

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

9(145)
2008

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Александров А. Е. Разработка и проектирование прикладных программных приложений на основе порождающего программирования 2
Гаврилов А. В., Павлова Е. А. Формализация проектирования сложных информационных систем на основе анализа функциональных интерфейсов 9

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Шерыхалина Н. М. Применение фильтрации численных результатов для увеличения надежности САПР 16
Поляков Б. Н. К созданию САПР технологии правки и оборудования ролико-правильных машин 22
Зимнухова Ж. Е., Немтинов В. А. О подходе к построению автоматизированной информационной системы поддержки принятия решений для проектирования процессов производства изделий из металлов 29

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И ДИАГНОСТИЧЕСКИЕ СИСТЕМЫ

- Горшков А. П., Грызлова Т. П. Система диагностики состояния сложных технических объектов по характерным последовательностям цифровых сигналов . . . 35
Малыхина Г. Ф., Меркушева А. В. Метод анализа независимых компонент для восстановления формы сигналов по их смеси 39

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Балакирев Н. Е., Гирин Б. Б. Основные концепции языка описания алгоритмов анализа естественно-языкового текста 42
Семенов Н. Н. Спектральные методы обработки пространственных данных для классификации целей 47
Лапшин В. П., Жуков И. С., Апилов А. Х. Количественная оценка размытости и асимметрии функций принадлежности нечетких множеств 51
Ковшов Е. Е., Машковцев А. В. Рационализация подбора медицинского персонала в лечебном учреждении путем применения генетических алгоритмов 56

ОПТИМИЗАЦИЯ

- Филиппова А. С., Кузнецов В. Ю. Задачи о минимальном покрытии ортогональных многоугольников с запретными участками 60
Орехов Ю. В., Орехов Э. Ю., Заминова А. Р. Исследование эффективности операторов генетического алгоритма при поиске экстремальных решений в бинарных и вещественных пространствах 65

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

- Девятков Д. Х., Каплан Д. С., Иванов К. А. Информационная система руководителя металлургического предприятия 72
Contents 79
Приложение. Авдошин С. М., Савельева А. А. Криптотехнологии Microsoft

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.
ЗАЛЕЩАНСКИЙ Б. Д.
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.
МУХТАРУЛИН В. С.
НАРИНЬЯНИ А. С.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
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.

Аннотации статей размещены на сайте журнала по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

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

УДК 0.04.42

А. Е. Александров, д-р техн. наук, проф.,
Московский государственный университет
приборостроения и информатики

Разработка и проектирование прикладных программных приложений на основе порождающего программирования

Представлено описание архитектуры базового программного обеспечения, включающего расширяемую библиотеку компонентов, проблемно-ориентированную среду программирования, реализованную на основе высокоуровневого языка, позволяющего создавать предметно-ориентированные библиотеки моделей, поддерживаемые самим пользователем. Описана технология разработки прикладных программных приложений на основе предложенных принципов и разработанного базового программного обеспечения. Приведены результаты расчетов, выполненных с помощью разработанного программного приложения.

Ключевые слова: порождающее программирование, библиотека компонентов, проблемно-ориентированный высокоуровневый язык, метод конечных элементов.

Введение

В развитии программной технологии начинается новый виток эволюции, связанный с переходом от низкоуровневых методов и технологий исследований программных систем к высокоуровневому анализу их построения. Получившая развитие в середине 80-х годов парадигма структурного проектирования уступила место в 90-х годах объектно-ориентированным методам анализа и проектирования, обладающих потенциальной возможностью множественного использования.

Однако несмотря на многообещающие надежды на создание универсальных механизмов по разработке программных продуктов внедрение методов объектно-ориентированной технологии ограничено лишь низкоуровневыми конструкциями (графические интерфейсы, базы данных).

В последнее время появилось новое направление в программировании, получившее название *порождающее* [1]. Парадигма порождающего программирования основана на моделировании се-

мейства программных систем. Модели систем собираются из компонентов, которые образуют расширяемые многократно используемые библиотеки. Идея использования компонентов — это минимизация дублирования кода при максимальном его повторном использовании. Сборка модели семейства проводится с помощью высокоуровневого конфигурационного описания, хранящегося в базе знаний.

Схожие принципы были использованы группой авторов, работающих в Московском государственном университете приборостроения и информатики, при разработке и проектировании прикладных программных приложений, ориентированных на решение задач математической физики на основе метода конечных элементов (МКЭ). Отметим основные из них:

- компонентная технология, оформленная в виде базовой расширяемой библиотеки компонентов, обеспечивающей проблемно-ориентированную поддержку программирования;
- расширяемая среда программирования, одним из основных элементов которой является проблемно-ориентированный высокоуровневый язык — язык управления вычислениями (ЯУВ) [2];
- предметно-ориентированная библиотека моделей, поддерживаемая самим исследователем. Модель представляет собой высокоуровневое конфигурационное описание, реализованное на ЯУВ и хранящееся в библиотеке исследователя.

В настоящей статье представлены описание архитектуры базового программного обеспечения и технология разработки прикладных программных приложений на его основе.

Архитектура базового программного обеспечения

Базовое программное обеспечение организовано в виде расширяемой библиотеки компонентов. На основе анализа формирования дискретной модели с помощью МКЭ было выделено понятие "*программный блок*", представляющее аналог компонента программного обеспечения, обеспечивающего многократное использование. Совокупность программных блоков определяет множество элементов, на котором в рамках конечно-элементной модели могут быть построены разнообразные дискретные модели. Имея в своем распоряжении совокупность программных блоков и соответствующий инструмент, определяющий последовательность вхождения их в глобальную

матрицу и глобальный вектор, исследователь может самостоятельно формировать дискретную модель, которая ассоциирована с исходной математической моделью.

Реализация алгоритма формирования глобальной матрицы и глобального вектора обеспечивается с помощью проблемно-ориентированного высокоуровневого языка — ЯУВ. ЯУВ разрабатывался как объектно-ориентированный язык.

Множество программных компонентов (программных блоков) совместно с методами для создания, поддержки и сопровождения программного приложения образуют базовую библиотеку классов. Базовая библиотека классов имеет две ветви наследования: иерархию наследования классов описания отдельных задач (базовый класс **CTask**) и иерархию наследования классов описания комплексных моделей (базовый класс **CComplexModel**). В состав базовой библиотеки классов помимо базового класса иерархии **CTask** входит также класс описания задач, решаемых с помощью МКЭ — **CFEMTask**.

Отметим, что разработка программного обеспечения включает два цикла разработки: первый цикл — для проектирования и реализации базовой библиотеки, предназначенной для многократного использования, и второй цикл — для проектирования и разработки конкретного программного приложения.

Инициирование расширения базовой библиотеки может проводиться как исследователем в целях расширения функциональности предметной области — создания новых функциональных компонентов, так и системным программистом для реализации системных функций — визуального представления, диагностики ошибок, управления версиями, генерации кода.

Созданные на ЯУВ приложения образуют двухуровневую библиотеку моделей, которая пополняется самим исследователем. Содержание первого уровня составляет библиотека моделей, представляющая реализацию различного типа уравнений. Тип уравнения реализуется в рамках разработанной алгоритмической схемы, оформленной в виде метода соответствующего класса. При необходимости методы могут наследоваться от родительского класса. Например, класс решения пластической задачи наследует метод решения упругой задачи. Напомним, что все созданные исследователем классы являются наследниками базового класса — **CFEMTask**.

В качестве примера ниже приведены описания отдельных методов класса **CFEMTask**:

- **ADD_COMPONENT (Component ID)** — обеспечивает добавление компонента дискретного уравнения в описание задачи;

- **GENERATE_SLAU** — обеспечивает формирование системы линейных алгебраических уравнений (СЛАУ) МКЭ;
- **SOLVE_SLAU** — выполняет решение СЛАУ МКЭ.

Идентификатор метода **ADD_COMPONENT (ComponentID)** определяет соответствующий программный блок. Программные блоки разделены на группы в зависимости от типа решаемой задачи. Заметим, что при формировании системы линейных алгебраических уравнений (СЛАУ) некоторой отдельно взятой задачи существует ограничение на совместное использование программных блоков, например, тепловых задач и механических задач (в силу различной размерности СЛАУ). В таблице в качестве примера представлены программные блоки и соответствующие им идентификаторы компонентов для тепловой задачи.

В таблице использованы следующие обозначения: FE — число конечных элементов; $q \left[\frac{\text{Вт}}{\text{м}^2} \right]$ — плотность теплового потока; $[N]$ — вектор глобальных функций формы; $\lambda(T) \left[\frac{\text{Вт}}{\text{м} \cdot ^\circ\text{C}} \right]$ — коэффициент теплопроводности материала; $c(T) \left[\frac{\text{Дж}}{\text{кг} \cdot ^\circ\text{C}} \right]$ — теплоемкость материала; $\rho(T) \left[\frac{\text{кг}}{\text{м}^3} \right]$ — плотность материала; $Q_V \left[\frac{\text{Вт}}{\text{м}^3} \right]$ — плотность тепловыделения.

Создание классов-наследников класса CFEMTask и производных от них классов. Структура и возможности создаваемого исследователем класса-наследника **CFEMTask** определяются, во-первых, набором программных блоков, с помощью которых могут формироваться алгоритмы решения задач, и, во-вторых, множеством разработанных алгоритмов, которые можно использовать для построения решения комплексной задачи.

Группа	Идентификатор компонента	Компонент дискретного уравнения	Описание
Temp	cmQ	$\sum_{FE} \int_S [N]^T q ds$	Граничное условие — тепловой поток
Temp	cmK	$\sum_{FE} \int_V \sum_{i=1}^3 \frac{\partial [N]^T}{\partial x_i} \frac{\partial [N]}{\partial x_i} \lambda(T) dV$	Матрица теплопроводности
Temp	cmC	$\sum_{FE} \int_V [N]^T [N] c(T) \rho(T) dV$	Матрица теплоемкости
Temp	cmQV	$\sum_{FE} \int_V [N]^T Q_V dV$	Источник тепловыделения

Для построения алгоритма решения задачи используются методы, которые наследуются от класса **CFEMTask**. При этом заданный для конкретного класса набор программных блоков определяет допустимые значения параметра метода **ADD_COMPONENT**.

Описание в формальном виде создания класса-наследника представлено на рис. 1.

В качестве класса-родителя может выступать как класс **CFEMTask**, так и производные от него классы библиотеки исследователя.

В списке **COMPONENTS_LIST** между фигурными скобками через запятую перечисляются идентификаторы программных блоков, использование которых допустимо в методах класса-наследника. В данный список могут быть включены любые программные блоки, входящие в базовую библиотеку.

Создание классов-наследников класса CComplexModel и производных от них классов. Созданные на первом уровне модели могут рассматриваться как компоненты для создания более сложных — комплексных — моделей второго уровня. Второй уровень представляет библиотеку комплексных моделей.

Структура и возможности создаваемого исследователем наследника класса **CComplexModel** определяется перечнем задач, которые входят в комплексную модель, и множеством разработанных алгоритмов решения комплексных задач.

Описание создания класса-наследника представлено на рис. 2.

При создании иерархии наследования классов описания комплексных моделей исследователю предоставляется возможность использования множественного наследования, что позволяет создавать описание комплексной модели путем объединения ранее разработанных моделей. Для построения алгоритма решения комплексной задачи используются методы, которые наследуются от класса **CComplexModel** и обеспечивают передачу данных между различными задачами комплексной модели.

В качестве классов-родителей могут выступать как класс **CComplexModel**, так и производные от него классы библиотеки исследователя. В списке **TASK_LIST** между фигурными скобками через запятую объявляются объекты классов, входящих в иерархию наследования базового класса **CTask**. Объявление объекта класса предполагает указание имени класса объекта и задание уникального идентификатора объекта. Возможно включение в список нескольких объектов одного и того же класса. Недопустимым является использование объектов классов **CTask** и **CFEMTask**, которые используются для организации иерархии наследо-

```
CLASS ClassName(BaseClassName)
    COMPONENTS_LIST = {cmID_1, cmID_2, ... ,
                        cmID_N}
    METHOD MethodName_1()
    METHOD MethodName_2()
    ...
    METHOD MethodName_M()
END_CLASS
```

Рис. 1

```
CLASS ClassName(BaseClassName1,
BaseClassName2,...,BaseClassNameN)
    TASK_LIST = {ClassName_1 TaskID_1,
                  ClassName_2 TaskID_2,
                  ...
                  ClassName_N TaskID_N}
    METHOD MethodName_1()
    METHOD MethodName_2()
    ...
    METHOD MethodName_K()
END_CLASS
```

Рис. 2

вания классов описания задач, но не позволяют решать конкретные задачи.

При создании классов-наследников **CComplexModel** отметим две особенности:

- во-первых, в состав данных класса-наследника включаются объекты классов, входящих в иерархию наследования базового класса **CTask**;
- во-вторых, в перечень методов класса-наследника включается метод, реализующий алгоритм решения комплексной задачи.

Таким образом, каждый класс иерархии базового класса **CComplexModel** включает в себя перечень задач, входящих в комплексную модель, описание связей между этими задачами и алгоритм решения комплексной задачи.

На рис. 3 приведен пример реализации алгоритма решения теплового нестационарного расчета на ЯУВ.

В приведенном примере имеем два цикла: один цикл — по временным шагам и второй — по линейным задачам. При формировании СЛАУ матрицы теплопроводности и теплоемкости зависят от температуры, поэтому добавление компонентов cmK и cmC происходит внутри цикла по линейным задачам и по временным шагам. Граничное условие $cmALFA$ не меняется от температуры, но изменяется по времени, поэтому добавление компонента $cmALFA$ происходит внутри цикла по временным шагам, но вне цикла по линейным за-


```

CLASS CFEMTempTask(CFEMTask)
  COMPONENTS_LIST={cmK, cmC, cmQ, cmALFA}
  // Нестационарный тепловой расчет
  METHOD SolveTempNstTask( )
END_CLASS
CLASS CFEMTempTask METHOD SolveTempNstTask( )
ODU1(_KRANK_NIKOLSON_)
  BEGIN_CICLE
    GET_TIME()
    ADD_COMPONENT(cmALFA)
    BEGIN_CICLE
      ADD_COMPONENT(cmK)
      ADD_COMPONENT(cmC)
      GENERATE_SLAU()
      SOLVE_SLAU()
      CHECK_CONVERGENCE()
    END_CICLE
    SAVE_RESULT()
  END_CICLE
END_METHOD

```

Рис. 3

дачам. Таким образом, размещая методы добавления компонентов необходимым образом, исследователь имеет возможность создавать модификации используемых алгоритмов.

Метод **ODU1** (*MethodID*) совместно с методом **GENERATE_SLAU**() преобразует ОДУ первого порядка в СЛАУ. В рассматриваемом варианте использована схема Кранка—Николсона [3] — **_KRANK_NIKOLSON_**.

Метод **CHECK_CONVERGENCE**() проводит оценку сходимости итерационного процесса. В том случае, если сходимость достигнута, данный метод прерывает выполнение текущего цикла. Если сходимость не достигнута, то прерывание цикла осуществляется по максимальному числу итераций. Совместно с операторами начала и конца цикла (**BEGIN_CICLE** и **END_CICLE**) служит для организации циклов сходимости задачи.

Результаты решения СЛАУ, полученные с помощью метода **SOLVE_SLAU**, сохраняются с использованием метода **SAVE_RESULT** для каждого временного шага задачи.

```

CLASS CTempComplexModel(CComplexModel)
  TASK_LIST={CFEMTempTask TempTask}
  // Нестационарный тепловой расчет.
  METHOD SolveTempNstTask( )
END_CLASS
CLASS CTempComplexModel METHOD SolveTempTask( )
  TempTask LOAD_DATA(TempTask.prj)
  TempTask SolveTempNstTask( )
END_METHOD

```

Рис. 4

Заметим, что в рамках метода, создаваемого исследователем класса-наследника **CFEMTask**, реализуется только алгоритм решения задачи МКЭ, сама задача МКЭ реализуется в рамках решения комплексной задачи. Это связано с тем, что помимо алгоритма реализации в задачу МКЭ входят исходные данные: геометрическая модель дискретного объекта, описание физических параметров материалов и граничных условий, связи материалов и граничных условий с фрагментами и границами геометрической модели, начальные условия, расчетные времена задачи. Все эти исходные данные задаются при формировании конечно-элементной задачи обычными инструментальными средствами и оформляются в виде описания в файле проекта в виде **DATA** (< название проекта > .prj). Этот сформированный файл загружается при реализации комплексной модели конечно-элементной задачи (рис. 4).

Инициализация расчета, связанного с компонентной моделью, сводится к объявлению объекта класса, описывающего данную модель, и вызову метода, содержащего алгоритм решения комплексной задачи.

Пример разработки программного приложения

Далее приведен пример разработки программного приложения на основе предложенного подхода, взятый из работы [4]. Реализация описанной комплексной модели с помощью известных коммерческих пакетов конечно-элементного анализа (таких как, например, Ansys) либо невозможна, либо требует разработки нового программного приложения на уровне алгоритмического языка, что в свою очередь является весьма ресурсоемким, а потому дорогим средством.

Постановка задачи. Рассмотрим узел фланцевого соединения, герметичность которого обеспечивается с помощью прокладки, выполненной из пластичного материала (в рассмотренном варианте — из никеля). Начальная затяжка шпильки имеет такое значение, что прокладка переходит в пластичное состояние. При рабочей эксплуатации на пластичное состояние прокладки накладываются нестационарные температурные воздействия, которые, с одной стороны, изменяют теплофизические параметры материала прокладки, а с другой стороны, вызывают дополнительные термические деформации, обусловленные неравномерным температурным полем. Сочетание этих факторов, в конечном счете, может привести к разгерметизации фланцевого соединения. Предварительный анализ показал, что для оценки влияния этих воздействий необходимо выполнить:

- упругий изотермический расчет;

- пластический изотермический расчет;
- нестационарный тепловой расчет;
- термопластический расчет с учетом изменения механических характеристик;
- термоупругий расчет;
- термопластический расчет с учетом изменения механических характеристик и силовых граничных условий.

Первый этап. Упругий изотермический расчет

Первый этап является предварительным и служит для определения шага дискретизации для сети конечных элементов. В качестве такой оценки была выбрана величина

$$\Delta\sigma = \sigma_i - \sigma_T,$$

где σ_i — интенсивность вычисленных упругих напряжений; σ_T — предел текучести.

Области, в которых величина $\Delta\sigma$ принимает положительные значения, являются областями пластических деформаций, а большие или меньшие значения положительной величины $\Delta\sigma$ позволяют качественно оценить предполагаемые пластические деформации и, соответственно, определить плотность сгущения сети конечных элементов.

На основе проведенного на первом этапе анализа для геометрической модели строится новая сеть конечных элементов, которая в дальнейшем используется для проведения расчетов на последующих этапах.

Второй этап. Пластический изотермический расчет

Основная цель второго этапа состоит в определении характера деформации прокладки, обусловленного только силовым нагружением. Полученная на данном этапе информация о неравномерности деформации прокладки как вдоль радиального сечения, так и в осевом направлении позволит в дальнейшем при проведении термопластических расчетов оценить влияние на пластическое деформирование прокладки таких факторов, как температурные изменения механических характеристик материалов и температурные деформации. При пластическом изотермическом расчете фланцевого соединения задача сводилась к нахождению НДС деталей фланца.

Для определения пластических деформаций была использована теория течения [5]. Пластическая задача в этом случае рассчитывается как итерационная последовательность упругих задач с начальной деформацией.

Третий этап. Нестационарный тепловой расчет

Основное внимание при анализе результатов теплового расчета было уделено оценке влияния

нестационарного теплового процесса на температурные поля в области прокладки, обеспечивающей герметичность, а также в области затяжки, так как температурные деформации фланца и шпильки в состоянии существенно изменить картину силового нагружения фланцевого соединения.

Результаты теплового расчета на последующих этапах использовались для проведения термопластического и термоупругого расчетов.

Четвертый этап. Термопластический расчет с учетом изменения механических характеристик

Силовые граничные условия для термопластического расчета совпадают с условиями, заданными для пластического изотермического расчета. Но на данном этапе механические характеристики материалов берутся для значений температур, вычисленных в ходе нестационарного теплового расчета.

Пятый этап. Термоупругий расчет

Цель проведения термоупругого расчета, исходными данными для которого является температурное поле, рассчитанное в ходе нестационарного теплового расчета, состоит в определении температурных деформаций фланца и шпильки. Полученные в результате расчета значения напряжений в площади контакта гайки и фланца, обусловленные неравномерностью температурного поля в области затяжки, позволяют учесть изменение силовых граничных условий в данной области.

Шестой этап. Термопластический расчет с учетом изменения механических характеристик и силовых граничных условий в области затяжки

На заключительном этапе, так же как для ранее проведенного термопластического расчета (четвертый этап), расчет выполнялся с учетом температурных изменений механических характеристик материалов.

Кроме этого, при задании значений распределенных усилий в области затяжки были учтены изменения силовых граничных условий, обусловленные рассчитанными на предыдущем этапе температурными деформациями шпильки и фланца.

Сопоставление результатов, полученных на данном этапе, с результатами пластического изотермического расчета и термопластического расчета с учетом изменений механических свойств материалов позволяет сделать вывод о степени влияния температурных деформаций в области затяжки на увеличение неравномерности деформации прокладки.

Таким образом, информация, полученная при последовательном проведении расчетов по предложенной комплексной модели, позволяет про-

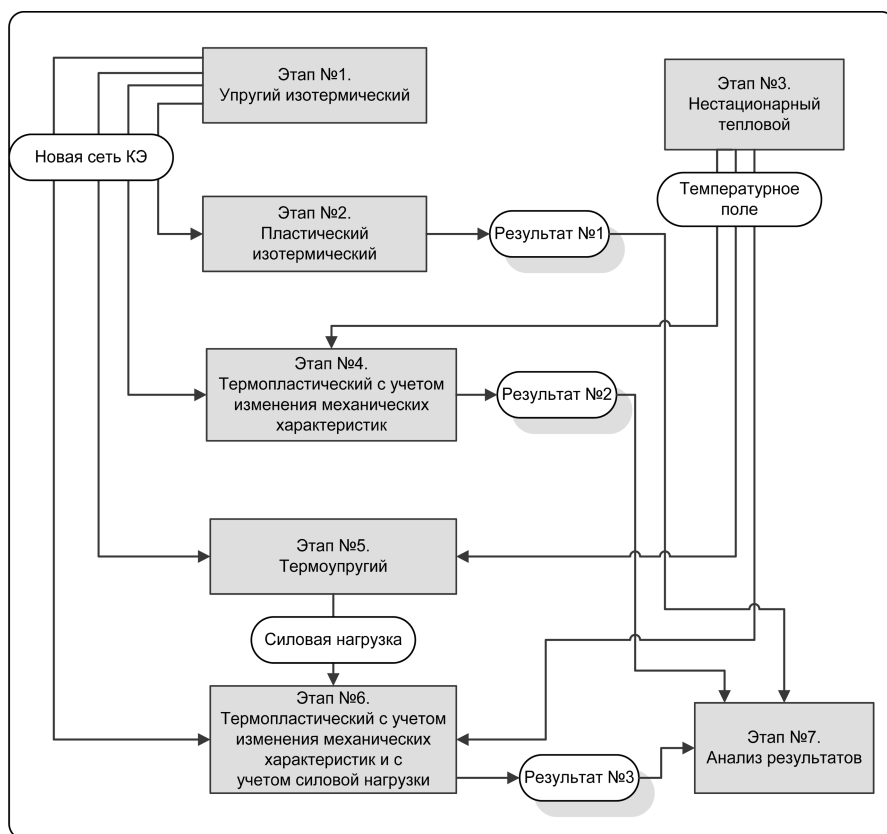


Рис. 5

вести всесторонний анализ процесса деформирования фланцевого соединения с учетом влияния на него тепловых воздействий.

Следует указать, что в силу особенности модели процесса расчета герметичности [4] каждый из указанных этапов должен быть реализован в рамках своей геометрической модели.

Разработка программного приложения. Создадим на основе описанной выше последовательности расчетов конфигурационное описание программного приложения в виде блок-схемы, представленной на рис. 5.

Отметим, что для реализации комплексных моделей второго уровня генерации необходимо создать три компонента первого уровня: решение упругой, пластической и нестационарной тепловой задачи.

Реализации упругой и нестационарной тепловой задачи могут быть выполнены как наследники на основе базового класса — **CFEMTask**. Реализация пластической задачи выполнена как наследник на основе реализации упругой задачи. Иерархия наследования классов описания этих задач представлена на рис. 6. Следует указать, что совокупность представленных реализаций первого уровня генерации можно рассматривать как конфигурационное описание первого уровня.

Выделенных трех компонент первого уровня достаточно для реализации комплексной задачи.

Рассмотрим комплексную реализацию задачи как множество, входящих в него моделей: этап 1 — **CElasticComplexModel**; этап 2 — **CPlasticComplexModel**; этап 3 — **CTempComplexModel**; этап 4 = этап 2 + этап 3 — **CTermoPlasticComplexModel**; этап 5 = этап 1 + этап 3 — **CTermoElasticComplexModel**; этап 6 = этап 3 + этап 5 — **CTermoPlasticComplexModel**.

Каждый этап решения комплексной задачи потребует создания соответствующего наследника иерархии описания комплексных задач, которые используют один или несколько классов описания отдельных задач:

- выполнение первого этапа (упругий изотермический расчет) потребует использования класса **CFEMElasticTask**;
- выполнение второго этапа (пластический изотермический расчет) — класса **CFEMPlasticTask**;
- выполнение третьего этапа (тепловой нестационарный расчет) — класса **CFEMTempTask**;
- на четвертом этапе (термопластический расчет с учетом изменения механических характеристик материалов) необходимо создать класс описания комплексной модели, с помощью которого последовательно реализуется решение тепловой нестационарной задачи (класс **CFEMTempTask**) и пластической задачи (класс **CFEMPlasticTask**), в которую как исходные данные передаются результаты решения тепловой задачи;
- на пятом этапе (термоупругий расчет) аналогичным образом необходимо создать класс

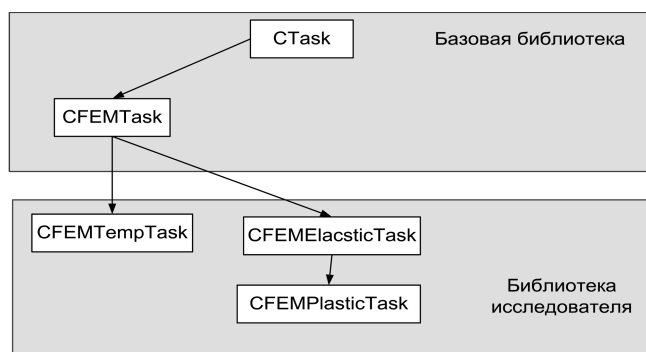


Рис. 6

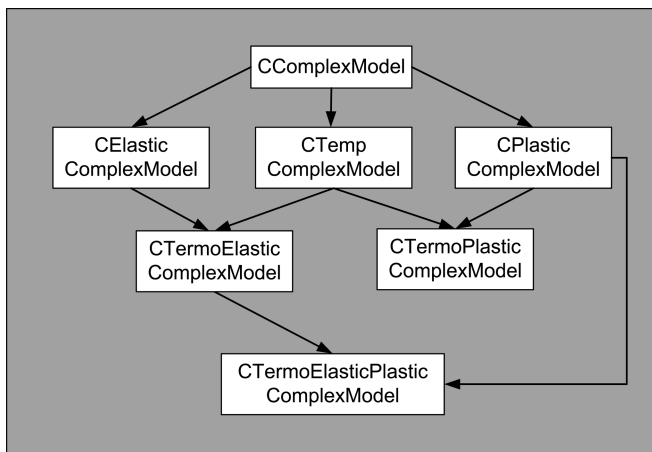


Рис. 7

```

CLASS CTermoElasticPlasticComplexModel
(CTermoElasticComplexModel, CPlasticComplexModel)
  TASK_LIST = {}
  // Шестой этап. Термопластический с изменением
  // силовых граничных условий
  METHOD SolveProblem( )
END_CLASS
CLASS CTermoElasticPlasticComplexModel METHOD
SolveProblem( )
  PlasticTask LOAD_DATA(PlasticTask.prj)
  SolveTermoElasticTask( )
  LINK_TEMP_WITH_MECH(TempTask, PlasticTask)
  LINK_MECH_WITH_MECH(ElasticTask,
PlasticTask)
  PlasticTask SolvePlasticNstPTask()
END_METHOD
  
```

Рис. 8

описания комплексной модели, включающий классы **CFEMTempTask** и **CFEMElasticTask**;

- на шестом этапе (термопластический расчет с учетом изменений механических характеристик и силовой нагрузки) комплексная модель должна включать классы **CFEMTempTask**, **CFEMElasticTask** и **CFEMPlasticTask**.

Иерархическая схема описания отдельных задач, включающая разработанные для решения комплексной задачи классы библиотеки исследователя, представлена на рис. 7.

В качестве примера на рис. 8 приведена реализация на ЯУВ класса описания комплексной модели для шестого этапа.

Решение задачи на шестом этапе помимо разработки соответствующего класса описания комплексной модели потребовала дополнение базового класса иерархии **CComplexModel** методом **LINK_MECH_WITH_MECH**, который должен обеспечивать передачу данных между механическими задачами, что необходимо для учета изме-

нений силовых нагрузок при решении пластической задачи.

Ниже приведены результаты расчетов, выполненные с помощью разработанного программного приложения.

Результаты расчетов. На рис. 9 представлены результаты расчетов, полученные на этапе "Упругий изотермический расчет" в области затяжки фланцевого соединения. Результаты представлены в виде поля распределения величины $\Delta\sigma$, которая определяется как разность между значениями предела текучести материала и интенсивностью вычисленных значений упругих напряжений для каждого узла конечного элемента.

Графики изменений значений перемещений (ось ординат) вдоль окружности (ось абсцисс) приведены на рис. 10. Там же для сравнения приведено изменение перемещений вдоль окружности для изотермического пластического расчета и для термопластического расчета при изменении

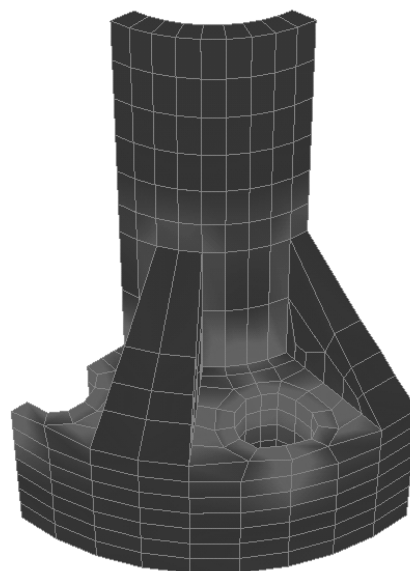


Рис. 9

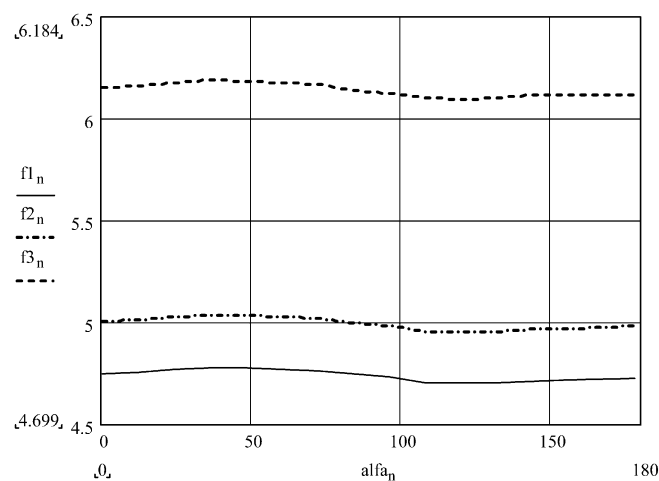


Рис. 10

механических параметров материалов. Термопластический расчет с учетом изменений характеристик материалов и силовых граничных условий выявил увеличение неравномерности деформации на прокладке по сравнению с изотермическим пластическим расчетом.

Результаты подобных исследований на математических моделях могут стать методической основой при определении ресурса узлов и деталей различных фланцевых соединений, их работоспособности в самых экстремальных условиях. Более подробно описание данной проблемы приведено в работе [5].

На рис. 10 обозначены f_1 — перемещения прокладки, полученные для изотермического пластического расчета; f_2 — перемещения прокладки, полученные для термопластического расчета с учетом изменений механических параметров материалов от температуры; f_3 — перемещения прокладки, полученные для термопластического расчета с учетом изменений механических параметров материалов и силовых граничных условий; α — угол в градусах, изменяемый по окружности.

Выводы

- На основе разработанной базовой библиотеки генерируется конкретное программное приложение по его конфигурационному описанию.
- Конфигурационное описание программного приложения сохраняется в программной реализации более высокого уровня иерархии, чем при использовании низкоуровневых языков (как, например, C++), что позволяет накапливать знания о предметной области в виде программного кода.

- Компоненты базовой библиотеки используются многократно, не для одного программного приложения, а для семейства программных систем.
- Расширение базовой библиотеки приводит к существенному расширению ее возможностей за счет создания на ее основе не одной, а целого семейства программных систем.
- В процессе разработки программного приложения используются не только функциональные компоненты базовой библиотеки для расширения функциональности предметной области, но и различные системные компоненты для реализации визуального представления, диагностики ошибок, управления версиями, генерации кода.

Список литературы

1. Чарнецкий К., Айзенкер У. Порождающее программирование: Методы, инструменты, применение: Пер. с англ. СПб.: Питер, 2005. 730 с.
2. Александров А. Е. Автоматизированная система формирования, анализа, хранения и сопровождения математических моделей. Принципы построения и архитектура системы // Приборы и системы. Управление, контроль, диагностика. 2004. № 12. С. 1—5.
3. Александров А. Е., Востриков А. А. Автоматизированная система формирования, анализа, хранения и сопровождения математических моделей. Средства создания математических моделей — язык управления вычислениями // Приборы и системы. Управление, контроль, диагностика. 2005. № 8. С. 1—5.
4. Стренг Г., Фикс Дж. Теория метода конечных элементов. М.: Мир, 1977.
5. Александров А. Е. Методология проектирования прикладных адаптивных программных систем с использованием многоуровневой инструментальной среды. // Дис. ... д. т. н. 05.13.11., 05.13.18. М., 2006.
6. Малинин Н. Н. Прикладная теория пластичности и ползучести. М.: Машиностроение, 1968.

УДК 004.05:004.512

А. В. Гаврилов, канд. техн. наук,

Е. А. Павлова, аспирант, Московский инженерно-физический институт (ГУ)

Формализация проектирования сложных информационных систем на основе анализа функциональных интерфейсов

Описывается комплексное использование методов формального проектирования для обеспечения синтаксической и семантической целостности компонентной модели архитектуры сложной информационной системы. Верификация корректности взаимодействия компонентов программного обеспечения проводится за счет проверки соответствия сигнатур функций в рамках предложенной алгебры функциональных интерфейсов. Для верификации обеспечения декларированной функциональности используется семантический анализ контрактов интерфейсов в рамках адаптированного аппарата аксиоматической системы Хоара. Приводится демонстрация подхода на примере верификации синтаксической корректности и семантической целостности модели архитектуры упрощенной системы электронной торговли.

Ключевые слова: верификация, целостность, формальный метод, компонентная модель, функциональный интерфейс, контракт.

Введение

При автоматизации комплексных производств и создании сложных информационных систем существенно важными становятся соответствие предоставляемой функциональности автоматизированных информационных систем (ИС) реальным задачам предприятия, обеспечение эффективности, гибкости, управляемости. На бизнес-уровне организации производства эта задача решается за счет использования сервисной модели и стандартизации коммуникаций. Аналогично, на технологическом уровне применение сервисно-ориентированной архитектуры и унификации сообщений обеспечивают эффективные средства построения интегрированных ИС, отвечающих современным требованиям индустрии. Использование архитектурных шаблонов проектирования и промышленных функциональных модулей и целых систем ряда компаний позволяет построить технологичный управляемый процесс разработки таких ИС в рамках комплексной автоматизации.

Однако открытыми остаются вопросы формального обеспечения корректности построения компонентной модели с точки зрения синтактико-семантической интерпретации реализуемой функциональности, позволяющего дальнейшую автоматизацию разработки подобных ИС. Частично проблема решается использованием методологии *разработки, управляемой моделями (Model-Driven Development — MDD)* [1]. При этом указанная методология, оставляя большую гибкость в средствах верификации модели, не предоставляет формальных методов проверки целостности компонентной архитектуры. Используемые в ряде современных MDD-систем разработки эвристические методы, предоставляя эффективные средства для конкретного класса задач, не дают общего решения указанной проблемы [2–5].

Целью работы являются формализация и апробация подхода к верификации корректности компонентной модели архитектуры информационных систем, на основе комплексного использования предложенной авторами алгебры функциональных интерфейсов и адаптированного аппарата аксиоматической системы Хоара.

Приведенный модельный пример позволяет продемонстрировать суть подхода при анализе модели архитектуры упрощенной системы поддержки электронной коммерции.

Комплексное применение теоретико-множественной алгебры функциональных интерфейсов и методов контрактного проектирования на основе аксиоматической системы Хоара [6] позволило обеспечить формальную корректность взаимодействия компонент программного обеспечения на уровне верификации синтактического протокола обмена [7, 8] и проверку предоставления декларированной функциональности на уровне семантического анализа контрактов [9]. При этом сквозное использование синтактико-семантического метааппарата

$\lambda\zeta$ -исчисления [10, 11] на всех этапах построения и анализа математической модели обеспечивает непротиворечивость и достаточность выбранной формальной системы. Краткое описание используемого $\lambda\zeta$ -исчисления приведено в приложении.

В дальнейшем, благодаря использованию единого формального аппарата для задания и анализа синтаксиса и семантики интерфейсов компонент, предложенный подход может быть легко автоматизирован и масштабирован для анализа моделей архитектуры сложных ИС, например, в рамках систем модельно-интегрированных вычислений (*Model-Integrated Computing*) [1].

Синтактическая совместность компонент

Адаптация предложенного М. Abadi и L. Cardelli [11] способа описания объектов путем фиксации множества интерфейсных функций дает возможность перейти к единообразному одноуровневому представлению компонент в терминах их функциональных интерфейсов. Такое представление компонентной модели допустимо, так как интерфейсы компонент, как и объекты, содержат методы и свойства, задаваемые функциями. При этом компонентный подход не предполагает использования таких средств теории объектов как делегирование, наследование и полиморфизм.

Для формального описания компонент будем пользоваться $\lambda\zeta$ -исчислением с типами, полученным в результате объединения типизированного λ -исчисления и типизированного ζ -исчисления второго порядка [11].

В рамках этой формальной системы полагаем, что всякий интерфейс компоненты является термом вида $I \stackrel{def}{\longleftrightarrow} [f_i = \zeta(x_i : A_i)g_i^{i \in 1..n}]$ типа $[f_i : B_i^{i \in 1..n}]$. При этом тело g_i само является термом. Пусть этот терм имеет вид $g_i = \lambda(p_i : t_i)b_p$, что можно интерпретировать как функцию с одним аргументом, где b_p , в свою очередь, является термом. Можно построить выражение вида $g_i = \lambda(p_{i1} : t_{i1})\dots\lambda(p_{in} : t_{in})b_p$, что соответствует функции с n аргументами. Соответственно, функция $f_i = \zeta(x_i)g_i : B_p$, имеющая тип возвращаемого значения B_p , будет принимать n параметров.

Интерфейсом будем называть терм вида
$$I = [f_1 = \zeta(x_1 : A_1)\lambda(p_{11} : t_{11})\dots\lambda(p_{1m_1} : t_{1m_1})b_1 : rt_1, \dots, \\ f_n = \zeta(x_n : A_n)\lambda(p_{n1} : t_{n1})\dots \\ \dots\lambda(p_{nmn} : t_{nmn})b_n : rt_n].$$

При таком подходе вызов функции интерфейса $I.f(a_1, \dots, a_{mf})$ имеет вид

$$I.f_i(a_1)\dots(a_{mf}) = I.\zeta(x_i : A_i)\lambda(p_{i1} : t_{i1})\dots \\ \dots\lambda(p_{imi} : t_{imi})b_i(a_1)\dots(a_{mf}).$$

Введем следующие обозначения: $\zeta(x : A)b$ — функция с параметром-ссылкой на содержащий

интерфейс; $I^{pr} = [f_1 = \zeta(x_1 : A_1)\lambda(p_{11} : t_{11})\dots\lambda(p_{1m_1} : t_{1m_1})b_1 : rt_1, \dots, f_n = \zeta(x_n : A_n)\lambda(p_{n1} : t_{n1})\dots\lambda(p_{nm_n} : t_{nm_n})b_n : rt_n]$ — предоставляемый интерфейс с n функциями, названными $f_1, \dots, f_n$. Каждая функция имеет тип возвращаемого значения rt_i и m_i ($i = 1\dots n$) параметров p_{ij} ($i = 1\dots n, j = 1\dots m_i$), каждый из которых имеет тип t_{mi} ; некоторые функции могут не иметь параметров; $I^{req} = [f_1 = \zeta(x_1 : A_1) \times \lambda(p_{11} : t_{11})\dots\lambda(p_{1m_1} : t_{1m_1})b_1 : rt_1, \dots, f_n = \zeta(x_n : A_n)\lambda(p_{n1} : t_{n1})\dots\lambda(p_{nm_n} : t_{nm_n})b_n : rt_n]$ — требуемый интерфейс с n функциями, названными $f_1, \dots, f_n$; $I.f(a_1, \dots, a_{mf})$ — вызов функции f интерфейса, где $a_1\dots a_{mf}$ — фактические значения параметров; $M = [I_1^{pr}, \dots, I_m^{pr}, I_1^{req}, \dots, I_k^{req}]$ — компонента с m предоставляемыми интерфейсами $I_1^{pr}, \dots, I_m^{pr}$ и k требуемыми интерфейсами $I_1^{req}, \dots, I_k^{req}$.

Таким образом компонента является термом вида $M \xrightarrow{def} [f_{in} = \zeta(x_{in} : A_{in})g_{in}, f_{js} = \zeta(x_{js} : A_{js})g_{js}]$ типа $[f_{in} : B_{in}, f_{js} : B_{js}]$, где $f_{in} \in I_i^{pr}$, $f_{js} \in I_j^{req}$, а n и s принимают значения от 1 до числа методов в соответствующем i -м или j -м интерфейсе.

Графическое представление компоненты $M = [I_1^{pr}, I_1^{req}]$, имеющей требуемый интерфейс I_1^{req} и предоставляемый I_1^{pr} , приведено на рис. 1.

Введение отношения частичного порядка и теоретико-множественных операций на заданных таким образом интерфейсах дает средства верификации корректности сигнатур и доказательства соответствия композиционной функциональности заявленной.

Пусть $I_1^{req} = [f_{11}, \dots, f_{1n}]$ — требуемый интерфейс компоненты M_1 и $I_2^{pr} = [f_{21}, \dots, f_{2m}]$ — предоставляемый интерфейс компоненты M_2 . Будем говорить, что интерфейс I_1^{req} входит в интерфейс I_2^{pr} ($I_1^{req} \subset I_2^{pr}$), если выполняется следующее условие:

$$\forall i \in [1, n] \exists j \in [1, m] : f_{1i} \equiv f_{2j}.$$

Введем отношение *частичного порядка*. Пусть $I_1^a = [f_{11}, \dots, f_{1n}]$, $I_2^b = [f_{21}, \dots, f_{2m}]$, причем $a = b$ (отношение определено только для интерфейсов одного типа, т. е. оба интерфейса должны быть либо предоставляемыми, либо требуемыми). Будем говорить, что интерфейс I_1 *меньше либо равен* интерфейсу I_2 ($I_1 \leq I_2$), если $\forall i \in [1, n], \exists j \in [1, m] : f_{1i} \equiv f_{2j}$. То есть интерфейс I_1 *меньше либо равен* интерфейсу I_2 , если интерфейс I_2 содержит все функции интерфейса I_1 .

Определим понятие равенства функций. Будем говорить, что функция $f_1 = \zeta(x_1 : A_1)\lambda(p_{11} : t_{11})\dots\lambda(p_{1m_1} : t_{1m_1})b_1 : rt_1$ *равна* функции $f_2 = \zeta(x_2 : A_2) \times$

$\lambda(p_{21} : t_{21})\dots\lambda(p_{2m_2} : t_{2m_2})b_2 : rt_2$ ($f_1 \equiv f_2$), если $f_1 = f_2$, $rt_1 = rt_2$, $m_1 = m_2$, $t_{1i} = t_{2i}$ ($i = 1\dots m_1$).

Таким образом, две функции считаются равными, если равны их имена, типы возвращаемых значений, число и типы параметров.

Определим понятие неравенства функций. Будем говорить, что функция $f_1 = \zeta(x_1 : A_1)\lambda(p_{11} : t_{11})\dots\lambda(p_{1m_1} : t_{1m_1})b_1 : rt_1$ не равна функции $f_2 = \zeta(x_2 : A_2) \times \lambda(p_{21} : t_{21})\dots\lambda(p_{2m_2} : t_{2m_2})b_2 : rt_2$ ($f_1 \neq f_2$), если хотя бы одно из условий $f_1 = f_2$, $rt_1 = rt_2$, $m_1 = m_2$, $t_{1i} = t_{2i}$ ($i = 1\dots m_1$) не выполняется.

Определим понятие суммы интерфейсов. Пусть $I_1^a = [f_{11}, \dots, f_{1n}]$, $I_2^b = [f_{21}, \dots, f_{2m}]$, причем $a = b$ (операция определена только для интерфейсов одного типа, т. е. оба интерфейса должны быть либо предоставляемыми, либо требуемыми). Тогда суммой интерфейсов $I_1^a + I_2^b$ называется интерфейс $I_3^a = [f_{11}, \dots, f_{1n}, f_{21}, \dots, f_{2m}]$, где $i_j \in [1, m]$, ($j = 1\dots k$), $\forall I_{2h}, h \in [1, \dots, i_k], \forall g \in [1, n]$ и выполняется условие $f_{2h} \neq f_{1g}$.

