systems*. 2004. V. 142. I. 2. P. 199–219.
2. **Кафаров В. В., Дорохов И. Н., Марков Е. П.** Системный анализ процессов химической технологии. Применение метода нечетких множеств. М.: Наука, 1986. 359 с.

3. **Бахитова Р. Х., Спивак С. И.** Нечеткие интервальные оценки в кинетике химических реакций // *Изв. вузов. Химия и хим. технол.* 1999. 42. № 3. С. 92–96.
4. **Уткин В. С.** Экспертная оценка качества материалов с использованием нечетких множеств // *Строит. материалы*. 2001. № 6. С. 34–35.
5. **Yang Zhao-hua, Yang Zhi, Wang Hui-zhon.** Исследование двух новых способов управления pH // *Gansu gongyong daxue xuebao: J. Gansu. Univ. Technol.* 2001. 27, № 3. С. 51–54.
6. **Гермашев И. В., Дербисер В. Е.** Свойства унимодальных функций принадлежности в операциях с нечеткими множествами // *Изв. вузов. Математика*. 2007. № 3. С. 77–80.
7. **Guha R., Jurs P. C.** Development of QSAR models to predict and interpret the biological activity of artemisinin analogues // *J. Chem. Inf. and Comput. Sci.* 2004. V. 44. N 4. P. 1440–1449.
8. **Theodorou D. N.** Understanding and predicting structure — property relations in polymeric materials through molecular simulations // *Mol. Phys.* — 2004. 102. N 2. P. 147–166.
9. **Скворцова М. И., Федяев К. С., Палюлин В. А., Зефирова Н. С.** Моделирование связи между структурой и свойствами углеводов на основе базисных топологических дескрипторов // *Изв. АН. Сер. хим. РАН*. 2004. № 8. С. 1527–1535.
10. **Филимонов Д. А., Акимов Д. В., Поройков В. В.** Метод самосогласованной регрессии для количественного анализа связи структуры и свойств химических соединений // *Хим.-фармац. журнал*. 2004. № 1. С. 21–24.

УДК 519.766.23:510.64:004.4:631.1

Л. М. Воронова, канд. техн. наук, доц.,
Владимирский государственный университет

Моделирование оптимальных планов земледелия с применением адаптированных пакетов технологий

Рассматривается новый подход к решению задачи оптимизации земледелия хозяйства в рамках автоматизированной системы технологического проектирования земледелия. На основе морфологического исследования объекта проектирования формируется морфологический ящик, отражающий синтаксическую структуру системы и включающий семантические связи между значениями компонентов. В результате морфологического синтеза строятся верные варианты планирования сельскохозяйственной деятельности для отдельных полей, отыскивается множество оптимальных решений. Программная система, реализующая предложенную методику, полезна для прогнозирования экономических показателей земледелия в условиях агротехнических ограничений конкретного земледельческого хозяйства.

Ключевые слова: система земледелия, поля, типы почв, технологии обработки, структура севооборотов, правила землепользования, моделирование, морфологический подход, основные компоненты, признаки компонентов, значения признаков, принципы генерации, правила подстановок, семантика связей, оптимизация, автоматизация планирования земледелия.

Введение

Глобальная информатизация всех отраслей производства коснулась и сельского хозяйства. Здесь применение информационных технологий (ИТ) возможно на разных уровнях: федеральном, региональном, местном вплоть до отдельного хозяйства. Задачи на этих уровнях разные, соответственно различаются и используемые данные, и математические, и программные средства работы с ними. Можно констатировать, что процесс внедрения в сельское хозяйство информационно-

прогнозирующих технологий находится на начальной стадии. Причем наименее разработан самый нижний уровень — уровень отдельного хозяйства. Как правило, ИТ реализуются через применение компьютерной техники и программного обеспечения по сбору, хранению, систематизации и передачи информации для бухгалтерского учета, для автоматизации делопроизводства в хозяйстве. Если говорить о земледельческих хозяйствах и их практических проблемах, то задача оптимизации земледелия всегда считалась весьма трудоемкой в

силу огромного количества исходной информации, которую необходимо обработать и проанализировать для принятия обоснованных решений по планированию и ведению процесса с получением достоверных оценок и нужных конечных результатов. Эта информация, как правило, весьма разнообразна, включает совокупность природных, экономических, социальных показателей, взаимодействующих и влияющих друг на друга, оказывающих комплексное воздействие на результат. Именно в силу указанных обстоятельств, а также благодаря созданию благоприятных объективных условий (в виде современных компьютерных систем и технологий) задача оптимизации давно требует машинного моделирования и автоматизации своего решения. Экспериментировать "вживую" — занятие опасное, агроном всегда выбирает надежный проверенный вариант, не подзревая порой, что оптимум лежит совсем рядом.

