

СРЕДА МОДЕЛИРОВАНИЯ ДЛЯ ПОСТРОЕНИЯ ИНЖЕНЕРНЫХ ТЕОРИЙ

© 1999 С.В. Смирнов

Институт проблем управления сложными системами РАН, г. Самара

Подводятся итоги исследований, целью которых являлась разработка методов и компьютерных средств, поддерживающих методологию построения инженерных теорий для компьютерной интеграции знаний. Формулируется проблема автоматизации конструирования предметно-ориентированных сред компьютерного моделирования, пригодных для поддержки этой методологии. Кратко рассматривается архитектура соответствующей общецелевой системы моделирования основанной на объектно-ориентированном представлении знаний.

Введение

В работах [1, 2] введено понятие *инженерной теории* и показана возможность его применения для решения проблемы интеграции знаний при исследовании сложных систем. Эта независимо разработанная концепция в главных чертах близка интенсивно разрабатываемому в западной литературе *онтологическому подходу* к интеграции знаний [3, 4]. Общие корни этих подходов лежат в *инженерии знаний* – признанной ветви научного направления обозначаемого как искусственный интеллект [5].

Представляется, что и инженерные теории, и онтологии предметных областей суть метафоры, пригодные при исследовании проблематики, связанной с синтезом *интегрированных моделей* объектов произвольной природы. При этом эпитет “инженерные” (или соответственно “производственные” для онтологии), употребляемый для характеристики создаваемых теоретических построений, прежде всего, обозначает наиболее актуальную сферу приложения интегрированных моделей: исследование и оценка систем искусственного происхождения, или артефактов [4, 6-8]. В других областях знания термин “инженерные теории” может быть оправдан отчетливой прагматической ориентацией соответствующих формальных конструкций и определяющей ролью компьютерных средств интеграции релевантных знаний.

В статье представлены результаты разработки методов и компьютерных средств, поддерживающих методологию построения

инженерных теорий. Практическая реализация обсуждаемой методологии интерпретируется как автоматизированное конструирование *предметно-ориентированных сред моделирования* (ПОСМ). Акцент ставится на выявлении структуры и актуальных составляющих *общей схемы* моделирования и *архитектурных компонентах* объектно-ориентированной системы компьютерного моделирования, обеспечивающих формирование и использование ПОСМ.

1. Компьютерная предметно-ориентированная среда моделирования как средство автоматизации научных исследований

В развиваемом подходе к автоматизации научных исследований эффективность применения компьютерных технологий определяется *качеством формируемых инженерных теорий*. Удел этих технологий - помощь в формировании теоретических представлений на основе ограниченного числа фундаментальных, проверенных практикой метамodelей. Однако внешне без труда обнаруживаемые эффекты достигаются лишь на этапе *автоматизации конструирования моделей объектов* из предметной области (ПО) построенной теории. Вне гипотетико-дедуктивного цикла формирования теории этот этап имеет чисто “вычислительный” характер, и в результате модель объекта лишь отражает наряду с истинным знанием рабочие гипотезы и заблуждения, зафиксированные компьютерными средствами. Характеризуя со-

временное применение компьютеров в науке, академик Краснощеков отмечает, что в подобных случаях легко провоцируется ситуация, когда “за истину принимается ее трансформация – правдоподобная иллюзия” [9]. На наш взгляд, наметившийся опасный перекокс в триаде “модель-алгоритм-программа” связан с недооценкой возможности эффективной компьютерной поддержки *всех фаз* осуществляемого любой отраслью науки познавательного процесса. В этом отношении автоматизация научных исследований как научная дисциплина должна включать проблематику *компьютерной поддержки построения теорий, моделей объектов познания на основе этих теорий и “стендов” для экспериментирования с такими моделями*. Другими словами, автоматизация научных исследований связывается с разработкой методов и средств, которые поддерживают возникновение в компьютере адекватной системы эпистемологических единиц, формирующих для пользователя *предметно-ориентированную среду моделирования* - ПОСМ.

Понятие среды моделирования является естественным развитием концепции предметно-ориентированных пакетов прикладных программ и систем моделирования. Будучи построенной, предметно-ориентированная среда моделирования поддерживает *разработку и исследование моделей уже конкретных объектов, или объектной модели, в избранной предметной области*. ПОСМ выступает в ролях:

1) *предметно-ориентированной базы знаний* при отражении *статической структуры* объектной модели;

2) *супервизора процесса моделирования* при исследовании объектной модели, проведении с нею экспериментов, т.е. какой-либо связанной с объектной моделью формой “вычислений” (имитацией поведения, расчетом, поиском экстремумов и т.д. и т.п.), понимаемой как реализация *модельной динамики*.

Технология построения *собственно* ПОСМ прежде всего связана с автоматизацией наиболее важного и сложного этапа научных исследований – *концептуального моделирования* предметной области. Оценка состояния компьютерного моделирования си-

стем и смежных областей (например, программирования вообще) позволяет указать не только успехи в решении имеющихся здесь проблем, но и открытые вопросы.

Классическая парадигма технологии компьютерного моделирования предполагает использованием некоторой готовой универсальной концептуальной модели, “защитой” в инструментальном средстве. ПО-ориентация ограничивается в таком случае разработкой *таблицы соответствия* [10] между априори заданными категориями универсальной модели и категориями, которые адекватны исследуемому предмету. Встроенные механизмы специализации инструментария касаются, как правило, расширения его операционного базиса (например, с помощью макроопределений [10, 11]). Относительно новой тенденцией является конструирование технологий компьютерного исследования путем предварительной предметной ориентации универсальных инструментов с помощью языковых процессоров [12, 13]. По сути, и в том, и в другом случае технологии моделирования обслуживают пользователя на *зуженном* участке процесса познания. Исследователь не получает надлежащей поддержки или отстраняется от наиболее “интеллектуального”, “наукоемкого” этапа познавательного процесса – формирования *сообразно* целям исследования такой абстракции, упрощенной точки зрения на мир, которая (и только она) принимается в расчет на последующих шагах познавательного процесса, связанных с отражением реального мира. Продуктом рассматриваемого этапа - концептуального моделирования, и является “инженерная теория”, или онтология, или в более нейтральных формулировках – представление, спецификация ПО.