Определим понятие равенства интерфейсов. Пусть $I_1^a = [f_{11}, \dots, f_{1n}]$, $I_2^b = [f_{21}, \dots, f_{2m}]$, причем $a = b$ (отношение определено только для интерфейсов одного типа, т. е. оба интерфейса должны быть либо предоставляемыми, либо требуемыми). Будем говорить, что интерфейс I_1 *равен* интерфейсу I_2 ($I_1 = I_2$), если выполняется следующее условие: $\forall f_2 \in I_2^b \exists f_1 \in I_1^a : f_2 \equiv f_1$ & $\forall f_1 \in I_1^a \exists f_2 \in I_2^b : f_1 \equiv f_2$. То есть интерфейсы равны, если совпадают множества входящих в них функций.

Вхождение требуемого интерфейса одной компоненты в предоставляемый интерфейс другой компоненты определяет возможность корректного взаимодействия данных компонент на уровне совместности сигнатур. То есть, если компонента M_1 имеет требуемый интерфейс I_1^{req} , а компонента M_2 — предоставляемый интерфейс I_2^{pr} , и при этом $I_1^{req} \subset I_2^{pr}$, то можно сказать, что компоненты M_1 и M_2 на уровне синтаксического протокола взаимодействия корректно. Если все требуемые интерфейсы компонент модели входят в агрегатные предоставляемые интерфейсы, то можно, обобщая, говорить об обеспечении синтаксической целостности компонентной модели информационной системы. В противном случае построенная компонентная модель противоречива либо недостаточна, т. е. нуждается в коррекции.

Верификация семантики контрактов

Использование триплетов Хоара для описания состояний взаимодействующих интерфейсов дает средства верификации семантической корректности сервисных контрактов в компонентной архитектуре.

В аксиоматической системе Хоара средства декларативного описания (формулы языка первого порядка) сочетаются со средствами процедурного описания (алгоритмические функции). Условия корректности программ формализуются средствами языка утверждений — формулами ви-



Рис. 1. Графическое представление компоненты

да $\{Q\}P_f\{R\}$, где P_f — некоторая программа, реализующая заданную функцию f , а Q и R — ее предусловие и постусловие соответственно. Этот триплет считается истинным ($P_f(Q) = R$) = *True*, если из того, что утверждение Q истинно перед исполнением программы и программа P_f завершается, следует, что утверждение R истинно. Согласно Ben-Ari [12]:

аксиома пустого утверждения $\{q\}\text{pass}\{q\}$; (1)

аксиома присваивания $\{q(x)\{x \leftarrow t\}x := t\{q(x)\}$; (2)

правило соединения $\frac{\{q\}p_1\{s\}, \{s\}p_2\{r\}}{\{q\}p_1; p_2\{r\}}$; (3)

правило альтернативы $\frac{\{q \wedge b\}p_1\{r\}, \{q \wedge \neg b\}p_2\{r\}}{\{q\}\text{if } b \text{ then } p_1 \text{ else } p_2\{r\}}$; (4)

правило цикла $\frac{\{q \wedge b\}p\{r\}}{\{q\}\text{while } b \text{ do } p\{r \wedge \neg b\}}$; (5)

правило следствия $\frac{q_1 \rightarrow q, \{q\}p\{r\}, r \rightarrow r_1}{\{q_1\}p\{r_1\}}$. (6)

Программа является семантически корректной, если в приведенной системе правил из корректно построенных исходных термов получен терм $\{Q\}P_f\{R\}$.

Под состоянием системы будем понимать множество утверждений, выполняющихся для данной системы. Состояние допустимо, если множество выполняющихся утверждений непротиворечиво.

С точки зрения анализа корректности всей информационной системы мы говорим, что система корректна, если в любой момент времени для триплета $\{Qs\}A\{Rs\}$, где Qs — исходное множество выполняющихся утверждений, задающих состояние системы; A — набор выполняемых программ $\{P_f^*\}$ при вызове интерфейсных функций, изменяющих состояние системы; Rs — результирующее множество выполняющихся утверждений, задающих состояние системы, истинны следующие утверждения:

- Qs и Rs принадлежат множеству допустимых состояний системы;
- набор и последовательность вызовов интерфейсных функций A определяется допустимым сценарием использования системы.

Отметим, что при этом триплеты $\{Qs\}P_f\{Rs\}$ задают описание сценариев в смысле А. Сокбурна [13].

В рамках формальной системы λ_5 -исчисления с типами триплеты Хоара задаются следующим образом: $T \xrightarrow{\text{def}} [Q : \text{Predicate}, P_f : A \rightarrow B, R : \text{Predicate}]$.

Предикаты пред- и постусловий Fpred задаются термами λ_5 -исчисления:

$\text{Fpred}(x) \xrightarrow{\text{def}} \lambda(x : A).\text{pred}\{x\} : \text{Predicate},$

$\forall x \text{Fpred}(x) \xrightarrow{\text{def}} \forall (X)\text{pred}\{X\} : \text{Predicate},$

$\exists x \text{Fpred}(x) \xrightarrow{\text{def}} \text{pack } X = A \text{ with pred}\{X\} : \text{Predicate},$
где *Predicate* — тип вида

$A \rightarrow B$

$\forall (X)B\{X\}$

$\exists (X)B\{X\},$

а A, B, X — типы.

Рассмотрим применения аксиоматической системы Хоара для анализа корректности компонентной модели.

Пусть $f_1 \in I_1^{\text{req}}$ задает $\{Q_{f1}\}P_{f1}\{R_{f1}\}$ и $f_2 \in I_2^{\text{pr}}$ задает $\{Q_{f2}\}P_{f2}\{R_{f2}\}$.

Полагаем контракт двух интерфейсных функций $D_f(f_1, f_2) : f_1 \in I_1^{\text{req}}$ и $f_2 \in I_2^{\text{pr}}$ допустимым (т. е. $D_f(f_1, f_2) = \text{True}$), если одновременно выполнены следующие условия: $(Q_{f2} = > Q_{f1}) \ \& \ (R_{f1} = > R_{f2}) = \text{True}$ и $f_1 \equiv f_2$.

Контракт двух интерфейсов I_1^{req} и I_2^{pr} $D_1(I_1^{\text{req}}, I_2^{\text{pr}})$ допустим тогда и только тогда, когда $\forall f_1 \in I_1^{\text{req}} \exists f_2 \in I_2^{\text{pr}} : D_f(f_1, f_2) = \text{True}$.

То есть, если компонента M_1 имеет требуемый интерфейс I_1^{req} , а компонента M_2 — предоставляемый интерфейс I_2^{pr} , и при этом $D_1(I_1^{\text{req}}, I_2^{\text{pr}}) = \text{True}$, то можно сказать, что компоненты M_1 и M_2 семантически совместны в рамках контракта интерфейсов. При этом множество $\{D_1^*\}$ задает сеть ограничений [14] для обеспечения совместности модели.

Полагаем контракт двух интерфейсных функций $C_f(f_1, f_2) : f_1 \in I_1^{\text{req}}$ и $f_2 \in I_2^{\text{pr}}$ выполненным (т. е. $C_f(f_1, f_2) = \text{True}$), если

$$D_f(f_1, f_2) = \text{True} \text{ и } (P_{f1}(Q_{f1}) = R_{f1}) \ \& \ (P_{f2}(Q_{f2}) = R_{f2}) = \text{True}.$$

Контракт двух интерфейсов I_1^{req} и I_2^{pr} $C_1(I_1^{\text{req}}, I_2^{\text{pr}})$ выполнен тогда и только тогда, когда

$$D_1(f_1, f_2) = \text{True} \text{ и } \forall f_1 \in I_1^{\text{req}} \exists f_2 \in I_2^{\text{pr}} : C_f(f_1, f_2) = \text{True}.$$

То есть, если компонента M_1 имеет требуемый интерфейс I_1^{req} , а компонента M_2 — предоставляемый интерфейс I_2^{pr} , и при этом $C_1(I_1^{\text{req}}, I_2^{\text{pr}}) = \text{True}$, то можно говорить о предоставлении декларированной функциональности в рамках контракта интерфейсов компонент M_1 и M_2 . При этом множество $\{C_1^*\}$ задает сеть ограничений для обеспечения достаточности модели.

В рамках статического анализа семантической корректности компонентной модели мы можем говорить о совместности построения и предоставлении декларированной функциональности в случае, если для каждой пары взаимодействующих компонент выполнен контракт на их интерфейсах и доказана допустимость всех таких контрактов.

Пример проверки целостности модели

Рассмотрим применение формальных методов компонентного проектирования на примере простой системы поддержки электронной коммерции. Упрощенная компонентная модель архитектуры информационной системы включает три компонента: магазин, склад и бухгалтерия. На рис. 2 приведена диаграмма, отражающая взаимодействие компонент в системе.

Магазин — компонента $M1 = [I_{11}^{req}, I_{12}^{req}, I_{11}^{pr}]$ имеет два требуемых интерфейса:

$$I_{11}^{req} = [OrderProduct],$$

где $OrderProduct = \zeta(x : I_{11}^{req})\lambda(ID : int)b_{11} : bool$ — функция, которая используется магазином для заказа товара на складе;

$$I_{12}^{req} = [Purchase],$$

где $Purchase = \zeta(x : I_{12}^{req})\lambda(ProductID : int)\lambda(ClientID : int)b_{12} : bool$ — функция, которая используется магазином для оплаты выбранного товара через бухгалтерию,

и предоставляемый интерфейс

$$I_{11}^{pr} = [AddToWindow, DelFromWindow],$$

где $AddToWindow = \zeta(x : I_{11}^{pr})\lambda(ID : int)b_{13} : bool$ — функция, которая предоставляется для добавления нового товара в магазин; $DelFromWindow = \zeta(x : I_{11}^{pr})\lambda(ID : int)b_{12} : bool$ — функция, которая предоставляется для удаления товара из магазина.

Бухгалтерия — компонента $M2 = [I_{21}^{req}, I_{21}^{pr}]$ имеет требуемый интерфейс $I_{21}^{req} = [DelProduct]$,

где $DelProduct = \zeta(x : I_{21}^{req})\lambda(ID : int)b_{21} : bool$ — функция, которая используется бухгалтерией для снятия оплаченного товара с учета на складе, и предоставляемый интерфейс $I_{21}^{pr} = [Purchase]$,

где $Purchase = \zeta(x : I_{21}^{pr})\lambda(ProductID : int)\lambda(ClientID : int)b_{12} : bool$ — функция, которая предоставляется для оплаты выбранного товара.

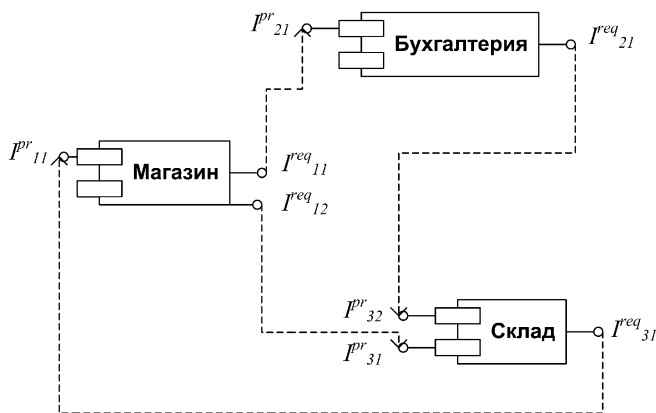


Рис. 2. Диаграмма компонент для системы поддержки электронной коммерции

Склад — компонента $M3 = [I_{31}^{req}, I_{31}^{pr}, I_{32}^{pr}]$ имеет требуемый интерфейс

$$I_{31}^{req} = [AddToWindow, DelFromWindow],$$

где $AddToWindow = \zeta(x : I_{31}^{req})\lambda(ID : int)b_{13} : bool$ — функция, которая используется складом для добавления нового товара в магазин; $DelFromWindow = \zeta(x : I_{31}^{req})\lambda(ID : int)b_{12} : bool$ — функция, которая используется складом для удаления товара из магазина;

и два предоставляемых интерфейса:

$$I_{31}^{pr} = [OrderProduct],$$

где $OrderProduct = \zeta(x : I_{31}^{pr})\lambda(ID : int)b_{11} : bool$ — функция, которая предоставляется для заказа товара на складе, и

$$I_{32}^{pr} = [DelProduct],$$

где $DelProduct = \zeta(x : I_{32}^{pr})\lambda(ID : int)b_{21} : bool$ — функция, которая предоставляется для снятия оплаченного товара с учета на складе.

Применим формальные методы для проверки корректности компонентной модели на уровне совместности сигнатур функций. Для корректности модели достаточно, чтобы $\forall I_i^{req} \subset \Sigma I_j^{pr}$, где $i = 1, \dots, n$ — число требуемых интерфейсов модели, $j = 1, \dots, m$ — число предоставляемых интерфейсов.

По определению операции суммы интерфейсов и отношения включения для модели на рис. 2:

$$I_{11}^{req} \subset I_{11}^{pr} + I_{21}^{pr} + I_{31}^{pr} + I_{32}^{pr};$$

$$I_{12}^{req} \subset I_{11}^{pr} + I_{21}^{pr} + I_{31}^{pr} + I_{32}^{pr};$$

$$I_{21}^{req} \subset I_{11}^{pr} + I_{21}^{pr} + I_{31}^{pr} + I_{32}^{pr};$$

$$I_{31}^{req} \subset I_{11}^{pr} + I_{21}^{pr} + I_{31}^{pr} + I_{32}^{pr}.$$

Таким образом, модель является синтаксически совместной.

Для доказательства семантической совместности модели достаточно показать допустимость контракта интерфейсов для каждой пары взаимодействующих компонент (M_i, M_j) .

Рассмотрим пару взаимодействующих компонент (M_1, M_2) на рис. 2. Проверим допустимость контракта $D_1(I_{11}^{req}, I_{21}^{pr})$. Контракт является допустимым, если $\forall f_1 \in I_{11}^{req} \exists f_2 \in I_{21}^{pr} : D_f(f_1, f_2) = True$. Это означает, что необходимо проверить контракты $D_f(f_1, f_2)$, где

$$f_1 = Purchase \in I_{11}^{req};$$

$$f_2 = Purchase \in I_{21}^{pr}.$$

Рассмотрим контракт $D_f(f_1, f_2)$. Приведем описание функций контракта в виде триплетов Хоара и затем применим постулаты аксиоматической системы для доказательства эквивалентности семантики программ, реализующих эти функции.

Триплет Хоара для $Purchase \in I_{21}^{pr}$

$Q_{f_2} = true$, λ_ζ — выражение, соответствующее программе P_{f_2} , имеет вид

$$\begin{aligned} & x. ClientList. GetClient(ClientID). account = \\ & = x. ClientList. GetClient(ClientID). account. subtract \\ & (x. ProductList. GetProductPrice(ProductID)), \end{aligned}$$

где знак $=$ обозначает операцию *update* [11] λ_ζ -исчисления; x — параметр функции *Purchase*, ассоциированный с компонентой, содержащей эту функцию; *Client* — поле параметра x , представляющее клиента, совершающего покупку для текущего вызова функции *Purchase*; *account* — поле *client*, представляющее счет клиента; *subtract* — метод *account*, позволяющий снять со счета клиента сумму денег, переданную в параметре вызова; *price* — поле параметра x , представляющее товар для текущего вызова функции *Purchase*; *ClientList* — поле параметра x , представляющее список клиентов; *GetClient* — метод *ClientList*, принимающий один параметр; *ProductList* — поле параметра x , представляющее список товаров; *GetProduct* — метод *ProductList*, принимающий один параметр.

$$\begin{aligned} R_{f_2} = & (x. ClientList. GetClient(ClientID). account = \\ & x. ClientList. GetClient(ClientID). account. subtract \\ & (x. ProductList. GetProductPrice(ProductID))). \end{aligned}$$

Для $Purchase \in I_{11}^{req}$

$$Q_{f_1} = true,$$

λ_ζ -выражение, соответствующее программе P_{f_1} , имеет вид

$$I_{21}^{pr}. Purchase(productID, clientID) \quad R_{f_1} = true.$$

Таким образом, выполнено

$$(Q_{f_2} = > Q_{f_1}) \ \& \ (R_{f_1} = > R_{f_2}) = True,$$

$f_1 = f_2$ (по определению равенства функций).

Следовательно, допустим контракт двух функций $D_f(f_1, f_2) = True$. Выполняется $\forall f_i \in I_{11}^{req} \exists f_j \in I_{21}^{pr} : C(f_i, f_j) = True$, следовательно, $D_f(I_{11}^{req}, I_{21}^{pr}) = True$.

Доказательство допустимости контрактов для остальных пар взаимодействующих интерфейсов аналогичное. Таким образом, модель является семантически совместной.

Рассмотрим вопрос предоставления декларированной функциональности в рамках контрактов интерфейсов компонент. Покажем, что для каждой пары взаимодействующих компонент выполнен контракт на их интерфейсах.

Для пары взаимодействующих компонент M_1 и M_2 контракт интерфейсов выполнен $C_f(I_{11}^{req}, I_{21}^{pr}) = True$ тогда и только тогда, когда контракт является допустимым и $\forall f_1 \in I_{11}^{req} \exists f_2 \in I_{21}^{pr} : C_f(f_1, f_2) = True$. Допустимость $D_f(I_{11}^{req}, I_{21}^{pr}) = True$ была доказана выше. Рассмотрим контракт $C_f(f_1, f_2)$. Контракт выполнен, если он является допустимым и $(P_{f_1}(Q_{f_1}) = R_{f_1}) \ \& \ (P_{f_2}(Q_{f_2}) = R_{f_2}) = True$. Допустимость контракта функций $D_f(f_1, f_2) = True$ была показана выше. Для проверки семантической корректности программы P_{f_2} , можно воспользоваться аксиомами (1-6). Если обе программы корректны $((P_{f_1}(Q_{f_1}) = R_{f_1}) \ \& \ (P_{f_2}(Q_{f_2}) = R_{f_2}) = True)$ и $D_f(f_1, f_2) = True$, то выполнен контракт функций $C_f(f_1, f_2) = True$.

Таким образом, выполнен контракт интерфейсов $C_f(I_{11}^{req}, I_{21}^{pr}) = True$ для компонент M_1 и M_2 . Аналогично можно показать выполнение контрактов интерфейсов для остальных пар взаимодействующих компонент. Значит, можно говорить о предоставлении декларированной функциональности в рамках контракта интерфейсов компонент.

Отметим, что благодаря использованию аппарата λ_ζ -исчисления для задания и анализа семантики компонент формализованный процесс верификации совместности может быть легко автоматизирован и масштабирован для анализа компонентных моделей архитектуры сложных систем, например, в виде системы ограничений модели в соответствующей MDD-среде разработки.

Итак, представленный подход предлагает средства формальной верификации корректности проектирования ИС, причем комплексное применение методов компонентного проектирования позволяет обеспечить синтаксическую и семантическую целостность разрабатываемой компонентной модели сервисно-ориентированной архитектуры сложной информационной системы.

Список литературы

1. **Gavrilov A., Pavlova E.** DSL Tools for Model-Driven Development // Proc. of the Software Engineering Conference (Russia), SEC(R) 2006, Moscow, RusSE-TEKAMA. — <http://secl.ru/2006/upload/files/107.pdf>, 2006.
2. **Onion F.** Extreme ASP. NET: Web Client Software Factory // MSDN Magazine. 2007. V. 22. N 8. San Francisco: CMP Technology.
3. **Smith D.** Introducing the WCF Service Factory. <http://www.codeplex.com/servicefactory/Project/FileDownload.aspx?DownloadId=9284>, 2007.
4. **Smart Client** Software Factory. <http://msdn2.microsoft.com/en-us/library/aa480482.aspx>, 2007.
5. **Mobile** Client Software Factory. <http://msdn2.microsoft.com/en-us/library/aa480471.aspx>, 2007.
6. **Hoare C.** An Axiomatic Basis of Computer Programming. Communications of the ACM. 1969. V. 12. New York: ACM Press.
7. **Гаврилов А. В.** Формализация проектирования объектно-ориентированных информационных систем в рамках UML-тех-

нологии разработки // XXVI Гагаринские чтения. М.: Изд-во ЛАТМЭС, 2000.

8. **Чернов А. В.** Формальное проектирование компонентной модели информационной системы и обеспечение ее корректности // Технологии Microsoft в теории и практике программирования. М.: Изд-во МГТУ, 2006.

9. **Мейер Б.** Объектно-ориентированное конструирование программных систем. М.: Рус. Редакция, 2005.

10. **Bruce K.** Foundations of Object-Oriented Languages: Types and Semantics. Cambridge, Massachusetts: The MIT Press, 2002.

11. **Abadi M., Cardelli L.** A Theory of Objects. New York: Springer, 1996.

12. **Ben-Ari M.** Mathematical Logic for Computer Science. London: Springer, 2001.

13. **Cockburn A.** Writing Effective Use Cases. NJ: Addison-Wesley Professional, 2001.

14. **Marriott K., Stuckey P.** Programming with Constraints: An Introduction. Cambridge, Massachusetts: The MIT Press, 1998.

Приложение

Приведенное ниже описание исчисления взято из работы М. Abadi и L. Cardelli [11].

FOb₂ исчисление

Синтаксис исчисления

$A, B, C, D ::=$	<i>типы</i>
X	<i>тип-переменная</i>
$[l_i : B_i^{i \in 1 \dots n}]$	<i>тип объекта</i>
$A \rightarrow B$	<i>функциональный тип</i>
$\forall(X)B$	<i>общий тип</i>
$\exists(X)B$	<i>экзистенциальный тип</i>
$a, b, c, d ::=$	<i>термы</i>
x	<i>переменная</i>
$[l_i = \varsigma(x_i : A_i) b_i^{i \in 1 \dots n}]$	<i>объект</i>
$a.l$	<i>вызов метода</i>
$a.l \Leftarrow \varsigma(x : A)b$	<i>обновление метода</i>
$\lambda(x : A)b$	<i>функция</i>
$b(a)$	<i>аппликация</i>
$\lambda(X)b$	<i>абстракция типа</i>
$b(A)$	<i>аппликация типа</i>
$\text{pack } X = A \text{ with } b\{X\} : B$	<i>упаковка абстракции данных</i>
$\text{open } c \text{ as } X, x : B \text{ in } d : D$	<i>распаковка абстракции данных</i>

Символ ς используется для связывания параметра; $\varsigma(x : A)b$ — это метод с параметром x и телом b . Параметр будет связан с объектом, содержащим этот метод; $a.l$ — это вызов метода. Часто наряду с методами используется понятие поля объекта.

Поля не являются частью $\lambda\varsigma$ -исчисления. Однако методы, которые не используют свой связанный параметр x , могут рассматриваться как поля. С учетом этого соглашения вызовы таких методов играют роль выбора поля, а обновление метода играет роль обновления поля.

Исчисление составлено из фрагментов формальных систем:

$$FOb_2 = \Delta_X \cup \Delta_{Ob} \cup \Delta_{\rightarrow} \cup \Delta_X \cup \Delta_{\forall} \cup \Delta_{\exists}.$$

$$\begin{array}{c} \text{(Env } \emptyset) \quad \text{(Env } x) \quad \text{(Val } x) \\ \hline \frac{}{\emptyset \vdash \circ} \quad \frac{E \vdash A \quad x \notin \text{dom}(E)}{E, x:A \vdash \circ} \quad \frac{E', x:A, E'' \vdash \circ}{E', x:A, E'' \vdash x:A} \end{array}$$

В каждом из приведенных здесь и далее правил над чертой находятся посылки, а под чертой — умозаключения; Δ_X — фрагмент, который описывает построение окружения (универсума) и извлечение типа переменной из окружения. Запись $x \in \text{dom}(E)$ означает, что переменная x определена в E . Утверждение $A \vdash \circ$ означает, что A — правильно построенное окружение; Δ_{Ob} — фрагмент, который описывает типы объектов;

(Type Object)(l_i определены)

$$\frac{E \vdash B_i \quad \forall i \in 1 \dots n}{E \vdash [l_i : B_i^{i \in 1 \dots n}]}$$

(Val Object)(где $A \equiv [l_i : B_i^{i \in 1 \dots n}]$)

$$\frac{E, x_i : A \vdash b_i \quad B_i \quad \forall i \in 1 \dots n}{E \vdash [l_i = \varsigma(x_i : A)b_i^{i \in 1 \dots n}] : A}$$

(Val Select)

$$\frac{E \vdash a : [l_i : B_i^{i \in 1 \dots n}] \quad j \in 1 \dots n}{E \vdash a.l_j : B_j}$$

(Val Update)(где $A \equiv [l_i : B_i^{i \in 1 \dots n}]$)

$$\frac{E \vdash a : A \quad (E, x : A \vdash b : B_j) \quad j \in 1 \dots n}{E \vdash a.l_j \Leftarrow \varsigma(x : A)b : A}$$

$\Delta_{\rightarrow}$ — фрагмент, который описывает функции и функциональные типы,

$$\begin{array}{c} \text{(Type arrow)} \quad \text{(Val Fun)} \quad \text{(Val Appl)} \\ \hline \frac{E \vdash A, E \vdash B}{E \vdash A \rightarrow B} \quad \frac{E, x : A \vdash b : B}{E \vdash \lambda(x : A)b : A \rightarrow B} \quad \frac{E \vdash b : A \rightarrow B \quad E \vdash a : A}{E \vdash b(a) : B} \end{array}$$

Δ_X — фрагмент, который описывает построение типа-переменной,

$$\begin{array}{c} \text{(Env } X) \quad \text{(Type } X) \\ \hline \frac{E \vdash \circ \quad X \notin \text{dom}(e)}{E, X \vdash \circ} \quad \frac{E', X, E'' \vdash \circ}{E', X, E'' \vdash X} \end{array}$$

$\Delta_{\forall}$ — фрагмент, который описывает применение квантора общности,

$$\begin{array}{c} \text{(Type All)} \quad \text{(Val Fun2)} \quad \text{(Val Appl2)} \\ \hline \frac{E, X \vdash B}{E \vdash \forall(X)B} \quad \frac{E, X \vdash b : B}{E \vdash \lambda(X)b : \forall(X)B} \quad \frac{E \vdash b : \forall(X)B \quad \{X\} \quad E \vdash A}{E \vdash b(A) : B\{\{A\}\}} \end{array}$$

$\Delta_{\exists}$ — фрагмент, который описывает применение квантора существования,

$$\begin{array}{c} \text{(Type Exists)} \\ \hline \frac{E, X \vdash B}{E \vdash \exists(X)B} \\ \text{(Val Pack)} \\ \hline \frac{E \vdash A \quad E \vdash b\{\{A\}\} : B\{\{A\}\}}{E \vdash \text{pack } X = A \text{ with } b\{X\} : B\{X\} : \exists(X)B\{X\}} \end{array}$$

(Val Open)

$$\frac{E \vdash c : \exists(X)B \quad E \vdash D \quad E, X, x : B \vdash d : D}{\text{open } c \text{ as } X, x : B \text{ in } d : D : D}$$

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УДК 519.6(07)

Н. М. Шерыхалина, канд. физ.-мат. наук, доц.,
Уфимский государственный авиационный
технический ун-т

Применение фильтрации численных результатов для увеличения надежности САПР

Идеи и методы выделения полезной информации из искаженного помехами сигнала путем его фильтрации, идентификации математических моделей информационного сигнала, принятия решений о достоверности выделенной информации применяются в области анализа результатов численного эксперимента для увеличения их точности и надежности. С помощью этих методов проведено исследование результатов работы одного из простейших методов численного решения задачи Коши. Найдены и объяснены некоторые, на первый взгляд парадоксальные, закономерности. Применение фильтрации и учет реальных свойств численных методов позволяет существенно улучшить надежность программных комплексов моделирования и проектирования различных устройств и технологий.

Ключевые слова: численная фильтрация, оценка и уменьшение погрешности вычислений.

Введение

Применение САПР в разных областях привело к широкому использованию различных программных комплексов. Специфика САПР требует разработки надежных вычислительных алгоритмов решения сложных задач. В связи с недостаточной надежностью этих комплексов при их использовании часто возникают трудности и отказы. В качестве примера можно привести работу [1], в которой авторы приходят к необходимости применения надстройки к комплексу OrCAD, которая довольно сложным способом помогает избежать отказов, связанных с точностью, устойчивостью вычислительных процессов, с одной стороны, и накоплением погрешности округления при численном интегрировании, с другой. При проектировании часто используются связанные с потерей точности операции, например численное дифференцирование [2]. Можно привести примеры получения парадоксальных результатов при использовании общеизвестными пакетами про-

грамм. Но парадоксальность позволяет увидеть ошибку; намного сложнее ее обнаружить, если парадокса нет, а результат вычисления грубый. Отсутствие возможности внешнего контроля точности расчетов, тенденция к освобождению пользователя автоматизированными комплексами программ от ответственности за результат усугубляют проблему. Результатам таких расчетов можно только верить, а это может привести к неверному описанию процессов, явлений и дальнейшим непредсказуемым ошибкам.

Проблема получения надежных результатов объясняется сложностью описания реального вычислительного процесса традиционными математическими методами. Традиционный аппарат вычислительной математики использует классические идеальные объекты и понятия (действительное число, предел, непрерывность и т. п.), которые не могут быть в строгом смысле применены к реальному вычислительному процессу, выполняемому компьютером в условиях ограниченности ресурсов по времени и памяти. Процессор и сопроцессор компьютера работают с конечными подмножествами рациональных чисел. Арифметические операции, выполняемые компьютером, не имеют тех свойств, которыми обладают операции над идеальными объектами — действительными числами (например, ассоциативностью сложения). Можно привести примеры, когда реальный объект имеет принципиально отличные свойства от объекта идеального (например, существование и единственность решения). В других случаях выводы, полученные на основе математического анализа, остаются верными, но оценки, полученные на их основе, являются слишком грубыми, а поэтому могут оказаться бесполезными для вычислителя.

Отсюда следует вывод: реальный вычислительный процесс отличается от идеального, как и процессы физические, химические, технические и т. д. Однако для последних создаются системы различных испытаний, проверок, методов анализа, технологий обработки результатов эксперимента. Для результатов вычислительного эксперимента обычно пытаются использовать грубые аналитические оценки, а если это невозможно (например, в связи с некорректностью задачи), то применяют весьма ненадежные способы, например, тестирование на частных решениях и распространение полученных оценок на общий случай.

Можно подойти к этой проблеме следующим образом. Признать, что численный эксперимент —

это реальность, а традиционный математический анализ дает приближенную модель этой реальности. Модель в разных условиях может хорошо или плохо соответствовать реальности. В этом случае возникает необходимость экспериментального изучения этой реальности, разработки методик проведения эксперимента и методов анализа его результатов. Математические модели, которые строят в результате такого эксперимента, могут быть более сложными и информативными, чем традиционные.

В результате эксперимента получают данные — информация, которая может быть подвергнута обработке специальными методами для выделения полезной (с какой-либо точки зрения) информационной составляющей.

Программа, комплекс САПР, информационная система — это технические объекты, поскольку используют реальное оборудование (компьютер). Поэтому нельзя требовать от них стопроцентной надежности, зато возможно, применяя известные информационные подходы, существенно увеличить надежность создаваемых комплексов.

Представляется перспективным использовать подходы и идеи, которые применяются в области информационных технологий, например, следующие:

- идентификация математической модели по результатам численного эксперимента;
- численная фильтрация результатов эксперимента;
- визуализация результатов эксперимента и распознавание образов;
- эвристические критерии для поддержки принятия решения о достоверности оценки в условиях неопределенности.

Для анализа численных данных представляются весьма перспективными методы численной фильтрации и экстраполяции зависимостей этих данных от параметров дискретизации, в качестве которых могут использоваться шаги по переменным, число интервалов разбиения, число слагаемых в сумме и т. п. Известны методы экстраполяции Ричардсона, Рунге, Ромберга, Эйткена [3, 4], которые, как показывает практика, иногда работают, а иногда дают неверные оценки [5]. Экстраполяция основана на предположении о конкретном виде зависимости результата вычислений от параметра дискретизации. Результаты численного решения несут информацию, как о решаемой математической задаче, так и о методе решения и используемом оборудовании. Отсюда возникает вопрос: как выделить необходимую составляющую?

Помочь ответить на данный вопрос должен анализ фундаментальных закономерностей процессов вычислений на реальном оборудовании.

1. Метод численной фильтрации

Во многих случаях математическую модель погрешности можно представить в виде [6]

$$z_n - z = c_1 n^{-k_1} + c_2 n^{-k_2} + \dots + c_L n^{-k_L} + \Delta(n), \quad (1.1)$$

где z — точное значение; z_n — приближенный результат, полученный при числе узловых точек, равном n ; $k_1, \dots, k_L$ — произвольные действительные числа ($k_1 < k_2 < \dots < k_L$).

В $\Delta(n)$ могут входить не вошедшие в сумму слагаемые степенного вида, остаточный член, погрешность округления и многие другие составляющие, обусловленные как численным методом, так и конкретной программной реализацией. Поэтому $\Delta(n)$ не стремится к нулю при увеличении n , а может даже возрастать.

Если при решении задачи можно допустить возможность разложения функций по формуле Тейлора, тогда k_j — это часть ряда натуральных чисел. Пусть имеется конечная последовательность вычисленных значений z_{n_i} , $i = 1, 2, \dots, M$. Тогда при предположении о возможности пренебрежения составляющей $\Delta(n)$ к задаче нахождения предельного при $n \rightarrow \infty$ значения z можно подойти как к задаче интерполяции зависимости $z(x)$ от параметра $x = 1/n$ алгебраическим многочленом с последующей экстраполяцией до $x = 0$ (метод Ромберга [3]). Есть и другой подход, приводящий в простых случаях при условии постоянства $Q = n_j/n_{j-1} > 1$ к тому же алгоритму, но не требующий целочисленности k_j . Это решение задачи численной фильтрации, т. е. последовательное устранение степенных слагаемых суммы (1.1) при сохранении значения константы z .

При увеличении n в целое число раз, т. е. $n_i = n_1 Q^{i-1}$, задача фильтрации решается точно и не требует пренебрежения $\Delta(n)$. Для этого рассмотрим два значения z_{n_1} , z_{n_2} , вычисленных при числе узлов, равном n_1 и $n_2 = Qn_1$ соответственно. Составим линейную комбинацию

$$z_{n_2}^{(1)} = \alpha_j z_{n_1} + \beta_j z_{n_2} = (\alpha_j + \beta_j)z + \dots + (\alpha_j n_1^{-k_j} + \beta_j n_2^{-k_j})c_j + \dots \quad (1.2)$$

и потребуем, чтобы суммарный коэффициент при z был равен 1, а при c_j (для определенного j) равен 0. Тогда $\alpha_j = -(Q^{k_j} - 1)^{-1}$, $\beta_j = Q^{k_j} (Q^{k_j} - 1)^{-1}$.

Отсюда получим формулу фильтрации, которая в данном случае совпадает с экстраполяционной формулой Ричардсона:

$$z_{n_2}^{(1)} = z_{n_2} + \frac{z_{n_2} - z_{n_1}}{Q^{k_j} - 1}. \quad (1.3)$$

Проводя последовательно фильтрацию по всем парам соседних значений z_{n_i} и $z_{n_{i-1}}$, получим отфильтрованную зависимость, не содержащую члена с n^{-k_j} :

$$z_n^{(1)} = z + c_1^{(1)} n^{-k_1} + \dots + c_{j-1}^{(1)} n^{-k_{j-1}} + c_{j+1}^{(1)} n^{-k_{j+1}} + \dots + c_L^{(1)} n^{-k_L} + \Delta^{(1)}(n), \quad (1.4)$$

где $c_l^{(1)} = \left(\alpha_j \left(\frac{n_1}{n_2} \right)^{-k_l} + \beta_j \right) c_l = c_l \frac{Q^{k_j} - Q^{k_l}}{Q^{k_j} - 1}$.

Заметим, что отфильтрованная последовательность $z_n^{(1)}$ содержит на один член меньше, чем исходная. Если она содержит больше одного члена, то ее также можно отфильтровать, устранив степенную составляющую с n^{-k_l} . Операции фильтрации можно повторять последовательно для n^{-k_1} , ..., n^{-k_L} , если исходная последовательность содержит достаточное число членов.

В случае $n_j \neq n_1 Q^{j-1}$ задача фильтрации сводится к решению системы $L + 1$ линейных уравнений типа (1.1) численным методом.

Поскольку зависимость z_n от n имеет нерегулярную составляющую $\Delta^{(0)}$, обусловленную, например, погрешностью округления, то изменение исходной нерегулярной части погрешности $\Delta^{(L-1)}$, содержащейся в вычисленных значениях z_j , при каждой фильтрации можно оценить следующим образом:

$$\begin{aligned} \Delta^{(L)} &\leq \Delta^{(L-1)} + \frac{\Delta^{(L-1)} + \Delta^{(L-1)}}{Q^{k_L} - 1} = \\ &= \frac{Q^{k_L} + 1}{Q^{k_L} - 1} \Delta^{(L-1)}. \end{aligned} \quad (1.5)$$

Для метода Ромберга, применяемого к последовательности (1.1) ($k_j = i$), произведение таких множителей ограничено числом, приблизительно равным 8, т. е. метод Ромберга является устойчивым к погрешности исходных данных.

Значение погрешности $\Delta^{(L)}$ можно оценить экспериментально, этому способствует визуализация результатов фильтрации (рис. 1—3).

Разработаны методы, позволяющие уменьшить влияние нерегулярной погрешности, не требующие информации о конкретных значениях k_j [5, 7], однако они требуют больших затрат ресурсов.

Следует заметить, что процедура оценки предела числовой последовательности по конечному числу известных ее членов некорректна с точки

зрения математического анализа. А именно этому и посвящен данный раздел статьи. Такими же некорректными являются задачи, которые приходится решать практически при проектировании реальных устройств и технологий. Поэтому возникает идея применить известные в информатике подходы, например, найти неформальные критерии, позволяющие в условиях неопределенности принять или отклонить оценку с достаточной для практических целей достоверностью.

2. Критерии размытости оценки

Оценка погрешности по правилу Рунге сводится к сравнению значения z_n с экстраполированным значением $z_n^{(1)}$. Поскольку эта оценка справедлива при допущении, что значение $z_n^{(1)}$ точнее, чем z_n , то необходима проверка справедливости этого допущения. Это можно сделать следующим образом. Повторим процесс экстраполяции и получим значение $z_n^{(2)}$. Разность $\Delta_n = z_n - z_n^{(1)}$ представляет собой оценку погрешности приближенного значения z_n . Разность $\Delta_n^{(1)} = z_n^{(1)} - z_n^{(2)}$ является оценкой погрешности экстраполированного значения $z_n^{(1)}$ или оценкой погрешности оценки погрешности. Отношение $r_n = |\Delta_n^{(1)} / \Delta_n|$ имеет смысл относительной размытости оценки погрешности.

Если $r_n \ll 1$, то это означает, что относительная размытость оценки Δ_n мала, и такой оценке можно доверять. Допустим, что результат расчета с оценкой погрешности представляется в виде интервала $z = z_n^{(1)} \pm \Delta_n$. В работе [7] приводится обоснование порогового значения $r_n = 1/3$ в предположении, что изменения значений $\Delta_n^{(j)} = z_n^{(j)} - z_n^{(j+1)}$ при последующих гипотетических фильтрациях мажорируются суммой геометрической прогрессии. Тем самым при $r_n \leq 1/3$ оценка принимается, иначе считается недостоверной. Ранее [8] этот вывод был получен чисто экспериментально.

Этот критерий назовем локальным критерием размытости. Наряду с этим предлагается ассоциативный критерий.

Можно все приближенные значения сравнивать с одним числом — эталоном $\bar{z}$, которое считается наиболее точным. Эталон $\bar{z}$ может быть получен экстраполяцией или каким-либо другим способом. Разность $\Delta_n = z_n - \bar{z}$ представляет собой оценку погрешности приближенного значения $\bar{z}$. В этом случае результат расчета с оценкой погрешности представляется в виде интервала $z = \bar{z} \pm \Delta_n$.

Оценки погрешности зависят от выбора эталона $\bar{z}$, с которым сравниваются все приближенные значения. Чтобы оценить это влияние, обеспечить надежность оценки и связать это со значе-

нием размытости, используем условие достоверности оценки

$$|\bar{z} - z| \leq |z_n - \bar{z}|.$$

Преобразуем, обозначив

$$\bar{r}_n = \frac{z_n^{(1)} - \bar{z}}{z_n - \bar{z}}, r_n' = \frac{\bar{z} - z}{z_n^{(1)} - z} : \quad (2.1)$$

$$\begin{aligned} \left| \frac{\bar{z} - z}{z_n - \bar{z}} \right| &= \left| \frac{z_n^{(1)} - \bar{z}}{z_n - \bar{z}} \right| \left| \frac{r_n'}{1 - r_n'} \right| = \\ &= |\bar{r}_n| \left| \frac{r_n'}{1 - r_n'} \right| \leq 1, |\bar{r}_n| \leq \left| \frac{1 - r_n'}{r_n'} \right|. \end{aligned} \quad (2.2)$$

Результаты фильтрации и оценки погрешности удобно представлять на графике (рис. 1) в виде зависимости $-\lg \delta$ (десятичного логарифма относительной погрешности, т. е. точности, выраженной в числе точных десятичных (знаков, равному количеству информации в децитах) от логарифма n . При этом каждая составляющая зависимости (1.1) представлялась бы на таком графике отрезком прямой.

Величина δ' связана с неопределенностью выбора эталона. Из практических соображений о диапазоне изменения этой величины можно сказать следующее. Пусть на графике (рис. 1) имеются две линии, линия 0 соответствует вычисленным значениям, линия 1 — отфильтрованным. Линии объединяют (ассоциируют) разрозненные точки в единый геометрический объект, форма которого (а именно, ее отличие от прямой) может служить сигналом о неправильном выборе эталона. Оценим влияние изменения эталона на кривые. Выразим $\bar{z}$ через r_n' из (2.1) и подставим в выражение

$$\begin{aligned} -\lg |\delta| &= -\lg \left| \frac{z_n^{(1)} - \bar{z}}{\bar{z}} \right| = \\ &= -\lg \left| \frac{z_n^{(1)}(1 - r_n') - z(1 - r_n')}{\bar{z}} \right| \approx \\ &\approx -\lg \left| \frac{z_n^{(1)} - z}{\bar{z}} \right| - \lg |1 - r_n'|. \end{aligned} \quad (2.3)$$

Линии 1' и 1'' на рис. 1 соответствуют $r_n' = 0,75$ и $r_n' = -2$. Поскольку отличие от прямой вполне заметно и в первом и во втором случаях, эти значения и можно принять за границы диапазона неопределенности r_n' , т. е. $-2 \leq r_n' \leq 0,75$. Тогда из (2.2) получим $|\bar{r}_n| \leq 1/3$. В случае наличия нерегулярной погрешности, "зашумляющей" верхнюю линию, диапазон неопределенности выбора эталона увеличивается, и это должно быть скомпенсировано соответствующим уменьшением $|\bar{r}_n|$. Но, поскольку для оценок используются не три значения, а относительное положение линий, влияние случайных факторов

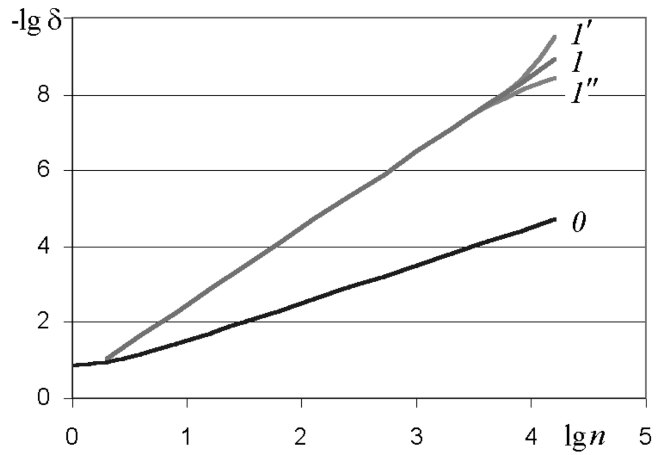


Рис. 1. Влияние выбора эталона на вид линий

на оценку погрешности и размытости существенно уменьшается. Для оценки r_n' может быть использован обычный или модифицированный метод наименьших квадратов [5, 7].

Этот критерий назовем ассоциативным.

Наличие нескольких критериев и, возможно, нескольких методов решения одной задачи с оценками погрешности позволяет существенно увеличить достоверность полученных результатов [8].

3. Численное решение задачи Коши

Задача Коши для обыкновенных дифференциальных уравнений часто используется для моделирования и проектирования динамических процессов. В качестве примера рассмотрим следующую задачу:

$$\begin{aligned} \frac{dy}{dx} &= f(y, t) = y \sin t, y(0) = y_0 = 1, \\ t &\in [0, \pi/2]. \end{aligned} \quad (3.1)$$

При использовании метода Эйлера 1-го порядка точности по рекуррентной формуле

$$y_j = y_{j-1} + hf(y_{j-1}, t_{j-1}) = h \sum_{i=0}^{j-1} f(y_i, t_i), \quad (3.2)$$

где $t_j = jh$, $j = 0, \dots, n$, $h = (b - a)/n$, вычисляется сумма и возникает ситуация, описанная в работе [8], когда складываются слагаемые разного порядка. Общая оценка погрешности численного интегрирования на основе модели предельной погрешности [4]

$$\begin{aligned} \left| y(x_n) - h \sum_{i=0}^{n-1} f(y_i, t_i) \right| &= \Delta_m + \Delta_c + \Delta_r \leq \\ &\leq x \rightarrow t \leq c_0(b - a)h + (b - a)f_{\max} 10^{-M} + \\ &+ f_{\max} \frac{(b - a)^2}{h} 10^{-M}, \\ f_{\max} &= \max_{t \in [a, b]} |f[y(t), t]|, \end{aligned} \quad (3.3)$$

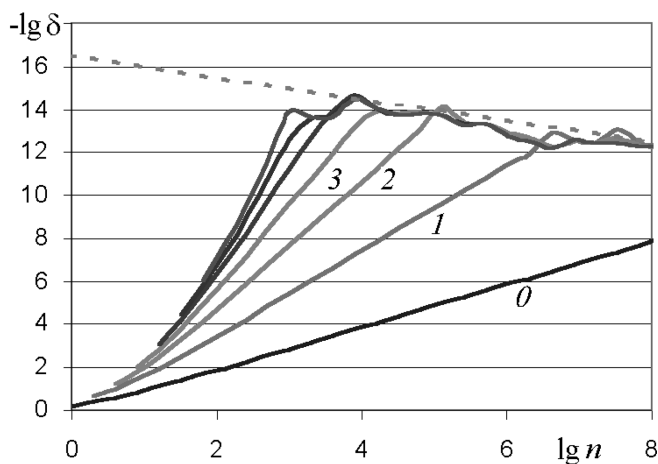


Рис. 2. Фильтрация результатов численного решения задачи (3.1)

где M — длина мантиссы машинного слова в десятичном эквиваленте; Δ_m — погрешность численного метода; Δ_c — погрешность исходных данных; Δ_r — погрешность округления при вычислении суммы.