Применение общепринятых математических моделей и методов к данной задаче не позволяет учесть все взаимосвязи компонентов системы, которые чаще всего представлены нечисловыми значениями. Предлагаем новый подход к компьютерному моделированию системы земледелия на предстоящий период на основе морфологического исследования объекта проектирования и формирования базы специального вида, включающей необходимую информацию о землях конкретного хозяйства и все принципы и правила ресурсосберегающего землепользования, адаптированные в хозяйстве. Практическая направленность развиваемой темы предполагает создание квалифицированного помощника-консультанта в хозяйстве в виде автоматизированной компьютерной системы. Она в доступной и понятной форме позволит использовать работающему на земле человеку достижения земледельческой науки и накопленные практические знания в виде руководящего материала, а также в виде "умного" проектировщика системы земледелия. В реально сложившихся условиях роста частного земледелия весьма актуальна проблема сохранения природоохранного землепользования.

Подход к анализу задачи

В качестве метода морфологического исследования объекта проектирования выбран метод морфологического ящика (ММЯ), разработанный швейцарским астрономом Ф. Цвики и позволяющий решить задачу классификации многопризнаковых объектов в виде морфологического ящика (МЯ). Метод предполагает два этапа: 1) анализ (исследование) объекта проектирования с выделением основных функциональных составляющих-компонент, их основных (классифика-

ционных) признаков и выявлением известных, реально существующих значений последних, представляемых своими векторами (открывается заманчивая перспектива дополнить эти списки новыми реализациями); 2) синтез как генерирование альтернатив перебором всех возможных сочетаний этих значений, взятых по одному из каждого вектора (для каждого классификационного признака). Допустимое сочетание реализации всех признаков компонентов системы и составляет вариант технического решения. Таким образом, МЯ потенциально вмещает все множество возможных объектов (даже практически непригодных с точки зрения опытного специалиста). Для исключения на уровне автоматического синтеза семантически неверных сочетаний значений признаков использован разработанный в работе [1] новый метод (названный вариантным), являющийся развитием ММЯ. Он использует логические закономерности, присущие предметной области, обуславливающие семантику связей признаков компонентов объекта проектирования, значения которых дискретны и чаще всего дают его качественное описание. Суть заключается в расширении каждого вектора МЯ по Цвики до матрицы, состоящей из "гнезд" — расширенных значений признака вместе с характеристиками, отражающими существование этого значения в семантически правильно построенных объектах, и условиями связей со значениями или характеристиками соседних признаков. Предлагается в работе [1] и схема структуризации информации для компактного компьютерного хранения описанной модели, "вмещающей" все множество синтаксически возможных и семантически правильных объектов проектирования в рамках вложенных в систему знаний. Эта схема позволяет легко вносить изменения и дополнения при появлении новых вариантов значений признаков с семантикой их вхождения в состав целого объекта. При этом требования к собственным знаниям конечного пользователя о поле новых вариантов объектов минимальны, можно полагаться на знания системы при выполнении синтеза. Обоснована модель автоматизированной системы порождения вариантов земледелия по образцу генерации правильных предложений в грамматике.

В работе [2] показана возможность, целесообразность и реализация применения изложенного выше подхода к планированию сельскохозяйственной системы земледелия, как некой морфологической системы. В каждой задаче необходимо конкретизировать определение системы, подвергаемой морфологическому исследованию (это может быть материальный объект, явление, некоторый процесс и др.). Напомним, что под системой

земледелия следует понимать некоторый протекающий во времени процесс, включающий **способы обработки земли** в целях выращивания необходимых сельскохозяйственных **культур** (в соответствии с экономически обоснованным количеством и экологически обоснованным качеством последних), направленные на восстановление, поддержание, а иногда и на повышение плодородия почвы [3].

При использовании морфологических методов самым сложным и трудозатратным является первый этап — изучение предметной области, анализ объекта проектирования, результатом которого является построение его морфологической (синтаксической) структуры. Основанием деления морфологической системы на составляющие, как указано выше, могут служить представления о функционально-необходимых свойствах исследуемой проблемы. Для нашей морфологической системы они просматриваются в ее определении. Выделены три компонента: **земля, культура, технология (способы) обработки** (где, что и как выращивать). В работе [2] обосновывается и выделение основных признаков этих компонентов, важных при проектировании системы. В хозяйстве любого региона **земля** представлена конкретными типами почв, рабочими полями-участками и поделена под адаптированные севообороты (далее сокращенно с/о), **культуры** представлены группами культур для текущего выбора и культурой-предшественником, что важно для с/о, **технологии** — уровнем интенсификации и технологическими звеньями обработки земли.

В рамках совместной научной работы с Владимирским научно-исследовательским институтом сельского хозяйства (ВНИИСХ) изучались многочисленные данные, полученные в результате многолетних практических и теоретических исследований учеными ВНИИСХ на их природно-производственной базе, анализ этих данных обнаружил многие закономерности на агроэкологическом уровне. Все данные, полученные из ВНИИСХ, влияющие на эффективность земледелия в регионе (а точнее, во Владимирском Ополе), систематизированы как реальные значения выявленных признаков. Эти данные включают: 27 типов почв, свыше 120 рабочих участков-полей, три уровня интенсификации, 25 схем севооборотов, 13 возделываемых в зоне культур со всем их сортовым разнообразием, восемь видов основной обработки почвы, 35 комбинаций предпосевной обработки почвы и посевов, 15 комбинаций уборочных технологий, более 10 схем применения удобрений и столько же — средств химизации [3].

