

нутя ли мыслящие машины всего лишь машинами или мы вынуждены будем считать их членами нашего человеческого общества со всеми правами и обязанностями. Нужны ли вообще мыслящие машины или это – чисто научная, абстрактная цель, которую лучше не воплощать в реальную практику. Большинство этих проблем видится нам в таком отдаленном и фантастическом будущем времени ..., но на все эти вопросы ответы надо искать уже сейчас, уже сегодня, когда только формируются, прорисовываются цели теории активного отражения.

Выводы. Необходим переход от термина "искусственный интеллект" к термину "активное отражение". Предметом теории активного отражения являются процессы активного отражения, в том числе мышление и обработка информации в компьютерах. Ее методом является формализованное знаковое моделирование, а целью – изучение процессов мышления, как процессов порождения, накопления, обработки, обмена, хранения информации и, если это возможно, то создание искусственных систем, сравнимых по своим возможностям с человеческим мышлением. Предложен критерий определения степени интеллектуальности систем: если система способна к порождению, генезису знака значит она – интеллектуальна.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Варламов О.О.* Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. – М.: Радио и связь, 2002. – 288 с.
2. www.ovar.narod.ru.

Л.А. Гладков, С.Н. Щеглов

СОВРЕМЕННЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ТЕХНОЛОГИЙ ИЗВЛЕЧЕНИЯ ЗНАНИЙ*

Эффективность работы большинства современных информационных систем определяется мощностью и адекватностью баз знаний, которые используются для выработки решений и рекомендаций. До настоящего времени в качестве основного источника знаний рассматривались эксперты предметной области. Отсутствие проверенных источников является одной из главных причин относительно слабого промышленного и коммерческого использования систем, основанных на знаниях. Подобная информация является источником знаний, необходимых для развития науки и производства, создания новых технологий, управления бизнес-процессами и т.п.

Особое внимание в современных интеллектуальных информационных системах уделяется различным областям человеческой деятельности, где накоплено много данных. Но данные сами по себе не имеют практической ценности, поскольку они нуждаются в серьезной обработке. Пользователи прикладных систем сегодня предъявляют серьезные требования к результатам исследования данных, которые не могут быть получены чисто статистической обработкой. Их интересуют заключения, помогающие в изучении комплексных задач и в принятии важных решений [1].

Развитие технологий обработки данных позволило оптимизировать средства доступа и механизмы контроля целостности данных. Был стандартизирован язык обработки SQL, который дал разработчикам настольных приложений универсаль-

* Работа выполнена при поддержке РФФИ (гранты № 07-01-00511, № 06-01-00272) и программ развития научного потенциала высшей школы 2006-2008 гг. (РНП.2.1.2.3193, РНП 2.1.2.2238).

ное средство общения. Информационные системы, ориентированные главным образом на ввод информации, получили название OLTP-системы (On-Line Transaction Processing, оперативная транзакционная обработка данных). Функциональным расширением баз данных (БД) являются объектно-ориентированные, мультимедийные БД, а также многомерные хранилища (Data Warehouses) и витрины данных (Data Marts) [2].

Объектно-ориентированные БД позволяют распространить принципы объектно-ориентированного программирования (ООП) на уровень хранения данных. Появление таких БД обусловлено тем, что стандартные реляционные подходы плохо применимы для работы с данными новых типов. Для решения подобных задач эффективны объектно-ориентированные информационные системы (ООИС). Состояние объектов в них описывается как значения их атрибутов, а поведение и применимые к ним операции определяются реализацией этих методов. Это позволяет представлять объекты предметной области в естественном семантически значимом виде, что упрощает разработку и модификацию схемы данных [1,2]. Примерами таких приложений являются САПР, CASE- и мультимедиа системы.

Интеллектуализация информационных технологий влечет за собой соответствующие процессы по отношению к компьютерам и программному обеспечению [2]. Интеллектуализация вычислительной техники производится за счет разработки и применения специальных устройств, таких как устройства распознавания речи и рукописного текста, устройства голосового управления компьютером, нейрокомпьютеры, параллельные вычислительные устройства и т.д. Повышение интеллектуальности программного обеспечения связано с использованием систем извлечения, обработки и хранения знаний, программных пакетов на базе нечеткой логики, позволяющих принимать решения в сложных неформализованных условиях. Также широко используются методы нечеткой математики, математические модели на базе искусственных нейронных сетей и оптимизационные алгоритмы на основе методов эволюционного, бионического и квантового поиска [3].