Стремление предоставить исследователю технические средства познания в расширенном диапазоне этапов познавательного процесса, включая концептуальное моделирование ПО, сборку объектных моделей и осуществление процесса моделирования определяет *проблему автоматизированного конструирования ПОСМ*.

Решение этой проблемы позволит создавать в различных областях научных иссле-

дований специализированные системы моделирования, базирующиеся на знаниях. От существующих эти системы будут отличать развитые средства концептуального анализа и моделирования ПО, гибкая среда моделирования, допускающая простую модификацию и расширение спектра исследовательских задач. В конечном счете, автоматизированное конструирование ПОСМ позволит создавать эволюционные технологии исследований, обеспечивающие не только получение в результате моделирования новых знаний о ПО, но и их использование для развития технологии исследований путем уточнения и расширения концептуального базиса, на основе которого выполняется сборка объектных моделей и управление модельной динамикой. Именно такое эволюционное развитие теорий и средств моделирования отвечает потребностям интеграции знаний о сложных системах управления.

2. Система эпистемологических единиц предметно-ориентированной среды моделирования

Одним из результатов исследования проблем, связанных с программной инженерией с одной стороны и искусственным интеллектом с другой, стало утверждение точки зрения, что методы и средства построения программных продуктов (равно как сами такие продукты) опираются на *представление и обработку знаний*. Экспликация с этих позиций “знаниевой”, или эпистемической, роли привычных программно-технических понятий позволило найти новые, не использованные ранее методы программирования, которые оказались чрезвычайно полезными при решении сложных задач в различных прикладных сферах приложения компьютеров. Эпистемологическая роль программно-технических единиц обнаруживает себя в различных формах, начиная от знания последовательности шагов, необходимых для решения задачи, в обычном *императивном* программировании, до различных техник *декларативного* (логического, продукционного, базирующегося на ограничениях и т.п.) программирования, в основе которых лежит отделение описания задачи от интерпретации

этого описания.

В свете сказанного разработка новых методов и программных средств в актуальной прикладной области должна начинаться с выявления адекватного набора, системы эпистемологических единиц. Критерием такой адекватности становится “интеллектуальное” поведение созданной системы, интеллектуализация компьютерной среды для решения задач пользователя [14].

Применительно к автоматизации научных исследований на этапе построения моделей это означает *определение системы эпистемологических единиц в общей схеме компьютерного моделирования*.

Решение такой задачи с общих позиций научного познания мира является предметом *методологии науки* [15], методологических разделов фундаментальных естественных наук (прежде всего *математики* [16]), а также многочисленных прикладных дисциплин, изучающих кибернетические, психологические и др. закономерности креативной *деятельности* [17], различные аспекты автоматизации интеллектуального труда. В духе интеллектуализации ЭВМ общие варианты ответов на поставленный вопрос предложены в рамках *новой информационной технологии* [14, 18]. Конкретно проблему определения эпистемологических компонент в задачах компьютерного моделирования исторически раньше других специалистов начали разрабатывать идеологи машинного *имитационного моделирования сложных систем* [19, 20]. Позднее эти вопросы оказались в центре внимания разработчиков методов *объектно-ориентированного проектирования и программирования* [21, 22], результатом которых в области моделирования явилась методология объектно-ориентированного анализа и попытка стандартизации соответствующих языковых средств [23]. Далее излагается подход к решению поставленного вопроса на основе переосмысления с современных позиций материалов публикаций, затрагивавших эту тему, а также оригинального опыта разработки и использования систем компьютерного моделирования.

Моделирование как всеобщий научный метод включает в себя следующие фазы:

- 1) концептуальное моделирование ПО;
- 2) конструирование модели актуального объекта (системы объектов) ПО, или объектное моделирование;
- 3) экспериментирование с объектной моделью, трансформация объектной модели (трансформационное моделирование).

В полном объеме даже при условии компьютерной поддержки моделирование является человеком-машинным процессом, автоматически могут быть выполнены лишь отдельные шаги в рамках основных фаз. Во многом именно это обстоятельство определяет итерационный характер процесса моделирования и обусловлено наличием в каждой фазе обязательного *этапа анализа и интерпретации результатов*. Содержание этих этапов целиком определяется спецификой концептуального, объектного, трансформационного моделирования, а следствием выполнения, т.е. принятия пользователем решения, – повторение или смена фазы моделирования (рис. 1).

Концептуальное моделирование связано с анализом и структурированием имеющихся у исследователя данных. Эта фундаментальная фаза научного исследования, которая по существу направлена на выявление *допустимой семантики информации* о реальной действительности. В результате фиксируется некоторая абстракция, упрощенная точка зрения на мир, *концептуальная схема*, которая представлена в каких-то целях. В литературе для обозначения этой фазы часто используют термин “*концептуальный анализ*”, содержание которого интерпретируют как “*абстрагирование существа физических объектов и их взаимосвязи в виде некоторой модели или*

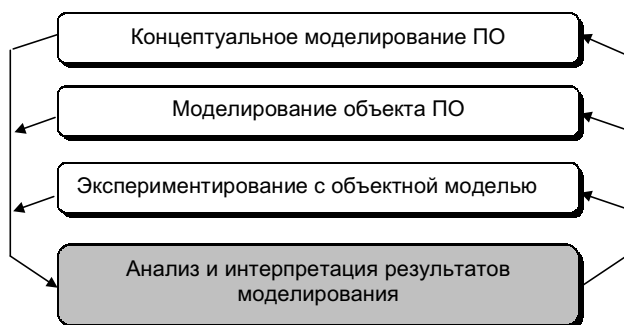


Рис. 1. Характер развития процесса моделирования

теории, которая соответственно называется *концептуальной моделью* изучаемого объекта” [24, с. 3]. Одной из разновидностей таких моделей являются *онтологии* предметных областей [3, 4]. Основным феноменом фазы концептуального анализа является возможность получения и сосуществование *нескольких различных* концептуальных моделей ПО¹ (рис. 2).