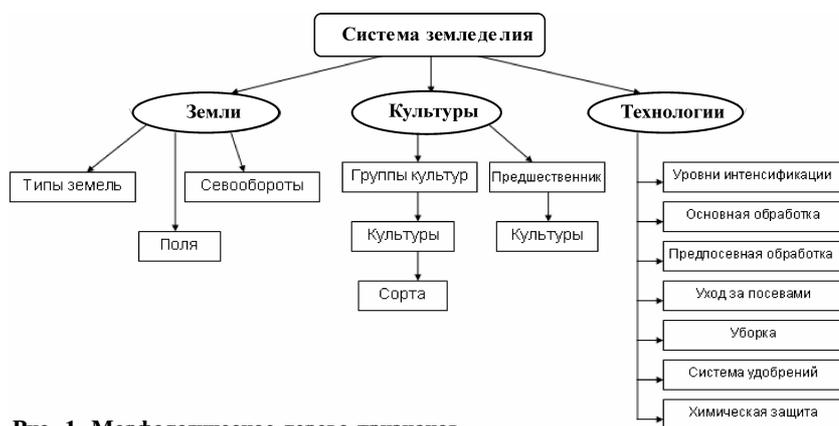


Рис. 1. Морфологическое дерево признаков

Представить результаты анализа на формальном уровне позволяет разработанное морфологическое дерево, отражающее синтаксис системы: основные компоненты, их признаки, важные для земледелия, возможные значения признаков (рис. 1). Нижний уровень ввиду громоздкости представления показан усеченно. Полнота набора признаков и их значений относительна, поскольку признаки отражают лишь выбранный для исследования тип системы и обнаруженные о ней знания. Необходимо обеспечить возможность расширения, изменения, настройки при углублении и расширении наших представлений о ней в процессе работы. Формальным описанием содержимого этого дерева является МЯ, отвечающий указанным выше пожеланиям.

Синтез проектных решений

Для автоматизированного порождения вариантов земледелия по методике в работе [1] (по образцу генерации синтаксически правильных предложений) разработана простейшая грамматика "Системы земледелия": алфавит, правила подстановок, порядок их применения:

$$G(Z) = (N, T, P, Z),$$

где N — алфавит нетерминальных символов (компоненты и их признаки); T — алфавит терминальных символов (значения признаков); Z — начальный символ ($Z \in N$); P — конечный набор правил подстановок.

Поскольку грамматика разработана в русском алфавите с ориентацией на конкретных пользователей, нетерминальные и терминальные символы, используемые в правилах подстановок, совпадают с первыми буквами имен компонентов, признаков, их значений и в дополнительных комментариях не нуждаются. Заметим, что в процессе выполнения работы возникла необходимость в дополнении дерева синтеза (см. рис. 1) новым компонентом B — **внешние требования** (по заданию агронома, для управления синтезом в соответ-

вии с реальными условиями). Приведем фрагмент правил P (в нормальной форме Бэкуса):

```

<Z> :: = <B><3><K><T>
<B> :: = <год_реализ><режим_синтеза><...>
<режим_синтеза> :: = одно_поле | один_с/о | все_хоз-во
<3> :: = <раб. уч-к><с/о>
<раб. уч-к> :: = № 1 | № 2 | ... | № 120
<с/о> :: = № 1 | № 2 | ... | № 35
<K> :: = (<Г1>|<Г2>|<Г3>|... <Г13>)<Предшествв.>
.....
<Предшествв.> :: = П1 | П2 | ... <Т> :: =
<Интенсиф.><Т1><Т2> ... <Т6>
<Интенсиф.> :: = Э | Н | И
.....

```

В скобках $\langle \dots \rangle$ — нетерминальные символы; остальные — термы; | знак выбора "или". Семантика имен Э, Н, И в последней приведенной строке подстановок связана с технологиями обработки земли: экстенсивной, нормальной и интенсивной.

Предложения, порождаемые грамматикой, это множество выводимых терминальных цепочек, представляющих варианты реализации земледелия, включая и заведомо "плохие". При использовании МЯ человеком такие решения будут отбрасываться. Но уже для числа признаков ~ 10 и 3—4 значений каждого, задача для человека становится практически неразрешимой.