Перейдем к результатам численного эксперимента, которые показаны на рис. 2. Линия 0, как и выше, соответствует погрешности вычисленных данных, линии 1, 2, ... — погрешностям отфильтрованных 1, 2, .. раз результатов.

Фильтрация позволяет устранить погрешность метода численного интегрирования с высокой точностью. Для сравнения на рисунок нанесена прямая $y = 16,5 - 0,5 \lg n$ (штриховая линия). Из результатов эксперимента следует, что погрешность округления накапливается по статистическому закону пропорционально $\sqrt{n}$ (т. е. характер погрешности округления более точно описывается случайной моделью).

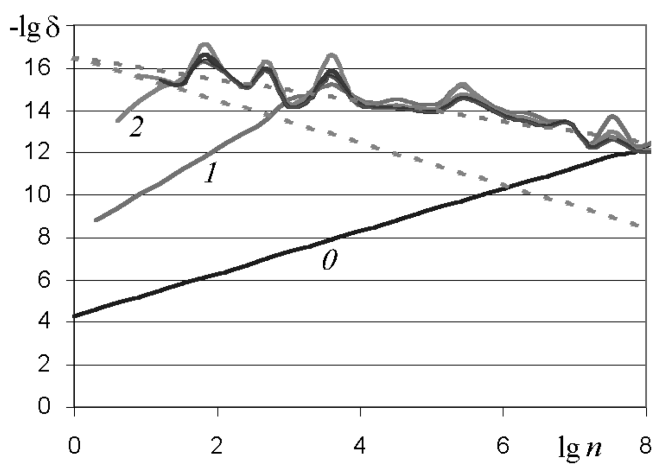
В отличие от сказанного выше, как следует из численного эксперимента, при интегрировании константы ($y' = c$) погрешность округления накапливается пропорционально n , т. е. более близкой к реальности оказывается модель предельной погрешности (3.3).

Рассмотрим результаты интегрирования в случае, когда функция $f(y, t)$ представляет собой сумму постоянной и переменной составляющих

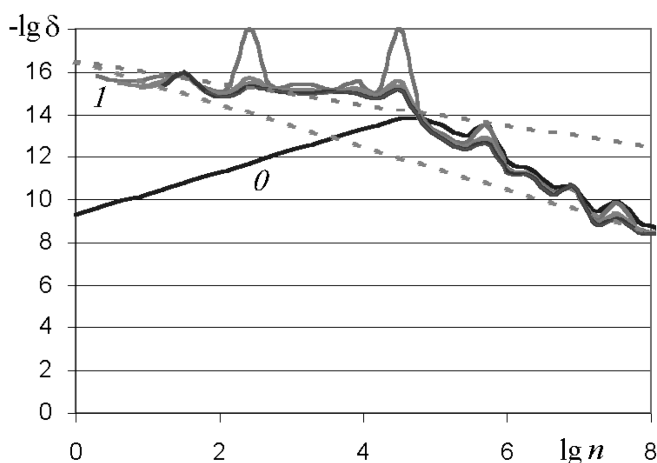
$$f(y, t) = Ay + 2/\pi$$

при $y(0) = 0, a = 0, b = 10$. (3.4)

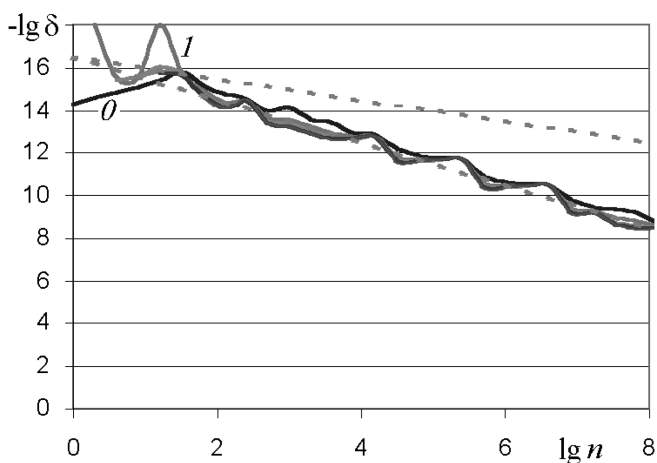
На рис. 3, а—в приведены графики относительной погрешности, соответствующие $A = 10^{-5}, 10^{-10}, 10^{-15}$. Видно, что для $A = 10^{-5}$ значение погрешности округления близка к зависимости $y = 16,5 - x/2$, при уменьшении A (вначале для больших n) погрешность округления приближается к зависимости $y = 16,5 - x$.



а)



б)



в)

Рис. 3. Результаты численного решения задачи (3.4):

а — при $A = 10^{-5}$; б — при $A = 10^{-10}$; в — при $A = 10^{-15}$

Из анализа результатов эксперимента можно сделать следующие выводы.

Характер погрешности округления переменной составляющей функции близок к случайному с математическим ожиданием, равным нулю. Поэтому

накопление этой погрешности происходит примерно как $10^{-M} \sqrt{n}$ (по статистическому закону).

При интегрировании функции, являющейся суммой константы и переменной части $f(x) = 0,1 + A \sin x$, общую погрешность округления нельзя представлять в виде суммы двух погрешностей округления. При увеличении n преобладает погрешность константы c , но при меньших n (конкретные значения зависят от соотношения c и A) переменная часть подавляет влияние константы и уменьшает скорость накопления общей погрешности. Отметим, что в данном случае роль играет не математическое представление функции в виде постоянной и переменной частей, которое не единственно, а фактическая запись ее на языке программирования.

Для объяснения этих эффектов рассмотрим в качестве примера процесс суммирования константы, значение которой близко к 0,1 в десятичной системе счисления, т. е.

$$c = 0,1c_2 \dots c_M.$$

В результате суммирования получается

$$S_k \approx \sum_{i=1}^k c = kc, S_k = 0, a_1 a_2 \dots a_M \cdot 10^p.$$

Отметим, что числа хранятся в памяти ЭВМ в нормализованной форме, т. е. значение порядка p выбирается так, что перед запятой должен быть ноль, после запятой — не ноль ($a_1 > 0$). Это максимизирует количество информации, которое может быть сохранено в машинном слове ограниченного размера.

При суммировании может возникнуть погрешность округления, вызванная сдвигом мантиссы меньшего слагаемого при выравнивании порядков с последующим усечением или округлением.

В таблице даны результаты, характеризующие процесс суммирования: k — числа слагаемых; S_k — сумма; $\Delta_{\text{окр}}$ — десятичный порядок погрешности округления, возникающей при каждой операции суммирования; m — число операций, при которых возникает такая погрешность; Δ_p — накопленная погрешность при данном порядке частичной суммы.

Отметим, что при суммировании константы погрешность округления повторяется (неизменно) определенное число раз до тех пор, пока не изменится порядок частичной суммы. После увеличения порядка суммы на единицу увеличивается на единицу и порядок погрешности округления, так как сдвиг при выравнивании порядков происходит на большее число разрядов. Кроме того, увеличивается на порядок и число слагаемых, при котором сохраняется порядок p частичной суммы. В результате накопления порядок погрешности округления при данном p на две единицы

Результаты суммирования константы

k	s_k	p	$\Delta_{\text{окр}}$	m	Δ_p
1...9	0,1...0,9	0	0	9	0
10...99	1...9,9	1	10^{-M}	90	10^{-M+2}
100...999	10...99,9	2	10^{-M+1}	900	10^{-M+4}
1000...9999	100...999,9	3	10^{-M+2}	9000	10^{-M+6}

превышает порядок погрешности, накопленной ранее. В двоичной системе изменение на два порядка означает увеличение в 4 раза, но и это весьма значительная величина.

Тем самым, при интегрировании константы погрешность округления определяется в основном суммой последних, больших по порядку, одинаковых погрешностей, поэтому и накапливается приблизительно пропорционально n^2 . Поскольку при численном решении задачи Коши сумма (3.2) умножается на h , то погрешность решения увеличивается пропорционально n .

При интегрировании суммы константы и переменной функции переменная составляющая последней вносит возмущение в округляемом разряде, что и приводит к тому, что результирующая погрешность имеет квазислучайный характер. Но при увеличении n порядок максимального значения A переменной составляющей функции становится меньше порядка округляемого разряда. Тогда накопленная погрешность округления суммы возрастает и определяется скоростью накопления погрешности при интегрировании константы. Пусть длина мантиссы машинного слова равна

$M = 16,5$. Пусть также $k = -\lg\left(A \frac{2}{\pi} b\right)$. Тогда влия-

ние переменной составляющей исчезает при $\lg n = M - k$. Это и подтверждается экспериментом.

Выводы

Методы численной фильтрации позволили практически устранить влияние погрешности численных методов и изучить характер погрешности округления, который отличается от общеизвестных (более грубых) оценок.

Результаты эксперимента показывают, что погрешность округления в ряде случаев оказывается существенно меньшей, чем это можно было ожидать. Во-первых, это объясняется несоответствием модели предельной погрешности (предполагающей "наихудший случай") фактической реальности. Во-вторых, это связано с особенностями программирования на языках высокого уровня и использованием ресурсов компьютера трансляторами с этих языков.

Предложены критерии размытости, сравнение численных результатов с единым эталоном, кото-

рый может быть получен путем многократных filtrаций и применением разработанных методов. Сравнительным анализом множества численных данных и результатов фильтрации обеспечивается повышение надежности оценок.

Результаты исследований показали, что на самом деле методами фильтрации можно достичь существенного увеличения точности и надежности оценок.

Использование разработанных методов улучшения численных оценок при разработке промышленных САПР позволит существенно улучшить их надежность практически без увеличения затрат ресурсов.

Список литературы

1. Болотовский Ю. И., Таназлы Г. И. Опыт моделирования систем силовой электроники в среде OrCAD 9.2 // Силовая электроника. 2004. № 1. С. 90–95.

2. Zhitnikov V. P., Zinnatullina O. R., Sherykhalina N. M., Oshmarin A. A. Numerical filtration as method of precising of computing results and error estimation // Proc. of the 8-th Workshop on Computer Science and Information Technologies CSIT'2006. Vol. 1. Karlsruhe, Germany. 2006. P. 134 —137.

3. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. М.: Наука, 1987. 598 с.

4. Волков Е. А. Численные методы. М.: Наука, 1982. 256 с.

5. Шерыхалина Н. М. Методы обработки результатов численного эксперимента для увеличения их точности и надежности // Вестник УГАТУ. Сер. Управление, вычислительная техника и информатика. 2007. Т. 9. № 2 (20). С. 127–137.

6. Житников В. П., Шерыхалина Н. М. Обоснование методов фильтрации результатов численного эксперимента // Вестник УГАТУ. Сер. Фундаментальная и прикладная математика. 2007. Т. 9. № 3 (21). С. 71–79.

7. Шерыхалина Н. М., Ошмарин А. А. Численная фильтрация данных, искаженных нерегулярной погрешностью // Вестник УГАТУ. 2006. Т. 8. № 1 (17). С. 138–141.

8. Житников В. П., Шерыхалина Н. М., Ураков А. Р. Линейные некорректные задачи. Верификация численных результатов: Учеб. пособие. Уфа: УГАТУ, 2007. 100 с.

УДК 004.925.83

Б. Н. Поляков, д-р техн. наук, проф.,

Российский государственный профессионально-педагогический университет, г. Екатеринбург

К созданию САПР технологии правки и оборудования роlikоправильных машин

Предлагаются математическая модель и основные положения современной научной методологии расчета и оптимизации на ЭВМ технологического процесса деформации металла сортового (или листового) проката при правке на роlikоправильной машине (РПМ). Приводятся некоторые результаты применения созданной методологии для условий производства рельсов, с целью совершенствования технологии и оптимизации режимов правки и обоснования конструктивных параметров РПМ с позиции минимума конечной кривизны и уровня остаточных напряжений. В частности, предлагаемые "мягкие" режимы деформирования существенно снижают энергосиловые параметры и повышают потребительские свойства рельсов, а оптимальные режимы правки могут в 25 раз понизить конечную кривизну термоупрочненных рельсов; доказанная высокая чувствительность потребительских свойств рельсов к изменению конструктивных параметров РПМ позволила обосновать их рациональные значения, а также целесообразность автоматизации технологии и оборудования правильных комплексов.

Ключевые слова: САПР, рельс, роlikоправильная машина, напряженно-деформированное состояние, кривизна, оптимизация, остаточные напряжения.

Заключительным этапом производства горячекатаных профилей является процесс правки. Данную технологию относят к числу отделочных операций основного производства. Необходимость правки объясняется тем, что во многих случаях предупредить искривление и нарушение требуемой формы изделия не удастся. Вероятны также случаи, когда предупреждение искривлений принципиально возможно, но неэффективно по технико-экономическим соображениям. Известно, что правка оказывает большое влияние и на формирование

потребительских свойств проката, таких как конечная кривизна и остаточные напряжения, которые, например, применительно к рельсам определяют их контактно-усталостные свойства, а следовательно, срок службы. Указанные обстоятельства делают правку одной из ответственных операций в технологии изготовления прокатных изделий.

Наиболее широко распространенным способом правки в настоящее время является правка знакопеременным упругопластическим изгибом на роlikо-правильных машинах (РПМ).

В данной статье показаны определяющие положения современной научной методологии¹ расчета деформации металла при правке на РПМ и некоторые полученные на ее основе практические рекомендации [1]. Конкретные расчеты и сравнение с экспериментальными данными выполнены для правки рельсов в холодном состоянии на шести роликовых РПМ Нижнетагильского металлургического комбината (НТМК).

Правке на НТМК подвергают закаленные и незакаленные рельсы. Закаленные рельсы проходят последовательно правку в плоскостях максимальной и минимальной жесткости. Незакаленные рельсы правят лишь в плоскости максимальной жесткости.

Под настройкой РПМ понимают задание значений вертикальных перемещений роликов относительно "нулевого" положения. При различных отклонениях в физико-механических свойствах материала профиля (предел текучести, модуль упругости, структура и т. д.), при колебаниях начальной искривленности, биения бандажей и т. д. режимы настройки РПМ изменяются. Поэтому на практике для одинаковых профилей, изготовленных из одного и того же материала, но с различными свойствами необходимо каждый раз подбирать настройку РПМ. Это связано со значительными затратами, так как при эмпирическом подборе настройки несколько профилей бракуют (правку в холодном состоянии можно проводить только один раз). Следовательно, возникает задача выбора рациональных режимов правки для различных профилей.

Также с учетом того, что живучесть рельса, его контактно-усталостные свойства в значительной степени определяются фоном, уровнем и распределением по сечению остаточных напряжений, необходимо, чтобы математическая постановка задачи обоснования режимов правки отражала и этот важный технологический фактор. Кроме того, на металлургических заводах, например на НТМК, сохраняется определенный уровень брака при правке, в частности, появляются трещины на рельсах и наблюдается их поломка, поэтому остается актуальной и экономически обоснованной задача дальнейшего совершенствования существующей технологии правки.

Перечисленные выше соображения приводят к необходимости создания современной математической модели процесса правки и разработки ме-

тодики поиска оптимальных режимов процесса. В данном случае (при пренебрежении концевыми эффектами) задача исследования напряженно-деформированного состояния (НДС) в процессе правки сводится к стационарной задаче упруго-пластичности. Следует особое внимание обратить на сложность создания современной методологии теоретического исследования НДС профиля при правке, идентификации граничных условий и исходных параметров процесса (с позиции минимума конечной кривизны и остаточных напряжений). Это усугубляется тем, что в отечественной и зарубежной литературе практически отсутствует экспериментальная информация по дифференциальным характеристикам (изменение напряжений, деформаций, кривизны и т. п.), известны лишь единичные замеры интегральных параметров (изгибающих усилий, крутящих моментов и т. п.) процесса правки сортовых профилей.

Таким образом, цели теоретических исследований процесса правки сортовых профилей заключались в рационализации технологии и конструкций правильных машин, в оценке целесообразности их автоматизации. Для достижения этих целей была создана методика и программа расчета на ЭВМ НДС профилей в стационарной стадии процесса правки.

Исследованиям процесса правки сортовых профилей посвящено достаточно большое число интересных теоретических и экспериментальных работ [2]. Необходимо отметить особую сложность теоретического описания процесса деформации металла при правке, связанную с решением соответствующей задачи механики деформируемого твердого тела. Поэтому обычно это вынуждает исследователей принимать различные упрощения, часто довольно грубые, существенно влияющие на результаты, что и содержится во многих ранее выполненных теоретических разработках [2]. Ниже предлагается новая постановка задачи исследования НДС металла при правке на РПМ, позволяющая избежать некоторых существенных недостатков.

Исследование НДС сводится в данном случае к решению объемной стационарной изотермической задачи упругопластичности. При постановке задачи здесь и далее будем использовать прямоугольные декартовы системы координат, а также соглашение о суммировании по повторяющимся индексам (правило Эпштейна). Индекс после запятой означает дифференцирование по соответствующей координате.

Обозначим исследуемую область через Ω , ее границу через S . Предположим, что на части границы S_u заданы перемещения, на части S_p — усилия. Объединение области Ω и границы S обозначим как $\bar{\Omega}$. Задача заключается в определении перемещений, напряжений и деформаций в области Ω .

¹ Работа выполнялась сотрудниками Инженерно-конструкторской лаборатории станков горячей прокатки НИИТяжмаша ОАО "Уралмаш" совместно с сотрудниками кафедры "Теоретическая механика" Пермского государственного технического университета под руководством проф., д-ра техн. наук Ю. И. Няшина. Автор выражает им искреннюю признательность и благодарность за многолетнее плодотворное творческое сотрудничество и дружеские взаимоотношения.

В систему уравнений, записанную для удобства последующего решения в приращениях, входят [3]: уравнения равновесия

$$d\sigma_{ij,j} + dF_i = 0, x \in \Omega; i = \overline{1,3}; \quad (1)$$

определяющие уравнения, которые объединяют теории упругости и пластического течения

$$d\sigma_{ij} = D_{ijkl}d\epsilon_{kl}, x \in \bar{\Omega}, i = \overline{1,3}; \quad (2)$$

геометрические соотношения

$$d\epsilon_{ij} = d\epsilon_{ij}^l + d\epsilon_{ij}^p = \frac{1}{2}(dU_{ij} + dU_{ji}), x \in \bar{\Omega};$$

$$i, j = \overline{1,3}. \quad (3)$$

Начальные условия (задача — стационарна, однако нужно задать условия на поверхности Γ входа частиц металла в область Ω):

$$\sigma_{ij} = \sigma_{ij}^0, \epsilon_{ij} = \epsilon_{ij}^0, U_i = U_i^0, x \in \Gamma; i, j = \overline{1,3}. \quad (4)$$

Граничные условия:

$$dU_i = d\bar{U}_i, x \in S_u, i = \overline{1,3}; \quad (5)$$

$$d\sigma_{ij}n_j = d\bar{P}_i, x \in S_p, i = \overline{1,3}. \quad (6)$$

В уравнениях (1)–(6) приняты следующие обозначения: σ_{ij} , ϵ_{ij} — компоненты тензоров напряжений и деформаций; U_i , F_i , P_i — компоненты тензоров перемещений, массовых сил и поверхностных нагрузок; $d\sigma_{ij} = D_{ijkl}d\epsilon_{kl}$, $x \in \bar{\Omega}$, $i = \overline{1,3}$ — компоненты четырехвалентного тензора упруго-пластических свойств; n_j — направляющие косинусы внешней нормали к поверхности; индексы l и p в уравнении (3) и далее относятся к упругим и пластическим деформациям; $\bar{U}_i$ — перемещение недеформированного состояния на границах; $\bar{P}_i$ — граничные значения усилия на роликах.

Перемещения U отсчитываются от недеформированного состояния. Однако в рамках геометрически линейной теории их можно отсчитывать и от значений U^0 на границе Γ (4).

Для определения остаточных напряжений может быть использована теорема о разгрузке в изотермических условиях [4], в том числе в случае появления вторичных пластических деформаций [5]. При использовании теоремы предполагается, что разгрузка начинается во всех точках области одновременно, что в практических задачах редко имеет место. В общем случае произвольного НДС аналогичные теоремы отсутствуют, в силу чего остаточные напряжения будем находить с помощью решения задачи упругопластичности, соответствующей полному снятию силовой нагрузки.

Аналитическое решение поставленной задачи (1)–(6) представляет значительные математические трудности, а ее численное решение требует

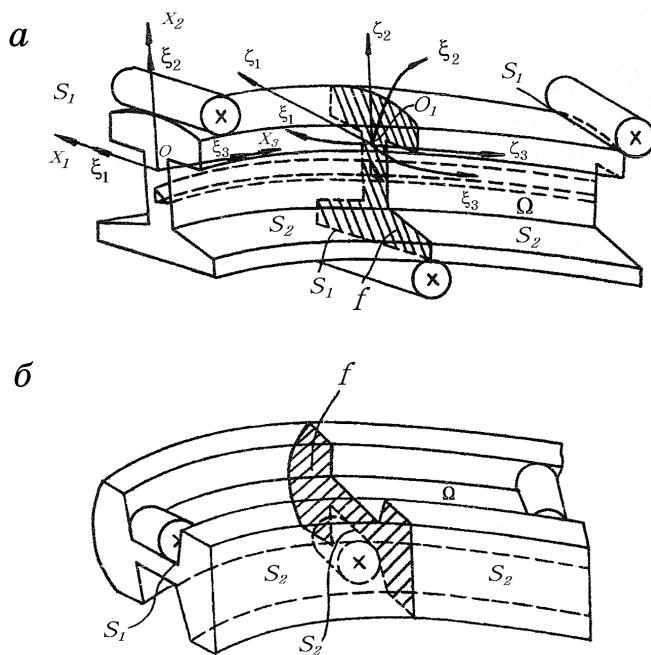


Рис. 1. Неподвижная система отсчета $0x_1x_2x_3$ и движущаяся система координат $0_1\xi_1\xi_2\xi_3$ (а), границы исследуемой области S_u (б):

S_1 — поверхности контакта рельса с правильными роликами при правке в горизонтальной и вертикальной плоскостях; S_2 — поверхности рельса, свободные от контакта с правильными роликами

большого количества машинного времени и значительного объема памяти ЭВМ. Поэтому несколько изменим постановку задачи.

Введем неподвижную систему отсчета $0x_1x_2x_3$ и движущуюся как жесткое целое систему координат $0_1\xi_1\xi_2\xi_3$ (рис. 1). Начало системы $0_1\xi_1\xi_2\xi_3$ находится в центре тяжести рассматриваемого сечения, причем оси $0_1\xi_1\xi_2\xi_3$ и $0_1\xi_1\xi_2\xi_3$ лежат в плоскости поперечного сечения и система $0_1\xi_1\xi_2\xi_3$ движется как жесткое целое так, что одна из осей ($0_1\xi_1\xi_2\xi_3$ или $0_1\xi_1\xi_2\xi_3$) остается параллельной оси $0x_1$ (при изгибе в плоскости $0x_2x_3$) или $0x_2$ (при изгибе в плоскости $0x_1x_3$).

Для определенности изложения будем считать, что изгиб происходит в плоскости $0x_2x_3$. Для учета истории нагружения необходимо ввести так же отсутствующую лагранжеву систему координат $0_1\xi_1\xi_2\xi_3$ (для индивидуализации частиц сплошной среды).

Будем считать, что углы поворота при правке малы. В силу этого систему $0_1\xi_1\xi_2\xi_3$ можно было бы не вводить, а изложение методики вести в системе $0x_1x_2x_3$.

Воспользуемся гипотезой плоских сечений, согласно которой (при изгибе в плоскости $0x_2x_3$) продольная деформация ϵ_{33} любого волокна равна

$$\epsilon_{33} = a_0 + a_2\xi_2, \quad (7)$$

где a_0 — продольная деформация линии центров тяжести.

Величина a_2 с точностью до знака равна кривизне проекции линии центров тяжести на плоскость изгиба $0x_2x_3$ (см. рис. 1) [6]:

$$a_2 = -\kappa_2. \quad (8)$$

Найдем приращение продольной деформации $\Delta\varepsilon_{33}$ для любой точки поперечного сечения профиля:

$$\Delta\varepsilon_{33} = -\xi_2\Delta\kappa_2 + \Delta a_0. \quad (9)$$

Из гипотезы плоских сечений следует

$$\Delta\varepsilon_{13} = \Delta\varepsilon_{23} = 0. \quad (10)$$

Введем предположение

$$\Delta\sigma_{13} = \Delta\sigma_{23} = 0. \quad (11)$$

Запишем определяющее соотношение для $\Delta\sigma_{33}$:

$$\Delta\sigma_{33} = D_{33ij}\Delta\varepsilon_{ij}, \quad i, j = \overline{1, 3}. \quad (12)$$

Соотношение (12) можно записать в виде

$$\Delta\sigma_{33} = D_{3333}\Delta\varepsilon_{33} + \Delta\sigma'_{33}, \quad (13)$$

где

$$\Delta\sigma'_{33} = D_{33ij}\Delta\varepsilon_{ij}, \quad i, j = \overline{1, 2}.$$

Отметим, что $\Delta\sigma_{33}$ удовлетворяет следующим интегральным уравнениям равновесия:

$$\left\{ \begin{array}{l} \int_F \Delta\sigma_{33}\xi_2 dF = -\Delta M_2, \\ \int_F \Delta\sigma_{33} dF = 0. \end{array} \right. \quad (14)$$

$$(15)$$

При малых изгибах можно записать

$$\kappa_2 = \frac{d^2 U}{dx^2}, \quad (16)$$

где U — поперечное перемещение центра тяжести сечения.

Граничные условия имеют вид

$$U_i = \widehat{U}_i, \quad x \in S_u; \quad M_2^0 = M_2^N = 0, \quad (17)$$

где U_i — заданные перемещения роликов РПМ; M_2^0 и M_2^N — изгибающие моменты на первом и последнем роликах РПМ соответственно.

К граничным условиям необходимо также добавить условия сопряжения на промежуточных роликах РПМ [1].

Теперь при известных D_{33kl} и $\Delta\varepsilon_{ij}$ ($i, j = \overline{1, 2}$) можно определить из решения задачи (9)—(17) $\Delta\varepsilon_{33}$ и $\Delta\sigma_{33}$.

Вместе с тем, при выполнении предположений (10) и (11) краевая задача (1) —(6) для определения НДС в области поперечного сечения F принимает вид

$$\Delta\sigma_{ij,j} + \Delta F_i = 0, \quad x \in F; \quad i, j = \overline{1, 2}; \quad (18)$$

$$\Delta\sigma_{ij} = D_{ijkl}\Delta\varepsilon_{kl} + D_{ij33}\Delta\varepsilon_{33}, \quad x \in F; \\ i, j, k, l = \overline{1, 2}; \quad (19)$$

$$\Delta\varepsilon_{ij} = \Delta\varepsilon_{ij}^l + \Delta\varepsilon_{ij}^p = \frac{1}{2}(\Delta U_{ij} + \Delta U_{ji}), \quad x \in F; \\ i, j = \overline{1, 2}; \quad (20)$$

$$\Delta U_i = \Delta \widehat{U}_i, \quad x \in S_u; \quad \Delta\sigma_{ij}n_j = \Delta \widehat{P}_i, \quad x \in S_p; \\ i = \overline{1, 2}. \quad (21)$$

Таким образом, при принятых предположениях исходная задача (1)—(6) сводится к решению системы уравнений (9)—(21). Обращаем внимание, что ее можно "расщепить" на две связанные между собой задачи. Первая задача (назовем ее задачей А) описывает задачу изгиба профиля при учете поперечных напряжений и деформаций. Она сводится к решению системы уравнений (9)—(17). Вторая задача (задача В) описывает исследование НДС частиц элементов поперечного сечения профиля при их движении по линиям тока в процессе правки на РПМ, с учетом продольных напряжений и деформаций, и сводится к решению системы уравнений (18)—(21). Для решения задачи (18)—(21) используется алгоритм, подробно описанный в работе [3].

Совокупность задач А и В позволяет решить задачу правки. При решении задачи не учитываются контактные напряжения и вся поверхность (граничные условия для задачи В) считается свободной, так как это воздействие носит локальный характер (хотя контактные напряжения возникают в области максимальных пластических деформаций) и вследствие принципа Сен-Венана не внесет существенного изменения в результаты расчета НДС. Задача исследования НДС при правке является прямой задачей для задачи выбора рациональных режимов РПМ (задачи оптимизации). Реализация решения задачи упругопластического знакопеременного изгиба требует значительного времени счета даже на современных ЭВМ. Это неприемлемо для задачи оптимизации, которая требует многократного решения прямой задачи. В то же время результаты расчета по приведенной методике показывают, что компоненты тензора напряжений (за исключением продольной) почти во всех точках поперечного сечения, исключая область контакта, существенно меньше σ_{33} . Это свидетельствует о возможности использования схемы одноосного напряженного состоя-

ния (все компоненты напряжений, за исключением продольной, полагаются равными нулю).

Физические уравнения в этом случае примут вид

$$\Delta\sigma_{33} = D^*\Delta\varepsilon_{33}, \quad (22)$$

где D^* — секущий модуль [1]; $\Delta\sigma_{33}$ — приращение продольного напряжения в исследуемой точке поперечного сечения профиля.

Дифференциальное уравнение изогнутой линии центров тяжести получаем подстановкой, в частности, уравнения (22) в (14) и (15), и которое имеет вид

$$\frac{d^2x_{C_2}^{(n)}}{dx_3^{(n)2}} = \frac{\Delta a_0 \int_F D^*\xi_2 dF + \kappa_2^0 \int_F D^*\xi_2 dF + \Delta M_2}{\int_F D^*\xi_2^2 dF}. \quad (23)$$

При рассмотрении знакопеременного упруго-пластического изгиба необходимо учесть эффект Баушингера. В данной работе эффект учитывался путем использования принципа Мазинга, согласно которому кривая повторного знакопеременного напряжения совпадает с соответствующей кривой при первом нагружении, построенной в осях с удвоенным масштабом и обратным направлением (рис. 2).

Если при первом нагружении были достигнуты напряжения и деформации ε_{33}^n , то при повторном

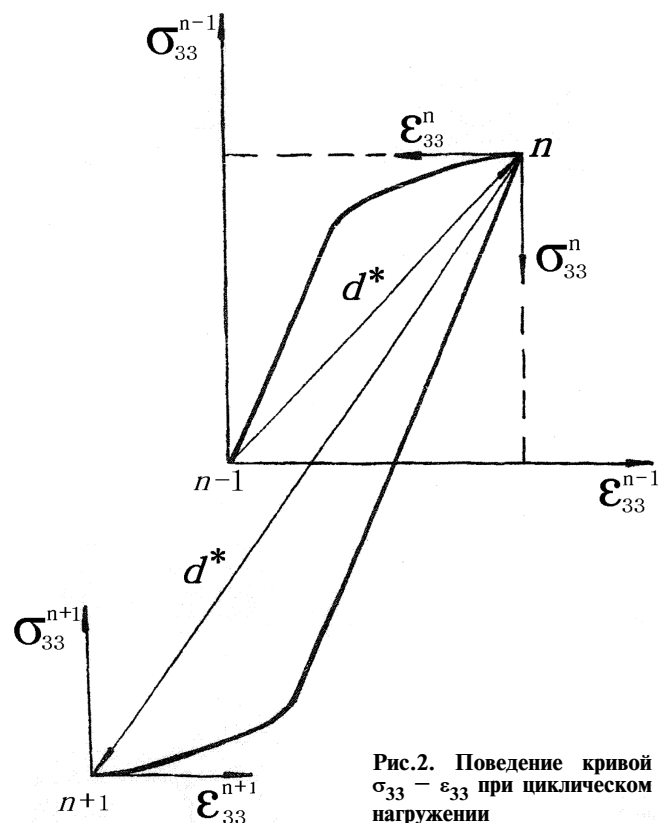


Рис.2. Поведение кривой $\sigma_{33} - \varepsilon_{33}$ при циклическом нагружении

нагружении в обратном направлении предел текучести определяется параметрами σ_s^{n+1} , ε_s^{n+1} , удовлетворяющими уравнениям:

$$|\sigma_{33}^n - \sigma_s^{n+1}| = 2\Phi\left(\frac{\varepsilon_{33}^n - \varepsilon_{33}^{n+1}}{2}\right) \quad (24)$$

при наличии вторичных пластических деформаций,

$$|\sigma_{33}^n - \sigma_s^{n+1}| = 2\Phi\left(\frac{\varepsilon_{33}^n - \varepsilon_{33}^{n+1}}{2}\right) \quad (25)$$

при упругой разгрузке.

Поскольку $\sigma_s^n = E\varepsilon_s = \Phi(\varepsilon_s)$, то

$$\sigma_s^{n+1} = \sigma_{33}^n - 2\sigma_s^n, \quad \varepsilon_s^{n+1} = \varepsilon_{33}^n - 2\varepsilon_s^n. \quad (26)$$

Отсюда следует

$$-\sigma_s^{n+1} < \sigma_s^n, \quad (27)$$

так как $\sigma_{33}^n > \sigma_s^n$.

Таким образом, принцип Мазинга учитывает эффект Баушингера. В случае квазиобъемной постановки вместо σ_{33} , ε_{33} необходимо брать ε_{ii} , σ_{ii} .

Для получения кривой деформационного упрочнения $\sigma_{33} = f(\gamma_{33})$ для термоупрочненных рельсов производства НТМК были выполнены экспериментальные работы [1]. Эксперименты проводились на растяжение и сжатие, и зависимость $\sigma_{33} = f(\gamma_{33})$ была получена в виде

$$\sigma_{33} = k_1 \varepsilon_{33}^{k_2}. \quad (28)$$

Предел текучести при сжатии (повторное нагружение) был определен согласно принципу Мазинга.

Для решения задачи оптимизации режимов правки на основе представленной математической модели применен метод штрафных функций [1]. Здесь в качестве целевой функции используется конечная искривленность профиля. В вектор управления U вошли перемещения роликов РПМ, а ограничения на эти перемещения формируют соотношения типа неравенств

$$|U_i| \leq |\bar{U}_i|. \quad (29)$$

В качестве дифференциальных связей выступают уравнения краевой задачи упругопластичности (9)–(21). Целевая функция при использовании метода штрафных функций записывается в виде

$$I_1(U, V) = \kappa_{\text{кон}}(U, V) + k_1 \sum_{i=1}^N U_i^2 \delta_i, \quad (30)$$

где $\kappa_{\text{кон}}(U, V)$ — конечная искривленность профиля; δ_i — весовые коэффициенты; k_1 — коэффициент штрафа,

$$k_1 = \begin{cases} 0, & \text{при выполнении условия (29),} \\ +\infty, & \text{если условие (29) не выполняется.} \end{cases}$$

Для решения поставленной задачи применен метод безусловной оптимизации — метод деформируемого многогранника Нелдера—Мида [1].

Оценка достоверности приведенной выше математической модели НДС профиля при правке на РПМ была выполнена на основе решения ряда тестовых задач. В частности, для проверки упругого решения методом Верещагина был рассчитан изгиб профиля прямоугольного сечения на семи роликах РПМ. Результаты расчета интегральных характеристик процесса по квазиобъемной и стержневой методикам сопоставлены с известными экспериментальными данными. Также обоснована степень полинома, аппроксимирующего радиус искривленности профиля на пролете РПМ [2].

На основе разработанной современной научной методологии исследования и оптимизации НДС профиля в процессе знакопеременного изгиба создана программа расчета на ПЭВМ дифференциальных и интегральных параметров технологии правки на РПМ рельсов и других фасонных профилей. Программа позволяет вычислить компоненты векторов перемещений, тензоров напряжений и деформаций и их распределение в любой точке подвергаемого правке профиля; распределение по роликам усилий, изгибающих и крутящих моментов, мощностей; распределение остаточных напряжений и значения конечной кривизны. Исходными данными для программы являются: взаимное расположение роликов РПМ (настройка), шаг и диаметр роликов, конфигурация и исходное значение кривизны выправляемого профиля, предел текучести и форма кривой деформационного упрочнения материала профиля.

Применение программы позволяет с достаточной степенью точности определить рациональную настройку роликов, обеспечивающую минимальную остаточную кривизну, рассчитать интегральные характеристики режима правки, сократить трудозатраты на выбор параметров настройки, исключить проведение трудоемких ручных расчетов технологических и энергосиловых параметров РПМ. Программа применялась для обоснования правки двутавровых и рельсовых профилей и конструктивных параметров РПМ при их проектировании для металлургических предприятий ряда зарубежных государств [1, 2].

С помощью программы на ЭВМ выполнены численные и параметрические исследования некоторых аспектов применяемых на НТМК техно-

гий правки "сырых" (незакаленных) и термоупрочненных рельсов Р65, в результате которых предложен ряд технологических и конструктивных рекомендаций. Наиболее важные из них следующие.

1. Нецелесообразно уменьшение применяемых в настоящее время в конструкциях РПМ значений шагов и диаметров роликов, так как это отрицательно повлияет на формирование остаточных напряжений.

2. В противовес вынужденно применяемым (для уменьшения разброса значений конечной кривизны) "жестким" режимам правки (со степенями деформации более 1,0 %), которые хотя и позволяют достичь большей прямолинейности, но имеют ряд существенных недостатков, а именно: повышают неоднородность деформации, что способствует формированию худшей картины распределения остаточных напряжений, приводят к возникновению значительной разницы в поперечных размерах между непрямыми концами и правленной частью рельса, а также к образованию и развитию трещин, а иногда и к разрушению по имеющимся дефектам поверхности, предложены "мягкие", щадящие режимы (максимальные деформации на уровне 0,5—0,6 %), которые уменьшают поврежденность металла, т. е. повышают служебные свойства рельсов, сокращают вероятность поломок при правке, что в конечном итоге положительно отражается на живучести и сроке службы рельсов.

3. Целесообразен переход к технологии правки как минимум на восьмироликовых РПМ, которые за счет большего числа упругопластических перегибов (в сравнении с шестироликовыми) формируют условия меньшей чувствительности остаточной (конечной) кривизны к колебаниям размеров поперечного сечения рельса, механических свойств (предела текучести) его материала и погрешностям настройки. При этом за счет применения "мягких" режимов можно дополнительно снизить нагруженность подшипниковых узлов, оборудования и приводов, например по моментам изгиба и усилиям правки — на 21 % (рис. 3), и уменьшить поврежденность металла (рис. 4).

Рекомендация о числе роликов в конструкции РПМ удачно корреспондируется с параметрами правильных машин, применяемых на сравнительно новом (июнь, 2002 г.) рельсопрокатном стане фирмы "Steel DYNAMICS, INC" ("SDI", Columbia City, USA), где горизонтальная машина имеет девять роликов, диаметром 1060 мм, а вертикальная — семь роликов, диаметром 750 мм.

4. Решение задач оптимизации технологии правки показало, что полученный рациональный режим правки "сырого" рельса Р65 наиболее близок к режиму, применяемому в настоящее время в рельсо-балочном цехе НТМК (что дополнительно

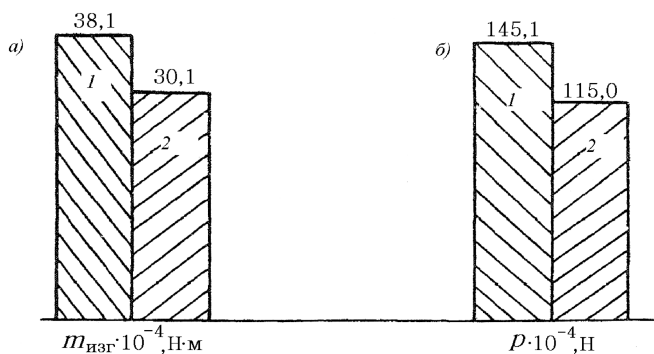


Рис. 3. Снижение средних значений изгибающих моментов (а) и усилий (б) при правке на восьмироликовых РПМ: 1 — "жесткие" режимы деформирования; 2 — "мягкие" режимы деформирования

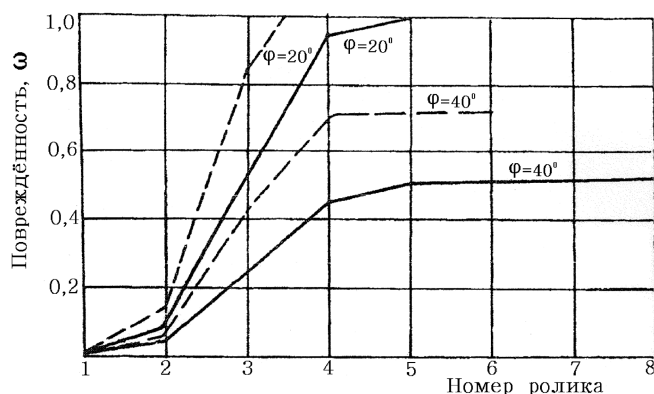


Рис. 4. Накопление поврежденности при правке (φ — угол исходного поверхностного дефекта): ----- — на шестироликовой РПМ; — на восьмироликовой РПМ

но подтверждает достоверность методики и результатов), а оптимальный режим правки термоупрочненного рельса в плоскости максимальной жесткости позволяет уменьшить конечную кривизну (в сравнении с действующим) примерно в 25 раз [2].

5. Доказано, что заметное изменение остаточной кривизны рельса и остаточных напряжений происходит уже при отклонении роликов от заданного положения на 0,1...0,2 мм (т. е. имеет место высокая чувствительность потребительских свойств проката к конструктивным параметрам правильных машин), поэтому точность настрой-

ки, формируемая точностью перемещения роликов, значениями биения бандажей и люфтов в подвижных частях и жесткостью конструкции, должна обеспечиваться конструкцией РПМ, а определяющие значения должны находиться в следующих пределах: точность перемещения роликов и биение бандажей должны быть не более 0,1...0,2 мм, суммарный люфт — не более 0,5 мм, а жесткость РПМ — не менее 2000 кН/мм. Кроме того, с целью эффективного управления остаточными напряжениями в правящемся профиле, целесообразно перейти от настройки РПМ по перемещениям роликов к настройке по напряжениям (по усилиям на роликах), предусмотрев в конструкции РПМ установку датчиков давления на каждом ролике.

6. Показана целесообразность автоматизации правильного комплекса и предложен состав автоматических систем контроля и микропроцессорного программного управления [2].

В современных сложных экономических условиях для обеспечения конкурентоспособности рельсов и других профилей проката (а это абсолютно реально) целесообразно продолжить работу по совершенствованию технологии правки и оборудования, а также внедрению автоматических систем с учетом приведенных выше результатов.

Список литературы

1. Поляков Б. Н., Няшин Ю. И. К созданию САПР роликоправильных машин. САПР в тяжелом машиностроении: Сб. науч. тр. / Под ред. Б. Н. Полякова. Екатеринбург: Изд-во Урал. гос. проф.-пед. ун-та, 2000. С. 45—68.
2. Обследование нагруженности механического оборудования УБС НТМК. Совершенствование технологии и оборудования для правки рельсов тяжелых типов и балочных профилей. Исследование процесса правки сложных профилей на роликоправильных машинах и термомеханической правки растяжением: Отчет по НИР / Науч.-исслед. конструк.-технол. ин-т тяжелого машиностроения (НИИтяжмаш); Руководители работы — Б. Н. Поляков, Ю. Д. Макаров; Отв. исполнители — Ю. Д. Макаров, С. В. Колмогоров; т. ГР 80011770; Инв. т. 0286.0040780. Свердловск, 1985. Т. 3. 79 с.
3. Поздеев А. А., Няшин Ю. И., Трусов П. В. Остаточные напряжения: теория и приложения. М.: Наука, 1982. 112 с.
4. Ильюшин А. А. Пластичность: В 3 ч. Ч. 1: Упругопластические деформации. М.; Л.: Гостехиздат, 1948. 376 с.
5. Москвитин В. В. Циклические нагружения элементов конструкции. М.: Наука, 1982. 394 с.
6. Работнов Ю. Н. Механика деформируемого твердого тела. М.: Наука, 1979. 744 с.

Ж. Е. Зимнухова,
В. А. Немтинов, д-р техн. наук, доц.,
Тамбовский государственный
технический университет

О подходе к построению автоматизированной информационной системы поддержки принятия решений для проектирования процессов производства изделий из металлов

Предложен подход к построению автоматизированной информационной системы поддержки принятия решений при проектировании технологических процессов класса технических систем, для которых при получении целевой продукции могут быть использованы различные технологии, виды оборудования, приспособлений и вспомогательных материалов.

Ключевые слова: информационная система, поддержка принятия решений, проектирование технологических процессов.

Введение

Новые эффективные методы и технологии проектирования, создания и модернизации процессов изготовления изделий из металлов призваны обеспечить готовность машиностроительного производства к выпуску изделий заданного качества в минимальные сроки при наименьших трудовых, материальных и финансовых затратах. В настоящее время широко применяются интегрированные системы компьютерного проектирования и технологической подготовки производства. Применение их значительно повышает эффективность труда конструктора и технолога, автоматизируя рутинные операции проектирования и оформления документации для конкретно принятого варианта технологического процесса (ТП) получения целевой продукции. Однако при этом остается нерешенной задача получения оптимального варианта (или нескольких близких к оптимальному вариантам) ТП для реальных условий производства изделий из металлов.

На основании проведенного анализа известных подходов к построению автоматизированных информационных систем (АИС), информационных моделей ТП производства изделий из металлов [1—5] отмечены общие недостатки, имеющие

место при автоматизации разработки ТП производства изделий из металла, а именно, отсутствие:

- представления всей совокупности задач проектирования ТП машиностроительного производства — автоматизированного выбора марки металла, способа получения и вида заготовки в зависимости от вида упрочнения; автоматизированного выбора ТП, оборудования, приспособлений, вспомогательных материалов и режимных параметров для механообработки и для упрочняющей обработки — в виде единого комплекса с использованием теории сложных систем и, как следствие, отсутствие единой информационной базы;
- комплексного оценивания (с экономических, экологических и технологических позиций, а также с учетом профессионального риска) конструкторских и технологических решений, а также интеллектуализации обработки информации в известных АИС.