Объектное моделирование соответствует *синтетической* стадии познания реальности. Результатом этой фазы моделирования является отражение, воспроизведение, *описание конкретного объекта* или системы объектов ПО, выполненное в рамках избранной концептуальной схемы (т.е. на “языке” этой схемы, или, что более употребительно, но неточно, – “языке” ПО). С общих позиций необходимо указать на возможность конструировать согласно данной концептуальной схеме объектные модели *различных типов*: физические, графические, математические и т. п. [19]. Рис. 2 иллюстрирует это обстоятельство. Компьютерное моделирование сужает подобную типологию объектных моделей, но не устраняет ее, оставляя открытым исключительно важный для практики вопрос о выборе типа объектной модели².

Трансформационное моделирование представляет и реализует знания о “вычислениях”, связанных с объектной моделью. Эту фазу (см. рис. 2) с весьма общей точки зрения можно рассматривать как процесс *решения задачи, выбор* воздействий на стартовую объектную модель, в результате которых она приобретает некоторые удовлетворяющие исследователя свойства. Операционным базисом подобных трансформаций, аппаратом решения служит *объекто-ориентированное исчисление*, которое, вообще говоря, составляет часть концептуальной схемы. Результат рассматриваемой фазы может рассматриваться в двух аспектах:

- 1) методической ценностью для исследователя может обладать зафиксированная *последовательность воздействий* на объект или *цепочка промежуточных объектных моделей*, начиная со стартовой;
- 2) уже названная выше финишная объектная модель, которую естественно называть

интегрированной моделью, поскольку для ее построения вовлекаются все знания, которые актуализируются исследователем в процессе моделирования.

Концептуальная и объектная модели, описание (протокол) трансформации объектной модели, интегрированная модель составляют *систему эпистемологических единиц ПОСМ.*; в этом смысле общая схема моделирования дополнительно включает *метаконцепцию, пригодную для конструирования концептуальных схем* (рис. 2).

Автоматизированное конструирование предметно-ориентированных сред компьютерного моделирования предполагает разработку методов и программных средств, обеспечивающих *возникновение в компьютере системы эпистемологических единиц общей схемы моделирования*, т.е. поддерживающих все фазы процесса моделирования, включая представление получаемых на каждой шаге результатов.

Анализ возможных подходов к реализации подобного компьютерного инструментария показывает определяющее значение требований к *содержанию и возможности эффективной компьютерной поддержки метаконцепции описания ПО*. В [1, 2] роль такой метаконцепции принадлежит *методу балан-*

сов, в основе которого лежит составление уравнений балансов различных ресурсов ПО. Этот подход эффективен при интеграции “глубоких” знаний о ПО [1, 25] и в целом близок к так называемому *программированию в ограничениях* [26]³. В сложных системах большее значение имеют “поверхностные” (плохо формализуемые) знания, а организация знаний оказывается более сложной [28]. Поэтому в качестве метаконцепции для описания ПО целесообразно использовать более универсальную методологию, на роль которой в настоящее время претендует методология *объектно-ориентированного анализа* [22, 23].

3. Компьютерные средства конструирования предметно-ориентированных сред моделирования

В ходе работ по созданию компьютерных средств поддержки методологии построения инженерных теорий создан прототип объектно-ориентированной системы моделирования общего назначения, которая в основных чертах реализует предложенный подход к интеграции знаний. При разработке системы использованы оригинальные приемы объектно-ориентированного расширения языков программирования [29] и результаты иссле-