В работе [2] на конкретном примере демонстрируется необходимость дополнения МЯ семантикой, тем более, если генерацию поручить автомату (ЭВМ). Указанные дополнения возможны, так как на базе богатейшего опыта, достижений земледельческой науки учеными сформулированы многие опорные правила в рассматриваемой предметной области в виде некоторых рекомендаций, ограничений, которые требуют реального учета при формировании жизнеспособных систем земледелия в конкретных хозяйствах. Ограничения касаются как типов почв по допустимым пакетам технологий, так и культур по рекомендуемым типам и технологическим операциям. Получены как базовые урожайности культур на разных уровнях интенсификации, на разных типах почв, так и поправки к ним для каждой технологической операции, каждого предшественника, соотношения для экономических оценок и т. д. Все это должно быть учтено при автоматическом синтезе и оптимизации выбора.

Основываясь на морфологическом подходе, нам удалось выполнить целостное, взаимоувязанное описание многообразными семантическими связями всех данных, внесенных в МЯ, учесть реальные факторы земледелия в нашем регионе (точнее, Владимирском Ополье) как агроэкологические, так и экономико-технологические (чего не удастся сделать в рамках классических математических моделей). В работе [2] в развернутом виде описывается МЯ, как совокупность матриц для

Условия	Признак				
	Значение	Характеристики			
		№ с/о	№ поля в с/о	Площадь	Тип земли
.....
Уч-к № 15 с/о № 3	15	3	2	112	1.1
.....

каждого признака объекта, структура которых состоит из гнезд-значений в виде: $\langle \text{условия} \rangle \langle \text{значение} \rangle \langle \text{характеристики} \rangle$ и пример семантической генерации. Одно гнездо матрицы "рабочие_поля" представлено в таблице.

Теперь, если по дереву синтеза выбрано конкретное поле 15 и задан агрономом год его реализации, по данному гнезду определится севооборот и будет рассчитан общий сдвиг относительно его начала по году реализации и характеристике "№ поля в с/о". В следующей матрице севооборота ставится в соответствие последовательность групп культур, а по сдвигу будет выбрана требуемая для поля группа. По соответствующей матрице разрешительными культурами для выбора из этой группы будут лишь те, условия которых содержат тип земли, уже стоящий в цепочке строящегося решения как характеристика поля 15. В то же время, при включении подходящей культуры в эту цепочку вписываются и ее характеристики "допустимые уровни технологий". Так что, при выборах по дереву синтеза технологий на каждом этапе просматриваются соответствия записей в цепочке условиям операции из очередной матрицы по типам, культурам, а также конкретным наполнениям операциями всего ряда предшествующих звеньев обработки (технологии звеньев очень сцеплены друг с другом).

Таким образом, семантический МЯ является своеобразным фильтром для выбора "правильного" значения признака в текущий вариант проектного решения, активным участником направленного перебора при синтезе. Кроме того, характеристики значений многих признаков включают еще и числа — поправочные коэффициенты к урожайностям (на базе которых постепенно формируются данные для построения реальной оценки результатов). Они тесно связаны именно с конкретным набором условий, а не просто со значением признака, поэтому включены в МЯ.

Дадим формальное описание изложенных рассуждений. Если правую часть гнезда любой матрицы (расширенный признак) рассматривать как некоторое логическое высказывание, то на языке логики предикатов оно включает субъект (значе-

ние признака) и предикат — функцию субъекта, выражающую его основные свойства, характеристики. Поскольку каждый K_i -признак — это множество значений $\{l_j^i\}$, определим на нем предикат P — "правильные характеристики значения" (правильные подстановки). Причем каждая матрица включает лишь истинные значения предиката для любого l_j^i :

$$\forall l^i \in K_i P(l^i).$$

Включая в рассмотрение левую часть гнезда (условия), применим язык логики предикатов для записи всего гнезда-высказывания в виде формулы

$$U(l^i) \rightarrow P(l^i).$$

Это новый предикат — "от условия для каждого значения i -го признака к его правильной характеристике". Так как условия формируются из значений или характеристик других признаков (субъектов), определяемых предыдущими подстановками, то в общем случае они отражают отношения связи между предметами (в нашем случае между признаками системы). Поскольку для любого l_j^i указаны лишь истинные условия его выбора, можно констатировать, что каждая матрица МЯ содержит лишь множество истинности своего признака:

$$\forall l^i \in K^i (U(l^i) \rightarrow P(l^i)).$$

Тогда выводимые терминальные цепочки из Z будут содержать только "истинные" подстановки на основе семантической согласованности друг с другом, т. е. будут являться "правильными" вариантами земледелия.

Несмотря на строгость, лаконичность, удобства логики предикатов для записи "предложений", намеренно оставляем выбор за разработанной грамматикой языка, обогащенной семантическими связями. На то есть несколько причин. Во-первых, понятие вывода в грамматиках в значительной степени опирается на здравый смысл, а значит, легче поддается интеллектуализации (хотя в основе любой грамматики лежит логика высказываний). Во-вторых, и это главное, надо учитывать практическую направленность выстраиваемой системы, т. е. обеспечить ее понятность, расширяемость, возможность быстрой настройки на новое хозяйство (и не силами разработчиков системы — опытных программистов, а обычными пользователями среднего уровня). Строгость логики в значительной мере обеспечивается формализацией символики, а это сильно затруднит любые модификации при работе с системой. Значительная громоздкость в записи условий, характеристик, значений признаков объекта проектирования, их перегруженность привычными для модели человеческого мышления терминами и понятиями поглотит лаконичность логики, но вполне

допустима в алфавите и подстановках грамматики языка.