Это приводит к тому, что не удастся рассмотреть все многообразие альтернативных вариантов ТП, применяемых на всех этапах изготовления изделий из металла, и выбрать среди них вариант, оптимальный с различных позиций.

При проведении анализа математических методов решения задач разработки ТП производства машиностроительных изделий были отмечены основные характеристики этих задач: большая размерность, множество критериев оценки различной природы, наличие многоуровневой с большой размерностью информационной базы и т. д..

В связи с этим описываемая здесь работа посвящена развитию подхода к построению АИС поддержки принятия решений, используемой при решении задач проектирования экологически безопасных ТП производства изделий из металлов.

1. Методологические основы подхода

Основу подхода к построению АИС поддержки принятия решений для проектирования ТП производственных технических систем (ПТС) составляет реализация возможности представления всей совокупности решаемых задач с позиций теории иерархических систем на всех этапах принятия конструкторско-технологических решений и их комплексной оценки [6, 7].

Комплекс особо значимых задач, решаемых АИС при проектировании ТП технической системы, на примере производства изделий из металлов представлен в виде иерархической структуры (рис. 1).

Используя терминологию теории систем, систему автоматизированного проектирования ТП машиностроительного производства можно пред-

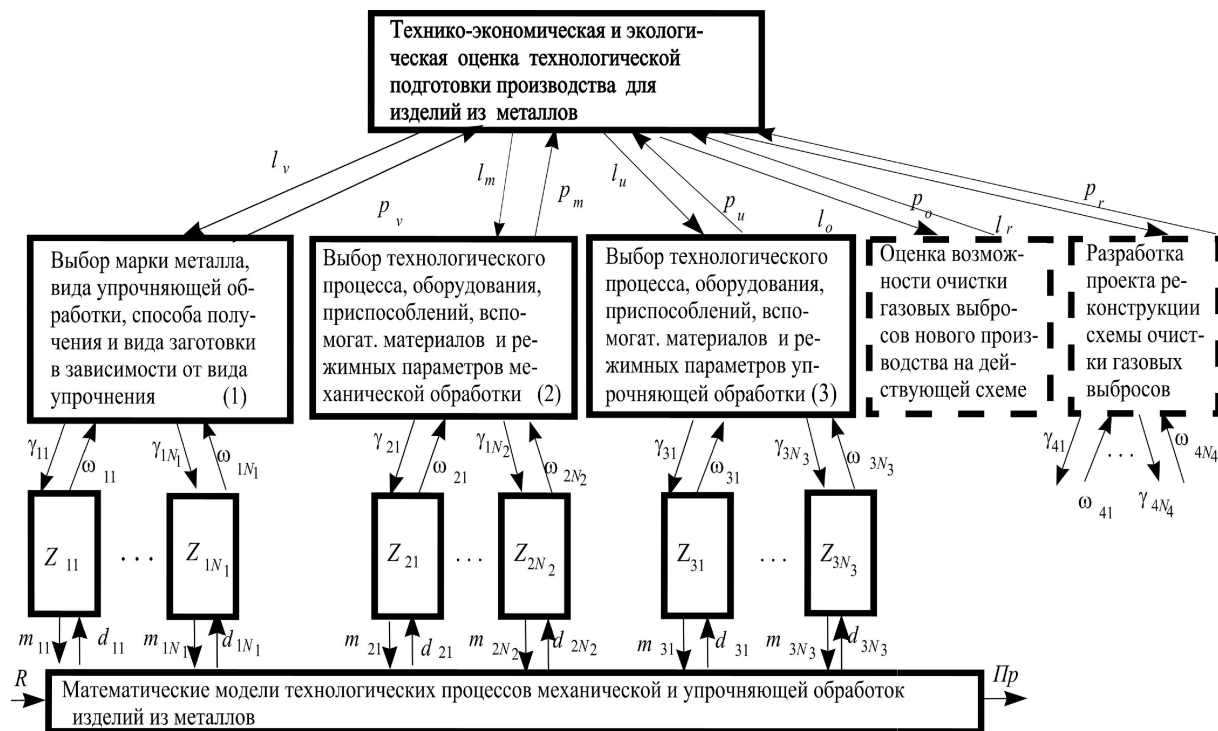


Рис. 1. Структурная схема подзадач, решаемых АИС, при проектировании ТП производства изделий из металлов

ставить как отношение на декартовом произведении-множеств:

Pr — множество решений задачи проектирования ТП производства изделий из металлов;

M_v, M_m, M_u — множества управляющих сигналов для процесса конструирования в задачах нижнего уровня (выбора материала и вида его упрочнения, вида заготовки, определения характеристик допустимого метода получения заготовки, наборов оборудования и вспомогательных материалов, а также технологических операций для обработки детали и др.), например, геометрические размеры детали, технологические свойства и прочностные характеристики детали и др.;

$D_v, D_m, D_u, W_v, W_m, W_u, P_v, P_m, P_u$ — множества информационных сигналов о решении локальных задач, например, свойства выбранных марок стали; ТП механической и упрочняющей обработок; типы, характеристики станочного оборудования и печей для упрочняющей обработки; значения критериев локальных задач оптимизации и др.;

$\Gamma_v, \Gamma_m, \Gamma_u, L_v, L_m, L_u$ — множества координирующих сигналов для локальных задач нижестоящих уровней, например, категория значимости и серийность детали, наличие металла на складе, длительности отдельных ТП упрочняющей обработки, времени пребывания деталей в печи, норма загрузки деталей в печь и др.

Наличие множеств различных критериев оптимальности при принятии решений этих задач привело к необходимости использования методов многокри-

териальной оптимизации. При этом в каждом конкретном случае решаются проблемы выбора: альтернативных вариантов; методов решения задачи с учетом оценки вариантов по всем рассматриваемым критериям; принципа нормализации, приводящего все критерии к единому масштабу измерения и позволяющего проводить их сопоставления; принципа учета приоритета, позволяющего отдавать предпочтение более важным, по мнению технологов, критериям.

В формализованном виде задача проектирования ТП ПТС заключается в поиске минимума целевой функции $F(w)$

$$w^{\text{opt}} = \arg \min_{w \in W} F(w), \quad (1)$$

при выполнении следующих ограничений:

- детерминированных ограничений на выходные переменные

$$w_i^* \leq w_i \leq w_i^{**}, i = \overline{1, n}; \quad (2)$$

функциональных ограничений

$$c_j^* \leq f_j(w_1, w_2, \dots, w_{n_j}) \leq c_j^{**}, j = \overline{1, k}; \quad (3)$$

ограничений на значения показателей технологических процессов производственных технических систем

$$F^v(w) \leq F^{v, \text{lim}}, v = \overline{1, K_1}, \\ F^t(w) \geq F^{t, \text{lim}}, t = \overline{1, K_2}; \quad (4)$$

операторов, описывающих математические модели принятия решения при проектировании ТП ПТС,

$$\bar{\varphi}_l : \Delta_{l1} \times \Delta_{l2} \times \dots \times \Delta_{lm} \times \dots \times \Delta_{lp} \rightarrow W_1, \\ l = \overline{1, K_3}. \quad (5)$$

Здесь w_i^* , c_j^* , w_i^{**} , c_j^{**} — соответственно минимальные и максимальные значения в ограничениях (2) и (3); $f_j(\circ)$ — некоторые функции от w_1 , w_2 , ..., w_{n_j} , заданные в явном виде; $F^v(w)$, $F^t(w)$, $F^v, \lim$, $F^t, \lim$ — соответственно значения показателей технологических процессов изготовления детали (процент брака, технологичность и т. п.) и их заданные значения; K_1 , K_2 — соответственно число показателей, для которых задается условие (4); $\bar{\varphi}_l$ — функциональный оператор; Δ_{lm} — множество данных; K_3 — число функциональных операторов; p — число множеств данных.

Задача (1)–(5) относится к классу задач дискретного программирования. Ввиду высокой размерности задачи и традиций организации труда для многих классов ПТС, в том числе и машиностроительных производств, она, в соответствии с разработанной нами структурной схемой, разбивается на ряд подзадач меньшей размерности.

Обобщенный критерий $F(w)$ можно записать как

$$F(w) = \sum_{i=1}^{K_0} \rho_i \omega^i(w) = \rho_1 \omega^1(w) + \dots + \rho_{K_0} \omega^{K_0}(w), \quad (6)$$

где $\rho_1, \dots, \rho_{K_0}$ — весовые коэффициенты,

$$\rho = \{\rho_i\} = \{\rho_i : \rho_i > 0, i = 1, \dots, K_0, \sum_{i=1}^{K_0} \rho_i = 1\}; \quad (7)$$

$\rho_i \omega^i(w)$ — взвешенные потери по i -му критерию; $\omega^i(w) = \omega^i(F^i(w))$, $i = 1, \dots, K_0$, $w \in W$ — монотонные функции, преобразующие каждую функцию цели $F^i(w)$, $i = 1, \dots, K_0$, $w \in W$ к безразмерному виду;

$$\omega^k(w) = \frac{F^k(w) - F^{k0}}{F_{(\max)}^k - F^{k0}}, \quad w \in W, k = \overline{1, K_1'},$$

$$\omega^k(w) = \frac{F^{k0} - F^k(w)}{F^{k0} - F_{(\min)}^k}, \quad w \in W, k = \overline{1, K_2'},$$

где $F_{(\max)}^k$ и $F_{(\min)}^k$ — соответственно наибольшее значение минимизируемых и наименьшее значение максимизируемых функций $F^k(w)$, $w \in W$ на множестве допустимых альтернатив W ; F^{k0} — оптимальное значение функций цели $F^k(w)$, $w \in W$. Значения $\omega^i(w)$, $i = 1, \dots, K_0$, $w \in W$ лежат в пределах от 0 до 1.

Для выбора единственного решения в задаче принятия сложного решения требуется задать весовые коэффициенты ρ_i , $i = 1, \dots, K_0$, удовлетво-

ряющие соотношению (7) и отражающие относительную важность функций цели $F^1(w)$, ..., $F^{K_0}(w)$, $w \in W$. Наиболее эффективными подходами к определению этого предпочтения являются методы ранжирования и приписывания баллов [7] (последний применен в работе).

Для формализованного описания информационных массивов данных, необходимых для решения описанных выше задач, создана структурированная база данных. Структура данных области исследования отображается информационно-логической моделью (ИЛМ) технологического процесса ПТС рассматриваемого класса и представляет собой объединение множеств данных, локальных моделей стадий ТП и связей между ними. Она представлена следующим кортежем:

$$M_n = (\mu_{n1}, \dots, \mu_{ni}, \dots, \mu_{n, I_n}, \delta_{n1}, \dots, \delta_{nj}, \\ \delta_{n, J_n}, \rho_1^n, \dots, \rho_{S_n}^n), \quad (8)$$

где M_n — оператор ИЛМ; $\mu_{n1}, \dots, \mu_{n, I_n}$ — множество локальных моделей; $\delta_{n1}, \dots, \delta_{n, J_n}$ — множество данных ИЛМ; $\rho_1^n, \dots, \rho_{S_n}^n$ — множество правил модели M_n .

Каждая локальная модель, в свою очередь, описывается кортежем аналогичной формы. Локальная модель нижнего уровня включает в себя множества данных и связей между ними в виде правил:

$$M_n = (\mu_{n1}, \dots, \mu_{n, I_n}, \delta_{n1}, \dots, \delta_{n, J_n}, \rho_1^n, \dots, \rho_{S_n}^n) \\ \downarrow \\ \mu_{ni} = (\mu_{n-1, 1}, \dots, \mu_{n-1, I_{n-1}}, \delta_{n-1, 1}, \dots, \\ \delta_{n-1, J_{n-1}}, \rho_1^{ni}, \dots, \rho_{S_{ni}}^{ni}), \quad (9) \\ \downarrow \\ \dots \downarrow \dots \\ \mu_{1b} = (\delta_1^{1b}, \dots, \delta_{S_2}^{1b}, \rho_1^{1b}, \dots, \rho_{S_1}^{1b}),$$

где $\rho_1^{1b}, \dots, \rho_{S_1}^{1b}$ — множество правил модели μ_{1b} , $b = \overline{1, B}$.

В свою очередь, правила, входящие в ИЛМ, построены по типу: если ... (условия выполняются), то ... (реализация следствия), и в формализованном виде описываются следующим образом:

$$\rho_i^{1b} : \{if((\delta_1^{1b'} A_1 z_1^{1b'}) \wedge (\delta_2^{1b'} A_2 z_2^{1b'}) \wedge \dots \\ \dots \wedge (\delta_n^{1b'} A_n z_n^{1b'})) \text{ then } (\delta_{1k}^{1b''} A'_1 z_{1k}^{1b''})\},$$

где *if* — обозначение условия "если", *then* — обозначение следствия "то"; $A'_1, A_i \in \{=, >, \geq, <, \leq\}$,

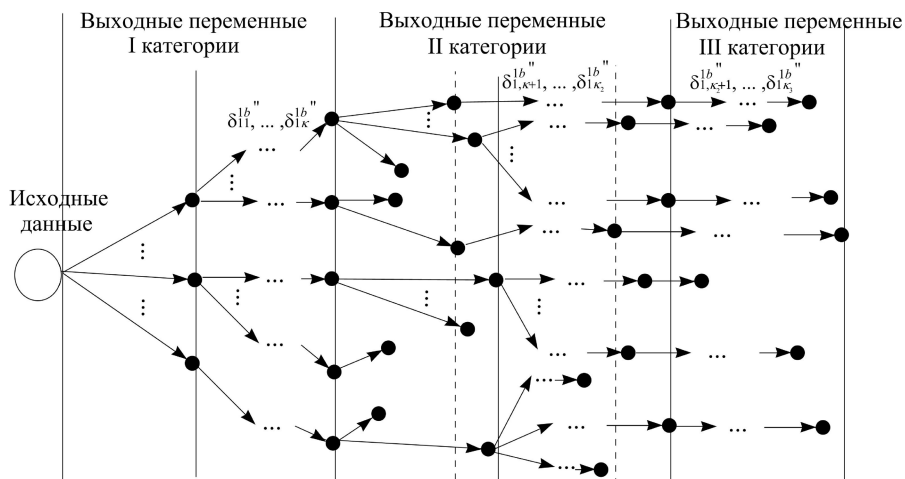


Рис. 2. Схема реализации процедурной модели принятия решения

$i = \overline{1, n}$ — арифметический оператор ИЛМ; $\Lambda_i \in \{\wedge, \vee, !\}$ — логический оператор ИЛМ; $\delta_s^{1b'}$, $\delta_{1k}^{1b''}$ — соответственно входные и выходные данные модели $\mu_{1b'}$; $Z^{1b'} = \{z_1^{1b'}, \dots, z_n^{1b'}\}$ — множество значений входных данных $\delta_s^{1b'}$, $z_1^{1b''} \in \{z_{11}^{1b''}, \dots, z_{1k}^{1b''}\}$ — значение для выходных данных $\delta_{1k}^{1b''}$; n — число условий.

В тех случаях, когда множество вариантов решений не превышает 10^4 , учитывая быстродействие современных ПЭВМ, искомое решение можно находить методом полного перебора вариантов. При более высокой размерности задач предлагается процедурная модель, общая схема реализации которой основана на последовательном анализе и отсеивании части элементов, составляющих вариант решения, путем исключения бесперспективных, как по ограничениям, так и по целевой функции.

Исходя из специфики проектирования ТП для данного класса производственных систем, все выходные переменные должны быть разбиты на три категории. К первой категории относятся выходные переменные, для которых при формировании множества вариантов решения используются все их возможные значения. Вторая категория объединяет выходные переменные, для которых при формировании множества вариантов решения используются только те значения, которые попадают в окрестность "оптимистичных" значений локального критерия. Данная окрестность определяется следующим правилом: $F^h r \leq F^{\text{opt}}$, $r < 1$, $h = 1, 2, \dots, H$, где r — коэффициент задаваемый лицом, принимающим решение (ЛПР), для формирования окрестности "оптимистичных" значений локального критерия F ; F^h — значение кри-

терия для h -го варианта формирования значений выходных переменных; H — множество их допустимых значений. В третью категорию попадают наименее значимые выходные переменные, для которых при формировании множества вариантов решения используется лишь одно значение выходной переменной.

При проектировании ТП деление на категории осуществляется ЛПР в процессе формирования множества вариантов решения задачи.

Схема реализации процедурной модели приведена на рис. 2.

2. Практическая реализация подхода к построению АИС

Для практической реализации предлагаемого подхода к построению АИС детализируем информационные и процедурные модели для класса машиностроительных производств.

Для решения задачи, в соответствии с иерархической структурой, заменим общую задачу оценки технико-экономической эффективности и экологической безопасности при проектировании процессов производства изделий из металлов последовательным рассмотрением подзадач меньшей размерности, обозначенных на рис. 1 цифрами 1, 2, 3, имеющих и самостоятельное значение в процессе технологической подготовки машиностроительного производства [3].

В случае отсутствия решения на каждом следующем этапе проектирования ТП ЛПР выбирается другой "оптимистичный" вариант решения задачи предыдущего этапа.

В работе предлагается следующая постановка задачи 1. Для конструируемой детали с заданными геометрическими размерами L и массой G , а также условиями эксплуатации U^d , серийностью производства Sp^d и категорией значимости (степенью ответственности) Kz на множестве $W_1 = M^d \times Tu^d \times Z^d \times P_z^d \times V_z^d$ найти такой вариант $w_1^* \in W_1$, для которого стоимость получения заготовки из выбранных марок стали с соответствующей упрочняющей обработкой имеет минимальное значение. Множество W_1 представляет собой декартово произведение подмножеств допустимых видов: материалов, используемых для изготовления детали M^d ; упрочняющей обработки, обеспечивающей заданные показатели качества изделия Tu^d ; заготовок Z^d ; способов получения заготовок P_z^d и вспомогательных материалов для проведения методов получения заготовок V_z^d .

В формализованном виде задача заключается в поиске минимума целевой функции

$$F_1^{\text{opt}} = \min_{W_1} ((S_M + S_{TZ} + S_{VS} + S_{OB})K_{SS} + S_{TR} + S_{\text{НУТ}}) \quad (10)$$

при выполнении ограничений на эксплуатационные свойства и прочностные характеристики изделия; оператора, представляющего математическую модель принятия решения при проектировании ТП получения заготовки,

$$\bar{\psi}: M^d \times Tu^d \times Z^d \times P_z^d \times V_z^d \times V_u^d \rightarrow W_1^{\text{доп}}. \quad (11)$$

Здесь S_M — стоимость материала, используемого для изготовления детали; S_{TZ} — трудозатраты; S_{VS} — стоимость вспомогательных материалов; S_{OB} — стоимость обработки (снятие технологических прибылей); K_{SS} — коэффициент, учитывающий срок службы детали; S_{TR} — транспортные расходы на доставку металла от поставщика на склад предприятия; $S_{\text{НУТ}}$ — стоимость риска ущерба здоровью, обусловленного работой в неблагоприятных условиях труда; $W_1^{\text{доп}}$ — множество допустимых решений задачи 1.

Для формализованного описания информационных массивов данных и связей между ними в соответствии со схемой построения ИЛМ (9) предложена двухуровневая модель ТП, включающая в себя две локальные модели, совокупность данных о способах получения и видах заготовок, стойкости материала, условиях эксплуатации, марочника сталей, процессах механической и упрочняющей обработок и т. д., а также производственные правила. Локальные ИЛМ описывают данные и связи между ними, которые используются при определении марки материала и вида упрочняющей обработки. Приведем для примера несколько правил, записанных в формализованном виде: if ((категория значимости k_z = "ответственная") $\wedge$ (класс деталей Vd = "диски") $\wedge$ (группа деталей Gd = "средние")) then (способ получения заготовки p_z = "прокат"); if ((способ приложения нагрузки u_{pr} = "объемные") $\wedge$ (время приложения нагрузки u_{vr} = "статические") $\wedge$ (среда u_{sr} = "атмосфера") $\wedge$ (диапазон температур u_t = $(-80) \div (0)$ °C)) then (вид химико-термической обработки t_u = "Азотирование").

Процедурная модель принятия решений задачи включает ряд эвристических процедур, реализованных в соответствии с обобщенной процедурной моделью, описанной выше (см. рис. 2). Исходя из специфики проектирования ТП машиностроительного производства, к выходным переменным первой категории отнесены способ и метод получения заготовки, марка материала и

вид упрочняющей обработки, ко второй категории — вид заготовки, а к третьей категории — вид вспомогательных материалов для проведения методов получения заготовки.

В связи с тем, что постановка задач 2 и 3 и модели принятия их решений во многом совпадают, ограничимся детальным рассмотрением задачи 3, а именно задачи выбора ТП, оборудования, приспособлений, вспомогательных материалов и режимных параметров определенного ранее вида упрочняющей обработки.

Для конструируемой детали с заданными геометрическими размерами L и массой G , серийностью производства Sp^d , а также выбранным видом упрочняющей обработки $tu^d \in Tu^d$ и маркой материала m^d на множестве $W_3 = Tp^d \times O_u^d \times P_u^d \times V_u^d$ найти такой вариант $w_3^* \in W_3$, для которого критерий оптимальности имеет минимальное значение. Множество W_3 представляет собой декартово произведение подмножеств ТП Tp^d для выбранного вида упрочняющей обработки tu^d , допустимых наборов оборудования O_u^d и приспособлений P_u^d и видов вспомогательных материалов V_u^d при выполнении следующих ограничений: для ТП на температурный режим; для материала детали на глубину слоя химико-термической обработки; для оборудования на габаритные размеры упрочняемой детали; для приспособления на вес упрочняемой детали и оператора, представляющего математическую модель принятия решения при проектировании ТП упрочняющей обработки:

$$\bar{\phi}: Tp^d \times O_u^d \times P_u^d \times V_u^d \rightarrow W_3^{\text{доп}}, \quad (12)$$

где $W_3^{\text{доп}}$ — множество допустимых решений задачи 3.

В большинстве работ используется экономический критерий, однако наряду с экономическими показателями не менее важными являются другие количественные и качественные показатели, такие как оценка варианта $w_3^* \in W_3$ на процент брака при изготовлении машиностроительных деталей, технологичность совокупности процессов их изготовления и надежность используемого оборудования, поэтому в данной работе задача 3 является многокритериальной задачей.

Для каждого варианта решения задачи 1, принадлежащего области "оптимистичных" значений критерия F_1 , решается задача автоматизированного выбора ТП, оборудования, приспособлений, вспомогательных материалов и режимных параметров для упрочняющей обработки. Для решения задачи предложены информационно-логическая и процедурная модели принятия решений, которые во многом аналогичны используемым в задаче 1. При реализации процедурной модели к

выходным переменным первой категории отнесены вид упрочняющей обработки и вид оборудования; ко второй категории — вид приспособления, а к третьей категории — вид вспомогательных материалов для проведения упрочняющей обработки.

Результатом решения задачи является маршрутная карта ТП упрочняющей обработки конструируемой детали.

3. Программная среда и результаты использования системы

При разработке программных модулей за основу взят типовой набор технических средств ПЭВМ *Pentium III*. В качестве базовой системы программирования была выбрана СУБД *Clipper*.

С помощью АИС, реализующей разработанные информационно-логические и процедурные модели, осуществлено решение ряда практических задач проектирования ТП производства изделия из металлов, в частности, форматоров-вулканизаторов (ФВ2-130-940-185/280, ФВ1-500-1800-305, ФВ2-140), редукторов (МР2-315, МПО1М-10, МПО2М-15Щ, МПО2-18) на ЗАО "Завод Тамбовполимермаш"; установки подземного и капитального ремонта скважин УРАН-20.1, предназначенной для проведения технологических и ремонтно-восстановительных работ на нефтяных, газовых и газоконденсатных скважинах, на ОАО "Первомайскхиммаш" и др.

Заключение

На базе системного анализа, методов математического моделирования и теории оптимального управления авторами развит подход к построению АИС поддержки принятия решений для класса технических систем, для которых при получении целевой продукции могут быть использованы различные технологии, виды оборудования, приспособлений и вспомогательных материалов. Этот подход увеличивает уровень интеллектуализации обработки информации в данной области. Разработанная АИС поддержки принятия решений позволяет при реализации основных этапов технологической подготовки производства рассмотреть каждый вариант решения задачи с технологических, экологических и социально-экономических позиций, осуществить комплексную оценку всех вариантов при принятии решений.

Осуществлена постановка задачи одного из этапов жизненного цикла изделия — совместного автоматизированного проектирования технологи-

ческих процессов производства изделий из металлов. В качестве составляющих обобщенного критерия оптимальности использованы: приведенные затраты на реализацию ТП производства изделий из металлов; экономический ущерб, наносимый окружающей среде газовыми выбросами в атмосферный воздух; уровень профессионального риска; технологичность и экологическая безопасность ТП.

Разработана обобщенная ИМЛ ТП рассматриваемого класса технических систем, представляющая собой объединение данных и локальных моделей стадий ТП, которая включает информационные массивы о назначении и качественных показателях целевой продукции, особенностях проектирования и реализации ТП, технической оснащенности предприятия, а также производственные правила, реализующие их взаимосвязь. Осуществлена детализация ИМЛ ТП, лежащих в основе: автоматизированного выбора марки металла, вида и способа получения заготовки в зависимости от характера упрочнения изделий; автоматизированного выбора экономичного и экологически безопасного ТП, оборудования, приспособлений, вспомогательных материалов и режимных параметров для упрочняющей обработки.

Разработана обобщенная процедурная модель принятия решений для проектирования рассматриваемого класса производственных систем, которая при пошаговом конструировании решения с учетом значимости и категоричности выходных переменных для дальнейшего анализа использует либо все их значения, либо только те, которые попадают в окрестность "оптимистичных" значений локального критерия, либо единичные значения — для наименее значимых переменных.

Список литературы

1. Яблочников Е. И. Автоматизированные системы управления технической подготовкой производства на машиностроительных предприятиях // Информационные технологии. 1998. № 8. С. 17—19.
2. Соломенцев Ю. М., Павлов В. В. Моделирование технологической среды машиностроения. М.: Изд. МГТУ Станкин, 1994. 104 с.
3. Капустин Н. М., Дьяконова Н. П., Кузнецов П. М. Автоматизация машиностроения. М.: Высш. шк., 2003. 223 с.
4. Грувер М., Зиммерс Э. САПР и автоматизация производства: Пер. с англ. М.: Мир, 1987. 528 с.
5. Новые принципы построения и организации автоматизированной системы конструкторско-технологической подготовки производства / А. П. Афанасьев, В. И. Галкин, А. А. Лисов и др. // Автоматизация и проектирование. 1999. № 2. С. 18—24.
6. Месарович М., Мако Д., Такахара Я. Теория иерархических многоуровневых систем. М.: Мир, 1973. 344 с.
7. Михалевич В. С., Волкович В. Л. Вычислительные методы исследования и проектирования. М.: Наука, 1982. 288 с.

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И ДИАГНОСТИЧЕСКИЕ СИСТЕМЫ

УДК 004.931:004.891.3

А. П. Горшков, аспирант, ассистент каф.,

Т. П. Грызлова, канд. техн. наук, доц.,

Рыбинская государственная авиационная
технологическая академия им. П. А. Соловьева

Система диагностики состояния сложных технических объектов по характерным последовательностям цифровых сигналов

Представлена архитектура системы технической диагностики с возможностью автоматизированного формирования признаков пространств и синтеза подсистемы принятия решений. Используются признаки на основе характерных последовательностей цифровых сигналов, зарегистрированных при различных состояниях технических объектов. На примерах диагностики нарушений процесса производства полупроводниковых приборов и диагностики состояния подшипников трансмиссии газотурбинных двигателей показана высокая информативность характерных последовательностей и разработанной системы в целом.

Ключевые слова: классификация временных рядов, техническая диагностика, вибродиагностика, автоматизированное формирование признаков, характерные последовательности, распознавание образов.

Введение

Разработка технологий и систем автоматической диагностики состояния сложных технических объектов по цифровым сигналам уже давно является актуальной и очень важной научной и практической задачей. В последнее время наблюдается изменение принципов создания таких систем. Если раньше диагностические признаки и признаковые пространства (пространства образов) формировались специалистом предметной области [1, 2], то в ряде современных работ исследуются методы автоматизации формирования образа [3, 4]. В настоящей статье описаны концепции, положенные в основу технологии автоматизированного формирования образов по классифицированным записям цифровых сигналов. По этой технологии построен программный комплекс "Гиперкуб", позволяющий в короткие сро-

ки выполнить автоматизированный синтез системы диагностики состояния сложных технических объектов. Представлены результаты исследования двух систем диагностики, синтезированных с помощью комплекса "Гиперкуб".

Семейство диагностических признаков на базе характерных последовательностей

В основу синтезируемых систем диагностики состояния технических объектов и технологии их автоматизированного синтеза легло предложенное авторами семейство диагностических признаков по характерным последовательностям. Диагностические признаки формируются на основе анализа появления похожих подпоследовательностей сигналов. Характерная последовательность (ХП) — это элементарная последовательность (ЭП), которая неоднократно встречается в сигнале (кластер "одинаковых" ЭП). Для формализации процедуры сопоставления последовательностей вводится расстояние на множестве ЭП одинаковой длины и максимально допустимый порог расстояния, при котором ЭП считаются одинаковыми. Распространенными расстояниями для сравнения последовательностей являются метрики Чебышева (1) и СКО (2):

$$\rho_{\text{ч}}(x, y) = \max_{i=1 \dots l} |x_i - y_i|; \quad (1)$$

$$\rho_{\text{ско}}(x, y) = \frac{1}{l} \sum_{i=1 \dots l} (x_i - y_i)^2. \quad (2)$$

Наряду с ними используется расстояние на основе вычисления коэффициента корреляции

$$\rho_{\text{кор}}(x, y) = 1 - \frac{1}{n\sigma_x\sigma_y} \sum_{i=1 \dots n} (x_i - m_x)(y_i - m_y). \quad (3)$$

Основная операция, выполняемая над сигналом, — это сопоставление его ЭП с эталоном. При сопоставлении осуществляется перебор ЭП сигнала с заданным в отсчетах шагом и их сравнение с эталоном по заданному порогу. Под частотой ХП понимаем отношение числа ЭП сигнала, расстояние между которыми и эталоном не превосходит порога, к общему числу перебранных ЭП. Частоты ХП образуют эффективное семейство признаков для распознавания цифровых сигналов.

Архитектура системы

Система диагностики — это система с переменной структурой, зависящей от режима работы

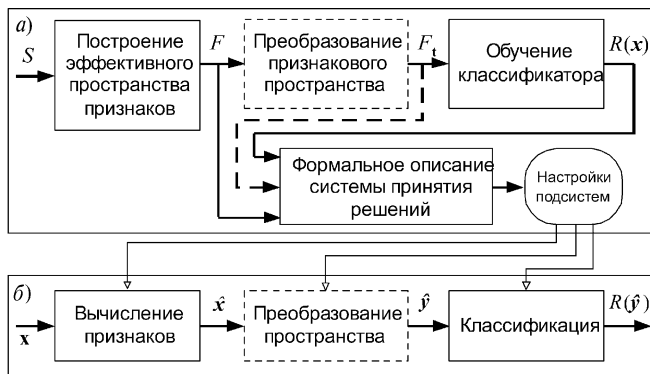


Рис. 1. Структура системы диагностики в режимах:
а — синтеза системы принятия решений; б — принятия решения

(рис. 1). В режиме синтеза системы принятия решения осуществляется поиск эффективного пространства признаков F на основе ХП, преобразование кластеризации этого пространства F_t (если это целесообразно) и формирование решающего правила $R(x)$ одним из математических методов теории распознавания образов (рис. 1, а). В режиме принятия решения система по оцифрованному сигналу x оценивает состояние технического объекта $R(\hat{y})$ (рис. 1, б). Обучающая выборка S при синтезе системы принятия решений — это множество диагностических сигналов с метками о состоянии технического объекта.

Анализ систем диагностики состояния сложных объектов показывает, что наибольшее влияние на эффективность принятия решений оказывает выбор признакового пространства, а не метод классификации [3, 4, 7–9], поэтому в нашей реализации системы в качестве классификатора применяется метод ближайшего соседа. Обучение состоит в запоминании значений признаков для всех сигналов обучающей выборки.

Одним из средств повышения эффективности систем распознавания образов является применение преобразований кластеризации [1]. Это преобразование исходного признакового пространства в пространство меньшей размерности с сохранением кластерных свойств. Блоки преобразования признакового пространства в системе являются необязательными. Если блоки преобразования признакового пространства не используются, то данные передаются без изменения: $F = F_p$, $\hat{x} = \hat{y}$. Наши исследования показывают, что преобразование Карунена—Лоэва часто повышает эффективность пространств на основе ХП.

Для построения признакового пространства формируется множество возможных признаков F_2 и ищется информативное признаковое подпространство F (рис. 2). Алгоритм поиска эффектив-

ной системы признаков может быть построен независимо от семейства признаков.

На основе множества ХП CS , построенного с помощью кластерного анализа последовательностей сигналов обучающей выборки S , формируется множество возможных признаков F_2 (рис. 3). Для проведения кластерного анализа задается размер последовательностей l , расстояние ρ и порог Th_p . Формирование ХП осуществляется для нескольких размеров последовательностей, расстояний и, иногда, для нескольких значений порогов. Таким образом, кластерный анализ последовательностей выполняется неоднократно, а полученные множества ХП объединяются. Длины последовательностей задаются на основе визуальной оценки графиков сигналов.

Далее определяются пороги ХП для повышения информативности множества признаков. Для каждой ХП порог оптимизируется по критерию максимума заданной меры информативности признаков Q_f после чего для полученных признаков удаляются парные корреляции. При этом задается пороговое значение коэффициента корреляции $Th_{кор}$.

При автоматизированном формировании множество возможных признаков имеет большие размеры (больше 500), поэтому выполняется отбор заданного числа N наиболее информативных признаков. Далее в полученном признаковом пространстве F_3 выполняется случайный поиск с адаптацией (метод СПА) [2], для которого необходимо задать меру информативности признаков пространств Q_F (рис. 4). Информативность

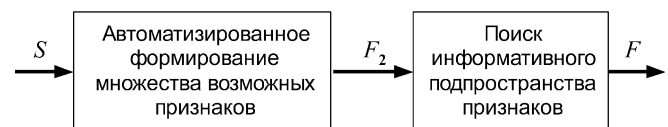


Рис. 2. Структурная схема подсистемы построения эффективного пространства признаков



Рис. 3. Структурная схема алгоритма формирования множества возможных признаков для семейства частотных признаков на основе ХП

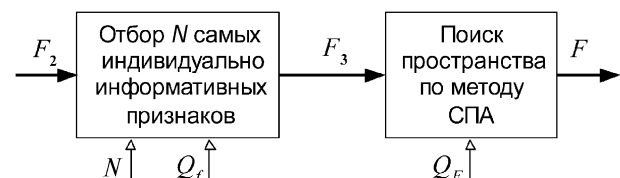


Рис. 4. Структурная схема алгоритма поиска информативного признакового пространства

признакового пространства определяется кластерными свойствами расположения множеств точек, соответствующих разным классам. При использовании в системе преобразования кластеризации оно включается в Q_F и информативность оценивается для преобразованного пространства. В отличие от Q_F , алгоритмы вычисления мер информативности признаков Q_f удается сделать более быстрыми — с линейной оценкой вычисления относительно размера обучающей выборки, что и определяет необходимость их отдельного использования.

Апробация системы

Диагностика нарушений процесса производства полупроводниковых приборов. Для испытания и сопоставления методов классификации временных рядов на специализированных страницах англоязычного Интернета проводится сбор наборов данных для реальных и модельных задач [6]. Одной из таких задач является задача диагностики процесса производства полупроводниковых устройств [4–6]. В процессе напыления регистрируются интенсивности излучения двух длин волн (405 и 520 нм) [4]. Каждый сигнал классифицируется по типу процесса: с нарушениями и без нарушений — как аномальный и нормальный соответственно. Примеры диагностических сигналов показаны на рис. 5. Всего имеется 7164 сигнала, из них 762 — аномальных.

По обучающей выборке 717 сигналов (10 % от исходной выборки), из которых 77 аномальных, построена система диагностики процессов напыления. Найдено двумерное пространство обобщенных признаков, построенное с помощью преобразования Карунена—Лоэва 15 частотных признаков на основе ХП с расстоянием, определяе-

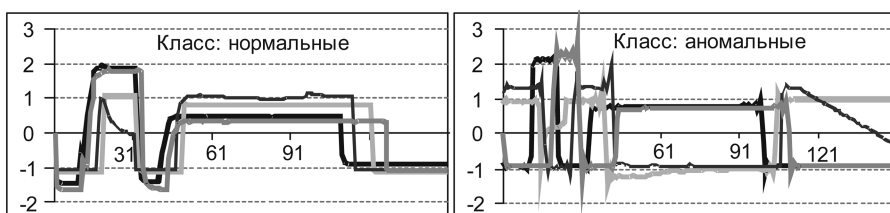


Рис. 5. Примеры диагностических сигналов процесса напыления

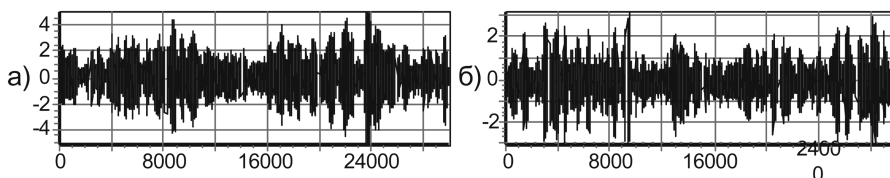


Рис. 6. Примеры вибрационных сигналов для диагностики состояния подшипников трансмиссии ГТД:
а — кондиционного; б — некондиционного

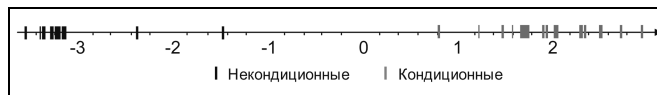


Рис. 7. Распределение образов в одномерном пространстве, сформированном системой диагностики для распознавания состояния подшипников трансмиссии

мым формулой (3). Размеры эталонов ХП от 5 до 40 отсчетов. Оценка вероятности правильного распознавания на тестовой выборке составляет 99,98 %, что выше вероятности правильного распознавания по методу динамической трансформации времени [4, 6] и обобщенного формирования структурных признаков [5] (см. таблицу), причем в работах [4, 6] оценки точности классификации более оптимистичны, поскольку завышен объем обучающей выборки.

Диагностика состояния подшипников трансмиссии газотурбинных двигателей (ГТД). Некоторые эффективные признаки для диагностики состояния подшипников трансмиссии ГТД по оцифрованным сигналам представлены в [7–9]. Из-за небольшого размера обучающей выборки сопоставление эффективности предложенных алгоритмов выполняется на основе анализа разделимости образов кондиционных и некондиционных подшипников в различных признаковых пространствах. Для исследования доступны 36 сигналов, среди которых 14 некондиционных. Примеры кондиционного и некондиционного сигналов приведены на рис. 6.

Пространство на основе статистических характеристик не обеспечивает полной разделимости образов подшипников [7]. Двумерное пространство признаков, полученное методом главных компонент из пространства мощностей шести частотных диапазонов, обеспечивает почти полную разделимость образов [8].

По обучающей выборке построена система диагностики состояния подшипников трансмиссии ГТД. Найден один обобщенный признак, построенный с помощью преобразования Карунена—Лоэва 25 частотных признаков на основе ХП с расстояниями $\rho_{\text{ч}}$ (1), $\rho_{\text{СКО}}$ (2) и $\rho_{\text{кор}}$ (3) (рис. 7). Даже один этот признак позволяет полностью разделить образы подшипников (рис. 7).

Программная реализация системы диагностики "Гиперкуб 1.0" создана на платформе C++ Builder 6 и позволяет в короткие сроки (от одной недели) получить алгоритм автоматической диагностики и приложение или

Сопоставление точности классификации

Метод классификации временных рядов	Критерий (алгоритм оценки)	Точность
На базе ХП	Обучение на 10 % выборки. Тестирование на 90 %	99,98 %
[4]	Разбиение выборки на 10 подмножеств (по 10 %). Рассматривается 10 разбиений выборки на тестовую и обучающую, где тестовая выборка является одним из блоков, а обучающая — все оставшиеся. Используется средняя оценка по этим разбиениям	98,6 %
[5]	Алгоритм не указан	93 %
[6]	Обучение на 14 % выборки. Тестирование на 86 %	99,5 %

библиотеку с соответствующей функциональностью. Для синтеза процедуры диагностики необходима обучающая выборка — множество классифицированных сигналов. Время синтеза зависит от таких параметров задачи, как число классов, размер обучающей выборки и длина реализации сигналов. Программный комплекс "Гиперкуб" позволяет решать задачу повышения качества уже существующей системы диагностики, работающей на основе вычисления признаков. Для этого должна быть предоставлена таблица значений признаков существующей системы диагностики для всех сигналов обучающей выборки. Помимо технической диагностики система "Гиперкуб" позволяет решать и другие задачи классификации временных рядов. Готовятся публикации с успешными результатами по классификации временных рядов на четырех задачах, включающих модельные данные, сигналы из областей распознавания изображений и видеонаблюдения.

Заключение

Система диагностики состояния сложных технических объектов по ХП цифровых сигналов позволяет автоматизировать процесс формирования эффективных признаков пространств и синтеза системы принятия решений. Эффективность автоматической диагностики состояния подшипников трансмиссии ГТД и диагностики процесса напыления в разработанной системе превосходит аналоги.

Система диагностики является гибкой и при наличии классифицированной обучающей выборки позволяет найти эффективные признаковые пространства и синтезировать системы принятия решений для сигналов различного происхождения.

С авторами можно связаться по электронной почте hypercube@iss-lab.ru и ktntpgryzlova@mail.ru.

Список литературы

1. **Ту Дж., Гонсалес Р.** Принципы распознавания образов. М.: Мир, 1978.
2. **Загоруйко Н. Г.** Методы распознавания и их применение. М.: Сов. радио, 1972.
3. **Eads D., Gloer K., Perkins S., Theiler J.** Grammar-guided feature extraction for time series classification // *Neural Information Processing Systems*, 2005.
4. **Olszewski R. T.** Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data // PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.
5. **Xiaopeng Xi, Keogh E., Shelton C., Li Wei & Ratanamahatana C. A.** Fast Time Series Classification Using Numerosity Reduction // *International Conference on Machine Learning*, 2006.
6. **Keogh E.** UCR Time Series Archive: www.cs.ucr.edu/~eamonn/TSDMA/, 2007.
7. **Шепель В. Т., Комаров Б. И., Грызлова Т. П.** Выбор признаков для диагностики технического состояния трансмиссионных подшипников ГТД // *Авиационно-космическая техника и технология*. Харьков: ХАИ, 2005. № 8 (24).
8. **Шишкин В. Н., Комаров Б. И., Гайдай М. С., Шепель В. Т.** Диагностика технического состояния трансмиссионных подшипников ГТД методом структурного анализа их вибросигналов // *Контроль и диагностика*. 2000. № 4.
9. **Горшков А. П., Грызлова Т. П., Комаров Б. И., Шепель В. Т.** Диагностика состояния подшипников трансмиссии газотурбинных двигателей в пространствах статистик характерных последовательностей вибраций // *Авиационно-космическая техника и технология*. Харьков: ХАИ, 2006. № 10/36.

Г. Ф. Малыхина, д-р техн. наук, проф.,
А. В. Меркушева, канд. техн. наук,
ст. инженер-исследователь,
Санкт-Петербургский политехнический университет

Метод анализа независимых компонент для восстановления формы сигналов по их смеси

Рассмотрены задача восстановления формы набора сигналов по данным измерений их линейной смеси и метод решения этой задачи, основанный на анализе независимых компонент. Процедура разделения сигналов из смеси и восстановления их формы базируется на представлении многомерного вероятностного распределения в виде аппроксимации совокупностью распределений независимых сигналов — компонент смеси. При анализе формы сигналов применен критерий, построенный на основе расстояния Кульбака — Лейблера для элементов функционального пространства распределений.

Ключевые слова: сигналы смеси, восстановление, независимые компоненты, критерий Кульбака—Лейблера, матричный градиент, процедура, верификация.

Введение

Разделение сигналов смеси и восстановление их формы необходимо в различных приложениях: анализе речевого сигнала, телекоммуникации, выделения информационного сигнала из нестационарного шума, связанного нелинейным образом с основным сигналом [1—4]. Модель задачи восстановления формы сигналов отражает наблюдение набора из K сигналов $x_1(t), x_2(t), \dots, x_K(t)$, регистрируемых на K датчиках (сенсорах). Каждое наблюдение $x_i(t)$ является линейной комбинацией K' независимых источников $s_1(t), \dots, s_{K'}(t)$. Таким образом, $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, где $\mathbf{x}(t)$ и $\mathbf{s}(t)$ — векторы с компонентами $x_1(t), \dots, x_K(t)$, и $s_1(t), \dots, s_{K'}(t)$, а $\mathbf{A}$ — некоторая матрица размерности $K \times K'$. Требуется восстановить форму сигналов источников $s_1(t), \dots, s_{K'}(t)$, являющихся первичными информационными сигналами, среди которых может быть сигнал сопутствующего фона. При этом предполагается, что первоначально ничего не известно о сигналах-источниках, кроме их взаимной независимости¹⁾.

Использование метода анализа независимых компонент для разделения сигнала и фона применяется обычно для двухкомпонентной смеси в

виде информационного сигнала и сопутствующего шума. Кроме этого метода для подавления шума и выделения сигнала могут эффективно использоваться адаптивные системы с нелинейными фильтрами различной структуры [5—8] и системы, включающие нейронные сети [9]. Поэтому при математическом описании метода анализа независимых компонент (МАНК) для разделения сигналов в составе многоканально измеряемой линейной смеси предполагается незначительный уровень шума в информационных сигналах — компонентах смеси, т. е. такой уровень, которым можно практически пренебречь. В то же время, если число сенсоров значительно больше числа информационных сигналов, то посредством МАНК возможно восстановление формы почти чистых сигналов и получение отдельных компонент шума.