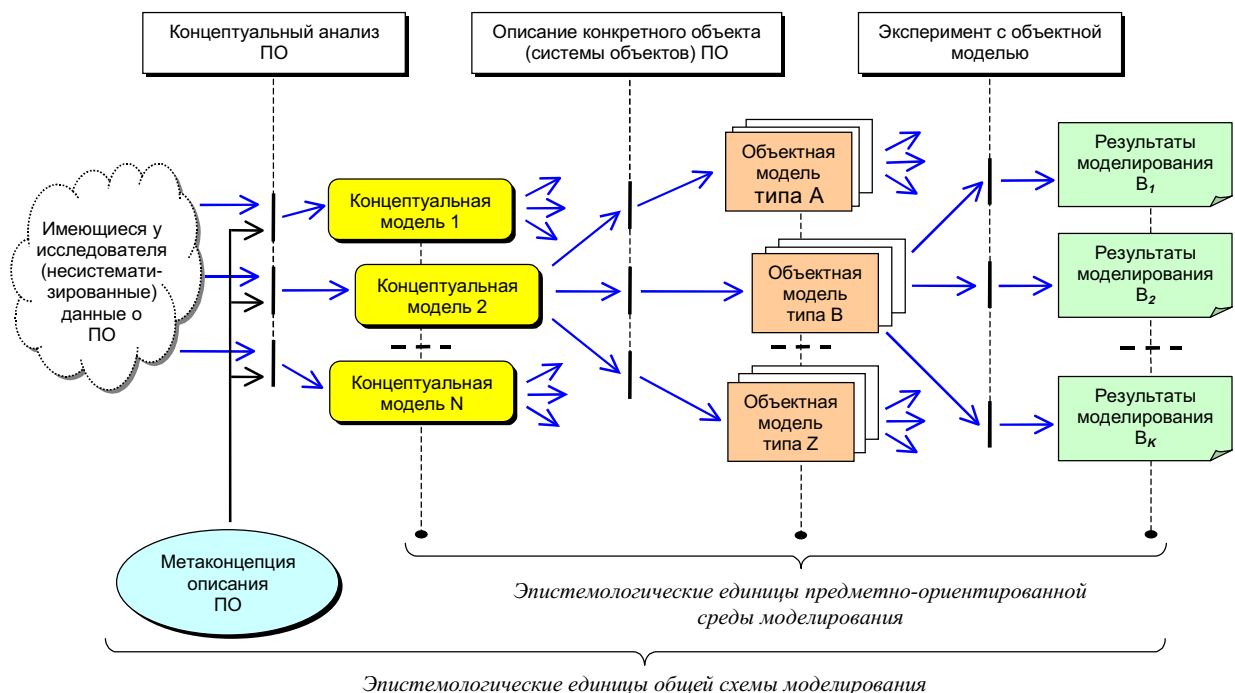


Рис. 2. Общая схема моделирования

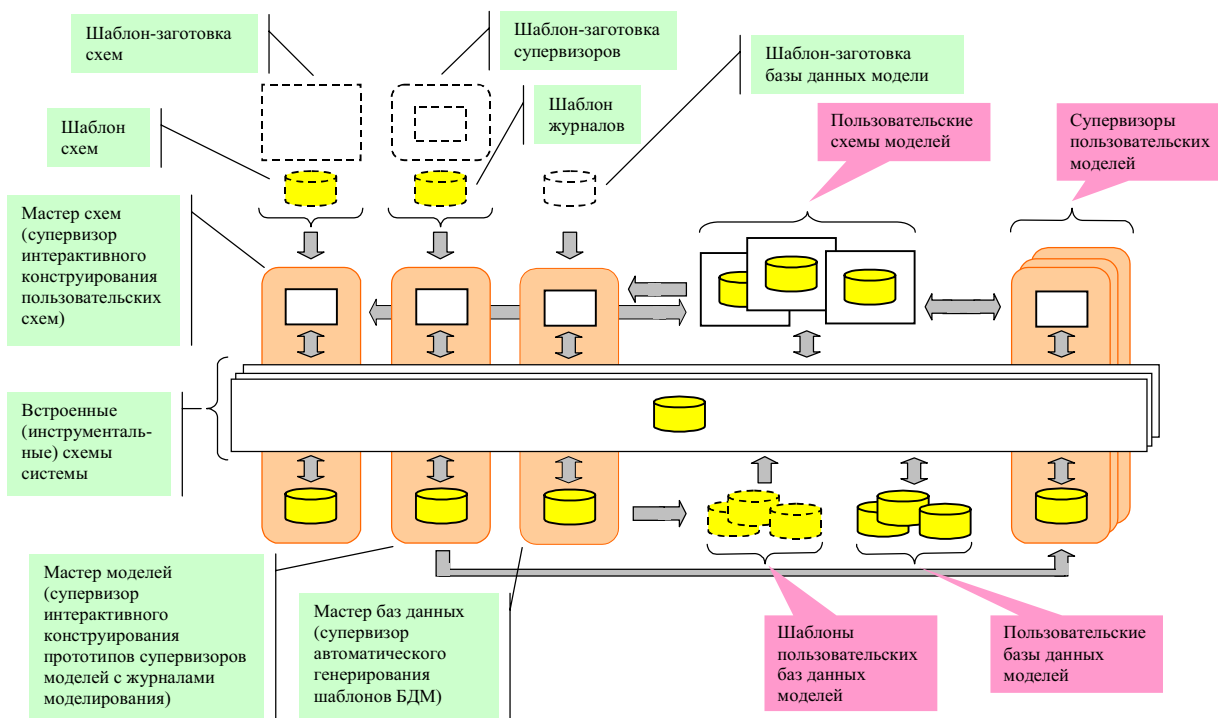


Рис. 3. Архитектура системы моделирования

ований в области создания объектно-ориентированных СУБД [30, 31].

Архитектура этой системы моделирования подробно обсуждалась в [32] и представлена здесь на рис. 3, где фигурными стрелками обозначены *информационные потоки* между *архитектурными компонентами*, т.е. чтение и/или запись данных и использование процедур. Архитектурные структурно-функциональные компоненты (имеется всего три их основных типа: *объектно-ориентированные базы данных - ООБД, схемы моделей и супервизоры моделей*) по отдельности или в определенных сочетаниях отображают *эпистемологические единицы* общей схемы моделирования, и *технологические составляющие*, с помощью которых эти единицы порождаются и увязываются между собой.

ООБД, сохраняющая конкретную объектную модель, или база данных модели (БДМ), конструируется на основе предметно-независимого шаблона, но содержательная сторона этого акта определяется концептуальной моделью ПО, которая именуется *схемой модели*. В свою очередь схема модели также хранится как БДМ, построенная на основе *схемы схем*, определение которой в этом смысле рекурсивно.

Супервизор моделей в эпистемологическом плане представляет и реализует знания о "вычислениях", связанных с объектной моделью. Объем соответствующего исчисления определяется связанной с каждым супервизором специальной БДМ - *журналом моделирования*, где регистрируется множество схем, с объектными моделями которых может оперировать супервизор. Дополнительно журнал используется для организации *эволюционной технологии исследований на моделях*, понимаемой как смена состава *допустимых контекстов* работы данного супервизора, т.е. используемых супервизором схем моделей. История подобных изменений регистрируется в журнале, а выполняемая при этом репликация БДМ позволяет осуществлять бэктрекинг процесса моделирования с возможностью выбора иного пути развития этого процесса.

В технологическом отношении схема - эквивалент предметно-ориентированного ППП, выстроенного в соответствии с объектно-ориентированной парадигмой, а супервизор - *ассоциация* процедурной части, которая всегда имеет характер *главной (основной) программы* и специальной объектной модели - *журнала моделирования*.

Основные архитектурные компоненты делятся на *инструментальные (встроенные)* и *пользовательские* (см. тип выносок на рис. 3). Первые входят в состав системы априори, вторые создаются с помощью инструментальных для решения пользовательских задач. К числу встроенных относятся прототипы БДМ, схем и супервизоров, схемы схем и журнала моделирования, а также вспомогательные схемы, описывающие предметные области, связанные с представлением и обработкой сложных структур данных.