Реализация разработанного семантического МЯ в виде электронного варианта знаниеориентированной базы данных сама по себе уже имеет большое практическое значение. Кроме того, она положена в основу программной (т. е. автоматизированной) генерации проектных решений. На ее основе возможно порождение вариантов планирования сельскохозяйственной деятельности на поле в зависимости от условий заказчика, а именно: типа почвы, площади поля, культуры, предшественника, технологических и экономических требований (зависящих от доступных технических и финансовых ресурсов). Эти данные будут являться входными параметрами для генератора. Порождение вариантов планирования (с их оценками) для группы полей, образующих севооборот, осуществляется на базе сгенерированных вариантов для отдельных полей в группе. Как показывает практика, общее число полученных решений может достигать нескольких тысяч. При таком объеме результатов землепользователь не в состоянии самостоятельно выбрать подходящие для него варианты. Поэтому ему необходимо предоставить возможность поиска оптимальных (для условий в его хозяйстве) решений.

Выбор оптимальных решений

Наша задача относится к разряду многокритериальных задач дискретной оптимизации, так как области определения аргументов дискретны и на множестве возможных вариантов проектирования целесообразно формировать несколько функций или критериев оценки: затраты, урожайность, как следствие — урожай, доходы, прибыль (их вычисление выполняется пошагово в самом процессе генерации). Основная особенность таких задач: решением является не одна, а множество эффективных точек, удовлетворяющих поставленным требованиям. Для их отбора мы использовали принцип оптимальности по Парето — выбрать множество лучших (с точки зрения лица, принимающего решение) вариантов, имея все допустимые решения.

Пусть варианты сравниваем по двум оценкам (одна — минимизируется, другая — максимизируется): затраты z и доход d . В общем случае оценки и порядок их перечисления можно менять: доход и затраты; урожай и затраты; затраты и прибыль. Из принципа оптимальности вариант i будет доминировать над вариантом j , если:

$$d_i > d_j, \quad \forall j: z_i = z_j.$$

Оптимальные варианты по каждому значению затрат определяют паретовское множество опти-

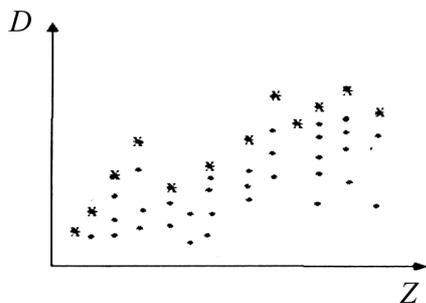


Рис. 2. Паретовское множество 1

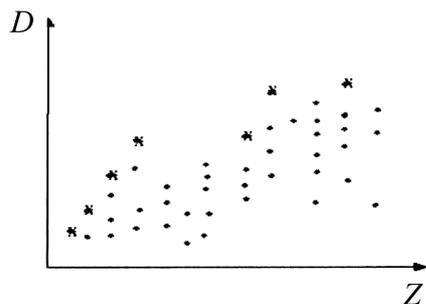


Рис. 3. Паретовское множество 2

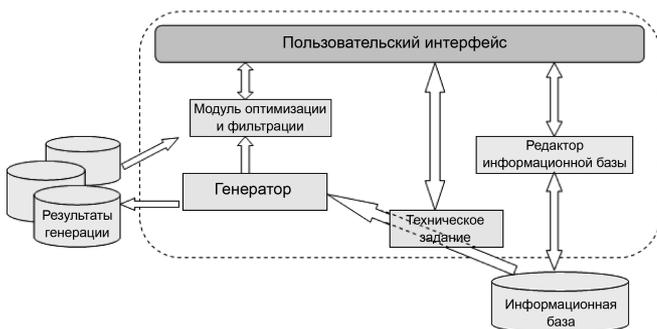


Рис. 4. Обобщенная логическая схема системы проектирования

мальных решений (рис. 2, выделены). Однако для множества возможных альтернатив абсолютно одинаковые оценки затрат встречаются в задаче довольно редко. Поэтому поиск наиболее доходного d_i варианта ведется не при равенстве затрат, а при затратах, попадающих в соответствующий интервал Δ_k . Интервалы одинаковы, зависят от общего разброса значений z , числа сгенерированных вариантов n и в силу особенностей группирования решений нашей задачи определены так:

$$\Delta_k = (z_{\max} - z_{\min}) / (n/2), k = \{1, n/2\}.$$

На разных Δ_k возможно: отсутствие вариантов — такие интервалы пропускаются; наличие единственного варианта — он включается в оптимальные решения; наличие нескольких реализаций,

среди которых выбирается доминирующий i -й вариант так, что:

- 1) $d_i^{(k)} = d_j^{(k)}, \forall j: z_i^{(k)} < z_j^{(k)}$,
- 2) $d_i^{(k)} > d_j^{(k)}, \forall j: |z_i^{(k)} - z_j^{(k)}| \leq \Delta_k$.