С учетом отмеченных положений для определенности далее МАНК будет построен для случая $K = K'$. Отсутствующие сигналы-источники на выходе системы восстановления формы могут замещаться сигналами шума или их суммой.

В основе МАНК для восстановления формы сигналов лежат идеи Тонга, Иноя и Лиу [1], Комона (Comon) [10], Пхама, Гарата и Джуттен (Pham, Garat, Jutten) [11]. Исходным является не модель сигналов, а распределение вероятности $P_{\mathbf{x}}$ (на множестве K -мерных распределений P_K) нескольких случайных векторов $\mathbf{x}(t)$. Целью является получение такой $(K \times K)$ -матрицы $\mathbf{B}$, чтобы компоненты вектора $\mathbf{B}\mathbf{x}$ были (возможно более) независимыми²⁾. При такой вероятностной трактовке анализа независимых компонент неизвестная величина $P_{\mathbf{x}}$ оценивается по наблюдаемой последовательности векторов $\mathbf{x}(t)$ ³⁾. Если для рассматриваемой задачи МАНК выполняется соотношение $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ при независимых компонентах у $\mathbf{s}(t)$, то $\mathbf{B}(t) = \mathbf{A}^{-1}$ является решением задачи анализа независимых компонент. При наличии среди компонент вектора $\mathbf{s}(t)$ не более одной с Гауссовым распределением полученное решение единственно с точностью до общего масштаба компонент и перестановки их порядка [10].

Таким образом, решение задачи анализа независимых компонент (АНК) может использоваться для разделения сигналов источника, и наоборот, процедура разделения сигналов источника служит для решения задачи АНК. Однако эти две задачи

²⁾ Сокращенные обозначения $(K \times K)$ -матрица или K -вектор, используются для указания размерности матрицы или вектора. Независимость понимается в обычном смысле: среднее значение (математическое ожидание) любой линейной комбинации компонент вектора $\mathbf{B}\mathbf{x}$ равно нулю или (возможно более) близко к нулю.

³⁾ При этом для $\mathbf{x}(t)$ неявно предполагается стационарность и эргодичность (независимость распределения от времени) на интервале определения анализируемых сигналов — компонент вектора $\mathbf{s}(t)$.

¹⁾ Интуитивно ясно, что нельзя восстановить форму более, чем K сигналов источника, если имеется только K каналов наблюдения. Поэтому далее предполагается (без специального упоминания), что $K' \leq K$.

различаются по используемым предположениям и по целям:

- при разделении сигналов источника (РСИ) требуется восстановить эти сигналы наилучшим образом, предполагая их существование⁴⁾;
- при АНК процедура выполнения менее результативна — источника может просто не быть.

Наиболее важной для АНК является робастность: анализ должен давать почти независимые компоненты для широкого диапазона распределений. Тогда он может использоваться для разделения сигналов в более сложных ситуациях (например, при числе сигналов источника больше, чем число сенсоров) и сможет выделять основные сигналы при наличии примеси слабых сигналов. В то же время процедура РСИ ориентирована на идеальную модель, встречает затруднения, если ее предположения не выполнены, и в отдельных случаях может приводить к неоднозначному решению. Получение робастного метода АНК, который устраняет недостатки традиционных процедур РСИ и АНК, основано на использовании меры Кульбака—Лейблера — статистического критерия, введенного для оценки различия распределений в многомерном вероятностном пространстве [11].

Метод анализа независимых компонент на основе критерия Кульбака—Лейблера

В методе АНК для восстановления формы сигналов по их смеси рассматриваются вероятностные распределения P_x ($P_x \in R_K$) случайного вектора $\mathbf{x}(t)$ и определяется такая $(K \times K)$ -матрица $\mathbf{B}$, чтобы компоненты вектора $\mathbf{Bx}$ были независимы. Для меры близости к независимости используется критерий Кульбака—Лейблера (ККЛ). Структура этого критерия такова, что он оценивает "расстояние" (или меру близости) между двумя многомерными распределениями с одинаковыми наборами переменных. Мера близости компонент $\mathbf{Bx}$ к независимости оценивается сравнением распределения вероятности $P_{\mathbf{Bx}}$ с распределением $P_{\mathbf{Bx}}^{\text{НК}} = P_{(\mathbf{Bx})_1} \times \dots \times P_{(\mathbf{Bx})_K}$ — для совокупности независимых компонент, где $P_{(\mathbf{Bx})_i}$ — распределение i -й компоненты вектор-сигнала $\mathbf{Bx}$. Такое сравнение осуществляется с помощью ККЛ, определяющего расстояние между двумя распределениями P и Q в виде $\text{ККЛ}(P, Q) = -\int \ln \frac{dQ}{dP} dP$ ⁵⁾.

⁴⁾ Эффективностью выполнения процедуры может служить оценка, насколько близко к независимости восстановленные компоненты $s(t)$, но "близость к независимости" — размытое понятие, для нее нет общепринятой меры.

Метод АНК для восстановления формы сигналов источника состоит в определении такой матрицы $\mathbf{B}$, которая минимизирует ККЛ ($P_{\mathbf{Bx}}, P_{\mathbf{Bx}}^{\text{НК}}$). В связи с этим удобно обозначить $C(\mathbf{B}) = \text{ККЛ}(P_{\mathbf{Bx}}, P_{\mathbf{Bx}}^{\text{НК}})$

- **Структура критерия Кульбака—Лейблера для восстановления сигналов смеси методом АНК.** Если наблюдаемый вектор-сигнал $\mathbf{x}$ имеет плотность вероятности f_x , то плотность для $\mathbf{Bx}$ определяется выражением

$$f_{\mathbf{Bx}}(s) = f_{\mathbf{Bx}}(s_1, \dots, s_K) = \frac{f_x(\mathbf{B}^{(-1)}s)}{|\Delta \mathbf{B}|}, \quad (1)$$

где $\Delta \mathbf{B}$ — определитель $\mathbf{B}$.

Плотность вероятности $P_{\mathbf{Bx}_i}$ (маргинальная плотность для $(\mathbf{Bx})_i$) получается интегрированием $f_{\mathbf{Bx}}(s_1, \dots, s_K)$ относительно всех переменных, кроме s_i . С учетом этого, $C(\mathbf{B}) = \text{ККЛ}(P_{\mathbf{Bx}}, P_{\mathbf{Bx}}^{\text{НК}})$ выражается соотношением

$$\begin{aligned} C(\mathbf{B}) = & - \int_{R_K} \ln \frac{\prod_{i=1}^K f_{(\mathbf{Bx})_i}(s_i)}{f_{\mathbf{Bx}}(s_1, \dots, s_K)} f_{\mathbf{Bx}}(s_1, \dots, s_K) ds_1, \dots, \\ & ds_K = \int_{R_K} |\ln f_{\mathbf{Bx}}(s)| f_{\mathbf{Bx}}(s) ds - \\ & - \sum_{i=1}^K \int_{R_K} |\ln f_{(\mathbf{Bx})_i}(s_i)| f_{(\mathbf{Bx})_i}(s_i) ds_i. \end{aligned} \quad (2)$$

Последнее слагаемое в (2) является суммой энтропий, вычисленных по маргинальным плотностям распределения $P_{\mathbf{Bx}_i}$. Первое слагаемое в (2) в соответствии с соотношением (1) представляется в виде $\int_{R_K} |\ln f_x(x)| f_x(x) dx - \ln \Delta \mathbf{B}$. Поэтому $C(\mathbf{B})$ представим в форме

$$\begin{aligned} C(\mathbf{B}) = & \int_{R_K} |\ln f_x(x)| f_x(x) dx - \ln \Delta \mathbf{B} - \\ & - \sum_{i=1}^K \int_{R_K} |\ln f_{(\mathbf{Bx})_i}(s_i)| f_{(\mathbf{Bx})_i}(s_i) ds_i, \end{aligned} \quad (3)$$

причем первое слагаемое в (3) не влияет на процесс минимизации при определении матрицы $\mathbf{B}$, и его можно опустить⁶⁾.

- **Процедура получения матрицы $\mathbf{B}$ (и соответствующего линейного преобразования) для восстановления сигналов из смеси.** Определение

⁵⁾ Предполагается, что dQ/dP хорошо определено, иначе считается, что $\text{ККЛ} = \infty$. Кроме того, ККЛ является не отрицательным и он равен нулю только при равенстве распределений ($\text{ККЛ}(P, Q) = 0 \Leftrightarrow P = Q$).

⁶⁾ В таком виде величина $C(\mathbf{B})$, называемая контрастом [10], остается неизменной при преобразовании $\mathbf{B}$ с помощью матрицы перестановок или масштабирования. В связи с этим восстановление формы вектора сигналов источника с помощью МАНК возможно только с точностью до нумерации его компонент вектора и масштаба.

матрицы $\mathbf{B}$ путем минимизацией $C(\mathbf{B})$ (в усеченном виде) проводится градиентным методом. Этот метод целесообразно применить с использованием "относительного градиента" (∇^*), который введен в [12, 13] и связан с обычным градиентом (∇) соотношением⁷⁾:

$$\nabla^* C(\mathbf{B}) = \nabla C(\mathbf{B}) \mathbf{B}^T. \quad (4)$$

Для представления относительного градиента (ОГ) $\nabla^* C(\mathbf{B})$ в развернутой форме требуется введение двух промежуточных соотношений, которые определяют:

- среднее значение (математическое ожидание E) функции $g(\mathbf{x})$ при плотности $f_{\mathbf{B}\mathbf{x}}(\mathbf{s})$ с фиксированной компонентой $(\mathbf{B}\mathbf{x})_i = s_i$ в наборе ее аргументов:

$$E_{g(\mathbf{x})|(\mathbf{B}\mathbf{x})_i}^f(s_i) = \int_{R^{K-1}} g(\mathbf{B}^{-1}\mathbf{s}) f_{\mathbf{B}\mathbf{x}} ds_1 \dots ds_{i-1} ds_{i+1} \dots ds_N, \mathbf{s} = (s_1, \dots, s_N)^T \quad (5)$$

(это умноженное на $f_{(\mathbf{B}\mathbf{x})_i}(s_i)$ условное математическое ожидание функции $g(\mathbf{x})$ при фиксированном значении $(\mathbf{B}\mathbf{x})_i = s_i$);

- значение производной $\psi_{(\mathbf{B}\mathbf{x})_i}$ от $-\ln f_{(\mathbf{B}\mathbf{x})_i}$.

Введенные соотношения позволяют представить относительный градиент выражением [14]

$$\begin{aligned} \nabla^* C(\mathbf{B}) &= \begin{bmatrix} \int_{R_K} \psi_{(\mathbf{B}\mathbf{x})_1}(s) E_{X^T \mathbf{B}^T | (\mathbf{B}\mathbf{x})_1}^f(s) ds \\ \vdots \\ \int_{R_K} \psi_{(\mathbf{B}\mathbf{x})_K}(s) E_{X^T \mathbf{B}^T | (\mathbf{B}\mathbf{x})_K}^f(s) ds \end{bmatrix} - \mathbf{I} = \\ &= \int_{R_K} \begin{bmatrix} \psi_{(\mathbf{B}\mathbf{x})_1} \cdot [(\mathbf{B}\mathbf{x})_1] \\ \vdots \\ \psi_{(\mathbf{B}\mathbf{x})_K} \cdot [(\mathbf{B}\mathbf{x})_K] \end{bmatrix} (\mathbf{B}\mathbf{x})^T f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} - \mathbf{I}. \quad (6) \end{aligned}$$

В (6) диагональные элементы $\nabla^* C(\mathbf{B})$ равны нулю, поэтому для определения $\mathbf{B}$ (с точностью до масштаба) получается $K(K-1)$ уравнений, являющихся следствием условия минимизации ОГ $\nabla^* C(\mathbf{B})$:

$$\int_{R_K} \psi_{(\mathbf{B}\mathbf{x})_i} [(\mathbf{B}\mathbf{x})_i] (\mathbf{B}\mathbf{x})_j f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = 0, i \neq j = 1, \dots, K. \quad (7)$$

- **Верификация метода АНК для восстановления сигналов из линейной смеси.** Для проверки метода и алгоритма АНК моделировалось два источника сигналов с различными функциями распределения, поступающими на два сенсора. Данные $\mathbf{x}(t) = [x_1(t), x_2(t)]$ ($t = 1, \dots, 300$) смеси реализованы смешиванием независимых сигналов источника ($s_1(t)$ и $s_2(t)$) посредством ли-

нейного преобразования с матрицей

$$\mathbf{A} = \begin{bmatrix} 1 & 0,6 \\ 0,6 & 1 \end{bmatrix}. \text{ При этом для распределений от-}$$

счетов сигналов источника использовались равномерное, нормальное и составное нормальное распределения (с изменением некоторых параметров) и чередование сочетаний этих распределений в сигналах $s_1(t)$ и $s_2(t)$. Оценка

матрицы $\hat{\mathbf{B}}$, соответствующей градиентному методу минимизации ККЛ, получена с помощью итерации соотношений (7). Качество восстановления сигналов оценивалось по относительному значению среднеквадратичного "загрязнения" восстановленного сигнала (сравнительно с его действительным значением). Эта величина в зависимости от сочетания вариантов распределения сигналов источника находится в пределах от 3 до 10 %, а в наиболее неблагоприятном случае близких к идентичным двух распределений сигналов источника может достигать 14—20 %.

Заключение

Рассмотрен новый метод для разделения сигналов источника по данным измерения их смеси на группе датчиков (сенсоров). Метод основан на анализе независимых компонент — сигналов, составляющих анализируемую смесь.

- Метод имеет свойство адаптации к различным видам вероятностного распределения сигналов источника и определенную степень робастности — устойчивости к небольшому уровню зашумленности анализируемых сигналов.
- При разделении и анализе сигналов источника с небольшим уровнем шума требуется, чтобы число сенсоров превышало общее число входных сигналов источника.
- Алгоритм разделения сигналов построен на основе статистического критерия Кульбака—Лейблера (ККЛ), который определяет меру близости (расстояние) двух плотностей вероятностного распределения векторной переменной произвольной размерности.
- ККЛ построен таким образом, чтобы получить меру близости распределения разделяемых сигналов смеси к такому же распределению при условии независимости этих сигналов. В этом состоит основная концепция разделения сигналов методом анализа независимых компонент.
- ККЛ используется для получения матрицы, реализующей восстановление отдельных сигналов источника, поступающих в составе смесей на отдельные сенсоры измерительной системы. Линейное преобразование сигналов сме-

⁷⁾ Относительный градиент $\nabla^* C(\mathbf{B})$ фактически является матрицей как произведение K -вектора (матрицы столбца) $\nabla C(\mathbf{B})$ на вектор-строку $\mathbf{B}^T$ того же размера.

си на сенсорах измерительной системы и соответствующая матрица (**B**) этого преобразования получается минимизацией ККЛ.

- ♦ Минимизация ККЛ осуществляется градиентным методом в модифицированном виде с использованием относительного градиента (в форме Кардосо—Лакельда и Амари).
- ♦ Практическое использование алгоритма показывает достаточно быструю его сходимость: как правило, бывает достаточно 5—10 итераций.
- ♦ Восстановление сигналов получается с точностью до масштаба и изменения порядка нумерации компонент. Это связано с инвариантностью, зависящей от матрицы **B**, части ККЛ к преобразованиям с матрицей перестановок и с диагональной матрицей масштабирования. В приложениях такая степень неопределенности восстановленных сигналов обычно не имеет принципиального значения.
- ♦ Метод разделения сигналов проверен на примере моделированных смесей двух сигналов с различными (и не совпадающими для них) плотностями распределения: нормального, равномерного и ряда вариантов линейных смесей распределений этого типа.

Список литературы

1. Tong L., Inouye Y., Liu R. Waveform preserving blind estimation of multiple independent sources // IEEE Transaction of signal processing. 1993. Vol. 41. N 7. P. 2461—2470.

2. Меркушева А. В. Фильтрация нестационарного сигнала (речи) в вейвлет-области с адаптацией к виду и динамике шума // Научное приборостроение. 2003. Т. 13. № 2. С. 73—87.

3. Malychina G. F., Mercusheva A. V. Metrological aspects of non-stationary signal transformation for dynamic spectrum analysis // 10th IMEKO TC7 International Symposium on Advances of Measurement Science (June 30 — July 2, 2004, Saint-Petersburg, Russia). Vol. 1. P. 212—216.

4. Малыхина Г. Ф., Меркушева А. В. Концепции информации и энтропии в нейросетевых методах анализа сигналов в ИИС // Информационные технологии. 2005. № 12. С. 26—36.

5. Billings S. A., Alturki F. A. Performance monitoring in non-linear adaptive noise cancellation // Journal of Sound and Vibration. 1992. V. 157. N 1. P. 161—175.

6. Xie L. Robust Kalman Filtering for uncertain Systems // System and Control Letters. 1994. Vol. 22. P. 123—133.

7. Billings S. A. El-Hitmy Adaptive noise cancellation for class of nonlinear IIR filters // Journal of Sound Vibration. 1990. Vol. 140. N 3. P. 457—473.

8. Mathews V. J. Adaptive polynomial filters // IEEE Signal Processing Magazine. 1991. Vol. 8. N 3. P. 10—26.

9. Меркушева А. В. Нейросетевой метод цифровой обработки нестационарного сигнала (речи) в области вейвлет-отображения // Цифровая обработка сигналов. 2006. № 1. С. 18—30.

10. Comon P. Independent components analysis, a new concept // Signal processing. 1994. Vol. 36. Vol. 3. P. 287—314.

11. Кульбак С. Теория информации и статистика. М.: Наука, 1967. 407 с.

12. Cardoso J. F., Laheld B. Equivariant adaptive source separation // IEEE Transaction of signal processing. 1996. Vol. 44. N 10. P. 3017—3030.

13. Amari S. Natural gradient works efficiently in learning // Neural Computation. 1998. Vol. 10. P. 251—276.

14. Pham D. T. Blind separation of instantaneous mixture of sources via independent component analysis // IEEE Transaction of signal processing. 1996. Vol. 44. N 11. P. 2768—2782.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 004.912

Обзор, постановка задачи

Н. Е. Балакирев, канд. техн. наук, проф.,
Б. Б. Гирин, аспирант,
"МАТИ"-РГТУ имени К. Э. Циолковского
bgl234567@list.ru

Основные концепции языка описания алгоритмов анализа естественно-языкового текста

Предложен специализированный язык, предназначенный для описания алгоритмов синтаксического анализа, и показано его место в системе анализа естественно-языковых текстов.

Ключевые слова: синтаксис, естественные языки, искусственные языки, данные, троичная логика.

В наиболее общем случае под анализом текста понимается процесс получения некоторой информации из текста на естественном языке. Одним из направлений компьютерной лингвистики является автоматизация анализа текста. Автоматизированный анализ текста применяется в системах, решающих различные задачи: машинный перевод, естественно-языковое общение с компьютером, поиск в текстах, извлечение знаний из текстов, анализ и синтез речи.

При автоматизированном анализе текстов на естественном языке, как правило, исходят из предположения, что каждый текст содержит несколько формальных структур, разделенных на уровни [1]. Первая из них — это *поверхностная синтаксическая структура*, в которой каждое предложение рассматривается отдельно от других. Она определяется синтаксическими ролями слов

в предложении (подлежащее, сказуемое, дополнение и т. д.) и синтаксическими связями между словами.

Вторая — это *глубинная синтаксическая структура*. Она строится в терминах типа объект, субъект, инструмент и т. д. Выделение этой структуры связано с тем, что разные языки, имея характерные синтаксические особенности, могут передавать весь спектр взаимодействий и взаимоотношений между объектами и явлениями реального мира.

Далее следует *семантическая структура*, в которой можно выделить *поверхностный и глубокий семантические уровни*. На поверхностном уровне констатируется смысл высказывания, и этот уровень тесно связан с глубокой синтаксической структурой. На глубоком семантическом уровне смысл высказывания может связываться с целым классом однотипных ситуаций, могут делаться выводы, например, о причинах события, о котором идет речь и т. д.

Отдельно можно выделить *прагматическую структуру текста*. По сути, эта структура описывает действия, к которым побуждает текст, привязывает текст к окружающей действительности. Тогда собственно анализ текста можно представить как последовательность следующих этапов:

- графематический анализ;
- морфологический анализ;
- поверхностный синтаксический анализ;
- глубокий синтаксический анализ;
- поверхностный семантический анализ;
- глубокий семантический анализ;
- прагматический анализ.

Существуют и более тонкие способы деления на этапы (см. например, [2]), но приведенный является наиболее распространенным и универсальным.

Одним из этапов автоматизированного анализа текста является поверхностный синтаксический анализ. Именно этот этап анализа рассматривается в данной статье. Задачей этапа поверхностного синтаксического анализа (далее будем называть его просто синтаксическим) является построение поверхностной синтаксической структуры текста, которая, по сути, представлена членами предложения и их связями.

В любой универсальной системе анализа текста поверхностный синтаксический анализ является ключевым этапом [3]. Синтаксический анализ в качестве этапа обязательно присутствует при решении большинства задач компьютерной лингвистики: машинный перевод, извлечение знаний из текста, создание вопросно-ответных систем, голосовые технологии (анализ, синтез речи), проверка правильности пунктуации и т. д. В последнее время синтаксический анализ используется даже в системах, основанных на ста-

тистике языка [4], по крайней мере, для разрешения омонимии частей речи.

На практике синтаксический анализ, как правило, представляет собой переход от морфологических характеристик слов и расположения знаков препинания, входящих в предложение, к одной из трех структур [5]: дерево подчинения, система составляющих, синтаксические группы. Синтаксический анализ можно разделить на несколько этапов, следующий метод деления на этапы предложен в [6]:

- предсинтаксис:
 - ♦ постморфология (морфосинтаксис);
 - ♦ снятие омонимии частей речи;
- предсегментация — построение необходимых для сегментации проективных отрезков атрибутивных именных, предложных групп, сложного сказуемого и т. д.;
- сегментация — построение сегментов;
- внутрисегментный анализ — построение связей слов в сегментах;
- межсегментный анализ — построение связей сегментов.

Такое деление является наиболее подробным и, по-видимому, наиболее полным.

В процессе разработки алгоритмов синтаксического анализа возникает проблема необходимости проверки различных пробных вариантов алгоритмов и порядка их применения [7]. То есть необходимо средство, позволяющее разработчику видеть и изменять алгоритмы анализа текста без значительного изменения системы. Такая подсистема (назовем ее инструментальной средой описания алгоритмов синтаксического анализа) должна обеспечивать минимизацию трудоемкости написания алгоритмов синтаксического анализа, позволять изменять алгоритм анализа без перестройки всей системы, наглядно представлять алгоритм в целом и его составные части.

В настоящее время первая причина необходимости создания инструментальных средств для систем анализа текста — технологическая: упрощается процесс создания и отладки алгоритма. Вторая причина состоит в том, что в зависимости от структуры исходного текста (например, исходный текст — словарная статья) или от той информации, которую требуется извлечь (например, контекст омонима), может потребоваться различная "глубина" анализа текста и, следовательно, различными будут алгоритмы анализа.

В [6, 7] описана реализация подобной инструментальной среды для экспериментов с алгоритмами поверхностно-синтаксического анализа. Она представляет собой объектную систему, являющуюся расширением модели многофункциональных словарей. В такой модели лингвистические единицы, информация о них и отношения

между ними представлены в виде объектов, объединенных в динамическую многоссылочную структуру. Применяемая в моделировании словарей объектная библиотека расширена объектами для представления правил поверхностно-синтаксического анализа. Интерфейс системы реализован с помощью документов MSVisio®, правила анализа представлены структурами, сходными с блок-схемами.

В [8] правила синтаксического анализа записываются на определенном подязыке языка C++. При этом основными объектами являются: уже построенный набор синтаксических групп, к которому нужно добавить новую группу и характеристики отдельных слов — омонимов входного отрезка текста.

Существуют и другие подходы к созданию инструментальных систем, примерами могут служить использование грамматик, относящихся к классу мягко контекстно-зависимых формализмов [9], словарь предложных и предложно-глагольных фреймов [10].

При разработке системы анализа текста, предназначенной для извлечения знаний из неструктурированных источников [11], авторы пришли к необходимости создания удобных средств для отладки алгоритмов анализа текста. Особенностью созданной инструментальной системы было введение в ее состав специализированного языка описания алгоритмов синтаксического анализа текста.

Такой язык должен удовлетворять следующим требованиям: быть максимально простым, ориентированным на решение только задач анализа естественно-языкового (русского) текста и, кроме того, обеспечивать открытый доступ пользователя к алгоритмам анализа текста. Исходя из этих требований создавался набор средств языка: встроенные функции реализуют операции, используемые при автоматизированном анализе текста; введение троичной логики позволяет работать с присущей языку неоднозначностью; встроенные в язык морфологические средства обеспечивают алгоритм требуемыми исходными данными.

Отдельно следует указать и на то, что ввод в систему средства отладки алгоритмов анализа и существование специализированного для описания алгоритмов, и библиотеки алгоритмов анализа текстов делают систему откры-

той и позволяют пользователям компактно и наглядно представлять используемые в системе алгоритмы и модифицировать их. Система делится на две части: первая закрытая и статичная, вторая — открытая, структура которой видна пользователю и может быть им изменена. Вторая часть при этом реализует именно правила синтаксиса, присущие естественному языку, в нашем случае — русскому.

Место языка в системе анализа текста. Входные и выходные данные

Систему анализа текста можно представить как последовательность переходов от исходного текста к его формальным структурам — синтаксической, семантической и прагматической. Включение транслятора, как видно на рисунке, не меняет общую последовательность анализа текста.

С точки зрения преобразования информации имеем следующую цепочку преобразований: $T0 \rightarrow T1 \rightarrow T2 \rightarrow T3 \rightarrow T4 \rightarrow \dots$, где $T0$ — исходный текст (в некотором формате); $T1$ — текст, подготовленный для анализа, в виде последовательности символов (буквы, цифры, пробелы, знаки препинания); $T2$ — текст после графематического анализа; $T3$ — текст после морфологического анализа; $T4$ — текст после синтаксического анализа. Далее могут следовать результаты других этапов анализа (семантического, прагматического).

В записи $T2 = T1 + W1 + S1$ и далее знак "+" обозначает объединение элементов, где $W1$ — массив "слов" текста; $S1$ — массив предложений



Схема системы анализа текста с использованием языка описания алгоритмов анализа

текста. Под "словом" в $W1$ понимаются: группы букв (кириллица, латиница); группы цифр; знаки препинания; смешанные группы; пробелы. Предложение в $S1$ представлено номером первого "слова" и числом "слов" в нем.

$T3 = T1 + W1 + M1 + S1$. На морфологический анализ подается, по сути, только составляющая $W1$, на выходе анализа получаем $M1$ — морфологическое представление. $M1$ — массив морфологических записей, каждая из которых представляет собой начальную форму некоторого "слова" из $W1$ и грамматические характеристики этого слова. "Слову" из $W1$ могут соответствовать одна или несколько записей, если "слово" относится к группе "слов", записанных русскими буквами, и действительно существует в русском языке; в остальных случаях либо "слово" не соответствует ни одной записи, либо существует одна запись, где начальной формой принимается та, в которой "слово" встретилось в тексте.

$T4 = T1 + W2 + M2 + S1 + G1$. $W2$ — массив слов, в котором остались только слова, имеющие соответствующие им записи в $M1$, и знаки препинания; $M2$ — массив морфологических записей с исключенными из него при разрешении омоформ и омографов записями; $G1$ — массив синтаксических групп. Синтаксическую группу можно рассматривать как подмножество множества слов предложения, при этом каждая из таких групп будет иметь слово (корень), от которого зависят в синтаксическом смысле остальные слова группы, а сами группы могут зависеть друг от друга (группа может быть родителем для одной или нескольких других групп).

Для определения синтаксической структуры предложения можно использовать несколько "маркеров": расположение знаков препинания, морфологическая информация о словах, грамматическая согласованность слов, дополнительные семантические данные [13]. Поэтому входной информацией системы синтаксического анализа являются морфологические описания слов входного предложения и информация о наличии и местоположении в предложении знаков препинания, а выходной информацией — набор синтаксических групп, связей между ними и внутри них.

Концепции языка

Первоначально средство описания и отладки алгоритмов анализа виделось авторам состоящим из двух частей: языка описания правил синтаксического анализа и языка комбинирования этих правил [12]. В терминах этой системы был построен алгоритм извлечения из статьи толкового словаря родового понятия для определяемого слова. Но для решения других задач, в частности, сег-

ментации предложения, такая структура оказалась малопригодной. В предлагаемой системе был определен единый язык, в котором синтаксическое правило, по сути, стало подпрограммой.

В основу разрабатываемого языка были положены три концепции.

1. *Простота и специализация.* Язык должен быть, с одной стороны, максимально прост, с другой — максимально специализирован для решения задач анализа текста. Работающий с системой пользователь должен уделять максимальное внимание собственно алгоритму анализа, а не проблемам языка.

2. *Троичная логика.* В языке используется логический тип данных с тремя состояниями: истина, ложь, неопределенность. Необходимость такого типа обусловлена в первую очередь свойственной естественному языку неоднозначностью.

3. *Расширяемость и обозримость.* Данная концепция возникла в связи с тем, что число правил, написанных на ранее созданном "языке описания правил", резко возрастало, а сами они были слишком мелкими, следовательно, общий алгоритм анализа разрастался и становился необозримым. Решением стало создание единого языка и предоставление возможности внесения условных переходов и/или циклов внутрь правила.

Таким образом, был разработан новый язык описания правил обработки текста, наиболее важными свойствами которого стали: использование троичной логики и ориентированность на решение задач анализа естественно-языкового текста.

Простота и специализация. Исходя из первой концепции было принято решение о введении в язык четырех типов данных, которых достаточно для решения задач данного типа: `int` — целые неотрицательные числа, `bool3` — логическая переменная с тремя состояниями, `string` — текстовая строка, `ling` — строка лингвистического описания, специализированный тип для обработки естественно-языкового текста.

После анализа доступных алгоритмов синтаксического анализа [3, 8, 13] и ранее созданного алгоритма синтаксического анализа текста авторами был выделен набор атомарных операций, на основе которого и были определены функции, вводимые в разрабатываемый язык. Их можно разделить на четыре группы:

- **Функции, работающие с исходным предложением:** поиск по морфологии, поиск по слову, поиск по обобщенной морфологии, проверка морфологии текущего или заданного слова в предложении, перемещение по словам предложения, проверка грамматической сочетаемости слов, удаление морфологических записей, ставших неактуальными, и корректировка

обобщенной морфологии, работа со списками (составные вводные слова, предлоги, союзы).

- **Управляющие функции:** два вида условного перехода, безусловный переход, цикл типа while, выход из функции с возвращением или без возвращения значения. Сюда же, условно, можно отнести и функции, реализующие логическое сложение и умножение (and, or), а также операции "+" и "-" для чисел; "+" для обоих видов строк.
- **Функции вывода и ввода:** функции вывода — аналог print cout и т. п., вывод в специализированный файл и сохранение в оперативной памяти синтаксической структуры предложения; функции ввода — загрузка предложения и переход к обобщенной морфологии, открытие файла. К этой же группе можно отнести функцию остановки, которая используется для остановки выполнения программы и выводит на экран текущие значения переменных.
- **Функции формирования синтаксических групп:** создание группы, добавление слова в группу, изменение типа группы, удаление группы, объединение групп, задание главного слова в группе.

Таким образом, в настоящий момент в языке реализованы четыре типа данных и четыре группы функций, достаточных для описания алгоритмов постморфологии, сегментации, синтаксического анализа.

Троичная логика. Использование троичной логики является основной особенностью языка и основным отличием языка от других систем. В качестве логического типа в языке принят тип с тремя возможными значениями: "ложь", "истина" и "неопределенность". Это позволяет средствами среды обрабатывать присущую естественному языку неоднозначность, которая на уровне синтаксиса выражена, в первую очередь, омонимией частей речи. В случае синтаксического анализа принимаемые логической переменной состояния, как правило, идентифицируют следующие ситуации: "истина" — слово точно соответствует поисковому запросу или заданному условию; "ложь" — слово однозначно не соответствует поисковому запросу или заданному условию; "неопределенность" — слово, возможно, соответствует поисковому запросу или заданному условию.

Следовательно, в языке должны быть два оператора условного перехода — в первом переход происходит, если значение проверяемого условия "истина", во втором — если значение "истина" или "неопределенность". Таким образом, во втором случае можно провести дополнительные проверки для снятия неоднозначности.

Тогда систему анализа можно описать следующим образом: если в некоторой точке программы

t_1 отсутствует достаточная информация для принятия решения, запускается подпрограмма получения дополнительных данных. Если после выполнения подпрограммы в точке t_2 однозначного решения не найдено, то может быть запущена еще одна подпрограмма; если решение найдено, то выполнение программы продолжается по одному из путей в зависимости от значения условия, которое может быть либо "истиной", либо "ложью".

Если подпрограмм, способных разрешить неоднозначность, не остается, то либо существует *реальная* неоднозначность внутри предложения, либо система недостаточно полна для разрешения неоднозначности и ее необходимо дополнить некоторыми подпрограммами. Кроме того, при выводе результатов анализа они могут быть помечены особым образом.

Расширяемость и обозримость. Как было сказано выше, синтаксическое правило записывается как подпрограмма на специализированном языке. Каждое правило представляется в виде, позволяющем пользователю видеть наглядное и компактное представление описания правила. Кроме того, при такой форме записи можно компактно и наглядно представить и весь алгоритм анализа.

Подпрограмма на языке описания алгоритмов анализа по своему синтаксису более всего походит на функцию языка C/C++.

Заключение

С использованием описываемого языка был разработан алгоритм обработки статьи толкового словаря (получение родового понятия и видовых отличий), алгоритмы разрешения омонимии частей речи (существительное — прилагательное, существительное — глагол). В настоящее время разрабатывается алгоритм сегментации предложения. А целью является построение алгоритма полного синтаксического анализа.

Инструментальная среда может быть доработана для получения возможности на той же основе представлять алгоритмы семантического анализа текста. Для этого должны быть добавлены новые списки, морфологический словарь должен быть расширен за счет включения в словарь семантических категорий слов. Словарь в этом случае из чисто морфологического становится морфосемантическим.

Кроме того, должны быть добавлены функции и, возможно, типы данных для работы с синтаксическими группами. После этих дополнений инструментальная среда должна позволить отлаживать как алгоритмы синтаксического анализа, так и алгоритмы семантического анализа.

Список литературы

1. **Информатика.** Энциклопедический словарь для начинающих / Под общ. ред. Д. А. Поспелова. М.: Педагогика-Пресс, 1994.
2. **Толпегин П. В.** Информационные технологии анализа русских естественно-языковых текстов // Информационные технологии. 2006. № 8. С. 41—50; № 9. С. 2—7.
3. **Полякова И. Н.** Синтаксический анализатор адаптивной системы общения человека с ЭВМ на естественном языке. Пунктуационный компонент // Дис. ... канд. физ.-мат. наук. МГУ, 1994.
4. **Ермаков А. Е., Плешко В. В.** Синтаксический разбор в системах статистического анализа текстов // Информационные технологии. 2002. № 7. С. 30—34.
5. **Баталина А. М., Елифанов М. Е., Кобзарева Т. Ю., Кушнарёва Е. В., Лахути Д. Г.** Опыт экспериментальной реализации алгоритмов поверхностно-синтаксического анализа // Компьютерная лингвистика и интеллектуальные технологии: Тр. Междунар. конф. "Диалог'2006". М.: Наука, 2006.
6. **Гладкий А. В.** Синтаксический структуры естественного языка в автоматизированных системах общения. М.: Наука, 1985.
7. **Баталина А. М., Елифанов М. Е., Ивлиева О. О., Кобзарева Т. Ю., Лахути Д. Г.** Инструментальная среда для экспериментов с алгоритмами поверхностно-синтаксического анализа // Компьютерная лингвистика и интеллектуальные технологии: Тр. Междунар. конф. "Диалог'2004". М.: Наука, 2004.

8. **Синтаксический** анализ в системе РМЛ [Электронный ресурс] // Группа АОТ: Технологии: Синтаксис [web-сайт] < <http://aot.ru/docs/synan.html> > (20.11.2007).
9. **Перекрытченко А. А.** Разработка парсера и модуля унификации для синтаксического процессора // Компьютерная лингвистика и интеллектуальные технологии: Тр. Междунар. конф. "Диалог'2005". М.: Наука, 2005.
10. **Люстиг И. В.** Реализация семантико-синтаксического анализа естественно-языковых конструкций в интеллектуальной поисковой системе MEDSEARCH // Компьютерная лингвистика и интеллектуальные технологии: Тр. Междунар. конф. "Диалог'2005". М.: Наука, 2005.
11. **Балакирев Н. Е., Гирин Б. Б.** Построение семантической сети на основе толкового словаря // Новые материалы и технологии — 2006: Тез. докл. Всерос. конф. Т. 2. М.: МАТИ, 2006. С. 127—128.
12. **Гирин Б. Б.** О разработке языка описания алгоритмов анализа текста // Тез. докл. "Междунар. молодежной науч. конф." XXXI Гагаринские чтения". Т. 4. М.: МАТИ, 2005. С. 53—54.
13. **Грязнухина Т. А.** Синтаксический анализ научного текста на ЭВМ. Киев: Наукова думка, 1999.
14. **Русский** морфологический словарь [Электронный ресурс] // Группа АОТ: Технологии: Русская морфология [web-сайт] < <http://aot.ru/docs/rusmorph.html> > (20.11.2007).

УДК 004.451.7.031.43

Н. Н. Семенов,
СПбГМУ
nsemenoff@nwgsm.ru

Спектральные методы обработки пространственных данных для классификации целей

Обнаружение и классификация целей в современной гидро- и радиолокации — это две неразрывно связанные задачи. Если оптимальный прием и обнаружение эхо-сигнала хорошо описано в литературе, то оптимального или единственного работоспособного классификатора до сих пор нет. И данная статья рассматривает один из возможных методов обработки данных для последующей классификации — спектральную обработку пространственных данных.

Ключевые слова: классификация, спектральный метод, эхо-сигнал, гидролокатор, координаты, объект локации.

Введение

Обнаружение и классификация целей в современной гидро- и радиолокации — это две неразрывно связанные задачи. Если оптимальное обнаружение эхо-сигнала хорошо описано в литературе [1, 2, ...], то соответствующего классификатора до сих пор нет. И данная статья рассматривает один из возможных методов обработки данных

для последующей классификации — спектральную обработку пространственных данных.

Наиболее значимой информацией для классификации объектов локации является их внешний контур. Выделение контура из множества "блестящих точек" на поверхности целей возможно множеством разных методов, хорошо описанных в литературе [2, 4, 5, 6]. Это и методы кластеризации, и графические методы, связанные с размытием ярких "блестящих точек" до состояния единого пятна с последующим подчеркиванием границ полученного пятна и выделением его контура фильтрами, и самообучающиеся нейронные сети, выделяющие группы радиолокационных точек по тому или иному признаку. Поэтому считаем задачу выделения контура решенной.

1. История происхождения метода

Метод анализа и синтеза контуров как замкнутых кривых на плоскости с помощью Фурье-описания был разработан Р. Л. Косгрифом [1] как метод анализа топологии карт. Кривая представляется параметрически как функция длины дуги или накопленное, начиная с известной начальной точки, изменение кривизны. После нормализации эта периодическая функция разлагается в ряд Фурье, и коэффициенты усеченного разложения используются в качестве признаков формы. Коэффициенты более высокого порядка описывают изменения в направлении кривой на протяжении

небольшого по длине фрагмента контура и отбрасывание их эквивалентно эффекту низкочастотной фильтрации. При этом снижается уровень зашумленности контура, а оставшиеся коэффициенты Фурье концентрируют в себе макроскопическую информацию о форме.

За работой Р. Л. Косгрифа в этом направлении появился ряд работ, из которых следовало, что похожие формы в пространстве Фурье-описания с евклидовой метрикой образуют кластер. В более поздних работах [2, 3, 5, 9] внимание уделялось другим теоретическим и практическим аспектам спектрального анализа контуров.

Например, в работе [9] для получения Фурье-описания вводится нормализованная интегральная функция, устанавливающая связи между геометрией формы и алгебраическими описаниями свойств Фурье-описания контура. Во всех этих работах, использующих понятие спектра Фурье, большое значение играет комплексный гармонический сигнал $\exp\{i\omega n\}$. Но так как контур изображения является визуальным образом, то с позиции наглядности проводимого спектрального анализа и интерпретации его результатов целесообразно представить базисные функции также в виде визуальных образов, т. е. в виде контуров. Кроме наглядности это позволяет оперировать при спектральном анализе и фильтрации контуров сигналами, относящимися к одному и тому же классу.

2. Адаптация метода к задачам локации целей

Попробуем воспользоваться сведениями, полученными из теории распознавания изображений, для начальной обработки радиолокационных данных на основе БПФ и дальнейшей классификации.

При распознавании радиолокационных образов подлежащий распознаванию элемент определяется множеством координат (x, y) его контура. Координатное описание контура представляется комплексным числом:

$$z(n) = x(n) + jy(n), \quad (1)$$

где n — номер очередной пары измерительных данных, описывающих образ. Для их обработки будем использовать дискретное преобразование Фурье (DFT) в виде:

$$F_k = F(k) = \sum_{n=0}^{M-1} z(n) \exp\left(-j \frac{2\pi}{M} kn\right) \quad (2)$$

для $k = 1, 2, \dots, M-1$, где M — число точек описания структуры; $z(n)$ — комплексное число, определенное выражением (1). Отдельные компоненты преобразования Фурье образуют вектор F :

$$F = [F_0, F_1, \dots, F_{M-1}].$$

Этот вектор также определяет структуру образа, но в совершенно другом пространстве параметров. Компоненты этого описания позволяют легко преобразовывать данные независимо от их положения, масштаба, угла поворота, а также выбранной начальной точки и общего их количества. Следует подчеркнуть, что знания составляющих вектора F достаточны для полного восстановления формы кривой с помощью обратного преобразования Фурье (IDFT).

Нулевой компонент F_0 преобразования Фурье представляет собой среднее значение (центр тяжести) измерительных выборок (x_i, y_i) , поскольку

$$F_0 = \frac{1}{M} \sum_{n=0}^{M-1} z(n). \quad (3)$$

При приравнивании этого выражения к нулю образ, представленный вектором F , перемещается на стандартную позицию относительно системы координат, не зависящую от фактического первоначального расположения в пространстве данных. По этой причине вектор F после такого преобразования имеет вид $F_{xy} = [0, F_1, F_2, \dots, F_{M-1}]$, инвариантный относительно смещения.

Использование в преобразовании Фурье различного количества оригинальных выборок (x_i, y_i) отражается на размерности формируемых векторов F . Для унификации процесса обработки данных число наиболее значимых компонентов устанавливается априорно. Согласно теории преобразования Фурье, наиболее значимыми для отображения структуры компонентами считаются пары координат F_1 и F_{M-1} , следующими компонентами — F_2 и F_{M-2} и так далее. При определении K таких пар формируется редуцированное представление вектора $F_k = [0, F_1, F_2, \dots, F_K, F_{M-K}, \dots, F_{M-2}, F_{M-1}]$, которое независимо от числа измерительных выборок, использованных в преобразовании Фурье, имеет одну и ту же априорно установленную размерность $(2K + 1)$.

Инвариантность относительно масштаба образа можно обеспечить нормированием всех высших компонентов разложения Фурье, амплитудой компонента, соответствующего паре F_1 и F_{M-1} . Если обозначить коэффициент масштабирования K_S , то его можно определить выражением

$$K_S = \sqrt{|F_1|^2 + |F_{M-1}|^2}. \quad (4)$$

В этом случае нормализация компонентов F_k вектора F выполняется согласно формуле

$$F_{KS} = \frac{F_k}{K_S}. \quad (5)$$

При таком преобразовании данных полученная форма вектора $\mathbf{F}$ не зависит от размера образа, что очень удобно для дальнейшей классификации.

Преобразование Фурье состоит из компонентов, допускающих оригинальную качественную интерпретацию. Пары компонентов (F_1, F_{M-1}) , (F_2, F_{M-2}) и т. д. имеют свой эквивалент в обратном преобразовании IDFT, которое, в частности, для только одной (первой) пары можно представить в виде:

$$z_1 = \frac{1}{M} \left(F_1 \exp\left(-j\frac{2\pi k}{M}\right) + F_{M-1} \exp\left(-j\frac{2\pi(M-1)k}{M}\right) \right) = \frac{1}{M} \left(F_1 \exp\left(-j\frac{2\pi k}{M}\right) + F_{M-1} \exp\left(-j\frac{2\pi k}{M}\right) \right). \quad (6)$$

Уравнение (6) описывает эллипс. Первая пара (F_1, F_{M-1}) задает главный эллипс с наиболее длинной осью, вторая пара (F_2, F_{M-2}) — следующий по значению и так далее. Поворот кривой относительно начальной позиции вызывает поворот главной оси эллипса. Поэтому для обеспечения неизменности измерительных данных относительно угла их поворота следует нормализовать положение этой оси. Коэффициент нормализации угла поворота может быть определен выражением

$$K_r = \exp\left(-j\frac{\psi_{F_1} + \psi_{F_{M-1}}}{2}\right), \quad (7)$$

где ψ_{F_1} и $\psi_{F_{M-1}}$ — это углы степенного представления комплексных чисел F_1 и F_{M-1} соответственно. Нормализация данных, обеспечивающая их инвариантность относительно угла поворота, основана на умножении каждого компонента вектора преобразования Фурье F_k на коэффициент K_r :

$$F_{kr} = K_r F_k. \quad (8)$$

После такого преобразования вектор характеристик образа, подаваемый на последующий классификатор, не будет зависеть от угла поворота этого образа. Последовательное выполнение описанных преобразований применительно к исходным компонентам вектора $\mathbf{F}$ обеспечивает полную инвариантность относительно перемещения, поворота и масштабирования. Компоненты преобразованного таким образом вектора называются дескрипторами образов. Пример результатов работы преобразователя показан на рис. 1.

Как видно из рис. 1, центр тяжести фигуры действительно находится в координатах (0,0), размер отнормирован до постоянного значения, и последнее действие — нормирование поворота фигуры до горизонтального положения.

Пример таких преобразований на языке MatLab показан на рис. 2.