Другим важным направлением интеллектуализации информационных систем является моделирование знаний о предметных областях. Разработка новых моделей представления знаний, семантических сетей, онтологий, систем интеллектуальной верификации знаний позволяют реализовать принципиально новые информационные системы извлечения и обработки знаний для решения сложных плохо формализованных задач.

Интеллектуальные информационные системы, такие как экспертные системы, системы обработки естественного языка и нейронные вычислительные системы увеличивают производительность и облегчают выполнение сложных задач. Они также обеспечивают поддержку, когда информационный поток неполный или «нечеткий» [1-3]. Интеллектуальные системы могут использоваться индивидуально, но во многих случаях они интегрированы между собой и с другими информационными системами.

История развития и анализ современных тенденций в области создания сложных технических систем на базе искусственного интеллекта позволяет сделать вывод о том, что наиболее эффективными на сегодняшний день являются системы, построенные на основе гибридизации различных направлений, методов и технологий [4]. Свидетельством того, что речь в данном случае идет не о механическом объединении разнородных структур, является синергетический (нелинейный) эффект, наблюдаемый у подобных систем. Он приводит к сглаживанию недостатков отдельных составляющих (методов, технологий) и повышению эффективности системы в целом [5]. Примерами подобной интеграции могут служить нейро-

логические, нейро-нечеткие модели, нечеткие логические регуляторы, контроллеры и т.д. [6].

Так для нечетких множеств основной областью применения являются задачи представления и обработки знаний, они также используются для управления неопределенностью.

Нейронные сети обладают способностью к самообучению и активно используются для решения таких задач искусственного интеллекта как классификация и распознавание образов. Поскольку нейронные сети позволяют решать задачу аппроксимации функций с нелинейными ландшафтами, их с успехом применяют при решении задач прогнозирования поведения в финансовой, экономической и социальной сфере.

Эволюционные вычисления, куда кроме генетических алгоритмов входят также эволюционное (генетическое) программирование и эволюционные стратегии [7] успешно применяются для решения задач структурной и параметрической оптимизации, возникающих в ходе управления и проектирования сложных интеллектуальных систем, а также при принятии решений в условиях неопределенной или неполной информации.

Методы теории нечетких множеств используются как для подбора параметров генетических алгоритмов, так и для выбора коэффициентов определяющих скорость обучения нейронных сетей.

Одной из таких гибридных технологий получившей в последнее время широкое распространение являются нечеткие генетические (адаптивные, эволюционные) алгоритмы.

Интеллектуальные технологии, используемые в искусственном интеллекте, основываются на символическом представлении. В искусственном интеллекте символ – это буква, слово или число, которые используются для представления объектов процессов и их отношений. Объектами могут быть люди, идеи, понятия, события или утверждения о фактах [8].

При использовании символов, можно создать базу знаний (БЗ), которая содержит факты, понятия и отношения между ними. Используются различные процессы при манипулировании символами для генерации советов или рекомендаций при решении задач. Несмотря на то, что решение задач при помощи ИИ не реализуется напрямую алгоритмически, алгоритмы используются для осуществления процесса поиска.

Предметно-ориентированные аналитические системы являются одним из наиболее популярных разновидностей систем, позволяющих извлекать знания [1]. Наиболее широкий их подкласс, получивший основное распространение в области исследования финансовых рынков, носит название систем «технического анализа». В него входят несколько десятков систем, предназначенных для прогноза динамики цен и выбора оптимальной структуры инвестиционного портфеля на основе различных эмпирических моделей развития рынка.

Системы, применяющие эволюционное программирование. Примером систем подобного типа является пакет *PolyAnalyst*. В нем гипотезы о зависимости целевой переменной от других переменных формулируются в виде программ написанных на внутреннем языке программирования [1]. Процесс построения программ похож на то, что используют генетические алгоритмы. При нахождении программы, близко отображающей необходимую зависимость, в нее вносятся небольшие модификации и среди полученных дочерних программ отбирают те, которые повышают точность работы. В результате создается несколько конкурирующих генетических линий. Специальный транслирующий модуль переводит найденные зависимости во внешний

язык, понятный обычному пользователю (формулы, таблицы, графики и пр.). Для повышения доступности программного пакета в нем предусмотрен широкий спектр различных средств визуализации. Также предусмотрена возможность контроля статистической значимости выводимых зависимостей с помощью различных средств тестирования.