По характеру выполняемых функций входящие в состав системы моделирования супервизоры в соответствии с современной терминологией могут именоваться *мастерами*. *Мастер схем* поддерживает в интерактивном режиме этап концептуального моделирования предметной области, *мастер моделей* автоматизирует разработку супервизоров моделей, *мастер баз данных* автоматически генерирует шаблон базы данных объектной модели в соответствии с указываемой ему схемой модели.

Супервизоры - единственный архитектурный компонент системы, с помощью которого *все* другие компоненты порождаются и увязываются между собой. Разделение супервизоров на инструментальные и пользовательские, обуславливается лишь их функциональной ролью, а не принципами построения и работы. Такая унификация обеспечивает *открытость архитектуры* системы моделирования, способность системы к *эволюции* методом "раскрутки".

Лабораторная версия системы моделирования, реализована как надстройка *Excel Microsoft Office 97*, т.е. представляет собой 32-разрядное приложение, использующее для хранения данных и процедур файлы XLS-формата. Базовым языком программирования является *Visual Basic for Application (VBA)* [33]. Как система автоматизации программирования система воплощает *единство* среды разработки пользовательских приложений и хранения данных. Ядро системы образует объектно-ориентированная СУБД, а средства разработки включают совокупность соглашений об использовании базового языка при написании процедурных составляющих раз-

личных архитектурных компонентов и пакет программ, реализующий специальное объектно-ориентированное расширение VBA и доступ к ООБД.

Заключение

1) Концепция предметно-ориентированной среды компьютерного моделирования интерпретирует и развивает современные теоретические и технологические аспекты компьютерного представления и использования знаний при исследованиях сложных систем.

2) Выявлена система эпистемологических единиц ПОСМ. Задача автоматизированного конструирования ПОСМ формулируется как задача разработки методов и программных средств, поддерживающих возникновение в компьютере системы эпистемологических единиц общей схемы моделирования, которая представляет собой соответствующую систему ПОСМ, дополненную метаконцепцией, пригодной для конструирования концептуальных схем предметных областей.

3) Разработаны архитектура системы, поддерживающей методологию построения инженерных теорий, и прототипы соответствующих программных инструментальных средств. Инструментарий поддерживает полный цикл исследований с помощью компьютерных моделей и пригоден для исследования и оценки систем произвольной природы.

СПИСОК ЛИТЕРАТУРЫ

1. *Vittikh V.A.* Engineering theories as a basis for integrating deep engineering knowledge // *Artificial Intelligence in Engineering*. 1997. Vol. 11. № 1. P. 25-30.
2. *Bumux B.A.* Интеграция знаний при исследованиях сложных систем на основе инженерных теорий // *Известия РАН. Теория и системы управления*. 1998. № 5. С. 132-139.
3. *Gruber T.R.* A translation approach to portable ontologies // *Knowledge Acquisition*. 1993. V. 5. № 2. P. 199-220.
4. *Uschold M., King M., Moralee S., Zorgos Y.* The Enterprise Ontology // *The Knowledge Engineering*

- Review. 1998. V. 13. № 1. P. 31-88.
5. *Искусственный интеллект*: В 3-х кн.: Справочник. Кн. 2. Модели и методы / Под ред. Д.А. Поспелова. – М.: Радио и связь. 1990. 304 с.
 6. *Kiryama T., Yamamoto F., Tomiyama T., Yoshikawa H.* Metamodel: An Integrated Modeling Framework for Intelligent CAD // *Artificial Intelligence in Design. Proc. of the fourth Int. Conf. on Applications of Artificial Intelligence in Engineering.* Cambridge, UK, 1989. P. 429-450.
 7. *Виттих В.А., Смирнов С.В.* Интегрированные модели артефактов в согласованной инженерной деятельности // Труды VI нац. конф. по искусственному интеллекту. Т. 2. – М.: Российская ассоциация ИИ, 1998. С. 398-403.
 8. *Виноградов И.Д., Кузнецов С.В., Смирнов С.В.* Приобретение знаний и моделирование для реорганизации инженерной деятельности // *Распределенная обработка информации: Труды шестого международного семинара.* – Новосибирск: СО РАН, 1998. С. 304-307.
 9. *Краснощеков П.С.* О чем умолчал Билл Гейтс // *Вестник Российской академии наук.* 1998. Т. 68. № 11. С. 980-985.
 10. *Кораблин М.А., Сидоров С.В., Смирнов С.В.* Функциональные модели вычислительных систем реального времени. – Самара: “Университет Наяновой”, 1997. 305 с.
 11. *Шрайбер Т.Дж.* Программирование на GPSS: Пер. с англ. – М.: Машиностроение, 1980. 592 с.
 12. *Кораблин М.А.* Конструирование специфицирующих оболочек для пакетов прикладных программ // *УСиМ.* 1990. № 2. С. 43-49.
 13. *Смирнов С.В., Колесов А.В.* Языковой процессор для порождения нейронных сетей // *Информационные системы и технологии: Межвуз. сб. научн. трудов.* – Самара: СГАУ, 1996. С. 138-146.
 14. *Перспективы развития вычислительной техники*: В 11 кн.: Справочное пособие / Под ред. Ю.М. Смирнова. Кн. 2. Интеллектуализация ЭВМ / Е.С. Кузин, А.И. Ройтман, И.Б. Фоминых и др. – М.: Высш. шк., 1989. 159 с.
 15. *Философия и методология науки* / Под ред. В.И. Купцова. – М.: Аспект Пресс, 1996. – 551 с.
 16. *Колмогоров А.Н.* Математика // В кн.: *Математический энциклопедический словарь.* – М.: Сов. энциклопедия, 1988. С. 7-38.
 17. *Щедровицкий Г.П.* Избранные труды. – М.: Школа культурной политики, 1995. 800 с.
 18. *Поспелов Г.С.* Искусственный интеллект – основа новой информационной технологии. – М.: Наука, 1988. 280 с.
 19. *Советов Б.Я., Яковлев С.А.* Моделирование систем. – М.: Высш. шк., 1985. 271 с.
 20. *Киндлер Е.* Языки моделирования: Пер. с чеш. – М.: Энергоатомиздат, 1985. 288 с.
 21. *Буч Г.* Объектно-ориентированное проектирование с примерами применения: Пер. с англ. – М.: Конкорд, 1992. 519 с.
 22. *Шлеер С., Меллор С.* Объектно-ориентированный анализ: моделирование мира в состояниях: Пер. с англ. – Киев: Диалектика, 1993. 240 с.
 23. *What is OMG-UML and why is it important? / Object management Group Press Release, 1997 // Интернет: www.omg.org/news/pr97/umlprimer.htm.*
 24. *Тамм Б.Г., Пуусенн М.Э., Таваст Р.Р.* Анализ и моделирование производственных систем. – М.: Финансы и статистика, 1987. 191 с.
 25. *Budyachevsky I.A.* Knowledge-based tools for development of engineering theories // *Artificial Intelligence in Engineering.* 1997. Vol. 11. № 1. P. 31-40.
 26. *Программирование в ограничениях и недоопределенные модели* / А.С. Нариньяни, В.В. Телерман, Д.М. Ушаков и др. // *Информационные технологии.* 1998. № 7. С. 13-22.
 27. *Технология системного моделирования* / Е.Ф. Аврамчук, А.А. Вавилов, С.В. Емельянов и др. – М.: Машиностроение; Берлин: Техник, 1988. – 520 с.
 28. *Виттих В.А.* Управление открытыми системами на основе интеграции знаний // *Автометрия.* 1998. № 3. С. 38-49.
 29. *Кораблин М.А., Смирнов С.В.* Наследование свойств в задачах объектно-ориентированного программирования на языке Модуль-2 // *Программирование.* 1990. № 4. С. 38-43.