Важным следствием применения преобразования Фурье в качестве препроцессора считается уменьшение зависимости результатов распознавания от шума, возмущающего измерения. Помехи, как правило, имеют характер высокочастотного шума. В преобразовании Фурье это соответствует полосе разложения в высокочастотном диапазоне (компоненты высшего порядка вектора $\mathbf{F}$). Отсечение этих компонентов вызывает автоматическое уменьшение уровня шума в образе сигнала после его воспроизведения. На рис. 3 и 4 иллюстрируется влияние конечного числа дескрипторов Фурье на форму воспроизведенных образов: оригинальный зашумленный образ; образ, воспроизведенный с использованием всех 64 дескрипторов Фурье; образ, воспроизведенный с использованием только пяти наиболее значимых нормализованных дескрипторов Фурье. Из рис. 4 видно, уменьшение числа дескрипторов Фурье автоматически повышает качество воспроизведенного образа. Коррекция качества заметна также и при анализе численных зна-

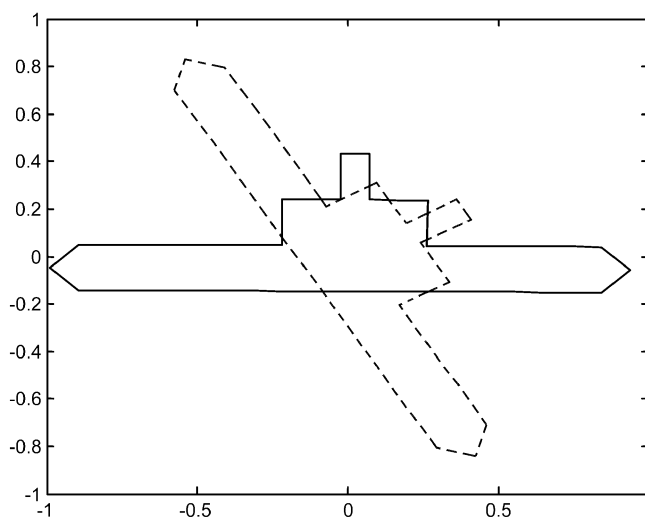


Рис. 1. Результаты работы преобразователя: штриховой линией показан образ до нормирования поворота, сплошной линией — после нормирования

```
z = x + 1i.*y; % Построили комплексные координаты

figure(1);
plot(z); % Нарисовали исходный набор данных

fz = fft(z); % Нашли пространственный спектр огибающей
fz(1)=0; % Перенесли центр тяжести в (0,0)
Ks = sqrt((abs(fz(2))^2+abs(fz(end))^2));
fz1 = fz./Ks; % Нормировка размеров

Kr = exp(-1i*(angle(fz(2))+angle(fz(end)))/2);
fz2 = fz1.*Kr; % Поворот к горизонтальному представлению

z2 = ifft(fz2); % Обратное преобразование
figure(2);
plot(z2,'g'); % Вывод результатов
```

Рис. 2. Пример программы для преобразований координат огибающей объекта

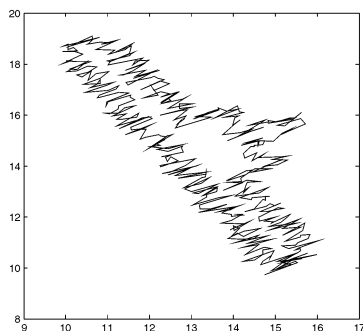


Рис. 3. Исходные данные

чений дескрипторов Фурье. Из анализа зашумленных данных следует, что даже при значительном присутствии шума в измерительных сигналах амплитудные характеристики дескрипторов изменяются очень незначительно. Это очень полезное качество для распознавания радиолокационных образов, скрытых помехами.

Выходные сигналы препроцессора в виде последовательности компонентов дескрипторов Фурье после преобразования, обеспечивающего инвариантность к перемещению, повороту и масштабированию, становятся выходными сигналами для многослойной нейронной сети, играющей роль системы распознавания образов и одновременно выполняющей их классификацию (отнесение каждого образа к соответствующему эталонному классу). Число входных узлов сети равно числу дескрипторов Фурье, учитываемых при классификации. Если допустить, что каждый входной нейрон представляет единственный класс, то их число тоже будет постоянной величиной, равной числу классов. Поэтому подбираться может только число скрытых слоев и число нейронов в каждом слое.

Классификатор тренируется методом обратного распространения с использованием одного из обучающих алгоритмов на множестве обучающих данных, последовательно представляющих все классы образов, подлежащих распознаванию. В режиме воспроизведения классифицируемых, образ, прошедший через все фазы препроцессора, подается на вход сети, возбуждая тот выходной нейрон, который соответствует требуемому классу.

Вследствие зашумленности образов на этапе их распознавания выходные сигналы нейронов сети могут принимать непрерывные значения из интервала $[0, 1]$

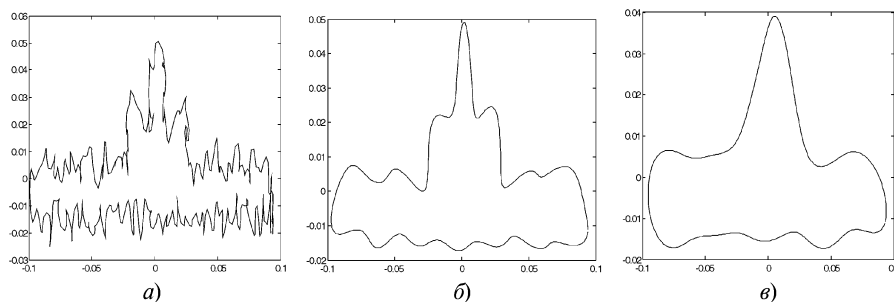


Рис. 4. Результат уменьшения числа компонентов частотного представления:
а — 25 компонентов; б — 15; в — 10

вместо ожидаемых нуль-единичных значений (с единицей, обозначающей распознанный класс).

Один из способов решения этой проблемы заключается в том, что в качестве представителя распознанного класса признается наиболее активный нейрон (выработавший самый сильный сигнал). Однако такой подход не позволяет сравнивать активность различных нейронов и приводит к ситуации, в которой решение о победе конкретного нейрона принимается даже тогда, когда активность всех нейронов близка к нулю. Это может приводить к ошибочной классификации.

Наилучшим подходом представляется двухуровневая классификация. Вначале проверяется, насколько максимальный сигнал превышает следующий за ним. Если разница достаточно велика, победителем признается наиболее активный нейрон. В противном случае, а также, если уровни активации всех нейронов не превышают определенного порога, интерпретатор при объявлении результата сообщает, что классификация считается неполной и тем самым предостерегает пользователя от возможной ошибки.

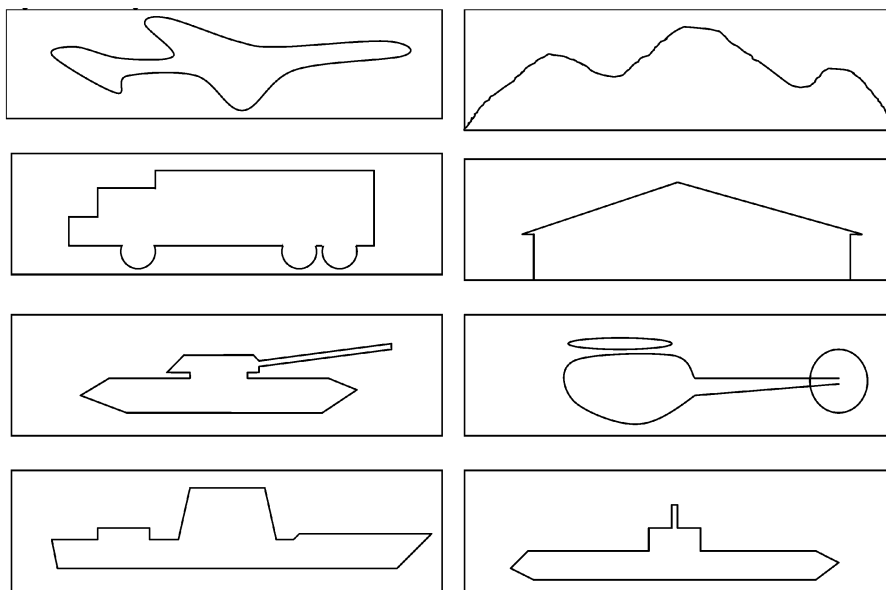


Рис. 5. Примеры объектов для распознавания

Выводы

Подобная реализация нейросетевого классификатора была апробирована при распознавании и классификации многих разнообразных образов, в том числе букв, цифр, предметов, объектов и т. п. На рис. 5 представлено тестовое множество объектов различных классов, которые распознавались с использованием персептронной сети. После предварительной обработки этих данных с помощью преобразования FFT были сформированы 18-элементные векторные дескрипторы (пять пар наиболее значимых коэффициентов Фурье для амплитуды и для фазы, при этом фазовые компоненты F_1 и F_{M-1} имеющие нулевые значения, не использовались). В ходе многочисленных экспериментов число скрытых нейронов выбрано равным 8. Применялась простейшая интерпретация результатов. Выходной сигнал в диапазоне 0...0,5 рассматривался как нулевой, а свыше 0,5 — как единица. После нормализации компонентов преобразования Фурье эффективность распознавания незашумленных сигналов составила 100 %. Только значительная зашумленность из-

мерительных сигналов (исходные данные зрительно почти не распознавались) с уровнем шума порядка 70 % уменьшила эффективность распознавания до 90 %.

Список литературы

1. Mahafza Bassem R. Radar Systems Analysis and Design. Chapman & Hall, 2000.
2. Osjwsky S. Signal flow graphs and neural networks // Biological Cybernetics. 1994. Vol. 70. P. 387—395.
3. Cossgriff R. L. Identification of Shape // Ohio, OSURF, 1960, rep. 820—11.
4. Фурман Я. А. Введение в контурный анализ. М.: Физматлит, 2003.
5. Webb Andrew B. Statistical Pattern Recognition. John Wiley & Sons, 2002.
6. Анисимов Б. В., Курганов В. Д., Злобин В. К. Распознавание и цифровая обработка изображений. М.: Высш. шк., 1983.
7. Ту Дж., Гонсалес Р. Принципы распознавания образов. М.: Мир, 1978.
8. Ray Liu K. J. Pattern Recognition and Image Processing. New York: Marcel Dekker, 2002.
9. Theodoridis S., Koutrumbas K. Pattern Recognition. Elsevier: Academic Press, 2003.
10. Duda R. O., Hart P. E., Stork D. G. Pattern Classification. John Wiley & Sons, 2001.
11. Патрик Э. А. Основы теории распознавания образов. М.: Советское радио, 1980.

УДК 681.519

В. П. Лапшин, канд. техн. наук,
И. С. Жуков, адъюнкт, **А. Х. Апилов**, курсант,
Ростовский военный институт
ракетных войск им. Главного маршала
артиллерии М. И. Неделина

Количественная оценка размытости и асимметрии функций принадлежности нечетких множеств

Дан анализ ряда функций принадлежности нечетких множеств с определением степени размытости и асимметрии этих множеств. Определены показатели размытости и асимметрии, которые позволяют получить количественную оценку этих свойств нечетких множеств. Определена граница размытости нечетких чисел, позволяющая ограничить дальнейшие операции над нечеткими данными.

Ключевые слова: функция принадлежности, нечеткие множества, степень размытости, асимметрия.

Введение

В последнее время при решении задач управления сложными организационными системами получил широкое распространение математиче-

ский аппарат теории нечетких множеств, наиболее часто применяемый для учета неопределенности в оценке обстановки экспертами. Необходимо отметить, что представление данных в виде нечетких множеств, связано в первую очередь с тем, что получение точной оценки данных во многих ситуациях невозможно, или такая оценка будет ошибочна. Данные, представленные в виде нечетких множеств, можно представить как совокупность достоверных данных и оценки погрешности их получения, при этом нечеткость данных (размытость) определяется погрешностью измерения. Для существующих методов представления данных в виде нечетких множеств характерно представлять погрешность измерения в таких понятиях, как размытость и асимметрия нечетких чисел. В связи с этим представляет интерес оценка степени размытости и асимметрии (точности метода), вводимой теми или иными методами представления и обработки нечетких данных.

1. Оценка нечетких множеств (чисел), представляемых в треугольной форме

1.1. Некоторые общие понятия

Наиболее используемой формой нечеткого числа является так называемая выпуклая нормальная треугольная форма, позволяющая суще-

ственно упростить все операции над нечеткими числами. При такой форме нечетких чисел исходные числа и результаты операций над ними могут быть представлены в виде нечеткого множества, состоящего только из трех членов. Рассмотрим такие числа [1].

Определение 1.1 Нечеткое число $\tilde{A}$ на действительной прямой — это нечеткий набор, характеризуемый функцией принадлежности $\mu_{\tilde{A}}: R \rightarrow [0,1]$. Нечеткое число $\tilde{A}$ может быть выражено как

$$\tilde{A}' = \int \mu_{\tilde{A}}(x)/x, \quad (1.1)$$

где $\mu_{\tilde{A}}(x) \in [0, 1]$ — степень принадлежности $x \in R$ множеству $\tilde{A}'$, $\int$ — объединение по всем $x \in R$; $\mu_{\tilde{A}}(x)/x$ означает, что степень принадлежности x множеству $\tilde{A}$ равна $\mu_{\tilde{A}}(x)$.

Определение 1.2 [1] Число a называется границей функции принадлежности, если выполняются следующие соотношения:

$$\forall \delta \mu(a) = 0; \mu(a - \delta) = 0; \mu(a + \delta) \neq 0.$$

Функция принадлежности имеет две границы: верхнюю и нижнюю. Таким образом, непрерывное нормальное выпуклое нечеткое число можно записать в виде

$$\tilde{A}' = \int_a^A (x - a)/x + \int_A^b (b - x)/x, \quad (1.2)$$

где a, b — нижняя и верхняя границы, A — мода функции принадлежности. Например, нечеткое число $\tilde{2}$ (приблизительно 2) можно представить в виде следующего выражения:

$$\tilde{2} = \int_1^2 (x - 1)/x + \int_2^3 (3 - x)/x, \quad (1.3)$$

или графически (рис. 1.1).

1.2. Оценка размытости нечеткого треугольного числа

Приведенное в предыдущем подразделе статьи нечеткое число могло быть и другим, например:

$$\tilde{2} = \int_{0,5}^2 (x - 0,5)/x + \int_2^{3,5} (3,5 - x)/x, \quad (1.4)$$

или представленным графически, как показано на рис. 1.2.

По своей сути это уже два различных числа, отличающихся границами, которые определяют степень размытости нечеткого числа. Для упрощения дальнейших рассуждений введем показате-

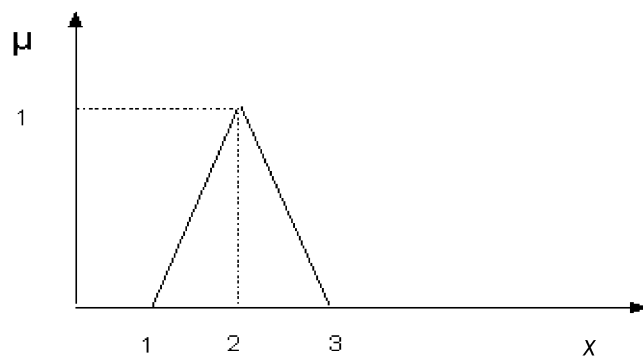


Рис. 1.1. Нечеткое число ПРИБЛИЗИТЕЛЬНО 2 с границами 1,3

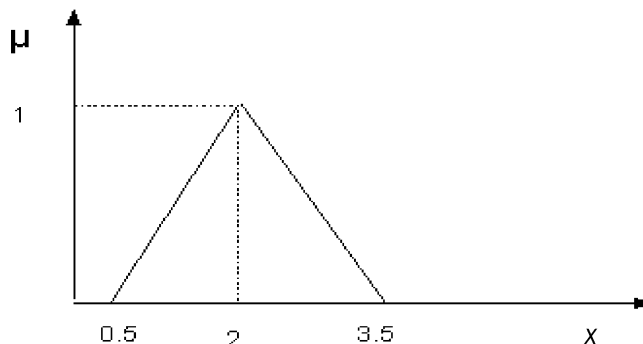


Рис. 1.2. Нечеткое число ПРИБЛИЗИТЕЛЬНО 2 с границами 0,5, 3,5

тель размытости выпуклого симметричного нормального треугольного нечеткого числа:

$$\Pi_p(\tilde{A}) = \frac{b-a}{A}, \quad (1.5)$$

где a, b — левая и правая границы нечеткого множества, A — значение моды функции принадлежности.

Приведенный в выражении (1.5) показатель размытости нечеткого числа определяется как разность между верхней и нижней границами носителя множества, взвешенная относительно моды функции принадлежности нечеткого числа. Так, для примера нечеткого числа, приведенного в выражении (1.3) и графически представленного на рис. 1.1, $\Pi_p(\tilde{2}) = 1$, а для примера, приведенного в выражении (1.4) и представленного графически на рис. 1.2, $\Pi_p(\tilde{2}) = 1,5$. Так как степень размытости нечеткого числа определяет степень достоверности тех данных, которые представляются в виде исследуемого нечеткого множества, то при $\Pi_p(\tilde{A}) > 1$ эти данные имеют достоверность ниже нормированной единицы. Необходимо отметить, что использование таких данных сомнительно в виду большой степени неопределенности, возникающей при принятии решений на основе этих данных. В работе [2] вводится понятие устойчивости операций, проводимых с нечет-

кими числами, обобщая это понятие на сами нечеткие числа, можно говорить о неустойчивости и устойчивости самих нечетких чисел. Действительно, если нечеткое число имеет высокую степень размытости ($\Pi_p(\tilde{A}) > 1$), то проведение ряда дальнейших операций с этим нечетким числом (скажем суммирование) приведет к еще более размытому результату, имеющему $\Pi_p(\tilde{A}) > > 1$. Использование такого результата представляется невозможным ввиду крайне низкой степени достоверности информации, отображаемой им.

Использование показателя размытости при анализе степени размытости нечетких данных, получаемых от различных методов представления и обработки данных, может позволить отбросить те данные, которые несут в себе чрезмерную погрешность.

1.3. Оценка асимметрии нечеткого треугольного числа

Приведенные ранее примеры нечеткого числа ПРИБЛИЗИТЕЛЬНО 2 были симметричными треугольными числами, но отметим, что это частный случай, в действительности такое число может быть несимметричным и иметь вид, представленный следующим выражением:

$$\tilde{2} = \int_{1,5}^2 (x - 1,5)/x + \int_2^{3,5} (3,5 - x)/x, \quad (1.6)$$

или графически, как показано на рис. 1.3.

Приведенное в выражении (1.6) и на рис. 1.3 нечеткое число представляет собой пример асимметрии в нечетких выпуклых нормальных треугольных числах. Асимметрия нечеткого треугольного числа может быть вызвана рядом причин: неуверенностью экспертов в наличии правильной треугольной формы данных, представляемых таким числом, спецификой рассматриваемой предметной области, сомнением экспертов в правиль-

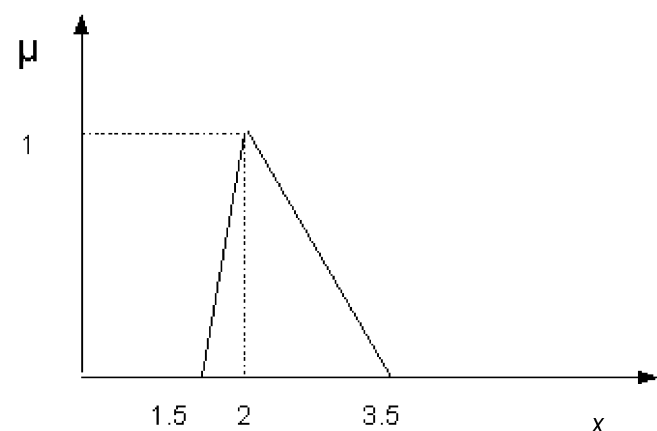


Рис. 1.3. Нечеткое число ПРИБЛИЗИТЕЛЬНО 2 с левой асимметрией

ности выбора нечеткого числа (для приведенного примера (1.6) возможно более правильное число ПРИБЛИЗИТЕЛЬНО 2,5). Оценка асимметрии нечеткого числа позволит учесть ошибку, связанную с возможным отклонением моды и формы нечеткого числа, отражающего те или иные данные. Для оценки асимметрии нечетких треугольных чисел в дальнейшем будем использовать показатель асимметрии, рассчитываемый следующим образом:

$$\Pi_a(\tilde{A}) = \frac{|(b - A) - (A - a)|}{A}. \quad (1.7)$$

Приведенный в выражении (1.7) показатель характеризует разницу между правой и левой частями нечеткого треугольного числа, взвешенную относительно моды функции принадлежности.

Показатель асимметрии имеет следующий ряд свойств:

- в случае симметричного нечеткого треугольного числа он равен нулю;
- в случае отсутствия левой или правой части (относительно моды) треугольного нечеткого числа (прямоугольный треугольник) показатель асимметрии равен показателю размытости этого нечеткого числа.

2. Нетреугольные формы функций принадлежности нечетких чисел

2.1. Вырожденная форма треугольного числа

Вырожденная форма треугольного числа является нечетким множеством, отображающим лингвистические переменные, например, переменная "Возраст" может принимать значения "молодой" или "старый", математически это будет так:

$$\begin{aligned} \tilde{A}' &= \int_0^{25} 1/x + \int_{25}^{30} (30 - x)/x, \\ \tilde{A}' &= \int_{60}^{65} (65 - x)/x + \int_{65}^{150} 1/x, \end{aligned} \quad (2.1)$$

а графически, как показано на рис. 2.1.

Каждое значение переменной "Возраст" несет в себе как нечеткую, так и четкую составляющую [см. выражение (2.1)]. В примере, приведенном на рис. 2.1, значение переменной "молодой" — четкая составляющая 0—25, нечеткая 25—30; значение переменной "старый" — нечеткая составляющая 60—65, четкая 65—150. Отметим, что и в случае выпуклой нормальной треугольной формы нечеткого числа присутствует четкая составляющая — мода функции принадлежности.

Дальнейшие рассуждения будут основываться на более обобщенных схемах вырожденных форм

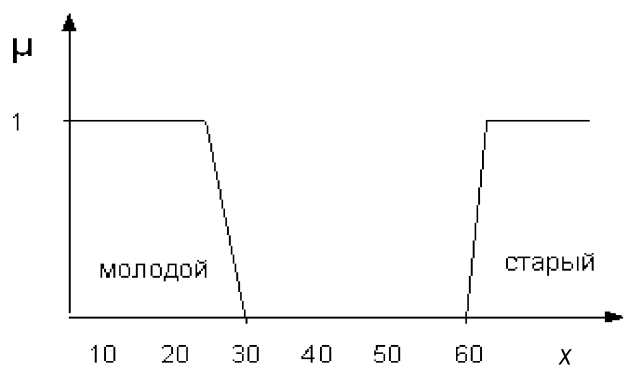


Рис. 2.1. Функции принадлежности нечетких термов переменной "Возраст"

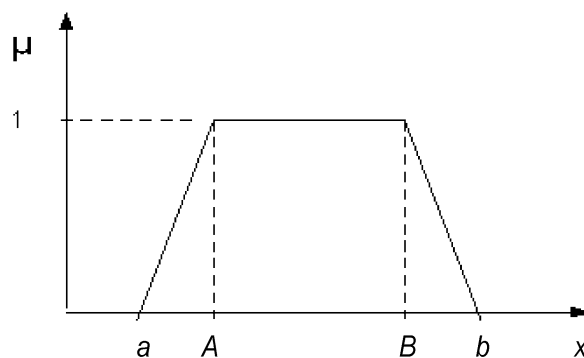


Рис. 2.3. Пример нечеткого трапецевидного числа

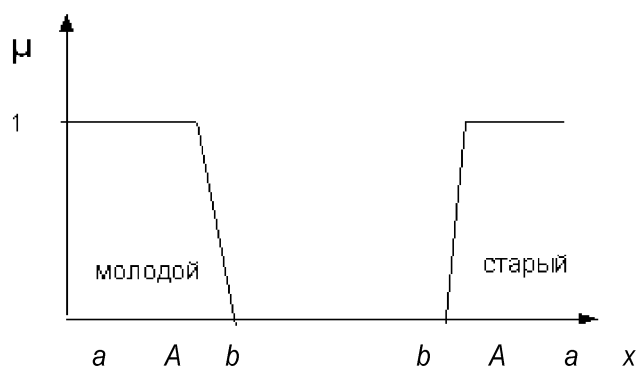


Рис. 2.2. Обобщенная схема функции принадлежности нечетких термов переменной "Возраст"

нечетких треугольных чисел, вид которых приведен на рис. 2.2.

Степень размытости вырожденной треугольной формы нечеткого числа определим как отношение нечеткой составляющей числа к четкой. Соответственно для значений переменной "Возраст" (представленных на рис. 2.2) выражение для расчета показателя размытости будет иметь следующий вид:

$$\Pi_p(\tilde{A}) = \frac{|b - A|}{A}. \quad (2.2)$$

Показатель асимметрии для вырожденных форм нечетких треугольных чисел будет равен показателю размытости, в связи с чем, показатель асимметрии для такой формы чисел определять не будем.

2.2. Трапецевидная форма функций принадлежности нечетких чисел

Нечеткие числа трапецевидной формы могут быть представлены следующим выражением:

$$\tilde{A} = \int_a^A (x - a)/x + \int_A^B 1/x + \int_B^b (b - x)/x, \quad (2.3)$$

или графически, как показано на рис. 2.3.

Как и в случае вырожденной формы нечеткого треугольного числа, при трапецевидной форме показатель размытости определяется только по нечеткой составляющей и рассчитывается по следующему выражению:

$$\Pi_p(\tilde{A}) = \frac{(A - a) + (b - B)}{(B - A)}. \quad (2.4)$$

Выражение для показателя асимметрии таких нечетких чисел имеет следующий вид:

$$\Pi_a(\tilde{A}) = \frac{|(A - a) - (b - B)|}{(B - A)}. \quad (2.5)$$

В случае трапецевидной формы, в отличие от вырожденной формы функции принадлежности нечетких треугольных чисел, имеются оба показателя — размытости и асимметрии, что объясняется наличием нечеткости с обеих сторон от интервальной оценки.

2.3. Сглаженные формы нормальных выпуклых функций принадлежности нечетких чисел

При построении функции принадлежности чисел, приблизительно равных некоторому числу A , можно использовать функцию следующего вида [1]:

$$\mu_{\tilde{A}}(x) = e^{-\alpha(A-x)^2}, \quad (2.6)$$

где α зависит от требуемой степени нечеткости $\mu_{\tilde{A}}(x)$ и определяется из выражения

$$\alpha = \frac{4 \ln 0,5}{\beta^2}, \quad (2.7)$$

β — расстояние между точками перехода для $\mu_{\tilde{A}}(x)$, т. е. точками, в которых функция вида (2.6) принимает значение 0,5.

Графически нечеткие числа, соответствующие выражению (2.6), представлены на рис. 2.4.

Показатель размытости для таких нечетких чисел определим как удвоенное отношение расстояния между точками перехода к моде функции принадлежности:

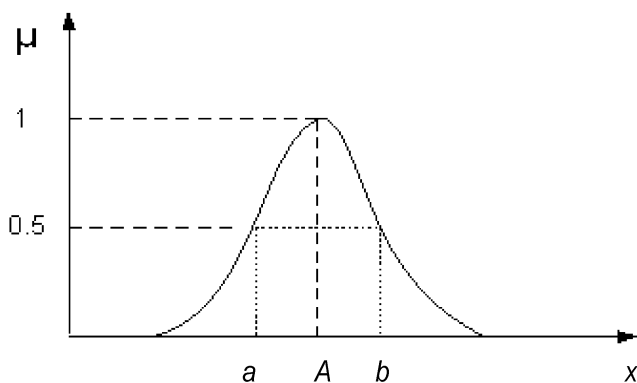


Рис. 2.4. Функция принадлежности, соответствующая сглаженной приближенной оценке

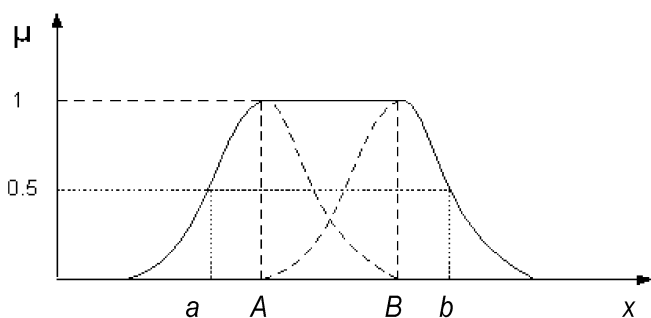


Рис. 2.5. Функция принадлежности нечеткого множества, соответствующего интервальной оценке ПРИБЛИЗИТЕЛЬНО В ИНТЕРВАЛЕ ОТ А ДО В

$$\Pi_p(\tilde{A}) = 2 \frac{(b-a)}{A}. \quad (2.8)$$

Показатель асимметрии для функций вида (2.6) определим как два отношения модуля разницы расстояний между точками перехода и модой функции к самой моде:

$$\Pi_a(\tilde{A}) = \frac{2|(b-A)-(A-a)|}{A}, \quad (2.9)$$

Для сглаженной формы интервальных оценок, представленной рис. 2.5, размытость и асимметрия функций принадлежности рассчитывается таким же образом, как и в предыдущем случае (трапецевидная форма). Показатель размытости определяется как

$$\Pi_p(\tilde{A}) = \frac{2((A-a)+(b-B))}{(B-A)}. \quad (2.10)$$

Показатель асимметрии соответственно представим как

$$\Pi_a(\tilde{A}) = \frac{2|(A-a)-(b-B)|}{(B-A)}. \quad (2.11)$$

Таким образом, мы рассмотрели сглаженные формы выпуклых нормальных функций принадлежности нечетких множеств и получили для них показатели размытости и асимметрии. При этом нами не рассмотрены сглаженные вырожденные формы треугольных функций принадлежности, но отметим, что показатель размытости для них практически не отличается от показателя, представленного в выражении (2.2), а показатель асимметрии для таких функций принадлежности не вычисляется, исходя из соображений, представленных в подразделе 2.1 этой статьи.

Заключение

При решении многих задач управления ошибка экспертизы той или иной предметной области учитывается методами представления полученных экспертных данных в виде нечетких множеств (чисел). Такая ошибка, строго говоря, не нужна и носит вынужденный характер, обусловленный рядом причин. При этом отметим, что чем больше размытость и асимметрия нечетких множеств (чисел), соответствующих данным экспертизы, тем больше такая ошибка.

В статье приводится количественная оценка размытости и асимметрии для различных форм функций принадлежности нечетких множеств (чисел). Использование показателей размытости и асимметрии позволяет количественно оценивать точность данных, представленных в виде нормальных выпуклых нечетких множеств на различных этапах представления и обработки этих данных. Дальнейшим развитием предложенного в статье подхода к оценке нечетких множеств (чисел) является практическое определение асимметрии и размытости нечетких чисел, получаемых в результате обработки экспертных данных теми или иными методами представления их в виде нечетких множеств. Задачей такого анализа будет являться определение сравнительной точности этих методов при условии использования ими одних и тех же данных экспертизы.

Список литературы

1. Борисов А. Н., Крумберг О. А., Федоров И. П. Принятие решений на основе нечетких моделей. Примеры использования. Рига: Зинайте, 1990. 184 с.
2. Федулов А. С. Устойчивая операция аккумуляции нечетких чисел // Нейрокомпьютеры: разработка, применение. 2007. № 1. С. 27–39.
3. Zadeh L. A. Fuzzy sets as a basis for a theory of possibility // Fuzzy Sets Systems. 1978. Vol. 1. N 1. P. 3–28.
4. Zadeh L. A. Fuzzy sets and information granularity // Advances in fuzzy sets theory and applications / Ed. M. M. Gupta, R. K. Ragade, R. R. Yager. Amsterdam: North-Holland Publ. Corp., 1979. P. 3–18.

Е. Е. Ковшов, д-р техн. наук., проф.,

А. В. Машковцев,

Московский государственный
технологический университет "СТАНКИН"

Рационализация подбора медицинского персонала в лечебном учреждении путем применения генетических алгоритмов

Рассматриваются вопросы применения аппарата генетических алгоритмов, являющегося одной из парадигм эволюционных вычислений, для рационализации подбора медицинского персонала в лечебном учреждении в целях повышения эффективности лечебного и организационного процессов.

Ключевые слова: генетические алгоритмы, эволюционные вычисления, подбор персонала, автоматизированная подсистема, обменный формат.

В современном обществе информация становится важнейшим фактором экономического роста, и информатизация охватывает все новые сферы деятельности, к которым принадлежит, например, работа кадровых служб лечебно-профилактических учреждений (ЛПУ). Одной из непростых задач при этом является подбор персонала на вакантные должности [2].

Большинство ЛПУ давно столкнулись с необходимостью повысить качество указанной процедуры. Для этого необходимо совершенствовать механизмы отбора кадров и сам процесс отбора реализовать на современной научной основе, учитывая как отечественный, так и зарубежный опыт.

Обычно персонал подбирают с помощью поэтапной процедуры (рис. 1). На каждом из этапов отсеивается часть претендентов. В зависимости от должности или специфики деятельности предприятия меняются последовательность и важность этапов, а также критерии отбора претендентов. Среди требований к кандидату со стороны ЛПУ — общие характеристики (пол, возраст, место проживания, семейное положение), образование, наличие ученых степеней и званий, опыт работы, специальные навыки (владение компьютером, знание иностранных языков и др.), личностные качества и т. д.

Общие характеристики и образование претендента на вакансию определяют на основе предоставляемых им документов. Опыт работы подтверждается записями в трудовой книжке, отзы-

вами и рекомендациями. Наличие специальных навыков выявляют специалисты в соответствующих областях с помощью методики, адаптированной к предметной области деятельности. Состояние здоровья кандидата определяется по результатам медицинского осмотра.

Сложнее всего определить, насколько соответствуют должности личностные качества претендента. Для их оценки наиболее часто применяют собеседование (интервью), анализ документов, тестирование. Надежность этих методов не абсолютна, так как ни один из них не дает полной и точной информации о кандидате. Каждый из подходов имеет свои преимущества и недостатки. На практике в настоящее время наиболее распространенным методом подбора кадров является собеседование.

Несомненно, собеседование — неотъемлемый этап выбора сотрудника. Однако существует ряд проблем, снижающих его эффективность как инструмента отбора персонала. Ключевой причиной этих проблем можно назвать субъективность интервьюера, т. е. их основа носит эмоциональный и психологический характер. На восприятие кандидата интервьюером оказывают влияние стереотипы, первое впечатление, физическая привлекательность (непривлекательность) кандидата, его манеры, одежда и прочие факторы.

На основе изложенного представляется необходимым дополнить процесс отбора новой процедурой, основанной на использовании генетических алгоритмов (ГА) на базе компьютерных технологий. Это, на наш взгляд, позволит получать устойчивый результат и оценку кандидата, максимальным образом не зависящую от субъективных факторов.

Задача подбора кандидата на вакансию может быть сформулирована как поиск оптимального значения, где значение — сложная функция, зависящая от множества входных параметров. Причем точный оптимум не требуется; решением может считаться любое значение, которое точнее некоторого заданного. Вследствие сложности задачи традиционные методы, ориентированные на поиск адекватных значений, непригодны, тогда как ГА позволяют решить данную задачу [1, 3]. Наиболее целесообразно использовать ГА на этапе 3 (см. рис. 1). ГА являются адаптивными методами поиска и в последнее время часто используются для функциональной оптимизации. Они имитируют генетические процессы, протекающие в биологических системах: биологические популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора. Подражая этим процессам, ГА способны находить решения реальных задач, если те соответствующим образом закодированы, т. е. ГА представляют собой простую модель эволюции в природе, реализованную в виде компьютерной про-

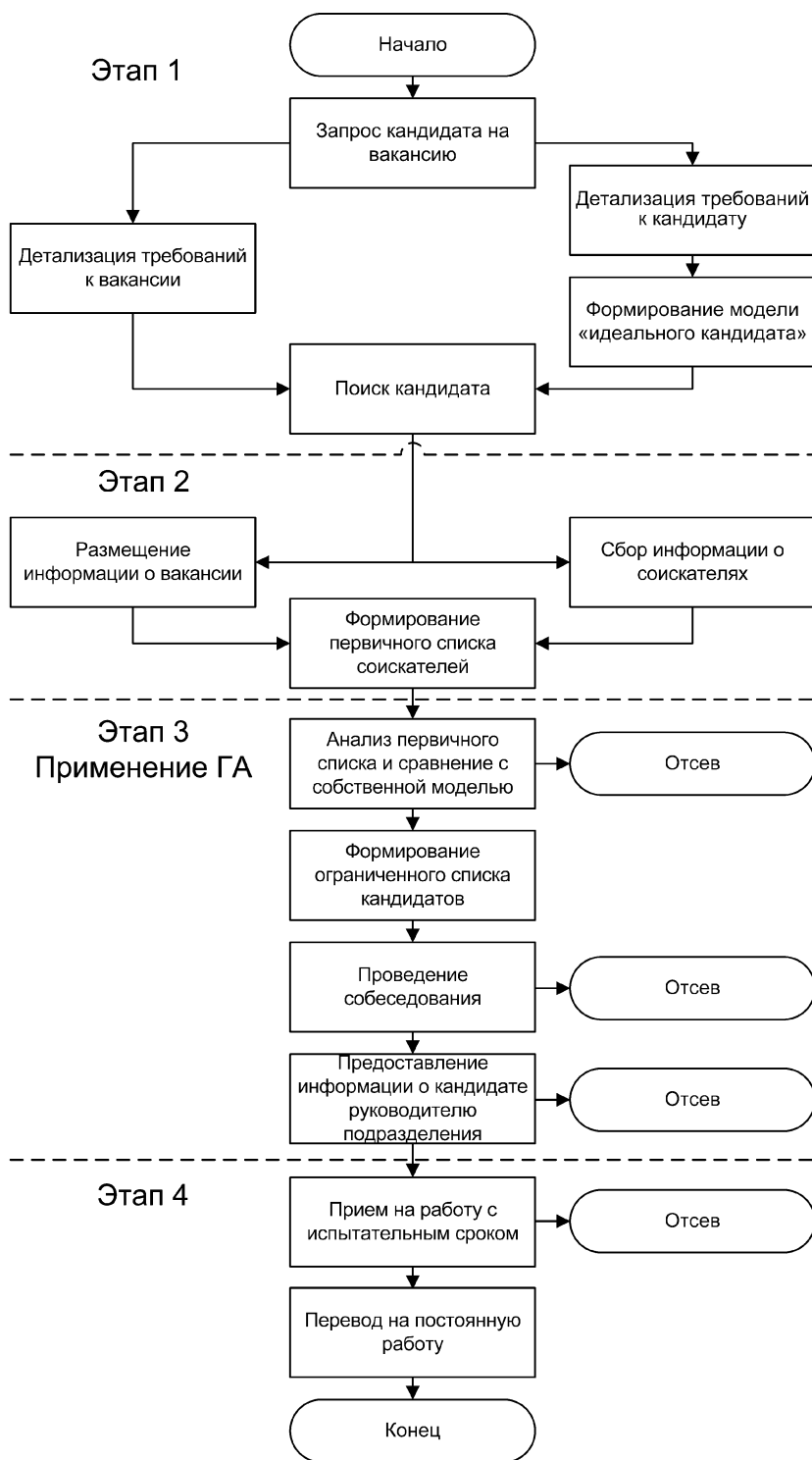


Рис. 1. Последовательность подбора кандидатов на вакансии

граммы. При этом используется несколько упрощенная биологическая терминология. Следует отметить, что ГА широко применяют во многих областях деятельности человека (дизайн продукции, мониторинг сложных систем, анализ финансового рынка и др.), так как они позволяют за сравнительно короткий промежуток времени ре-

шать сложные задачи с множеством постоянно изменяющихся параметров и сотнями формул.

Каждый объект (элемент любого множества из табл. 1) можно представить в виде совокупности атрибутов, численно характеризующих данный объект. Атрибуты определены на ограниченном множестве положительных значений.

Вектор подбора имеет вид $PODBOR = \{K_1, K_2, \dots, K_V, \dots, K_N\}$, где K_V — номер кандидата, подобранного на V -ю вакансию, $V = \{1, \dots, N\}$, $K = \{1, \dots, M\}$.

Таким образом, задача подбора кандидата на вакантную должность сводится к задаче выбора такого варианта вектора $PODBOR$ из числа возможных, который с учетом ограничений и критериев будет в максимальной степени способствовать достижению поставленной цели.

Чтобы определить критерии отбора, следует ясно сформулировать качества работника, необходимые для соответствующего вида деятельности. Критерии следует формировать так, чтобы они всесторонне характеризовали как работника, так и вакансию (табл. 2).

Ограничения

1. Весовой коэффициент кандидата должен быть больше или равен весу вакансии:

$$W_K \geq W_V$$

2. Число вакансий должно быть меньше или равно числу кандидатов.

Критерии

1. На конкретную вакансию должен быть выбран один кандидат.

Весовые коэффициенты кандидата и вакансии определяются по следующим формулам:

$$W_{K_i} = \prod_{j=1}^M (K_{i,j})^{n_j},$$

где $K_{i,j}$ — значение j -го атрибута у i -го объекта из вектора кандидатов, $i = \{1 \dots M\}$; n_j — относительный вес j -го атрибута у i -го объекта;

$$W_{V_i} = \prod_{j=1}^N (V_{i,j})^{n_j},$$

где $V_{i,j}$ — значение j -го атрибута у i -го объекта из вектора вакансий, $i = \{1...N\}$, n_j — относительный вес j -го атрибута у i -го объекта.

Дополнительное замечание: поскольку кандидат может быть назначен только на одну вакансию, в рассматриваемых хромосомах каждый ген (номер кандидата) должен встречаться только один раз. Такая разновидность хромосом называ-

Таблица 1

Исходные данные для процесса подбора кандидатов на вакантную должность

1	Кандидаты	$K = \{1...M\}$, M — число кандидатов
2	Вакансии	$V = \{1...N\}$, N — число вакансий

Таблица 2

Характеристики кандидата и вакансии

Атри- бут	Возможные значения атрибута	Значение коэффи- циента	Относитель- ный вес атрибута
Кандидат			
Образо- вание	Неполное среднее	1	3
	Среднее	2	
	Средне-специальное	3	
	Неполное высшее	4	
	Высшее	5	
	Несколько высших	6	
Квали- фика- ция	Низкая	1	4
	Средняя	2	
	Высокая	3	
Опыт работы	Не учитывается	1	3
	Нет	2	
	Есть	3	
Спец. навыки	Не учитывается	1	2
	Нет	2	
	Есть	3	
Долж- ность	Средний медперсонал	1	2
	Врачебный медперсонал	2	
	Руководитель	3	
Вакансия			
Образо- вание	Неполное среднее	1	3
	Среднее	2	
	Средне-специальное	3	
	Неполное высшее	4	
	Высшее	5	
	Несколько высших	6	
Квали- фика- ция	Низкая	1	4
	Средняя	2	
	Высокая	3	
Опыт работы	Не учитывается	1	2
	Нет	2	
	Есть	3	
Спец. навыки	Не учитывается	1	2
	Нет	2	
	Есть	3	
График работы	Любой	1	1
	Свободный график	2	
	Полный день	3	
Заня- тость	По совместительству	1	1
	Полное	2	

ется "перечислимые хромосомы с уникальными генами" и часто используется в комбинаторных задачах. Стандартная операция скрещивания для этого типа хромосом опять же некорректна, поэтому здесь используется более сложная схема двухточечного скрещивания.

Для организации работы ГА необходимо определить следующие понятия.

1. **Целевая функция** — функция, численно характеризующая результат подбора кандидата:

$$F(PODBOR) = F(PODBOR)_{\max} - \sum_{i=1}^V (W_{Vi} W_{Ki} Z(PODBOR)),$$

где **PODBOR** — вектор подбора; W_{Vi} — вес вакансии в i -м подборе; W_{Ki} — вес кандидата; $F(PODBOR)_{\max}$ — максимальное значение целевой функции,

$$Z(x) = \begin{cases} 1, & \text{если на одну вакансию из } x \leq 1 \text{ кандидатов,} \\ 0, & \text{если на одну вакансию из } x > 1 \text{ кандидатов.} \end{cases}$$

2. **Популяция** — совокупность из нескольких векторов **PODBOR**.

3. **Размер популяции** — общее число элементов в векторах **PODBOR**. Размер популяции задают до начала работы ГА и в течение всего периода работы он остается постоянным.

4. **Демос** — один вектор **PODBOR**. Популяция разделяется на несколько различных демосов (подпопуляций), которые впоследствии развиваются параллельно и независимо. Число демосов равно частному от деления размера популяции на число вакансий. Эта величина также постоянна и задается до начала работы ГА.

5. **Особь** — один элемент из вектора **PODBOR**.

6. **Ген** — элемент K или V из вектора **PODBOR**.

7. **Критерий прекращения работы ГА**. Работа ГА останавливается по истечении времени, отведенного для него, или при генерации заданного числа поколений.

Следующий шаг — это определение характеристик и значений атрибутов (см. табл. 2), от которых зависят весовые коэффициенты кандидата и вакансии. Кроме того, для организации работы ГА необходимо определить используемые генетические операторы.

1. Оператор селекции (*selection*). В рассматриваемом случае используется элитная селекция. В этом случае выбираются лучшие (элитные) элементы на основе сравнения значений целевой функции. Далее они вступают в различные преобразования, после которых снова выбираются элитные элементы. Процесс продолжается аналогично до тех пор, пока продолжают появляться элитные элементы.

2. Оператор кроссинг-овера (*crossing-over*). В данной работе используется двухточечное скрещивание. При этом две особи, выбранные оператором отбора, становятся "родителями" и обмениваются генами для воспроизведения новых особей — двух "потомков". При скрещивании особи обмениваются генами-вакансиями и генами-кандидатами. Учитывается, что весовой коэффициент гена-кандидата должен быть больше или равен весовому коэффициенту гена-вакансии. Оператор действует с вероятностью 50 %.