Системы на основе генетических алгоритмов. Как уже отмечалось, выше генетические алгоритмы являются эффективным средством для решения различных комбинаторных задач оптимизации, проектирования и принятия решений. Однако их вполне можно использовать и для решения задачи извлечения полезной информации и знаний из баз данных. Известно также, что в настоящее время генетические алгоритмы входят в состав стандартного инструментария методов *KDD*.

Например, требуется найти решение, наилучшим образом соответствующее некоторому заданному критерию. Каждое возможное решение полностью описывается неким набором атрибутов нечисловой природы. Они представляются в виде конечной последовательности «генов» (значений). «Хромосома» (решение), представленная последовательностью генов, является закодированным представлением одного варианта решения. Поиск оптимального решения моделируется процессом эволюции популяции индивидов [1,9].

Основными механизмами создания новых решений в процессе эволюции являются операторы кроссинговера и мутации. В результате смены поколений в процессе «эволюции» вырабатывается некое «квазиоптимальное» решение поставленной задачи.

Основными недостатками генетических алгоритмов является возможность схождения процесса поиска на решении далеком от оптимального, а также вероятность исключения перспективных решений из дальнейшего процесса поиска. Это особенно заметно при решении задач высокой размерности со сложными внутренними связями. Тем не менее, подобные алгоритмы и системы, построенные на их использовании являются достаточно эффективными. Примером коммерческого использования такой системы является пакет *GeneHunter* фирмы *Ward Systems Group* [1,3].

Еще одним перспективным направлением развития гибридных интеллектуальных систем и технологий являются «нечеткие генетические алгоритмы». Согласно определению [10] нечеткий генетический алгоритм – это генетический алгоритм, в котором некоторые компоненты реализованы с использованием инструментов нечеткой логики. Такими компонентами можно считать нечеткие операторы и нечеткие правила для создания генетических операторов с различными свойствами; системы нечеткого логического контроля параметров ГА в соответствии принятыми критериями; нечеткие критерии останки процесса генетического поиска. Математический аппарат теории нечетких систем используется в данном случае для кодирования, подбора оптимальных параметров генетических алгоритмов, значений вероятности генетических операторов, выбора функции пригодности и критерия останова, создания нечетких генетических операторов. Рассмотрим эти возможности более подробно.

Одним из ключевых моментов построения любого генетического алгоритма, это кодирование. Сущность кодирования заключается в преобразовании любого числового значения в некоторую последовательность символов конечного алфавита, состоящего обычно из небольшого числа элементов. Наиболее известный пример такого кодирования – это двоичное кодирование и представление решений в виде последовательности нулей и единиц.

Что касается нечеткого кодирования, то здесь можно провести аналогию между процессами кодирования и фаззификации, т.е. преобразования исходных числовых величин в распределения, соответствующие термам лингвистической пе-

ременной. При этом каждое числовое значение описывается одним или несколькими термами, а степень соответствия определяется степенью его принадлежности нечеткому множеству [5].

Термины конечного алфавита нечетких множеств могут включать такие понятия как «больше», «меньше», «немного больше (меньше)», «значительно больше (меньше)», «примерно ноль», «малая отрицательная (положительная)» и др. Использование таких алфавитов при кодировании решений в генетических алгоритмах дает возможность построить неоднородное разделение пространства поиска. Кроме того, закодированные последовательности могут иметь различную степень детализации.

Степень детализации и характер распределения решений в пространстве поиска может определяться на основе начальных знаний о решаемой задаче. Такое кодирование позволяет сосредоточить основные усилия на поиске в наиболее перспективных областях. В свою очередь для поиска в неперспективных областях (не содержащих элементов, которые оптимизируют функцию пригодности), определяют последовательности относительно низкой степени детализации.

Можно отметить, по крайней мере, два преимущества нечеткого кодирования. Во-первых, кодовые последовательности могут быть неоднородными и ориентироваться на отдельные многообещающие области поиска, что позволит сократить область поиска и соответственно вычислительные затраты. Кроме того, в закодированную последовательность может быть неявным образом включена функция пригодности [11].

Во-вторых, нечеткое кодирование позволяет выполнять так называемое слабое кодирование оптимизируемых структур. Слабое кодирование, в отличие от обычного (сильного) кодирования, не подразумевает жесткое соответствие типа «один-к-одному» между генотипом и фенотипом в кодируемой структуре.

Основная идея слабого кодирования состоит в использовании отношений типа «один-ко-многим» в процессе задания соответствия генотип – фенотип. Благодаря этому мы получаем единственный генотип, которому соответствует нечеткое семейство фенотипов. Это позволяет нам использовать все преимущества лингвистических переменных при сохранении двоичного кодирования.