30. *Боровицкий М.Д., Смирнов С.В.* Разработка объектно-ориентированной СУБД для представления знаний // Повышение эффективности средств обработки информации на базе математического и машинного моделирования: Труды 2-й Всесоюзн. конф. - Тамбов: ТВВАИУ, 1991. С. 239-240.
31. *Боровицкий М.Д., Смирнов С.В.* Реализация и исследование производительности объектно-ориентированной СУБД // Программирование. 1992. № 6. С. 18-28.
32. *Смирнов С.В.* Открытая архитектура инструментальных средств моделирования сложных систем // Проблемы управления и моделирования в сложных системах: Труды международной конф. – Самара: СНЦ РАН, 1999. С. 59-66.
33. *Visual Basic для приложений (версия 5) в подлиннике*: Пер. с англ. / Сання П. и др. – СПб.: BHV-Санкт-Петербург, 1997. 704 с.

¹ Без интерпретации в рамках некоторой релевантной концептуальной модели имеющиеся у исследователя данные не могут передавать знания (Канту приписывается фраза: "Anschauung ohne Begriffe ist blind" - "воззрение без понятий слепо"). При этом не существует никакого теоретически независимого, "чистого" языка наблюдений, который бы описывал предметы такими, какие они есть на самом деле. Для одних и тех же данных в зависимости от преследуемых целей можно представить различные толкования.

² Например, в рамках концепции "систем массового обслуживания" (СМО), с помощью математических пакетов класса Mathematica можно строить символические модели, характерные для математической теории массового обслуживания, однако практически ценными оказываются обычно имитационные модели СМО.

³ Другим известным примером развитой метаконцепции моделирования является *системная динамика* Форрестера [27], ориентированная на описание ПО системами дифференциальных уравнений 1-го порядка.

MODELING WORKBENCH FOR DEVELOPMENT OF ENGINEERING THEORIES

© 1999 S.V. Smirnov

Institute for the Control of Complex Systems of Russian Academy of Sciences, Samara

The purpose of the research was elaboration of methods and computer tools to support methodology of engineering theory development for computer knowledge integration. Results of the research are given in this paper. The problem of automated design of subject-oriented modeling workbench is formulated. The architecture of corresponding general-purpose modeling system, which is based on object-oriented knowledge representation, is briefly considered.

УДК 681.5

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕКОТОРЫХ МЕТОДОВ СЛУЧАЙНОГО ПОИСКА И ОПТИМИЗАЦИИ

© 1999 И.А. Минаков

Институт проблем управления сложными системами РАН, г. Самара

В статье рассматриваются методы случайного поиска, широко применяемые в настоящее время. После краткого обзора современных алгоритмов случайного поиска приведен эмпирический анализ возможностей их использования на примере некоторых проблем, возникших на практике. На основе анализа даны практические рекомендации о целесообразности использования различных методов в зависимости от решаемой проблемы.

1. Введение

В последние годы проблема оптимизации сложных систем, к которой сводятся многие социально-экономические, технические, организационно-управленческие, комбинаторные задачи и задачи теории игр, становится одной из ведущих в мире искусственного интеллекта. Она часто представима в виде функции цели, которую необходимо оптимизировать (причем функция цели задана не всегда аналитическим путем, а иногда и в виде «черного ящика»), и некоторого набора начальных данных и ограничений на решение. Этому представлению соответствуют и задача о путешествующем коммивояжере, и задача об оптимальном размещении, и задачи нахождения динамического баланса, и многие другие задачи из соответствующих областей.

Традиционно основой для решения подобного класса задач служили детерминированные алгоритмы, основными из которых являются: метод сканирования (слепой поиск), метод поочередного изменения параметров (метод Гаусса-Зейделя), метод градиента, метод наискорейшего спуска и некоторые другие. Каждый из них имеет свои преимущества и недостатки, и применение конкретного метода зависит от особенностей задачи, но общей их характеристикой является то, что последовательность действий алгоритма строго предопределяется сложившейся ситуацией.