3. Оператор мутации (*mutation*). В данной работе используется точечная мутация и инверсия. Заменяет ген у двух особей, выбранных оператором отбора, случайным значением с учетом его известной предельной величины. При этом происходит переход мутированных особей в следующее поколение. Учитывается то, что у особи, подвергшейся мутации, весовой коэффициент гена-кандидата должен быть больше или равен весовому коэффициенту гена-вакансии. Этот оператор также действует с вероятностью 50 %.

4. Оператор редукции (*reduction*). Выполняется после действия оператора скрещивания или мутации. Выбирает две лучшие особи из пар родителей и потомков (особей и особей-мутантов), имеющих максимальное произведение весовых коэффициентов, и помещает их в популяцию. При этом две оставшиеся особи в популяцию не попадают. Работа генетического алгоритма определяется следующей последовательностью действий.

1. Создание первой популяции заполнением случайными числами элементов вектора **PODBOR**, с учетом их возможных значений. Также необходимо учитывать, что вес кандидата должен быть больше или равен весу вакансии.

2. Проверка условия останова работы генетического алгоритма. Если критерий выполнен, осуществляется выбор особи, значение целевой функции $F(\text{PODBOR})$ у которой будет максимальным, в противном случае переходим к следующему пункту.

3. Выполнение отбора в популяции в целях выбора пары особей для их последующего скрещивания или мутации.

4. Применение оператора скрещивания или оператора мутации для пар родителей, выбранных оператором отбора, в каждом демосе.

5. Применение оператора редукции для выбора наилучших особей из пар родителей и потомков (особей и особей-мутантов).

6. Проверка условия останова работы генетического алгоритма, если критерий останова не выполнен, то необходимо вернуться к пункту 3, в противном случае перейти к пункту 7.

7. Выбор особи, значение целевой функции $F(\text{PODBOR})$ у которой будет максимальным.

Программная реализация математической модели процесса подбора кандидата на вакансию на основе ГА осуществляется с помощью системы программирования Borland Delphi 7.0 с использованием свободно-распространяемой библиотеки компонентов "GeneBase" [6]. Библиотека компонентов "GeneBase" — мощное средство решения задач многомерной непараметрической оптимизации. В наборе имеется компонент, реализующий генетический алгоритм, который может быть использован для решения задач многомерной непараметрической оптимизации. В частности позволяет находить субоптимальное решение *NP*-полных задач. Компонент позволяет выполнять минимизацию и максимизацию целевой функции.

Разработанная автоматизированная подсистема осуществляет подбор кандидата на вакансию на основе методов и алгоритмов, описанных выше, и функционирует в составе любой единой медицинской информационной системы за счет применения унифицированного формата хранения данных (XML). Данные о вакансиях и кандидатах хранятся в специализированных системах кадрового учета ЛПУ, таких как "1С:Зарплата", "Кадры 7.7 ПРОФ", "ПАРУС — Кадры 7.30", и поступают в автоматизированную подсистему через соответствующие шлюзы, что иллюстрирует схема, представленная на рис. 2.

Правильно построенная автоматизированная система подбора персонала позволяет ЛПУ не только справиться с текущими проблемами, но и решать возможные задачи в долговременной перспективе. Благодаря этому ЛПУ находит сотрудника, наибольшим образом удовлетворяющего всем



Рис. 2. Схема взаимодействия информационных систем

текущим требованиям с перспективой его профессионального (научного) и карьерного роста.

Достоинство ГА заключается в его способности манипулировать одновременно многими параметрами [4, 5]. Это свойство использовано во множестве прикладных программ и может быть успешно применено в решении вопроса подбора персонала на вакансии, поскольку это далеко не простой процесс, зависящий от множества входных параметров.

В рассматриваемом процессе подбора персонала исследуемое пространство достаточно велико, и задача не требует строгого нахождения глобального оптимума, т. е. достаточно за короткое время найти приемлемое, наиболее подходящее решение. В данном случае ГА имеют хорошие шансы стать эффективной процедурой поиска, конкурируя с другими методами, которые не используют знания о пространстве поиска, и даже превосходя их.

Использование информационных технологий на базе ГА в работе кадровых подразделений ЛПУ

позволит значительно улучшить их работу, в частности, сделать более рациональным процесс подбора медицинского персонала на имеющиеся вакантные должности.

Список литературы

1. Ковшов Е. Е., Машковцев А. В. Применение аппарата генетических алгоритмов для моделирования и формирования расписания операций в многопрофильной клинике // Информационные технологии. 2006. № 8. С. 63—66.
2. Мельникова А. В., Ковшов Е. Е. Автоматизация процесса подбора персонала для машиностроительного предприятия // СТИН. 2006. № 9. С. 33—37.
3. Корнеев В. В., Гареев А. Ф., Васютин С. В., Райх В. В. Базы данных. Интеллектуальная обработка информации. М.: Нолидж, 2001. 496 с.
4. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. М.: Горячая линия — Телеком, 2004. 452 с.
5. Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Харьков: ОС-НОВА, 1997. 112 с.
6. <http://www.basegroup.ru/download/genebase.htm>

ОПТИМИЗАЦИЯ

УДК 004.023

А. С. Филиппова, д-р техн. наук, доц.,

В. Ю. Кузнецов, аспирант,

Уфимский государственный авиационный
технический университет

Задачи о минимальном покрытии ортогональных многоугольников с запретными участками

Рассматривается одна из задач покрытия многоугольных областей объектами произвольной формы и ее приложения. Задача является NP-трудной и для ее решения предложен алгоритм с использованием прямоугольной аппроксимации. Проведен численный эксперимент, представлены его результаты с оценкой фактора аппроксимации.

Ключевые слова: методы оптимизации; задачи покрытия; технологии блочных структур; эволюционные алгоритмы; системы автоматизированного проектирования; карта вейпоинтов.

Введение

Задачи покрытия двумерных объектов (2D Bin Cover Problem, 2DBCP) представляют собой важный раздел комбинаторной оптимизации. Их

близость к задачам упаковки (2D Bin Packing Problem 2DBP) позволяет применять к решению 2DBCP и 2DBP сходные подходы. Сложность их решения обусловлена принадлежностью к классу NP-трудных задач. Рассматривается задача покрытия ортогонального многоугольника одинаковыми фигурами произвольной формы с запрещенными участками (дефектами). Многоугольник называется ортогональным *полигоном* P , если все стороны в нем являются горизонтальными или вертикальными. Полигон P простой, если он не имеет запретных участков. Необходимо покрыть ортогональный полигон заданными фигурами так, чтобы их число было минимальным.

Решение должно удовлетворять следующему условию: каждая точка покрываемой области принадлежит одному или нескольким покрывающим объектам. В работе [1] Y. Cheng, S. S. Iyengar и R. L. Kashyap показали, что эта задача имеет практическое применение в процессе сжатия изображений и в печати интегральных схем. Долгое время не была определена вычислительная сложность этой задачи [2]. Впервые W. J. Masek доказал, что задача является NP-полной [3]. Позднее J. C. Culberson и R. A. Reckhow [4] свели ее к более простой, чтобы доказать, что она остается NP-полной даже в случае, если полигон P является простым. P. Berman и B. Dasgupta [5] показали,

что если P имеет дефекты, то проблема является NP-трудной в сильном смысле. Поскольку задача является NP-трудной, то для получения решения, близкого к оптимальному, используют приближенные и эвристические алгоритмы.

Для оценки качества приближенного покрытия используется величина, именуемая "фактором аппроксимации" (Factor Approach, FA) [7]. Для задачи покрытия — это отношение числа покрывающих объектов в допустимом решении к числу покрывающих объектов в оптимальном решении. Задача покрытия прямоугольниками — частный случай приведенной проблемы. Ее FA — $O(\log n)$, где n — минимум из числа вертикальных и горизонтальных ребер многоугольника, стал известен благодаря D. S. Johnson и L. Lovasz [6, 7], использовавшим жадную схему. Таким образом, это значение FA считается лучшим до настоящего времени. Здесь рассматриваются некоторые прикладные задачи покрытия и приведены алгоритмы для их решения.

Задача о минимальном покрытии прямоугольной области с запретными участками

Дана прямоугольная область A ширины W и длины L . Введем прямоугольную систему координат так, чтобы оси Ox и Oy совпали соответственно с нижней и левой сторонами этой области. На области A определено множество Z запретных зон. Зоны заданы прямоугольниками, стороны которых параллельны граням области: $Z = \bigcup_{i=1}^n z_i$, где $z_i = \langle (x_i, y_i), (w_i, l_i) \rangle$ — прямоугольник с номером i , в котором (x_i, y_i) — координаты нижнего левого угла; (w_i, l_i) — ширина и длина прямоугольника.

Заданы круги радиуса r . Требуется покрыть фигуру $A \setminus Z$ наименьшим числом кругов или прямоугольников. Центры кругов должны лежать в области $A \setminus Z$. Пример покрытия изображен на рис. 1.

Приведем некоторые интерпретации этой задачи.

Проектирование технологических установок. Производственные установки по сбору и переработке газа представляют собой совокупность, как правило, несвязанного оборудования, размещенного на некоторой территории (полигоне). Для контроля возможной утечки используются газоанализаторы, их монтируют на территории установок. Они обладают круговой зоной действия. Требуется минимизировать число газоанализаторов при условиях покрытия зонами их действия всего полигона и размещения в обход запрещенных участков.

Размещение базовых станций сотовой связи. Требуется обеспечить сотовой связью определен-

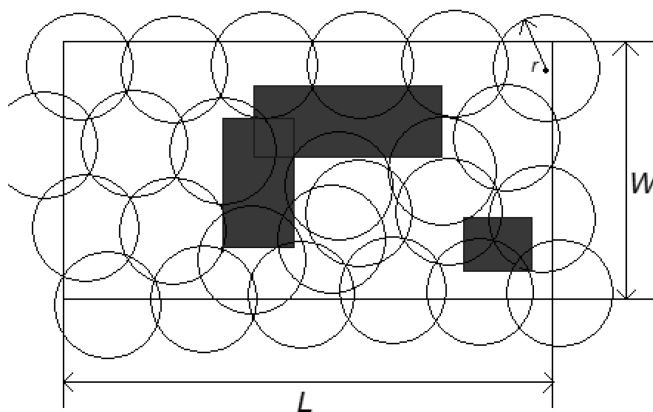


Рис. 1. Покрытие области кругами с запретными зонами

ную область, например город. Сотовая связь доступна внутри "зоны видимости" базовой станции, ее можно моделировать кругом заданного радиуса действия. Требуется разместить наименьшее число базовых станций в обход естественных преград (рек, дорог и т. д.).

Поиск пути для персонажей в компьютерных играх. Имеется трехмерный ландшафт сложной формы. На ландшафте могут размещаться препятствия. По ландшафту перемещаются персонажи, управляемые компьютером (ИИ, искусственный интеллект). Нужно разработать структуру данных и построить алгоритм, которые позволят в реальном времени прокладывать характерный для человека маршрут между любыми двумя точками ландшафта. Есть общепринятый подход к решению задачи поиска пути, основанный на *вейпоинтах* (waypoint). Создание карты вейпоинтов сводится к задаче покрытия произвольной области кругами заданного радиуса: карту можно разбить на зоны так, чтобы произвольная вертикальная прямая пересекала ландшафт не более чем в одной точке, препятствия — запретные зоны, центры кругов — вейпоинты, а ребра графа строятся по принципу — если два круга пересекаются, то между вершинами, образованными этими кругами, строится ребро. Автоматическая генерация карт вейпоинтов значительно ускоряет разработку компьютерных игр.

Передача информации. Квантование данных, полученных от непрерывно изменяющегося источника, тесно связано с задачей о минимальном покрытии пространства перекрывающимися шарами [8]. Например, чтобы выполнять квантование в двумерном пространстве, входные данные объединяют в пары и каждую пару рассматривают как координаты точки на плоскости. Затем каждую точку A , отвечающую паре данных, округляют до ближайшей отмеченной точки (B), лежащей в той же заранее выбранной области плоскости, что и A . Проблема состоит в том, чтобы выбрать

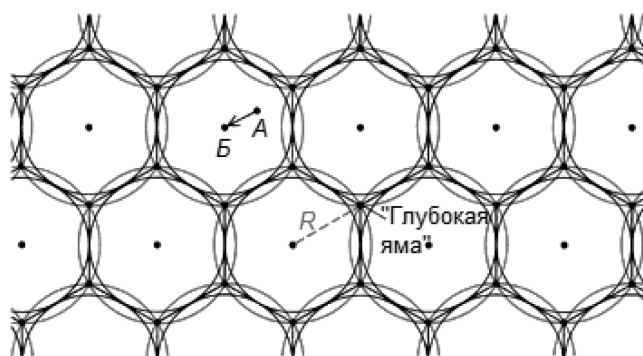


Рис. 2. Применение минимального покрытия плоскости кругами к решению задачи квантования данных

разбиение плоскости и отмеченные точки таким способом, при котором была бы минимальной средняя ошибка квантования. Если данные распределены равномерно, а отмеченные точки лежат в центрах квадратов, то средняя ошибка равна $1/12$. С помощью минимального покрытия плоскости кругами можно получить лучший способ квантования (рис. 2). Такое покрытие порождается гексагональной упаковкой (т. е. упаковкой, когда каждого круга касается еще шесть кругов) при увеличении радиусов кругов настолько, чтобы каждая точка плоскости оказалась внутри или на границе хотя бы одного круга. Радиус R кругов покрытия равен расстоянию от центра до самой "глубокой ямы" в соответствующей упаковке. Если соединить эти "глубокие ямы" подходящими прямыми, то плоскость разобьется на правильные шестиугольники. Выбор отмеченных точек в центрах правильных шестиугольников при квантовании равномерно распределенных данных дает среднюю ошибку $\frac{5\sqrt{3}}{108}$, т. е. около 0,0802, что немало меньше, чем $1/12$.

Алгоритм построения допустимого покрытия

Допустимое решение задачи о минимальном покрытии полигона кругами может быть получено в результате замены кругов квадратами с центрами в $A \setminus Z$. Это решение далее будет использовано в качестве верхней границы функции цели и для оценки некоторых характеристик. Покрытие осуществляется с использованием блок-структур (BS) [9].

Пусть имеется схема частичного покрытия: запретные зоны и часть размещенных квадратов. Через вертикальные стороны фигур проводим прямые. Таким образом, область разбивается на вертикальные блоки. Пример блок-структуры размещения объектов на полигоне ширины $W = 245$ и длины $L = 451$ приведен на рис. 3.

Блок-структура BS — это последовательность блоков различной длины и одинаковой ширины:

$BS = (B_1, B_2, \dots, B_n)$, где B_i — блок с номером i . Блок — совокупность объектов или их фрагментов заданной ширины: $B_i = ((i_1, i_2, \dots, i_j, \dots, i_m)L_i)$, где i_j — j -й элемент i -го блока; L_i — длина блока.

В блоке могут размещаться элементы трех типов: запретные участки (Z), покрывающие прямоугольники (D), и свободные участки (F). При этом j -й элемент задается парой (T_j, W_j) , где T_j — тип элемента (Z , F или D), W_j — ширина j -го элемента.

Блок-структура, представленная на рис. 3, имеет вид: $BS = ((D, 45), (F, 200)), 62; ((F, 162), (Z, 51), (F, 32)), 42; ((F, 77), (Z, 70), (F, 15), (Z, 51), (F, 32)), 57; ((F, 77), (Z, 70), (F, 98)), 62; ((F, 77), (Z, 70), (F, 9), (Z, 80), (F, 9)), 57; ((F, 156), (Z, 80), (F, 9)), 105; ((F, 245)), 66$.

По блок-структуре можно вычислить координаты любого элемента. Пусть требуемый элемент находится в i -м блоке в позиции j , тогда:

$$x = \sum_{k=1}^i l_k, y = \sum_{k=1}^j w_k, \quad (1)$$

где l_k — длина k -го блока; w_k — ширина k -го элемента в i -м блоке.

Алгоритм покрытия

Исходные данные: W — ширина покрываемой области; L — длина покрываемой области, R — радиус покрывающих кругов, запретные участки Z_i .

1. *Инициализация.* Блок-структура заполняется единственным блоком ширины W . В блок вставляется единственный пустой (F) элемент длины L .

2. *Размещение запретных участков.* Для каждого запретного участка к блок-структуре применяется процедура:

Вставка Прямоугольника с параметрами: тип = Z , позиция = (X_z, Y_z) , ширина = w_z , длина = l_z , где (X_z, Y_z) — координаты нижнего левого угла очередного запретного участка, w_z — ширина, а l_z — длина очередного запретного участка.

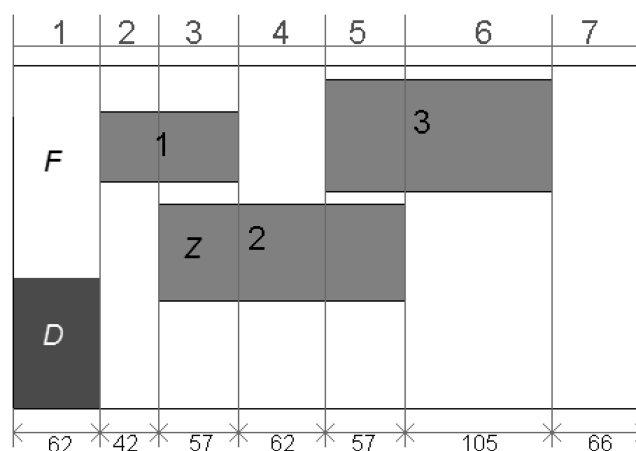


Рис. 3. Блок-структура частичного покрытия

3. *Вычисление стороны вписанного квадрата* $a = R\sqrt{2}$.

4. Если в блок-структуре нет F -элементов, то перейти к п. 8.

5. *Определение координат.* Получить по формуле (1) координаты (X_f, Y_f) F -элемента.

6. *Вставка.* Применить к блок-структуре процедуру **Вставка_Прямоугольника** с параметрами тип = D , позиция = (X_f, Y_f) , ширина = длина = a .

7. Перейти к п. 4.

8. *Построение кругов.* Для каждого квадрата S_i построить круг радиуса R с координатами центра $O_i = (X_i + \frac{\sqrt{2}}{2} R, Y_i + \frac{\sqrt{2}}{2} R)$, где X_i, Y_i — координаты левого нижнего угла квадрата S_i .

9. *Конец алгоритма.*

Процедура "Вставка_Прямоугольника"

Параметры: (x, y) — координаты вставляемого прямоугольника, который имеет тип t ; (w, l) — ширина и длина вставляемого прямоугольника.

1. *Поиск блока со свободной областью.* Пусть x_i — x -координата блока i . Ищем блок i такой, что $x_i \leq x$, а $x_{i+1} > x$; $iBegInd = i$. $iBegInd$ содержит номер первого блока, начиная с которого модифицируется блок-структура.

2. Если $x_i < x$, то разбиваем блок i на два блока, с ширинами $(x - x_i)$ и $(w_i - x + x_i)$, где w_i — ширина i -го блока; $iBegInd = i + 1$.

3. Ищем блок j такой, что $x_j \leq x + w \leq x_{j+1}$; $iEndInd = j$. $iEndInd$ — номер последнего модифицируемого блока в блок-структуре.

4. Если $x_j < x + w$, то разбиваем j на два блока ширины $(x + w - x_j)$ и $(w_j - x - w + x_j)$.

5. Для i от $iBegInd$ до $iEndInd$: применяем к блоку i процедуру **Вставка_Элемента** с параметрами: тип = t , позиция = y , длина = l .

6. *Конец процедуры.* Блок-структура модифицирована.

Процедура "Вставка_Элемента"

Параметры: вставляемый элемент типа $t = D$ или $t = Z$. Таким образом, вставлять в блок-структуру можно либо покрывающие квадраты, либо запретные участки. Позиция вставки — y , длина элемента — l .

1. Ищем элемент i такой, что $y_i \leq y < y_{i+1}$.

2. Если $y_i \neq y$, то разбиваем i -й элемент на два с длинами $(y - y_i)$ и $(y_i + l_i - y)$, $i = i + 1$.

3. Ищем элемент j такой, что $y_j \leq y + l < y_{j+1}$.

4. Если $y_j \neq y + l$, то разбиваем j -й элемент на два с длинами $(y + l - y_j)$ и $(y_j + l_j - y - l)$.

5. Для всех k от i до j : $t(k)$ — тип k -го элемента. Если $t(k) \neq Z$, то изменяем тип k -го элемента на t .

6. Для всех i от 1 до n (где n — число элементов в блоке), если $t(i) = t(i + 1)$, то объединить i -й и $(i + 1)$ -й элементы.

7. *Конец процедуры.* Блок модифицирован.

Численный эксперимент

Алгоритм был испытан на различных случайных задачах. Каждая задача описывается набором параметров. Обозначим параметры задачи через $P_i, i = 1, n$, где n — число параметров, P_i — множество возможных значений i -го параметра; $\mathbf{P} = P_1 \times P_2 \times \dots \times P_n$ — пространство задач. Вектор $p \in \mathbf{P}$ — индивидуальная задача. Число индивидуальных задач бесконечно, а число измерений пространства $\mathbf{P}$ — велико, поэтому на практике точно определить качество работы алгоритма невозможно. Для идентификации эффективности алгоритма, выполним следующие действия.

- выберем небольшое подмножество $\mathbf{PP} \subset \mathbf{P}$ существенных параметров задачи;
- каждый вектор $p \in \mathbf{PP}$ будем рассматривать как проекцию множества сходных задач из $\mathbf{P}$ на пространство $\mathbf{PP}$;
- будем вычислять значения характеристик не во всех векторах $\mathbf{PP}$, а только в узлах решетки, равномерно покрывающей $\mathbf{PP}$.

Были выбраны следующие параметры: длина и ширина покрываемой области; минимальные длина и ширина запретных участков; максимальные длина и ширина запретных участков; радиус покрывающих кругов (измеряется в % от максимального из размеров покрываемой области); заполненность $F = S_z/S_A$, где S_z — площадь фигуры

Z ($Z = \bigcup_{i=1}^k Z_i$, где k — число запретных участков,

Z_i — i -й запретный участок), S_A — площадь покрываемой фигуры A .

Для оценки качества работы алгоритма используется оценка фактора аппроксимации

$FA = \frac{N}{N_{opt}}$, где N — число кругов в покрытии,

N_{opt} — число кругов в оптимальном покрытии.

Фактор аппроксимации сложно вычислить, поскольку для вычисления требуется знать точное решение задачи. Однако нижнюю оценку FA легко получить, заменяя N_{opt} оценкой N'_{opt} минимального числа кругов, $N'_{opt} = \left\lceil \frac{S_A K}{S} \right\rceil$, где S_A —

площадь фигуры $A \setminus Z$, $K = 1,22$ — коэффициент, учитывающий перекрытие кругов в гексагональном покрытии, S — площадь круга. Число тестов на каждый вектор p — 100.

Результаты эксперимента, приведенные в таблице, позволяют сформулировать следующие выводы:

- фактор аппроксимации адекватен качеству полученного покрытия;
- с ростом заполненности возрастает фактор аппроксимации. Это может быть вызвано как не-

Результаты численного эксперимента

Покрываемая область		Запретные участки, %				Радиус кругов, %	Заполненность, %	Фактор аппроксимации
		Минимум		Максимум				
Длина	Ширина	Длина	Ширина	Длина	Ширина			
50	50	1	1	50	50	10	10	1,72
50	50	1	1	50	50	50	10	2
50	50	1	1	50	50	10	70	2,84
50	50	1	1	50	50	50	70	3,79
50	1000	1	1	50	50	10	10	4
50	1000	1	1	50	50	50	10	2
50	1000	1	1	50	50	10	70	8
50	1000	1	1	50	50	50	70	2
1000	50	1	1	50	50	10	10	4
1000	50	1	1	50	50	50	10	2
1000	50	1	1	50	50	10	70	8
1000	50	1	1	50	50	50	70	2
1000	1000	1	1	50	50	10	10	1,7
1000	1000	1	1	50	50	50	10	2
1000	1000	1	1	50	50	10	70	2,75
1000	1000	1	1	50	50	50	70	3,79

точностью оценки фактора аппроксимации, так и ограничениями блочной технологии;

- если радиус кругов намного меньше покрываемой области, то алгоритм находит решения с фактором аппроксимации, близким к 2. Потери в качестве решения вызваны переходом к покрытию квадратами.

Заключение

В статье описана постановка задачи покрытия полигона с запретами одинаковыми кругами (или квадратами). К ее решению сводятся прикладные ситуации, некоторые из них приведены в статье. Предложен и реализован метод покрытия с обходом препятствий, основанный на технологии блочных структур. Экспериментально показано, что полученные решения оказываются хуже нижней границы не более чем в 3 раза. Учитывая, что нижняя граница является грубой, реальная оценка должна быть лучше.

Намечены пути повышения эффективности метода: разработка и включение в алгоритм процедуры сдвига фигур с ориентацией на их вытал-

кивание из зоны покрытия; разработка блочной схемы покрытия области кругами; распараллеливание некоторых процедур, например, вставки фигур в блок-структуру.

Список литературы

1. **Cheng Y., Iyegnar S. S. and Kashyap R. L.** A New Method for Image compression using Irreducible Covers of Maximal Rectangles // IEEE Transactions on Software Engineering. 1988. Vol. 14. N 5. P. 651—658.
2. **Hegedüs A.** Algorithms for Covering Polygons with Rectangles // Computer Aided Geom. Design. 1982. N 14. P. 257—260.
3. **Masek W. J.** Some NP-Complete Set Covering Problems. Manuscript, MIT, 1979.
4. **Culberson J. C. and Reckhow R. A.** Covering Polygons is Hard // Journal of Algorithms. 1994. N 17. P. 2—44.
5. **Berman P. and Dasgupta B.** Approximating Rectilinear Polygon Cover Problems // Algorithmica. 1997. N 17 (4). P. 331—356.
6. **Johnson D. S.** Approximation Algorithms for Combinatorial Problems // Journal of Computing and Systems Sciences. 1974. N 9. P. 256—278.
7. **Lovasz L.** On the Ratio of Optimal Integral and Fractional Covers // Journal of Discrete Mathematics. 1975. N 13. P. 383—390.
8. **Слоэн Н. Дж. А.** Упаковка шаров, <http://ega-math.narod.ru/Nquant/spheres.htm>
9. **Филиппова А. С.** Моделирование эволюционных алгоритмов решения задач прямоугольной упаковки на базе технологии блочных структур // Информационные технологии. 2006. № 6. Приложение. 32 с.

Ю. В. Орехов, канд. техн. наук, доц.,
Э. Ю. Орехов, канд. физ.-мат. наук, доц.,
А. Р. Заминова,
 Уфимский государственный авиационный
 технический университет

Исследование эффективности операторов генетического алгоритма при поиске экстремальных решений в бинарных и вещественных пространствах

Рассматриваются операторы кроссовера и мутации генетического алгоритма, применяемые для поиска в бинарных и вещественных пространствах решений. Составлена многоэкстремальная целевая функция, позволяющая провести сравнение эффективности полученных конструкций генетического алгоритма в пространстве произвольной размерности. Приведенные числовые результаты подтверждают эффективность рассмотренных операторов кроссовера и мутации и их сочетаний.

Ключевые слова: генетический алгоритм, вещественное и бинарное кодирование, операторы кроссовера и мутации.

Введение

Генетический алгоритм (ГА) представляет собой метаэвристику, эффективно применяемую для решения NP-полных и NP-трудных задач.

Применение ГА для решения задачи оптимизации можно представить следующей схемой [1, 2].

1. Предварительный этап:

1.1) выбор представления пространства решения (способ кодирования);

1.2) задание целевой функции — функции оценки допустимости решения;

1.3) выбор начального подмножества решений.

2. Этап конструирования решений с использованием операторов кроссовера (ОК).

3. Этап случайного изменения решения с использованием операторов мутации (ОМ).

4. Этап отбора полученных решений в следующую выборку-поколение с использованием оператора селекции.

Конструирование конкретного типа ГА осуществляется выбором одного из возможных вариантов соответствующего оператора на каждом шаге выполнения ГА.

В данной статье проводится сравнение эффективности конструкций ГА, получаемых различ-

ными сочетаниями способов реализации первых трех этапов алгоритма. На четвертом этапе в качестве оператора селекции во всех конструкциях использовался один из простейших методов — селекция на основе заданной шкалы [1].

Задача нахождения экстремума функции нескольких переменных предполагает существование ограниченного пространства решений и заданной на этом пространстве целевой функции.

Традиционный ГА (представленный впервые в работах Д. Холланда [4]) предполагает кодирование значений аргументов целевой функции в виде бинарной строки, называемой "хромосомой". В этом случае операторы ГА обращаются и работают с каждым битом, из которых состоит текущее решение — хромосома. С развитием аппарата ГА эти алгоритмы стали все чаще использоваться и в более сложных специальных задачах, например, в задачах поиска оптимальных маршрутов, инженерного проектирования, построения систем управления, обучения нейронных сетей, в задачах раскроя и упаковки и многих других [3, 5—7]. Для подобных задач элементарной составляющей (геном) закодированного решения (хромосомы) удобнее считать уже не бит — единицу электронной информации, а некоторую элементарную структуру, соответствующую условиям задачи. В частности, для задач проектирования в качестве хромосомы во многих случаях выбирается допустимая схема размещения объектов [5, 6]. В задачах поиска оптимального алгоритма упаковки ГА используется для поиска лучшего решения в пространстве кодированных схем упаковок. Соответственно, в данном случае вид хромосомы определяется выбранным способом кодирования, например, искомое решение может представлять последовательность прямоугольников [7] или последовательность блоков, составленных из упаковываемых прямоугольников [3].

Для решения значительного числа оптимизационных задач, в частности, для задач компоновки, кластеризации, маршрутизации, размещения и синтеза расписаний, предложен метод представления решений-хромосом в виде последовательности используемых комбинаций эвристик [2].

Таким образом, выбор типа возможных решений определяет набор подходящих операторов ГА, которые могут быть использованы для поиска в данном пространстве решений.

1. Операторы генетического алгоритма

В данной статье рассмотрены и реализованы основные операторы для каждого из рассмотренных типов кодирования.

1.1. Операторы ГА, работающие в пространстве решений с бинарным кодированием [1].

Пусть $C_1 = (c_1^1, \dots, c_n^1)$ и $C_2 = (c_1^2, \dots, c_n^2)$ — две родительские хромосомы, выбранные для применения оператора кроссовера, где c_j^i — j -й бит закодированной i -й хромосомы; $C = (c_1, \dots, c_n)$ — хромосома, отобранная для проведения мутации, где c_j — j -й бит выбранной хромосомы.

Оператор одноточечного кроссовера. В результате применения данного оператора кроссовера к хромосомам C_1 и C_2 создаются два потомка вида

$C'_1 = (c_1^1, c_2^1, \dots, c_p^1, c_{p+1}^2, \dots, c_n^2)$ и $C'_2 = (c_1^2, c_2^2, \dots, c_p^2, c_{p+1}^1, \dots, c_n^1)$, где p — реализация дискретной случайной величины с набором равновероятных возможных значений $\{1, \dots, n-1\}$.

Оператор двухточечного кроссовера. В результате применения данного оператора кроссовера к хромосомам C_1 и C_2 создаются два потомка вида

$C'_1 = (c_1^1, c_2^1, \dots, c_{p_1}^1, c_{p_1+1}^2, \dots, c_{p_2}^2, c_{p_2+1}^1, \dots, c_n^1)$ и $C'_2 = (c_1^2, c_2^2, \dots, c_{p_1}^2, c_{p_1+1}^1, \dots, c_{p_2}^1, c_{p_2+1}^2, \dots, c_n^2)$, где p_1 — реализация дискретной случайной величины с набором равновероятных возможных значений $\{1, \dots, n-2\}$, а p_2 — реализация дискретной случайной величины с набором равновероятных возможных значений $\{p_1+1, \dots, n-1\}$.

Оператор одноточечной мутации. В результате применения данного оператора мутации к хромосоме C создается потомок вида $C' = (c_1, c_2, \dots, \bar{c}_p, \dots, c_n)$, где $\bar{c}_p$ — инвертируемый бит с номером p , представляющим собой реализацию дискретной случайной величины с набором равновероятных возможных значений $\{1, \dots, n\}$.

Оператор двухточечной мутации. В результате применения данного оператора мутации к хромосоме C создается потомок вида $C' = (c_1, c_2, \dots, c_{p_2+1}, c_{p_1}, \dots, c_{p_2}, c_{p_1-1}, c_n)$, где p_1 — реализация дискретной случайной величины с набором равновероятных возможных значений $\{2, \dots, n-2\}$, а p_2 — реализация дискретной случайной величины с набором равновероятных возможных значений $\{p_1+1, \dots, n-1\}$.

1.2. Операторы ГА, работающие в пространстве решений с численным кодированием

Рассмотрим классификацию операторов кроссовера для вещественного кодирования решений, представленную в [8].

- Дискретные операторы кроссовера представляют собой операторы, конструирующие хромосомы-потомки таким образом, что каждый из генов совпадает с некоторым геном хромосомы-родителя, новые гены не конструируются. К данной группе относятся операторы кроссовера для бинарного кодирования, применяе-

мые к вещественным числам, а также оператор равномерного перемешивания генов.

- Конструирующие операторы кроссовера объединяют операторы, использующие конструирующую функцию, которая численно комбинирует значения генов родителей для генерации значений генов потомков. В качестве примеров операторов данной группы можно привести арифметический и геометрический операторы кроссовера.
- Операторы кроссовера "в окрестности" включают в себя операторы кроссовера, которые определяют гены потомка, выбирая их значения из некоторого интервала, определяемого областью, задаваемой генами родителей в соответствии с заданным распределением вероятностей. В качестве примера могут быть приведены кроссовер BLX- α и кроссовер SBX.

Пусть $C_1 = (c_1^1, \dots, c_n^1)$ и $C_2 = (c_1^2, \dots, c_n^2)$ — две родительские хромосомы, выбранные для применения оператора кроссовера, где c_j^i — j -й параметр i -й хромосомы, заданный вещественным числом; $C = (c_1, \dots, c_n)$ — хромосома, отобранная для проведения мутации, где c_j — j -й параметр выбранной хромосомы, заданный вещественным числом.

Оператор арифметического кроссовера. В результате применения данного оператора кроссовера к хромосомам C_1 и C_2 создаются два потомка вида $H_k = (h_1^k, \dots, h_n^k)$, $k = 1, 2$, где $h_i^1 = \lambda c_i^1 + (1 - \lambda)c_i^2$ и $h_i^2 = \lambda c_i^2 + (1 - \lambda)c_i^1$, $j = \overline{1, n}$, $\lambda \in [0, 1]$ — параметр, задаваемый пользователем.

Оператор BLX-кроссовера. В результате применения данного оператора кроссовера к хромосомам C_1 и C_2 создаются два потомка вида $H_k = (h_1^k, \dots, h_n^k)$, $k = 1, 2$, где $h_j^k, j = \overline{1, n}$, — реализация случайной величины, подчиняющейся равномерному закону распределения на интервале $[C_{\min} - I\alpha, C_{\max} + I\alpha]$. Здесь

$$C_{\max} = \max\{c_j^1, c_j^2\};$$

$$C_{\min} = \min\{c_j^1, c_j^2\};$$

$$I = C_{\max} - C_{\min};$$

α — параметр, задаваемый пользователем.

Оператор равномерного кроссовера. В результате применения данного оператора кроссовера к хромосомам C_1 и C_2 создаются два потомка вида

$$H_k = (h_1^k, \dots, h_n^k), k = 1, 2,$$

где

$$h_j^k = \begin{cases} c_j^1, & u \geq 0,5; \\ c_j^2, & u < 0,5; j = \overline{1, n}; \end{cases}$$

u — реализация случайной величины, подчиняющейся равномерному закону распределения на интервале $[0, 1]$.

Оператор равномерной мутации [9]. В результате применения данного оператора мутации к хромосоме C создается потомок вида $C' = (c'_1, \dots, c'_n)$, где c'_i — реализация случайной величины, подчиненной равномерному закону распределения в заданной области допустимых значений параметра c_j .

Оператор неравномерной мутации [9]. В результате применения данного оператора мутации к хромосоме C создается потомок вида $C' = (c'_1, \dots, c'_n)$, где

$$c'_i = \begin{cases} c_i + \Delta(t, b_i - c_i), \tau = 0; \\ c_i - \Delta(t, c_i - a_i), \tau = 1; \end{cases}$$

τ — реализация дискретной случайной величины, принимающей с равными вероятностями значения 0 или 1;

$$\Delta(t, y) = y \left(1 - r \left(1 - \frac{t}{g_{\max}} \right)^b \right);$$

r — реализация случайной величины, подчиненной равномерному закону распределения на отрезке $[0, 1]$; b — параметр, задаваемый пользователем.

Операторы ГА, работающие с численным кодированием решений, зависят от выбора значения соответствующего им параметра. Значения параметров были подобраны экспериментально — проводился перебор в заданном диапазоне для минимизации получаемого значения целевой функции за заданное число итераций ГА.

2. Экспериментальная оценка эффективности операторов ГА

Целью данного исследования было определение пары операторов кроссовера и мутации, которые бы наилучшим образом решали задачу поиска минимума тестовой функции. Следовательно, можно выделить следующие подзадачи.

1. Определение критериев выбора лучшей комбинации операторов кроссовера и мутации.
2. Выбор целевой функции, позволяющей провести сравнение сконструированных алгоритмов.
3. Подбор лучших значений параметров в используемых операторах.
4. Тестирование сконструированных и настроенных алгоритмов.

Под *точностью* алгоритма будем понимать близость найденного значения целевой функции к значению ее глобального оптимума. Под *приемлемой точностью* найденного значения будем понимать принадлежность точки, в которой оно

достигается, некоторой достаточно малой окрестности точки глобального оптимума. В качестве критериев сравнения алгоритмов использовались критерии эффективности и устойчивости. Под *эффективностью* будем понимать лучшее (в данном случае минимальное) значение целевой функции, которое может быть достигнуто за фиксированное число запусков алгоритма. Под *устойчивостью* будем понимать среднее лучших значений, достигаемых за фиксированное число запусков алгоритма. Таким образом, понятие эффективности отражает потенциальные возможности алгоритма и используемых в нем операторов, а понятие устойчивости — средний достигаемый результат.

2.1. Выбор функции для тестирования

Для тестирования построенных алгоритмов была выбрана функция, удовлетворяющая следующим условиям:

- она имеет несколько (больше 1) хорошо выраженных оптимумов;

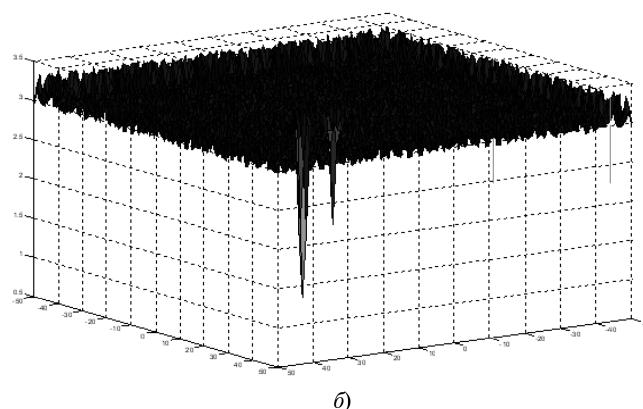
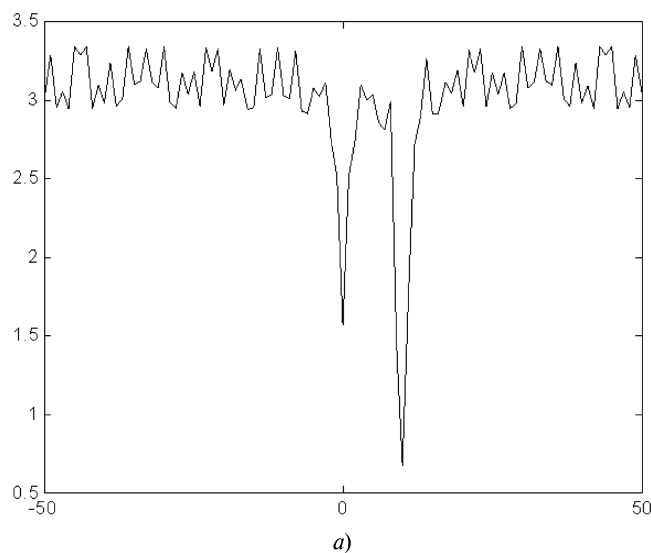


Рис. 1. Графики целевой функции в 2D- (а) и 3D-пространствах (б)

- значения функции в остальных точках незначительно колеблются в некотором небольшом диапазоне возможных значений функции;
- рассматриваемый диапазон значений аргументов функции превышает допустимый для нахождения решения любыми точными методами.

Примером функции, удовлетворяющей перечисленным выше условиям, может служить минимизируемая функция:

$$f(x_1, x_2, \dots, x_n) = A \left| \arctg \alpha \left(\sum_{i=1}^n x_i^2 \right) \right| + \\ + B \left| \arctg \beta \left(\sum_{i=1}^n (x_i - a_i)^2 - c \right) \right| + C \sin \left(\sum_{i=1}^n x_i^2 \right),$$

где A, B, C, a, b, α и β — некоторые заданные коэффициенты функции; $x_1, x_2, \dots, x_n$ — переменные, аргументы функции. Пусть $A = 1, B = 1, C = 0,2, a_1 = 10, a_2 = 2, a_3 = 1, a_4 = 4, a_5 = 1, c = 1, \alpha = 1, \beta = 1$ и рассматриваемое пространство поиска — ограничено: $x_i \in [-50, 50], i = \overline{1, n}$.

В двумерном и трехмерном пространствах функция с данными значениями параметров может быть представлена графиками (рис. 1).

Очевидно, что поиск глобального минимума данной функции не может быть осуществлен методом полного перебора вследствие больших диапазонов значений аргументов функции. А множество локальных экстремумов функции не позволяет использовать для поиска простейшие точные методы, как, например, метод градиентного спуска.

2.2. Подбор параметров в используемых операторах ГА

Для операторов ГА, содержащих свободные параметры, была проведена серия экспериментов по поиску оптимальных значений параметров. На графиках рис. 2 представлены зависимости от значения изменяемого параметра значения среднего оптимума целевой функции — оптимума, усредненного по заданному числу расчетов (100), и лучшего результата — лучшего полученного значения целевой функции по всем проводимым расчетам.

Зависимости достигаемых значений оптимумов целевой функции от значения параметра λ в формуле для оператора арифметического кроссовера представлены на рис. 2, а, от значения параметра α в формуле для оператора BLX-кроссовера — на рис. 2, б и от значения параметра b в формуле для оператора неравномерной мутации — на рис. 2, в.

В результате данных экспериментов были определены оптимальные значения параметров ка-

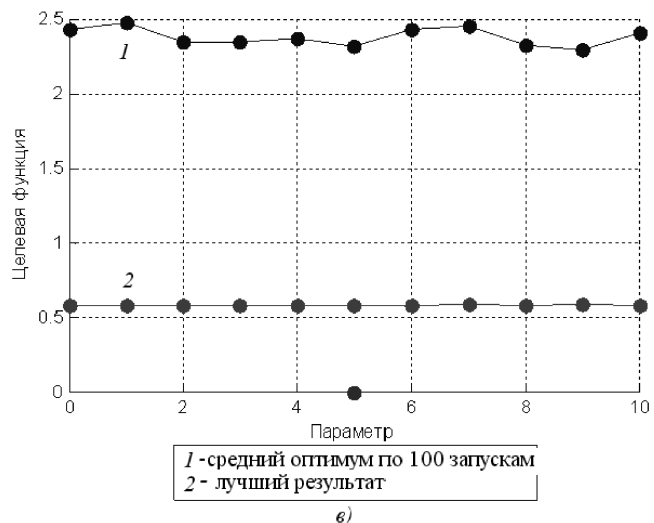
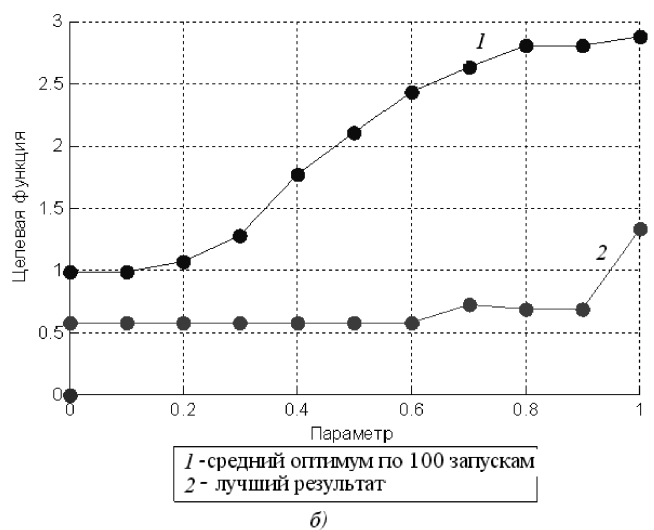
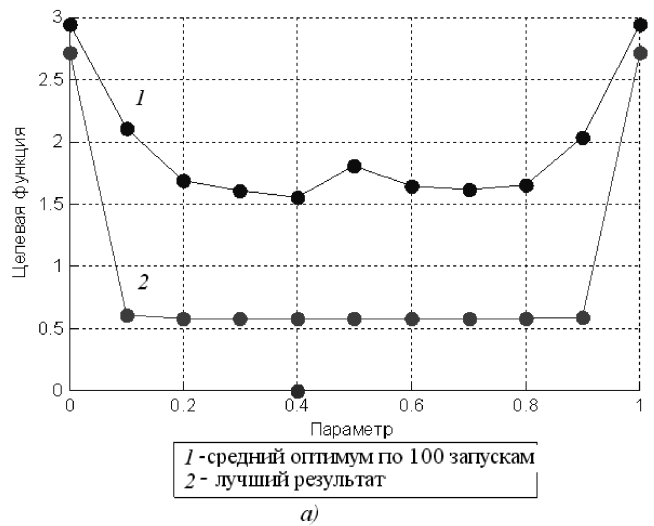


Рис. 2. Подбор оптимального значения параметра для: а — арифметического кроссовера; б — BLX-кроссовера; в — неравномерной мутации

жного оператора в заданных условиях: $\lambda = 0,4$ для арифметического кроссовера, $\alpha = 0,01$ для BLX-кроссовера (использовалось дополнительное укрупнение поисковой сетки) и $b = 5$ для неравномерной мутации. Далее полученные значения параметров использовались при сравнении эффективности альтернативных операторов ГА.