Кроме того, если i и j — варианты из разных интервалов, вошедшие в паретовское множество (см. рис. 2) и $d_i > d_j$ при $z_i < z_j$, то вариант j будет худшим по отношению к варианту i , и его необходимо исключить из рассмотрения (рис. 3, выделены).

Таким образом, если Ω — паретовское множество, то

$$\forall i, j \in \Omega: \text{если } d_i < d_j, \text{ то } z_i < z_j.$$

Оптимальные по оценкам (z_j, d_j) варианты могут не удовлетворять требованиям пользователя (исходя из условий в хозяйстве). Поэтому предоставлена возможность предварительно фильтровать генерируемые решения по ограничениям в техническом задании на технологические операции, на культуры и сорта, по ограничениям на основные оценки (затраты — допустимые, доходы, прибыль — не ниже желаемых) и т. д. То есть техническое задание, как и семантический МЯ, используется как фильтр для выбора в текущий вариант проектного решения "правильного" и имеющего реальную ценность значения признака. Это также сокращает время синтеза и число решений, оставляя лишь приемлемые. В дальнейшем именно среди них можно выбрать лучшие.

О структуре автоматизированной системы

При разработке логической модели системы были определены основные функциональные составляющие: модуль генерации и модуль поиска оптимальных вариантов, а также удобные и полезные средства связи с конечным пользователем (рис. 4). Генерация на любой интересующий год реализации полей севооборотов и севооборотов в целом осуществляется после ввода технического задания и учитывает ограничения на значения основных признаков, если они задаются агрономом. Сгенерированные решения можно сохранить в файл, среди этого множества найти оптимальные и также сохранить в файл. Кроме того, оптимизацию можно осуществлять над вариантами, загруженными в файл ранее. Как указывалось выше, отбор вариантов можно осуществлять по разным критериям: урожаем культур, затратам, доходу, прибыли. Для заказчика представляют интерес и реализованы такие режимы: z_{\min} при $d_{\text{орп}}$ снизу (доход $\geq d_{\text{орп}}$); d_{\max} при $z_{\text{орп}}$ сверху (затраты $\leq z_{\text{орп}}$); прибыль p_{\max} при $z_{\text{орп}}$ сверху; отбор при $d_{\text{орп}}$ снизу и $z_{\text{орп}}$ сверху; режим Парето (d_{\max} для каждого интервала по z или z_{\min} для каждого интер-

вала по d или p), а также их последовательные комбинации.

Несколько слов о перспективах использования и дальнейшего развития системы. Кроме использования системы по ее прямому назначению, возможно накопление библиотеки "верных" вариантов сельскохозяйственного земледелия в регионе в целях, опять-таки ее использования по прямому назначению, а также в научных целях для выявления, исследования ранее не изученных закономерностей и связей признаков, имея и анализируя при этом прогнозируемые результаты соответствующих взаимодействий. Желательна реализация

отдаленного доступа конечного пользователя к информационным ресурсам и к системе в целом для проектирования оптимального земледелия в соответствии с введенным техническим заданием.

Список литературы

1. Долматова (Воронова) Л. М. Вариантный метод автоматического синтеза функциональных схем типовых операционных узлов ЭВМ: дисс. ... канд. техн. наук. М., 1977.
2. Voronova L. M., Zhuravlev Yu. I. Variant Model of Long-term Planning of a Farm Agricultural Technology // Pattern Recognition and Image Analysis. 2004. Vol. 14. N 1. P. 150—154.
3. Адаптивно-ландшафтные особенности земледелия Владимирского Ополя / Под ред. А. Т. Волощука. М.: Агроконсалт, 2004. 448 с.

Поздравляем юбиляра!



Заслуженному деятелю науки и техники РФ, заведующему кафедрой САПР Московского государственного горного университета, заведующему базовой кафедрой ФГУП НПО «Альтаир», академику РАЕН, члену редколлегии журнала «Информационные технологии», доктору физико-математических, технических и экономических наук, профессору

Вячеславу Афанасьевичу ГОРБАТОВУ,

известному ученому в области дискретной математики и современных информационных технологий исполнилось **70** лет.

Вся трудовая деятельность В. А. Горбатова связана с отечественным высшим образованием: с 1961 по 1971 гг. — Московский энергетический институт, с 1971 по 1984 гг. — Московский инженерно-физический институт, с 1984 г. по настоящее время — Московский государственный горный университет.