Однако в последнее время все чаще встречаются задачи, в которых точные мате-

матические методы слабо применимы или неприменимы вовсе. Это связано с целым рядом проблем, основными из которых являются *проблема размерности* (особенно ярко она проявляется в NP-сложных задачах, к которым относится и задача о коммивояжере) и *проблема априорной неопределенности*, связанная с тем фактом, что в реальных задачах знания о проблеме могут быть трудноформализуемыми или неформализуемыми в принципе, иногда их трудно представить в числовом виде, а в некоторых случаях получение их представляется в принципе невозможным. Все эти проблемы относятся в первую очередь к тем моделям, в которых частью является человек или общество, т.е. задачи социологии, психологии, экономики, политики и т.п.

Итак, существует целый ряд задач (например, задача рекламирования конкурирующих товаров [1], задачи экономического прогнозирования [2], задачи моделирования политической ситуации [3], задачи размещения [4] и многие другие) для которых детерминистические методы решения неприемлемы или не обеспечивают необходимой степени точности.

Поэтому необходим альтернативный подход - использование эвристических методов и намеренное введение элемента случайности в алгоритм поиска. В этом случае случайность будет служить целям сбора информации о поведении объекта исследования и целям управления. Введение такого случайного поведения дает возможность построить

алгоритмы случайного поиска, которые в условиях априорной неопределенности, многопараметричности и большой размерности задачи зачастую превосходят известные регулярные алгоритмы поиска и оптимизации.

Основными достоинствами методов случайного поиска являются [5-6]:

- повышенное быстродействие;
- высокая надежность и помехоустойчивость;
- высокая робастность, т.е. малочувствительность к нерегулярностям поведения целевой функции, структуре множества оптимизации, наличию случайных ошибок при вычислении функции;
- сравнительно простая внутренняя реализация;
- малочувствительность к росту размерности множества оптимизации;
- возможность естественного ввода в процесс поиска операции обучения и самообучения;
- в рамках известных схем случайного поиска легко строятся новые алгоритмы, реализующие различные эвристические процедуры адаптации.

Далее в статье будут рассмотрены известные алгоритмы случайного поиска, а также новые, появившиеся в последнее время, но уже успевшие завоевать признание, и дан некоторый сравнительный анализ и рекомендации по применению этих методов.

2. Методы случайного поиска

По-видимому, впервые идея о пользе применения случайного поведения при поиске была сформулирована У.Р. Эшби в его книге «Конструкция мозга», одним из первых в России этой проблемой занимался Растрин [5 и др.]. В последнее время, в связи с развитием новых технологий этой проблеме уделяется пристальное внимание, поэтому представляется целесообразным дать краткое описание основных методов случайного поиска, применяемых в настоящее время.

Необходимо сразу подчеркнуть, что областью применения таких методов служат многопараметрические задачи, задачи с большим уровнем помех и возможностью сбоя, неустойчивые, с большой размерностью, с

множеством локальных экстремумов или негладкостью функции качества, с априорной неопределенностью или недоопределенностью, т.е. все те сложные случаи, в которых аналитические методы неприменимы. Это связано в первую очередь с тем, что в более простых и формализуемых задачах аналитические методы намного превосходят методы случайного поиска.

Также не будем рассматривать методы, которые сконструированы под конкретный класс задач (например, динамическое программирование), а рассмотрим так называемые «глобальные» методы, которые могут применяться с любым кругом проблем.

Для определенности будем полагать, что задачей алгоритма оптимизации является минимизация функции качества.

Грубый случайный поиск (метод Монте-Карло)

Это самый простой и в то же время самый известный алгоритм случайного поиска, состоящий из равномерного случайного «бросания» точек в пространство поиска. Основное его достоинство - простота, и в теории глобальной оптимизации этот алгоритм применяется в основном в качестве эталона при теоретическом или численном сравнении алгоритмов и в качестве составной части некоторых алгоритмов глобального случайного поиска.

Алгоритм с парной пробой

Алгоритм предполагает четкое разделение между поисковыми и рабочими шагами системы. В случайном направлении, определяемым случайным единичным вектором, по обе стороны от исходного состояния делаются пробы. Значение показателя качества в испытываемых точках и определяют направление рабочего шага, т.е. шаг делается в направлении наименьшего значения качества. В данном случае основные потери поиска связаны с двумя пробными определениями показателя качества. Предполагается, что рабочий шаг делается практически без помех.

Характерной особенностью данного алгоритма является его повышенная тенденция к «блужданию».

Алгоритм с возвратом при неудачном шаге

В пространстве оптимизируемых параметров делается шаг в случайном направлении. Если значение функции качества в новом состоянии больше или равно исходному (т.е. шаг неудачен), то система возвращается в первоначальное состояние, после чего вновь делается шаг в случайном направлении. Если же шаг удачен, то последующий случайный шаг делается уже из нового состояния.

Этот алгоритм, как и предыдущий, хорош для оптимизации «блуждающих объектов», когда функция качества по каким-то неизвестным, но определенным причинам изменяется во времени так, что в различные моменты времени система имеет различные значения функции качества.

Алгоритм с пересчетом при неудачном шаге

Это модификация предыдущего алгоритма для случая с низким уровнем помех. Возврат в исходное состояние при неудачной попытке в этом случае не производится, а просто снова делается случайный шаг, отсчитанный из старого состояния, т.е. возврат как бы пересчитывается вместе с последующим случайным шагом.

Все вышеуказанные алгоритмы легко модернизируются для случая с накоплением информации, что позволяет несколько понизить неопределенность результата оптимизации и улучшить характеристики процесса поиска. Так появляются *алгоритм с возвратом*, *алгоритм с «забыванием»* и *алгоритм с запоминанием результата*. Также, учитывая и храня в памяти неудачные и удачные комбинации шагов, можно повысить самообучаемость системы. Примерами таких алгоритмов могут быть *самообучение методом исключения*, *покоординатное экспоненциальное обучение*, *алгоритм покоординатного самообучения с произвольным законом изменения вероятности*. Все эти алгоритмы подробно описаны в [5].