Также при решении различных практических задач оптимизации используется кодирование на основе вещественных чисел [12]. Кодирование на основе вещественных чисел наиболее эффективно в решении задач параметрической оптимизации с переменными заданными в непрерывных областях значений. В таком кодировании хромосома обычно представляет собой вектор чисел с плавающей запятой, причем размер чисел соответствует длине вектора, который является решением рассматриваемой проблемы [13]. Генетические алгоритмы, основанные на вещественном представлении чисел, называют генетическими алгоритмами с вещественным кодированием (*real-coded GAs – RCGA*).

Одним из наиболее перспективных подходов к организации процесса поиска оптимальных решений является организация на основе мультиагентных архитектур. Под «агентом» может пониматься все, что способно воспринимать свою среду обитания с помощью датчиков (сенсоров) и воздействовать на нее с помощью исполнительных механизмов [8]. Внутреннее описание агента должно включать таблицу соответствия, определяющую, какая функция данного агента реализуется с помощью программы агента. При этом различают понятия функции и программы агента. Функция агента представляет собой абстрактное математическое описание, а программа агента – это конкретная реализация, действующая в рамках архитектуры агента [5,8].

Понятие агента применительно к различным прикладным системам может трактоваться по-разному. С рассмотренных выше позиций, агент может рассматриваться как искусственный организм в популяции себе подобных, стремящийся обучаться и адаптироваться к внешней среде, для того чтобы выжить в ней.

Тогда многоагентная система может рассматриваться как популяция простых и независимых агентов, каждый агент которой самостоятельно реализуется в локальной среде и взаимодействует с другими агентами. Связи между различными агентами являются горизонтальными, а глобальное поведение агентов определяется на основе расплывчатых правил [8].

При этом можно предложить использовать формальный механизм нечетких множеств для создания и вывода нечетких правил. С этой точки зрения такой механизм может восприниматься как функция агента, а программа, реализующая эту функцию применительно к решению каждой конкретной оптимизационной задачи, будет представлять собой нечеткий генетический алгоритм, настраиваемый в соответствии с текущими результатами.

В целом структура каждого агента может быть условно обозначена следующей формулой:

$$\text{Агент} = \text{Архитектура} + \text{Программа.}$$

Применение принципов построения многоагентных систем к задаче извлечения знаний позволит организовать распараллеливание основных технологических процессов при поиске и выборе решений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Городецкий В.И., Самойлов В.В., Малов А.О.* Современное состояние технологий извлечения знаний из баз и хранилищ данных // *Новости искусственного интеллекта*, №3, 2002. – С. 3-13.
2. *Башмаков А.И., Башмаков И.А.* Интеллектуальные информационные технологии. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005.
3. *Гаврилова Т.А., Хоросhevский В.Ф.* Базы знаний интеллектуальных систем. – СПб.: Питер, 2000.
4. *Ярушкина Н.Г.* Основы теории нечетких и гибридных систем. Учебное пособие. – М.: Финансы и статистика, 2004.
5. *Тарасов В.Б.* От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002.
6. *Bezdek J.C.* On the relationship between neural networks, pattern recognition and intelligence. *Int. J. Approximate Reasoning*, 6, 1992. – pp. 85-107.
7. *Емельянов В.В., Курейчик В.В., Курейчик В.М.* Теория и практика эволюционного моделирования. – М.: Физматлит, 2003.
8. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход. – М.: Издательский дом «Вильямс», 2006.
9. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы. – М.: Физматлит, 2006.
10. *Galantucci L.M., Percoco G., Spina R.* Assembly and Disassembly Planning by using Fuzzy Logic & Genetic Algorithms // *International Journal of Advanced Robotic Systems*, Vol. 1, №2, 2004. – pp. 67-74.
11. *Гладков Л.А.* Нечеткие генетические алгоритмы – новая гибридная технология решения оптимизационных задач // *Интеллектуальные системы. Коллективная монография. Выпуск второй.* – М.: Физматлит, 2007. – С. 31-47.
12. *Herrera F., Lozano M.* Adaptation of genetic algorithm parameters based on fuzzy logic controllers. In: *F. Herrera, J. L. Verdegay (eds.) Genetic Algorithms and Soft Computing*, Physica-Verlag, Heidelberg, 1996. – pp. 95-124.
13. *Lozano M., Herrera F., Krasnogor N., Molina D.* Real-Coded Memetic Algorithms with Crossover Hill-Climbing. *Evolutionary Computation* 12(3), 2004. – pp. 273-302.