Вячеслав Афанасьевич является автором научного открытия, сделанного в середине 60-х годов прошлого столетия — теории характеристического анализа, положенной в основу отечественной горбатовской научной школы по информационной математике и логическому управлению. Идеи В. А. Горбатова получили широкий резонанс как у нас в стране, так и за рубежом. В рамках этого научного направления он разработал теорию частично упорядоченных систем, новую интеллектуальную технологию решения проблемных задач большой размерности на дискретных структурах. Под его руководством разработан и внедрен в практику целый ряд системных, семантических прикладных программных пакетов и инструментальных компьютерных средств. Его основные научные результаты опубликованы в более чем 500 научных трудах, включая 38 книг (учебников, учебных пособий и монографий) по дискретной математике, информатике и вычислительной технике.

Научные достижения В. А. Горбатова отмечены Международной премией фон Неймана, и множеством как отечественных, так и зарубежных наград, включая медаль Петра I "За заслуги в деле возрождения науки и экономики России" и Георгиевский Крест I-й степени "За заслуги в деле развития науки и экономики России", а также Золотую медаль ООН за большой вклад в области информатизации мирового сообщества.

Вячеслав Афанасьевич уделяет большое внимание подготовке научных кадров: им подготовлено свыше ста кандидатов и докторов наук в области информатики и вычислительной техники.

Он является одним из организаторов журнала «Информационные технологии».

Талантливый ученый, блестящий педагог, высококвалифицированный специалист, чуткий, отзывчивый и обаятельный человек, Вячеслав Афанасьевич пользуется глубоким уважением учеников, коллег, товарищей.

Сердечно поздравляем юбиляра, желаем ему крепкого здоровья, благополучия и новых творческих успехов в научной и педагогической деятельности.

Редколлегия и редакция журнала

CONTENTS

Sharashov V. V. *Technique of a Choice of a Way of Mapping Inheritance Structures to Relational Database* . . . 2

The basic ways of mapping of object inheritance structures to relational database schema are considered. The analysis of influence of these ways on various aspects of system is made. On the basis of specifications the methodology of a choice of an optimum way of mapping is resulted.

Keywords: object-relational mapping, objects persistence, inheritance mapping, relational database management systems, object-oriented programming.

Redreev P. G. *Construction of Applications with the List Components* 7

In the article technology to automate construction from a source relational database a complex-structured tabular data representation with the list components is considered. Intermediate and destination models are formally defined. Examples of generation of user applications are given.

Keywords: relational database, online analytical processing, multidimensional data model.

Tumanov V. E. *Subject-Oriented Science Intelligence Systems* 12

In this paper a concept of the subject-oriented scientific intelligence systems, the basic requirements to such systems and their general technological architecture is discussed. A role of the subject-oriented scientific intelligence systems in scientific information resources is shown. A knowledge warehouse of the subject-oriented scientific intelligence systems and their general technological architecture is discussed. The example of a prototype of subject-oriented scientific intelligence system in physical chemistry of radical reactions is resulted.

Keywords: science databases, current research information systems, science intelligence systems, subject-oriented science intelligence system, knowledge warehouse, embedded expert systems, artificial neural networks.

Sokolov A. V. *Method for Estimation of Maximum Possible Values of Text Retrieval Efficiency Parameters* . . . 18

Method for constructing graphic chart of maximum possible values of search precision (in given data domain) from completeness and approach to estimation of efficiency of information retrieval system search mechanism on base of comparison of real parameters values with fundamentally possible ideal values are offered on base of expert estimated relevance as fuzzy relation.

Keywords: textual information, retrieval efficiency, relevance, completeness, accuracy, evaluation method, fuzzy relation, optimization.

Karpenko A. P., Fedoruk V. G. *A Class of Direct Adaptive Methods for Multi-Criteria Optimization* 24

In this work we propose a new class of direct adaptive methods for solving continuous problem of multi-criteria optimization using approximation of utility function.

Keywords: multi-criteria optimization problem, utility function, direct adaptive method.

Avvakumov V. D. *Optimum Accomodation of Flat Objects of Free Geometric Form* 31

We report about numerical way of solving the problem of the optimum accomodation of flat objects of free geometric form, based on the method of the joint of objects. We report the solvings of the problem of the single-in-line periodic accomodation of simple congruous objects on the band of the free width by Stoyan method and method of the joint. The use of the joint method for complex objects is shown.

Keywords: optimization, accomodation, flat objects of the complex form, area of the possible accomodations, R-function, thick accomodation, supporting function, method of the joint, function of the thick accomodation.

Knyazeva M. A., Plokhikh S. A. *Conception of the Control System of the Dedicated Knowledge Bank of the Program Transformation* 36

The article considers a concept of the control system of dedicated knowledge bank about programs' transformations, current status of the control system; represents and describes architecture-context diagram of the control system and its model and procedure of program's transformation process.

Keywords: control systems, program transformation, knowledge bank, optimizing compiler, transformation subsystem.

Volchenkov E. Ja. *System of the Combinatorial Generation of Test Cases Input/Output: Base Methods and Algorithms* 41

Article is devoted to the description of system of combinatorial test generation an input/output. Several new or modifications of known heuristic methods of selection of effective subsets of test cases is offered. The system is realized in program complex InputSpace TestGenerator.