2.3. Результаты тестирования

Для каждого сконструированного типа ГА (сочетание выбранного оператора кроссовера и выбранного оператора мутации) проводилось фиксированное число итераций (50) на поколении фиксированного размера в 50 особей. Усреднение результатов для каждого типа ГА проводилось по 100 измерениям, в результате которых были зафиксированы среднее по лучшим значениям целевой функции популяции, лучшее достигнутое значение за все измерения и среднее время, которое занимали вычисления.

Получаемое лучшее решение позволяет судить об эффективности алгоритма, а среднее лучших решений по проводимым экспериментам дает представление об устойчивости алгоритма — способности воспроизводить свой лучший результат. Ниже будет приведен пример оценки устойчивости для ГА с оператором арифметического кроссовера и оператором равномерной мутации.

Тестирование проводилось для всех возможных комбинаций реализованных ОК и ОМ внутри множества операторов для каждого способа кодирования. Сравнение бинарных и численных операторов кроссовера можно считать корректным с точки зрения объема используемой памяти, так как во всех случаях значение каждого гена представлено 32 битами. Расчет алгоритмом полного перебора с фиксированным шагом по каждому аргументу (0,05) позволяет оценить сложность каждого из рассматриваемых вариантов целевой

Таблица 1

Тестирование операторов ГА для функции одного аргумента

Тип ОК	Тип ОМ	t , с	$\bar{f}_{\text{средн}}$	$f_{\text{лучш}}$	x_1
Бинарное кодирование					
ОК	ОМ	0,8500	1,549914058	0,579278708	9,9404
ОК	ДМ	0,8036	1,678327294	0,576956570	9,9457
ДК	ОМ	0,8088	1,570354579	0,648214102	9,6344
ДК	ДМ	0,8030	1,655328637	0,582810938	9,9609
Численное кодирование					
АК	РМ	0,7729	0,782961561	0,576638162	9,9485
АК	НМ	0,7708	0,730728591	0,576638162	9,9485
BLX	РМ	0,8009	1,686540947	0,609887660	10,0496
BLX	НМ	0,7949	1,663641625	0,727245510	10,3593
РК	РМ	0,8391	1,984879106	0,807349205	10,2039
РК	НМ	0,8351	1,705339165	0,609163105	10,0698
Полный перебор		0,0000	0,576728463		9,9500

функции с точки зрения возможности отыскания ее минимального значения.

Результаты поставленных экспериментов приведены в табл. 1—5.

2.3.1. Результаты для функции от одного аргумента (табл. 1)

$$f(x_1) = \arctg \alpha(x_1^2) + \arctg((x_1 - 10)^2 - 1) + 0,2 \sin(x_1^2), x_i \in [-50; 50], i = 1.$$

Обозначим: ОК — однотоочный кроссовер, ДК — двухточечный кроссовер, ОМ — однотоочная мутация, ДМ — двухточечная мутация, АК — арифметический кроссовер, BLX — BLX-кроссовер, РК — равномерный кроссовер, РМ — равномерная мутация, НМ — неравномерная мутация.

Очевидно, что в данном случае небольшой размер области определения функции позволяет находить хорошее решение, используя точные методы. Использование же эвристик, в частности, генетического алгоритма, для подобной задачи нецелесообразно.

2.3.2. Результаты для функции от двух аргументов (табл. 2)

Таблица 2

Тестирование операторов ГА для функции двух аргументов

Тип ОК	Тип ОМ	t , с	$\bar{f}_{\text{средн}}$	$f_{\text{лучш}}$	x_1	x_2
Бинарное кодирование						
ОК	ОМ	0,8231	2,610613955	0,620247245	9,9525	2,1606
ОК	ДМ	0,8252	2,734379517	0,757412553	10,2875	1,8707
ДК	ОМ	0,8317	2,619316043	0,596820414	9,9825	2,0053
ДК	ДМ	0,7948	2,843439878	0,748110235	9,8911	2,3666
Численное кодирование						
АК	РМ	0,8223	1,804670828	0,575983226	10,0102	2,0079
АК	НМ	0,8100	1,718305320	0,579055905	10,0247	1,9259
BLX	РМ	0,8013	1,115427526	0,577723920	10,0589	2,0118
BLX	НМ	0,8431	1,440840269	0,575967491	10,0118	2,0013
РК	РМ	0,8263	1,830175130	0,632914364	9,9841	2,2477
РК	НМ	0,8156	1,762385825	0,577888966	10,0592	2,0194
Полный перебор		24,5160	0,577750146		10,0600	2,0100

$$f(x_1, x_2) = \arctg\alpha(x_1^2 + x_2^2) + \arctg((x_1 - 10)^2 + (x_2 - 2)^2 - 1) + 0,2\sin(x_1^2 + x_2^2),$$

$$x_i \in [-50; 50], i = \overline{1, 2}.$$

Увеличение числа узлов в поисковом пространстве привело к тому, что уже для двух аргументов поиск минимума данной функции при заданных ограничениях становится весьма трудоемким, что отражается во временных затратах. Большинство реализованных модификаций ГА показали приемлемые результаты при решении задачи поиска минимума. Значение показателя, отражающего устойчивость алгоритма (пятый столбец табл. 2), в целом выше для алгоритмов с вещественным типом кодирования, чем для алгоритмов с бинарным кодированием.

2.3.3. Результаты для функции от трех аргументов (табл. 3)

$$f(x_1, x_2, x_3) = \arctg\alpha(x_1^2 + x_2^2 + x_3^2) + \arctg((x_1 - 10)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 - 1) + 0,2\sin(x_1^2 + x_2^2 + x_3^2), x_i \in [-50; 50], i = \overline{1, 3}.$$

Размерность поискового пространства не позволяет использовать точные методы поиска. Все реализованные модификации ГА дают возможность получить приемлемые решения. Наиболее устойчивый результат обеспечивает применение арифметического кроссовера.

2.3.4. Результаты для функции от четырех аргументов (с размером популяции 70 особей) (табл. 4)

$$f(x_1, x_2, x_3, x_4) = \arctg\alpha(x_1^2 + x_2^2 + x_3^2 + x_4^2) + \arctg((x_1 - 10)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 - 1) + 0,2\sin(x_1^2 + x_2^2 + x_3^2 + x_4^2),$$

$$x_i \in [-50; 50], i = \overline{1, 4}.$$

При анализе функции четырех параметров с заданными ограничениями размер исследуемых популяций был увеличен до 70 особей в связи с увеличением числа точек исследуемого пространства. Для задачи поиска оптимума функции четырех параметров использование операторов кроссовера и мутации, применяющихся для бинарного кодирования, и операторов вещественного кодирования, работающих аналогично бинарным, в

Таблица 3

Тестирование операторов ГА для функции трех аргументов

Тип ОК	Тип ОМ	t, с	$\bar{f}_{\text{средн}}$	$f_{\text{лучш}}$	x_1	x_2	x_3
Бинарное кодирование							
ОК	ОМ	0,8714	2,720457703	0,605691612	10,0677	1,8311	0,9003
ОК	ДМ	0,8939	2,760866638	0,593895137	9,9885	2,0142	1,1871
ДК	ОМ	0,8839	2,737863098	0,597377896	9,9604	2,1460	1,0895
ДК	ДМ	0,9003	2,746345153	0,576541841	10,0136	1,9757	1,0132
Численное кодирование							
АК	РМ	0,8553	1,197158863	0,575966001	10,0114	2,0023	1,0012
АК	НМ	0,8580	1,328149690	0,575967729	10,0116	2,0018	1,0018
BLX	РМ	0,8332	1,470921751	0,575967968	10,0111	2,0037	1,0023
BLX	НМ	0,8417	2,125421858	0,575984657	10,0129	1,9970	0,9989
РК	РМ	0,8919	2,783566832	0,664298058	10,2466	2,1989	1,0544
РК	НМ	0,8853	2,861679318	0,633247018	9,9966	1,9074	1,3153

Таблица 4

Тестирование операторов ГА для функции четырех аргументов

Тип ОК	Тип ОМ	t, с	$\bar{f}_{\text{средн}}$	$f_{\text{лучш}}$	x_1	x_2	x_3	x_4
Бинарное кодирование								
ОК	ОМ	1,4492	2,911415035	0,933744550	9,3353	1,8555	1,0048	3,7323
ОК	ДМ	1,4325	2,906098873	0,830870926	10,168	1,9196	1,5808	3,8025
ДК	ОМ	1,4439	2,934946223	1,652047992	0,0427	-0,0714	0,0427	-0,0823
ДК	ДМ	1,4418	2,936462000	1,755180000	0,0977	0,1222	-0,192	0,3174
Численное кодирование								
АК	РМ	1,4217	2,522241918	0,616829574	10,137	1,9816	0,7827	4,0976
АК	НМ	1,4243	2,751386352	0,606008410	10,077	2,1553	1,0888	4,0523
BLX	РМ	1,4010	2,665949647	0,600986660	9,8586	1,9102	1,1161	3,9496
BLX	НМ	1,4019	2,903925716	1,260281920	9,9308	2,4720	1,5821	3,8511
РК	РМ	1,4598	2,941569758	2,203372240	10,054	3,1419	0,2579	3,7553
РК	НМ	1,4402	2,956255732	2,843212605	-0,253	-2,7939	0,0449	1,7549

Тестирование операторов ГА для функции пяти аргументов

Тип ОК	Тип ОМ	t , с	$\bar{f}_{\text{средн}}$	$f_{\text{лучш}}$	x_1	x_2	x_3	x_4	x_5
Бинарное кодирование									
ОК	ОМ	1,7279	2,950906916	2,901252031	1,0494	-3,183	1,8159	-2,055	3,3287
ОК	ДМ	1,6996	2,961833430	2,843942881	0,2512	-1,064	2,8748	1,0895	-0,5728
ДК	ОМ	1,7318	2,971221857	2,873420238	11,883	3,1921	-0,228	7,0374	5,4539
ДК	ДМ	2,558	2,969821101	2,918636560	10,104	-0,476	2,5452	3,8452	-2,6031
Численное кодирование									
АК	РМ	1,6683	2,698738000	0,930091000	9,704	2,456	1,174	4,37	4,745
АК	НМ	1,3906	2,943820125	1,973386920	0,172	0,067	-0,108	0,083	-0,548
BLX	РМ	2,4411	2,971955805	2,914386988	7,159	-2,92	-2,078	-1,080	3,8513
BLX	НМ	1,6922	2,959708186	2,902015209	4,7191	1,413	-1,826	4,9351	1,737
РК	РМ	1,7112	2,967155085	2,917963266	15,077	2,282	4,5183	1,4587	2,7039
РК	НМ	1,697	2,954208003	2,927164316	13,784	-0,909	-0,857	1,1139	12,295

частности, равномерного кроссовера, было признано неэффективным.

2.3.5. Результаты для функции от пяти аргументов (с размером популяции 80 особей) (табл. 5)

$$f(x_1, x_2, x_3, x_4, x_5) = \arctg \alpha(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2) + \arctg((x_1 - 10)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 5)^2 - 1) + 0,2 \sin(x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2),$$

$$x_i \in [-50; 50], i = \overline{1, 5}.$$

При анализе функции пяти параметров с заданными ограничениями размер исследуемых популяций был увеличен до 80 особей в связи с увеличением числа точек исследуемого пространства. Приемлемой точности результата позволил достичь только один из реализованных типов ГА, а именно, сочетание арифметического кроссовера и равномерной мутации. Следовательно, такая модификация ГА может считаться наиболее подходящей для поиска экстремума данной функции с данными ограничениями.

2.4. Зависимость устойчивости алгоритма от вида целевой функции. Проанализировав используемую целевую функцию, можно определить, что она обладает хорошо выраженными глобальным

минимумом и локальным минимумом, например, для одномерного случая это $f(9,9485) = 0,5766$ и $f(0) = 1,5607$ соответственно, остальные локальные оптимумы близки по значениям между собой и колеблются около 3,0000 (рис. 1, 2). Представленные ниже графики (рис. 3) позволяют оценить, какой процент запусков ГА с арифметическим кроссовером и равномерной мутацией выдает то или иное значение результата для данной целевой функции с различным числом аргументов.

Полученные результаты позволяют проследить, как с увеличением размерности поискового пространства ухудшается сходимость ГА с фиксированным размером популяции и фиксированным числом запусков. Так, для одномерного пространства преобладает сходимость к глобальному оптимуму (рис.3, данные 1-й группы) для пространств с числом измерений от двух до четырех преобладающей является сходимость к локальному оптимуму, а сходимость к глобальному оптимуму снижается с увеличением размерности пространства (рис. 3, данные 2—4-й групп), и, наконец, результаты тестов, проводимых в пятимерном пространстве (рис. 3, данные 5-й группы), показали, что имеющихся вычислительных возможностей, отраженных в выборе размера популяции, явно недостаточно для поиска оптимума целевой функции.

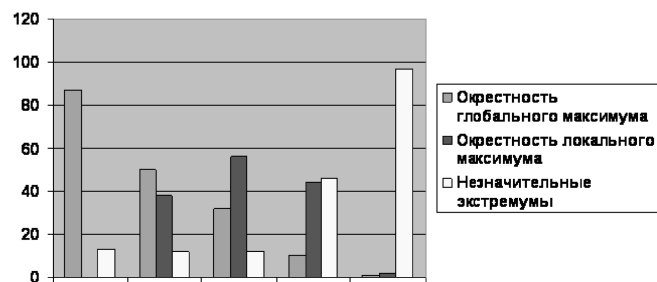


Рис. 3. Распределение результатов по 100 запускам для ГА с функцией от одного до пяти аргументов

Заключение

Проведенные эксперименты показали, что генетический алгоритм успешно решает задачу поиска оптимума функции в таких условиях, при которых затруднено использование точных методов. Полученные результаты также позволяют сравнить эффективности бинарного и численного способов кодирования решений. Недостаток бинарного кодирования заключается в том, что перебираемые алгоритмом возможные решения бе-

рутся лишь из некоторой сетки, частота которой зависит от числа битов, используемых для кодирования решения, что снижает его эффективность для задач со значительными размерами поискового пространства по сравнению с размерами популяции ГА.

Среди операторов кроссовера для вещественного кодирования наименее устойчивые результаты были получены при использовании оператора кроссовера дискретного типа (равномерный кроссовер), и наиболее устойчивыми были результаты, достигнутые с использованием оператора кроссовера конструирующего типа (арифметический кроссовер).

В данных реализациях решения с максимальной точностью были получены с использованием операторов численного кодирования, в частности, оператора арифметического кроссовера и равномерной мутации.

Список литературы

1. Емельянов В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. М. Курейчик, В. В. Курейчик. — М.: — Физматлит, 2003. — 431 с.

2. Норенков И. П. Эвристики и их комбинации в генетических методах дискретной оптимизации / И. П. Норенков // Информационные технологии. — 1999. — № 1. — С. 2—7.

3. Мухачева А. С. Задачи двумерной упаковки: развитие генетических алгоритмов на базе смешанных процедур локального поиска оптимального решения / А. С. Мухачева, А. В. Чиглинцев, М. А. Смагин, Э. А. Мухачева // Информационные технологии. — 2001. — № 9. — Приложение 24 с.

4. Holland J. H. Adaptation in Natural and Artificial Systems: An Introductory I Analysis with Application to Biology, Control, and Artificial Intelligence / John H. Holland — USA: University of Michigan, 1975.

5. Наместников А. М. Эффективность генетических алгоритмов для задач автоматизированного проектирования / А. М. Наместников, Н. Г. Ярушкина // Изв. академии наук. Теория и системы управления. — 2002. — № 2. — С. 127—133.

6. Иванов А. Б. Оптимизация архитектуры мониторинга сети телекоммуникаций на основе генетического алгоритма / А. Б. Иванов, В. В. Ширяев // Электросвязь. — 2002. — № 8. — С. 46—49.

7. Филиппова А. С. Проблемы декодирования прямоугольных упаковок: обзор современных технологий / А. С. Филиппова // Информационные технологии. — 2005. — № 12. — С. 13—20.

8. Herrera F. Hybrid Crossover Operators for Real-Coded Genetic Algorithms: An Experimental Study / F. Herrera, M. Lozano, A. M. Sanchez // Soft Comput. — 2005. — 9 (4). — P. 280—298.

9. Herrera F. Tackling real-coded genetic algorithms: operators and tools for the behaviour analysis / F. Herrera, M. Lozano, J. L. Verdegay // Artificial Intelligence Review. 1998. — Vol. 12. — № 4. — P. 265—319.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

УДК 004.78:658.51-0,57.17

Д. Х. Девятов, д-р техн. наук, проф.,

Магнитогорский государственный технический университет им. Г. И. Носова,

Д. С. Каплан, директор по информационным технологиям,

ОАО "Магнитогорский металлургический комбинат",

К. А. Иванов, вед. инженер Управления корпоративных разработок,

ОАО "Магнитогорский металлургический комбинат"

Информационная система руководителя металлургического предприятия

Построение системы стратегического управления являлось одной из главных целей проектов внедрения корпоративной информационной системы на многих металлургических предприятиях России. Ее основой является информационная система руководителя, которая обеспечивает топ-менеджеров полной, достоверной и своевременной информацией обо всех аспектах финансово-хозяйственной и экономической деятельности предприятия. В статье систематизированы основные принципы построения информационных систем такого класса и представлен опыт передовых металлургических предприятий.

Ключевые слова: информационная система руководителя, металлургическое предприятие, опыт использования, Магнитогорский металлургический комбинат, инструмент для управления, ключевые показатели эффективности, система стратегического управления, имитационная модель.

Соответствуя основным направлениям развития в мировой экономике, металлургические предприятия России продолжают укрупняться, и

автоматизация, без преувеличения, составляет неотъемлемую часть их бизнеса. Практически все из них прошли этапы автоматизации учета и управ-

ления и приступили к автоматизации производственных процессов, цепочек поставок и разработки стратегии развития. Однако увеличение количества получаемых данных на всех этапах производства и сбыта продукции не всегда переходит в качество, если речь идет об управлении предприятием. Несмотря на наличие ERP-систем и единого информационного пространства, руководители зачастую остаются вне процессов автоматизации и по-прежнему принимают много решений на основе интуиции и бумажных отчетов.

Для решения задач обеспечения топ-менеджмента металлургического предприятия полной, точной и своевременной информацией, необходимой для управления, используются следующие решения:

- решения класса *Business Intelligence*;
- системы бюджетирования;
- системы сбалансированных показателей;
- информационные системы руководителей (*Executive Information System*);
- комплексные системы управления рисками;
- системы управления цепочками поставок;
- корпоративные порталы.

Связующим звеном для всех перечисленных выше решений являются информационные системы руководителя, которые аккумулируют всю необходимую для управления информацию и предназначены для непосредственного использования топ-менеджерами предприятий.

Определение

Информационная система компании, являющаяся частью системы стратегического управления, служащая для анализа и предоставления данных о функционировании предприятия и поддержки принятия управленческих решений, называется информационной системой руководителя (ИСР). [1] Это инструмент, который обеспечивает быстрый и прямой доступ к важной информации в понятном и удобном формате. Важная информация — это достоверная, полная и своевременная информация обо всех аспектах хозяйственной деятельности предприятия, которая представляет особый интерес для топ-менеджеров предприятия. Понятный и удобный формат представления означает, что система специальным образом разработана для пользователей с минимумом свободного времени и не требует специальных знаний и проведения дополнительного обучения. ИСР позволяют топ-менеджерам быстро находить узкие места, критические отклонения, а затем детализировать информацию для определения источников проблемы.

История

Первые варианты таких систем появились в середине 80-х годов прошлого века и получили название *Executive Information System* (EIS). В то время EIS считалась современной и перспективной разработкой (красочные графические интерфейсы, управление мышью и сенсорные экраны), которая сопровождалась широкой рекламой. На фоне терминалов мэйнфреймов и первых персональных компьютеров с операционной системой DOS информационные системы руководителя смотрелись очень интересно. Использовались деловая графика, возможности интегрирования данных из множества структурированных и неструктурированных источников, гибкий анализ данных (*ad-hoc*-анализ), данные из внешних и внутренних источников, технологии ссылок, KPI (ключевые показатели деятельности), CSF (ключевые показатели эффективности). Все эти идеи до сих пор существуют и развиваются.

Однако в то время информационные системы руководителя не получили развития, какое им предсказывали. Это предопределили два основных недостатка. Во-первых, технологии того времени не могли обеспечить реализацию всех требуемых возможностей, поэтому получались дорогие, медленные и сложно управляемые системы. Во-вторых, разработчики ошибались в своем видении работы руководителей. Несмотря на убеждения о пользе таких систем и старания авторов топ-менеджмент по-прежнему предпочитал проводить совещания и читать распечатанные, специально подготовленные для них отчеты. Для многих руководителей компьютер был черным ящиком, и они не доверяли ему. Остальные, работая с ним, не хотели выглядеть как обычные пользователи, они хотели иметь специализированный инструмент, созданный только для них [2].

Сегодня, с развитием информационных технологий, бизнеса, с расширением возможностей по разработке, ИСР вновь набирают популярность. Появились упрощенные версии информационных систем руководителей — информационные панели для топ-менеджеров. Они строятся по тем же принципам, что и ИСР, но обычно имеют меньшие возможности для анализа и разрабатываются на основе коробочных программных продуктов. Согласно результатам последнего опроса OLAP Survey более 30 % респондентов создают информационные панели руководителей на основе OLAP-средств [1].

Назначение

Основное назначение информационной системы руководителя — это обеспечение топ-менеджеров полными знаниями о работе предприятия,

о его финансово-хозяйственной деятельности и его взаимодействии с внешним миром. Менеджер, владеющий знаниями, может лучше ставить задачи и лучше находить решения.

Второе назначение — обеспечение своевременного доступа к информации. Вся информация, которая содержится в ИСР, обычно может быть получена и традиционными способами, например, в виде отчетов. Однако время и средства, необходимые для сбора интересующей информации, зачастую специфической, часто меняющейся и из большого числа различных источников, обычно снижают желание менеджера получить эту информацию. Часто к тому времени, когда необходим отчет сформирован, вопросы менеджера уже изменились, и этот отчет уже никогда не будет прочитан. Своевременное получение информации оказывает влияние на более глубокое изучение проблемы. Когда руководитель получает ответ на свой вопрос, у него часто возникают новые вопросы, связанные с первым. Если их можно немедленно задать и получить ответ, то получается непрерывный процесс изучения проблемы. Если использовать традиционные средства, то к тому времени, как получен ответ на первый вопрос, условия могут измениться, и суть проблемы может оказаться невыясненной до конца.

На третье назначение информационной системы руководителя обычно не обращают должного внимания. ИСР обладает мощной возможностью быстро направлять внимание руководителя к проблемным местам в работе предприятия. Некоторые менеджеры используют эту возможность для того, чтобы дисциплинировать подчиненных. Иногда, какая-то часть подчиненных боится возможности менеджеров пользоваться информационной системой руководителя напрямую и всячески пытается дискредитировать ее, хотя все должно работать вместе с руководителем над решением проблем, найденных с помощью ИСР.

Отличительные особенности

Основные отличительные особенности информационной системы руководителя:

- она разработана с учетом всех требований руководителя;
- полностью понятна и удобна в использовании;
- используется непосредственно руководителем, без посторонней помощи и дополнительного обучения;
- предоставляет широкий спектр внешней и внутренней информации;
- вся информация представлена в графической форме, максимально удобной для восприятия,

с использованием различных диаграмм, гистограмм и графиков;

- обеспечивает исчерпывающие инструменты для оперативного (*on-line*) анализа, такие, например, как анализ тенденций, уведомления об исключительных ситуациях, возможности детализации (*drill-down*);
- имеет возможность предоставлять дополнительную информацию о проблемах и критических отклонениях, вместе с обычными отчетами.

Опыт использования системы в российской металлургии

Создание современной корпоративной системы управления для топ-менеджеров является одной из главных целей при внедрении ERP-систем на металлургических предприятиях. Основным элементом корпоративных систем управления являются информационные системы руководителя.

ОАО "ЗСМК". На ОАО "ЗСМК" был самостоятельно разработан программный комплекс, который сводит в единую схему всю текущую деятельность комбината — от поставок сырья и работы кислородного производства до отгрузки готовой продукции. Разработка этого комплекса была одной из целей проекта внедрения SAP R/3 на ОАО "ЗСМК" [3].

Эта система обеспечивает визуализацию производственного процесса с ключевыми производственными показателями для высшего уровня руководства предприятием. На рабочих местах директоров в реальном времени присутствует полная информация по всем аспектам жизни предприятия. Это позволяет видеть единую картину, где четко прослеживаются все проблемы выполнения планов и производственных потоков. Все показатели снимаются в полностью автоматическом режиме, зачастую даже без ведома работников и непосредственного руководства.

Топ-менеджмент предприятия получил возможность контроля над режимами работы доменных, загрузки шихты и передела металла в прокатном и обжимном цехах. Одним из главных плюсов такой системы стал оперативный (*on-line*) контроль (рис. 1, см. третью сторону обложки) над себестоимостью продукции [4].

После внедрения системы наметилось некоторое сопротивление на местах, так как начал рушиться сложившийся порядок "перекуров" и "авралов", когда планы выполнялись в экстренном режиме.

ОАО "НТМК". Информационная панель руководителя ОАО "НТМК" — решение, которое позволяет топ-менеджменту вести оперативный мониторинг деятельности комбината практически из любой точки мира. Ее главное назначение —

независимый контроль производственной, коммерческой и финансовой деятельности комбината (рис. 2, 3, см. третью сторону обложки).

Разработка и внедрение информационной системы руководителя в Нижнем Тагиле была одной из перспективных задач при внедрении SAP R/3 [5].

В этой системе можно наблюдать производственный процесс, какие агрегаты работают хорошо, какие остановлены на ремонт, соответствует ли количество фактически выплавленной стали плановым значениям. Есть данные по динамике производства, движению пасов и оборотных средств. Схематичность производственных процессов позволяет предоставлять эту информацию в интуитивно-понятном и наглядном виде [6].

ОАО "ММК". В 2004 году на стол руководителя ОАО "ММК" ежемесячно доставлялось свыше 600 различных документов. В большинстве случаев документы имели различное оформление, информация в них часто дублировалась. Объем некоторых документов, таких, например, как отчет основных показателей по труду, составлял около 30 страниц. Это все нужно было оперативно анализировать, поэтому был сделан вывод, что руководителям необходим новый инструмент для управления. Одной из целей проекта внедрения Oracle E — Business Suite на ОАО "ММК" было построение системы стратегического управления на базе ключевых показателей деятельности предприятия. Важнейшим элементом создания системы стратегического управления стала разработка информационной системы руководителя [7].

Созданный программный комплекс получил название ИАС "АРМ Руководителя" (рис. 4, см. четвертую сторону обложки). За год было построено отдельное централизованное аналитическое хранилище данных, куда была загружена архивная информация о работе предприятия за пять последних лет и налажены механизмы загрузки оперативной информации о работе предприятия из КИС, с периодом обновления в один час. Для работы руководителей на основе этого хранилища с учетом всех особенностей построения информационной системы руководителя был разработан интерфейс пользователя.

Система постоянно развивается и совершенствуется. В настоящее время она состоит из 13 разделов, которые целиком отражают деятельность как непосредственно самого ОАО "ММК" так и всех обществ Группы ММК.

Один из наиболее развитых и пользующихся популярностью у руководителей разделов — мониторинг производства. В нем представлено более 20 оперативных (*on-line*) мнемосхем работы основных агрегатов, некоторые из них являются

трехмерными и отображают точную модель процесса производства (рис. 5, см. четвертую сторону обложки). Можно смотреть сменно-суточный рапорт, рапорты о простоях, рентабельности и анализе отгруженной продукции, осуществлять контроль расхода тепловых и энергетических ресурсов, доступно даже промышленное телевидение, которое позволяет видеть в реальном времени, что происходит в цехах.

Другой, не менее популярный раздел — "Ключевые показатели", где представлено около 150 ежедневных и 150 ежемесячных показателей эффективности работы предприятия, сгруппированных по 12 разделам с графиками, комментариями к отклонениям от плана, справочной информацией по расчету показателя, ответственным за него и источником поступления данных.

Раздел "Транспорт" показывает время простоя железнодорожного транспорта на станциях ОАО "ММК", прибытие — отправка грузов, штрафы и плату за использование вагонов (рис. 6, см. четвертую сторону обложки).

Раздел "Денежные средства" позволяет видеть в оперативном режиме движение и остатки денежных средств на всех счетах в банках, с дополнительными фильтрами по группам и валюте. Раздел "Финансы" позволяет в удобном графическом виде смотреть динамику расчетов с дебиторами и кредиторами, есть цветовая индикация просроченных обязательств.

Один из наиболее крупных разделов — раздел "Запасы". Он содержит информацию о запасах сырья и материалов на ОАО "ММК" по 200 складов и 60 000 номенклатурным позициям. Графики отображают движение и остатки по месяцам, дням, вплоть до складской карточки, возможна фильтрация по складам и номенклатурным группам. Можно посмотреть графики прямого списания в производство. Есть возможность увидеть распределение запасов по периодам хранения без движения.

Несмотря на значительные объемы данных скорость доступа к информации находится на высоком уровне. Среднее время выполнения запроса к системе составляет 5 с, максимальное время — 15 с.

ИАС "АРМ Руководителя" является инструментом, который построен с учетом всех требований руководителей. Он позволяет быстро в удобной и наглядной форме с использованием графиков, диаграмм, технологии "*drill-down*" анализировать оперативную и историческую информацию [8]. Это дает возможность экономить самый ценный ресурс предприятия — рабочее время топ-менеджеров.

ИАС "АРМ Руководителя" пользуется большой популярностью у руководителей. В настоящее время в этой системе работают более 300 пользователей, в их число входит президент и вице-президенты ООО "Управляющая компания ММК", исполнительный директор ОАО "ММК" и директора по направлениям. Пользователи имеют возможность работать с этой системой из любой точки мира, где есть доступ в Интернет [9].

Развитие

Для определения оптимальных направлений развития финансово-хозяйственной и экономической деятельности металлургического предприятия, помимо анализа текущей информации, топ-менеджерам приходится рассматривать много различных вариантов направления развития и прогнозировать последствия принятия тех или иных управленческих решений. Это необходимо, как для выявления и отсеивания возможных ошибок, так и для обобщения удачных решений.

Учитывая сложность металлургического производства, это не простая задача. Универсальным средством анализа и прогнозирования, в том числе анализа и прогнозирования деятельности металлургических предприятий, являются методы математического и имитационного моделирования, которые в последние годы получили большое развитие.

Платформой для практического применения результатов математического моделирования может стать информационная система руководителя. Интеграция методов имитационного моделирования в ИСР с учетом всех особенностей построения таких систем может привлечь топ-менеджеров предприятия непосредственно в сам процесс моделирования. Например, ежемесячно на ОАО "ММК" приходит и уходит несколько десятков тысяч вагонов. За их использование предприятие платит большие суммы в виде платы за вагоны и штрафы за простои. Оптимизация и прогнозирование этих сумм являются актуальными задачами. Основные влияющие параметры на эти суммы: объемы и вид поступающего сырья, объемы и вид готовой продукции, время года, использование отопительных боксов в зимний период, объем осадков, приемка вагонов, число вагонов на станциях разгрузки и погрузки, число пустых вагонов. При построении качественной имитационной модели, и интеграции ее в информационную систему руководителя топ-менеджеры, могли бы, например, самостоятельно изучить возможность строительства новых погрузочно-разгрузочных станций в случае затоваривания существующих. Или сделать прогноз, как возрастут плата и штрафы при повышении объемов произ-

водства после строительства новых цехов (что предусмотрено инвестиционной программой ОАО "ММК" на ближайшие несколько лет), или смоделировать, каковы будут плата и штрафы, например, в случае массовых закупок сырья.

Заключение

Информационная система руководителя — это удобный инструмент для своевременного получения всей необходимой информации о работе предприятия с минимальными затратами времени. Система, помимо оперативного предоставления информации из ERP-систем, может интегрировать данные из различных аналитических, отчетных и вообще любых информационных систем, содержать исторические данные и позволять менеджеру осуществлять постоянный контроль над работой предприятия. Это необходимо руководителям металлургических предприятий ввиду специфики, размеров и сложности металлургического производства. На сегодняшний день некоторые крупные российские металлургические предприятия, например, ОАО "ММК", ОАО "НТМК" и ОАО "ЗСМК" уже внедрили у себя и используют такие системы и отмечают их высокую эффективность и значимость. Использование методов математического и имитационного моделирования позволит расширить возможности информационных систем руководителя и предоставить дополнительные возможности для анализа и прогнозирования деятельности металлургических предприятий.

Список литературы

1. **Pendse N., Bange C.** The missing "Next Big Things" // OLAP Report. 2007. 8 марта / <http://olapreport.com/Faileddozen.htm/>
2. **Pendse N.** The origins of today's OLAP products // OLAP Report. 2007. 23 августа. / <http://www.olapreport.com/origins.htm#top/>
3. **Материалы** компании HP. История успеха: развитие ИТ-инфраструктуры Западно-Сибирского металлургического комбината // http://www.hp.ru/openview/news_n_articles/stories/zapad-sibir.pdf/
4. **Борушевский Д.** "Записи" переходит на ИТ-контроль онлайн // Cnews.ru. 2004. 19 июля / <http://pr.cnews.ru/reviews/index.shtml?2004/07/19/161347/>
5. **История** успеха: Нижнетагильский металлургический комбинат // Металлоснабжение и сбыт / http://it.metalinfo.ru/2006/cases/SAP/nizh_tagil.pdf/
6. **Суковатин И. В.** Главное — знать, куда стрелять // Intelligent Enterprise. 2006. 19.
7. **Каплан Д. С.** Мы добились реального единого информационного пространства // Intelligent Enterprise. Спецвыпуск: Oracle интервью. 2005. С. 43—45.
8. **Иванов К. А., Кулагин С. Н.** Информационно-аналитическая система "АРМ Руководителя" // Создание и внедрение корпоративных информационных систем (КИС) на промышленных предприятия Российской Федерации. Вып. 1: Сб. тр. Всерос. науч.-техн. конф. / Под ред. Д. Х. Девятова. 2005. С. 84—91.

9. **Каплан Д. С., Иванов К. А., Девятов Д. Х.** Разработка информационной системы руководителей на основе Oracle Business Intelligence Suite // Создание и внедрение корпоративных информационных систем (КИС) на промышленных предприятиях Российской Федерации. Вып. 2: Сб. тр. Междунар. науч.-техн. конф. / Под ред. Д. Х. Девятова. 2007. С. 150—155.

10. **Максимов А.** Это не дань моде // Эксперт Урал. 2005. 19. (Доп. матер. к ст. "Правдоустроители с Магнитки").

11. **Kanclides T., Kimble C.** Executive Information Systems: A framework for their development and use // Proc. of GRONICS '95, Groningen, The Netherlands / Ed. T. Lourens. February 1995. P. 47—52.

12. **Белимов В.** Правдоустроители с Магнитки // Эксперт Урал. 2005. 19.

13. **Виер И. В.** Внедрять так, чтобы не было стыдно // Inteligent Enterprise. 2005. N 11.

14. **Материалы** конф. "Информационные технологии на службе руководителя металлургического предприятия. Системы поддержки принятия решений и выработки стратегии" // Металлоснабжение и сбыт. 2006. № 12.

15. **Материалы** лаборатории Postech (Postech Strategic Management of Information Systems Laboratory) / "EIS" definition // http://mis.postech.ac.kr/topic/eis_e.html /

16. **Носов С. К., Суковатин И. В.** Математическое моделирование в анализе хозяйственной деятельности металлургического предприятия // Изв. вузов. Черная металлургия. 2004. № 5. С. 74—79.

17. **Носов С. К., Суковатин И. В.** Стратегическое планирование развития производства металлопродукции в условиях рынка // Изв. вузов. Черная металлургия. 2004. № 3. С. 63—67.

18. **Суковатин И. В.** Развитие информатики и вычислительной техники на НТМК // Сталь. 2000. № 5. С. 80—81.

ВЫСТАВКА

13–15 НОЯБРЯ

**ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ**

РОСТОВ-НА-ДОНУ

digIT 2008



В ПРОГРАММЕ:

- Информационные технологии
- Телекоммуникации
- Информационная безопасность
- Электронные компоненты
(оборудование и технологии)

Официальный партнер:



ВЕРТОЛ
ВЫСТАВОЧНЫЙ ЦЕНТР **EXPO**

г. Ростов-на-Дону, пр. М. Нагибина, 30

Т./ф.: (863) 268-77-33; e-mail: digit@vertolexpo.ru; www.vertolexpo.ru



3- 5 декабря 2008

ВОРОНЕЖ

СВЯЗЬ. ИНФОРМТЕХ. БЕЗОПАСНОСТЬ. СРЕДСТВА СПАСЕНИЯ

12-я межрегиональная специализированная выставка

Тематические разделы:

Связь. Информтех

- Связь. Услуги связи
- Информационные технологии
- Техническая поддержка связи

Безопасность. Средства спасения

- Технические средства обеспечения имущественной и информационной безопасности
- Средства индивидуальной защиты
- Безопасность условий труда, отдыха и обучения. Экологическая безопасность
- Средства, техника и оборудование спасения

Услуги

- Услуги страховых компаний
- Услуги охранных компаний
- Услуги юридических компаний

Научные и учебные заведения

Профессиональные организации

Специализированные издания и литература

Организаторы:



ТОРГОВО-
ПРОМЫШЛЕННАЯ
ПАЛАТА
ВОРОНЕЖСКОЙ
ОБЛАСТИ

ВетЦ
ВЫСТАВОЧНЫЙ ЦЕНТР

Поддержка:

- Администрация Воронежской области;
- Администрация городского округа г. Воронеж;
- МЧС Воронежской области
- ГУВД Воронежской области
- Академия проблем безопасности, обороны и правопорядка РФ (Воронежское региональное отделение)
- Воронежский институт МВД России
- Воронежский государственный технический университет

Оргкомитет:

Т./ф.: (4732) 51-20-12 (многоканальный),
77-48-36, e-mail:mach@veta.ru

Подробная информация на **www.veta.ru**

CONTENTS

Aleksandrov A. E. <i>Development and Designing of Applied Program Appendices on the Basis of Generating Programming</i>	2
It is presented the description of architecture of the base software including expanded library of components, the problem-focused environment of programming realized on the basis of high level of language, allowing to create the subject-oriented libraries of models supported by the user. The technology of development of applied program appendices on the basis of the offered principles and the developed base software is described. Results of the calculations executed by means of the developed program appendix are resulted. Keywords: generating programming, library of components, problem-focused high level language, finite element method.	
Gavrilov A. V., Pavlova E. A. <i>Functional Interface Analysis for Formalization of Complex Information System Design</i>	9
In this paper we describe the comprehensive usage of formal design methods to provide syntactic and semantic integrity of the component model for complex information system architecture. At the same time the usage of the proposed functional interface algebra and the adaptation of the Hoare axiomatic system apparatus allow to provide formal correctness of software component interaction at function signature level and to provide declared functionality at interface contract semantic analysis level. The verification of syntactic correctness and semantic integrity of the architecture model for an electronic commerce system are cited as an example of the proposed approach. Keywords: verification, integrity, formal method, component, model, functional interface, contract.	
Sherykhalina N. M. <i>Numerical Results Filtration Application for CAD Systems Reliability Increase</i>	16
The ideas and methods of useful information allocation from a noised signal by the filtration, identification of the mathematical models of an informational signal, decisions of the information reliability are used in the field of numerical results analysis for its accuracy and reliability increase. With the help of these methods the investigation of the one simple method of the Cauchy problem solution is carried out. Some paradox regularities are found and explained. Filtration application and consideration of real properties of the numerical methods gives an opportunity of essential reliability improvement of the program complexes for simulation and design of different constructions and technologies. Keywords: numerical filtration, estimation and decrease of computation error.	
Poliakov B. N. <i>To Creation CAD of Technology of Straightening and the Equipment Roller-Straightening Machines</i>	22
The mathematical model and substantive provisions of modern scientific methodology of calculation and optimization on the COMPUTER of technological process of deformation of metal of billet (or sheet) rolling at straightening on the roller-straightening machine (RSM) are offered. Some results of application of the created methodology for conditions of manufacture of rails on NTMC, with the purpose of perfection of technology and optimization of modes of straightening and a substantiation of design parameters RSM from a position of a minimum of final curvature and a level of residual stresses are resulted. In particular, offered "soft" modes of deformation essentially reduce energy power parameters and raise consumer properties of rails, and optimum regimes of straightening can lower in 25 times final curvature thermostrengthened rails; the proved high sensitivity of consumer properties of rails to change of design parameters RSM allowed to prove their rational values, and also expediency of automation of technology and the equipment of straightening complexes. Keywords: rail, roller-straightening machine, stressed-deformed state, curvature, optimization, residual stresses, CAD.	
Zimnukhova Zh. E., Nemtinov V. A. <i>An Approach to Creation of Automated Decision Making Information System for Designing Metal Pieces Production Process</i>	29
The approach of automated decision making information system is suggested. The approach is used at designing technological processes of production systems. Different technologies, types of equipment, devices and auxiliary materials can be used to obtain production by applying the system. Keywords: information system, support of decision-making, designing technological processes.	
Gorshkov A. P., Gryzlova T. P. <i>The Complex Technical Object State Diagnostic System Based on Characteristic Sequences</i>	35
A new flexible technical diagnostic system is presented in this article. The system allows building decision procedures and make diagnosis. It contains automated feature extraction component based on the characteristic sequences of the digital signals. We test our system on two problems. The first problem is known as "Wafer" and is a short length time series classification problem. It is a problem of semiconductor microelectronics fabrication process diagnostics. Last problem is the transmission bearing fault diagnostics of gas turbine engine based on registration of the vibration. It was resulted high performance decision for those problems. Keywords: time series classification, technical diagnostic, vibrodiagnostic, feature extraction, characteristic sequences, pattern recognition.	
Malychina G. F., Merkusheva A. V. <i>Independent Component Analysis for Reconstructing the Signals Using Mixture Measurement</i>	39
The problem of reconstructing the form for a signal set by measuring linear mixture of them is considered. Method is based on the independent component analysis. Procedure for signal decomposition uses the manifold probability distribution representation in the form of approximation by a set of independent signal — components distributions. For signal forms analysis we apply criteria, designed on the base of Kullback — Leibler distance for elements of distribution functional space. Keywords: signal mixture, reconstruction, independent components, Kulback—Leibler criteria, matrix gradient, procedure, verification.	

- Balakirev N. E., Girin B. B.** *The Main Conceptions of Algorithm Description Language for Analysis Natural-Language Texts* 42
 Special language for description algorithms of syntactical analysis and its place in system of analysis natural-language texts was reviewed.
Keywords: syntax, natural-language, artificial language, dataware, ternary logic.
- Semenov N. N.** *Spectral Transforms of Sonar Coordinates* 47
 Detection and the target classification in contemporary radar are the two close connected tasks. While optimum echo signal detection is well described in the literature, there is no optimum or just fully operational classifier till now. This article exclaims one of the possible methods of data processing for the subsequent classification — spectral processing of coordinates data for the subsequent classification.
Keywords: sonar, location, coordinates, classification, spectral transforms, sonar target.
- Lapshin V. P., Zhukov I. S., Apikov A. H.** *Quantitative Estimation of a Degree of an Illegibility and Asymmetry of Functions of an a Belonging of Fuzzy Sets* 51
 In article are subjected to the analysis a number of functions of an a belonging of fuzzy sets with definition of a degree of an illegibility and asymmetry of these sets. Parameters of illegibility and asymmetry allow to receive a quantitative estimation of these properties of fuzzy sets. The border of a degree illegibility of not clear numbers allows to limit the further operations with indistinct data.
Keywords: function of an a belonging, fuzzy sets, degree of illegibility, asymmetry.
- Kovshov E. E., Mashkovtsev A. V.** *The Improvement of Medical Staff Recruitment in Medical Institution by Using Genetic Algorithms* 56
 Discussing themes of usage genetic algorithms, which belong to evolutionary computations, for improvement of medical staff recruitment in medical institution, which can be used for rising the effectiveness of medical and organizational processes.
Keywords: genetic algorithms, evolutionary computations, staff recruitment, computer-based subsystem, exchange format.
- Philippova A. S., Kuznetsov V. Yu.** *Problem of the Minimal Overlaying of Orthogonal Polygons with Prohibited Areas* 60
 A problem of polygon area overlaying with objects of random form is under investigation. The problem is NP-hard and the rectangular approximation algorithm is offered to solve it. The numeric experiment has been carried out and the results are shown with evaluation of the approximation factor.
Keywords: optimization methods, covering problems, block-structure technology, evolutionary algorithms, computer-aided design, waypoints map.
- Orekhov Yu. V., Orekhov E. Yu., Zaminova A. R.** *Genetic Algorithms Operators' Efficiency Research During Finding the Extreme Solutions in Binary and Real Spaces* 65
 The crossover and mutation operators used for search in binary and real spaces are presented. Fitness function with many extreme values allowed making comparison efficiency getting genetic algorithm constructions in space with variable dimensions. Numerical results presented show the efficiency of concerned crossover and mutation operators and there combinations.
Keywords: genetic algorithm, real-coded genetic algorithm, binary-coded genetic algorithm, crossover operator, mutation operator.
- Devatov D. H., Kaplan D. S., Ivanov K. A.** *Executive Information System for Steel Company* 72
 Creating strategic management system is the one of the main goals of deployment enterprise resource planning system at many metallurgic companies. It is based on executive information system, which provides top management with full, reliable and timely information about all sides of financial and economic activity. The article systematizes basic principles of creating executive information systems and describes wide experience of deployment it at leading steelmaking companies.
Keywords: executive information system, steelmaking company, experience of deployment, Magnitogorsk Iron and Steel Works Open Joint Stock Company (MMK), management method, Key Performance Indicator, strategic control system, simulation model.

Адрес редакции:

107076, Москва, Стромьинский пер., 4/1

Телефон редакции журнала (495) 269-5510

E-mail: it@novtex.ru

Дизайнер Т.Н. Погорелова. Художник В.Н. Погорелов.
 Технический редактор О. А. Ефремова. Корректор Е. Г. Волкова

Сдано в набор 10.07.2008. Подписано в печать 15.08.2008. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
 Усл. печ. л. 9,8. Уч.-изд. л. 11,31. Заказ 893. 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