Keywords: the automated testing of the software, combinatorial test generation, input space, multiport program slots, input domains, package of reference test values, compatibility of test values, boundary categories, profiles of usage, as much as possible differing random (MDPR) sampling, *c*-ary partial combinations, special *c*-MD-PR-coverages.

Gubaydullin I. I., Frid A. I. *Experimental Method of Software Reliability Estimation* 47

Program reliability estimation method depending on the input data distribution laws is considered. The method is based on modelling results of researching program. The method is applied to reliability estimation of the program solving the problem of velocity coefficients foundation for one chemical reaction.

Keywords: reliability, reliability estimation, program modelling.

Levashkina A. O., Porshnev S. V. *Investigation of Supervised Criteria for Image Segmentation Result Evaluation* 52

The methodology for comparing of supervised criteria used for image segmentation results evaluation is described. This methodology is used for comparing several supervised criteria: FOM, Hausdorff distance, Baddeley's distance, DKu, RMS, RI, GCE, VI. In the issue we conclude that it's worthwhile to use Dku, GCE, RI, RMS criteria for image segmentation results evaluation. The methodology for comparing of segmentation algorithms is described. We compare several segmentation algorithms using described methodology (Canny, Log, Zerocross, Prewitt, Roberts, JSEG-algorithm).

Keywords: image segmentation, segmentation evaluation, supervised criteria.

Masalovitch A. A. *Numerical Methods for Detection and Correction of Geometrical Warping on Text Document Images* 57

This paper attempts to propose a novel technique for approximation of document image deformation. The whole image warp is estimated by the means of integration of deformation approximation of each interlinear space of text. Continuous skeletal representation of white fields is used to detect interlinear spaces on the image and to approximate deformation of such spaces. The use of skeletal representation makes proposed technique stable to local irregularities in the text lines of the document. For approximation of interlinear spaces deformation cubic Bezier curves are used. For approximation of whole image warp 2-dimensional cubic Bezier patch is used.

Keywords: text recognition, OCR, image preprocessing, warped image restoration, continuous border-skeletal representation, outer skeleton, Bezier patch.

Arkhipova O. E. *The Concept of Regional Ecological Information System of Monitoring* 62

In clause main principles of creation of system multilevel regional-adapted complex monitoring are described. As modelling object it is considered sea and ground ecological system, where various water both ground complexes and kinds of water use are presented, and water resources act as one of limiting factors of social and economic development.

The result of the research spent within the limits of the Program of basic researches of Branch of sciences about the Earth of the Russian Academy of Science, will allow to define ways of stabilization of an ecological situation, to develop approaches and to give the recommendation on steady operation of resources in zones with an extreme environment.

Keywords: geoinformation technologies, geoinformation systems (GIS), Databases (DB), Knowledge bases (KB), Base of geodata (BGD), system of ecological monitoring.

Bushmeleva K. I., Plusnin I. I. *Computer-Aided Geoinformation System of Gas Pipelines Working Condition Monitoring* 68

The article highlights main features of geoinformation system of monitoring of working conditions of gas pipelines. The system has been developed on the base of aircraft soft and hardware diagnostic complex and it is used to detect flaws and gas leakages.

Keywords: monitoring, gas pipeline, laser locator of gas leakage, soft and hardware diagnostic complex, geoinformation system, software.

Germashev I. V., Vasilyeva V. D., Derbisher V. E., Derbisher E. V., Silina A. Yu. *The Application of Theory of Fuzzy Sets for Decision of Chemical-Engineering Problems* 72

The method of formalization and analysis of information about chemical materials and structures is presented with applying of the fuzzy data device theories device. This method is used for task solution of the chemical technology with the aim of material ranking depending on the chosen sign and their property prognosis.

Keywords: Fuzzy information, fuzzy data theory, chemical material, functional activity, formalization, forecasting characteristic, ranking, testing.

A new approach is studied to modeling the optimum agricultural designs. On the basis of agricultural system morphological research morphological box is formed. It's containing structure and semantic connections between basic components of the design object (including operations of cultivation technologies). The model is used to build a set of resolution for separated fields, by the method of morphological synthesis. Based on the set, the optimum planning variants of agricultural activity are found. The software system is offered optimal system design in accordance with the entered information and may be used for prognostication economics activities in conditions of any agriculture and technological restrictions of the farmer.

Keywords: agricultural system, farm plots, soil types, cultivation technologies, agricultural plants interchanging for sowing, rules of land use, modeling, morphological approach, basic components, marks for components, signals for marks, generative principles, the rules for putting out, semantic connections, optimization, computer-aided agriculture design.

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *О. А. Ефремова*.
Корректор *Е. В. Комиссарова*

Сдано в набор 06.03.2009. Подписано в печать 16.04.2009. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 10,78. Уч.-изд. л. 11,84. Заказ 312. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"
142110, Московская обл., г. Подольск, ул. Кирова, 15