Существует множество алгоритмов, чья эффективность повышена за счет введения в алгоритм дополнительных составляющих: например, *алгоритм многоэкстремальной стохастической аппроксимации* использует элемент адаптации, на идее образования по-

крытий строится *метод случайных покрытий*, а метод *случайный мультистарт* эффективен за счет комбинации случайного выбора точек с итерациями локального спуска. Также для повышения эффективности работы используют идеи отбрасывания малоперспективных подмножеств, уменьшения доли случайности при сохранении свойств равномерности, штрафные функции, более частый выбор точек в окрестностях перспективных из рассмотренных ранее и др. [7].

Теперь рассмотрим несколько современных методов решения оптимизационных задач, которые широко применяются на практике.

Генетические алгоритмы

Генетические алгоритмы (Genetic Algorithms) можно отнести к группе адаптивных методов, которые используются для решения проблем поиска и оптимизации. Они сочетают в себе элементы детерминистического и стохастического подходов. В силу этого генетические алгоритмы относятся не только исключительно к методам случайного поиска. Более того, они успешно применяются в комбинациях с аналитическими методами или другими алгоритмами поиска и оптимизации. В основе генетических алгоритмов лежит принцип естественного отбора (выживания сильнейшего или наиболее приспособленного). В процессе поиска анализируются сразу несколько ветвей эволюции. Применяя так называемую «функцию приспособленности», определяющую насколько хорошо найденное решение проблемы и выполняющую роль окружающей среды при моделировании эволюционного процесса, ГА «выращивают» новые популяции объектов, генная структура которых более приспособлена к текущей ситуации. Таким образом, генетическая модель имитирует эволюцию приспособления через механизмы изменчивости объектов [6].

Сфера применения ГА очень широка: от автоматизированного проектирования, решения комбинаторных задач и проектирования нейронных сетей до применения в экспертных и обучающихся системах. Основные области применения на практике:

а.) оптимизация сложных численных фун-

кций (наиболее традиционное использование GA);

б.) распознавание образов, речи и т.п. (например снимки из космоса или создание фотороботов);

в.) комбинаторные задачи (включая задачи о путешествующем коммивояжере, планирование работ и т.п.);

г.) планирование (от размещения мебели до сложных экономических и политических проблем);

д.) самообучающиеся системы (в так называемых классифицирующих системах, где GA создают набор правил типа *if ... then* для решения предложенной проблемы);

е.) контроль и управление (в больших, комплексных системах, таких, как фабрика или завод, они способны управлять множеством параметров для поддержания оптимального режима работы системы).

Моделируемый отжиг

Этот метод был разработан Киркпатриком в 1982 г. и хорошо описан в [8]. Как в основе генетических алгоритмов лежит идея биологической эволюции, так моделируемый отжиг (Simulated Annealing) появился в виде аналогии термодинамического процесса нагревания и медленного охлаждения субстанции для получения кристаллической структуры.

Это существенно усовершенствованная версия метода наискорейшего спуска. Начиная со случайно выбранной точки в пространстве поиска, делается шаг в случайном направлении. Если этот шаг приводит в точку с более низким уровнем значения функции оптимизации, то он принимается. Если же он приводит в точку с большим значением функции оптимизации, то он принимается с вероятностью $P(t)$, где t - время. Функция $P(t)$ сначала близка к единице, но затем постепенно уменьшается до нуля - по аналогии с охлаждением твердого тела.

Таким образом, в начале процесса моделирования принимаются любые ходы, но, когда «температура» падает, вероятность совершения негативных шагов уменьшается. Негативные шаги иногда необходимы в том случае, когда нужно избежать локального оптимума, но принятие слишком многих нега-

тивных шагов может увести в сторону от глобального оптимума.

На сегодняшний день этот метод является областью активных исследований (быстрый «переотжиг», параллельный отжиг) и успешно применяется во множестве областей, например, проектирование СБИС [8].

Поиск с запретами

Поиск с запретами (Tabu search) является еще одним стохастическим методом глобального поиска. Он основан на аналогии с человеческим поведением, т.е. на присутствии в поведенческой схеме человека случайных элементов, которые означают, что в одной и той же ситуации человек может поведи себя разным образом. Сохранение листа запретов, в котором, например, может храниться набор уже рассмотренных точек пространства поиска, является одной из основных характеристик данного подхода. Алгоритм заключается в выборе случайной точки в пространстве поиска, рассмотрения точек из окрестностей данной точки, и, при достижении определенного критерия, выбор новой точки в другом регионе поиска, который еще не рассматривался.

Все рассмотренные методы имеют общие характеристики: они все адаптивные, итеративные, стохастические алгоритмы, на каждом шаге работы все они оценивают значение функции качества и для каждого можно доказать его сходимость к глобальному оптимуму. В недавно опубликованной NFL (No Free Lunch) теореме утверждается, что для всех алгоритмов, ищущих экстремум функции качества, производительность их одинакова, если усреднить результаты по всевозможным функциям качества. [9]. Практическое значение этой теоремы состоит в том, что не существует панацеи на все случаи жизни, несомненный успех какого-либо оптимизационного метода в определенной области знаний не гарантирует такого же успеха в другой области. Это означает, что для каждой специфической области необходимо проводить исследования и отыскивать тот оптимизационный метод, который подходит ей более всего.

При описании методов решения оптимизационных задач нельзя не упомянуть под-

ходы, которые составляют альтернативу методам случайного поиска.

Подход с использованием самоорганизующихся алгоритмов, как антипод детерминистических методов, оперирует с такими понятиями как элементарный алгоритм, эвристический критерий и интегральное воздействие. На основе подобных алгоритмов смоделированы сложные технические системы, но они могут быть использованы и в оптимизационных задачах [10,11].

Подход с использованием нечеткой логики (Fuzzy logic) [12] успешно используется во многих промышленных областях, например в области экономического управления, распознавания образов и обработки изображения. В настоящее время наметилась тенденция применения нечетких множеств в гуманитарных науках, лингвистике, психологии и социологии (в целом нечеткие множества более характерны для гуманитарных наук, так как неоднозначные, субъективные данные там встречаются чаще) [11].

Нейронные сети (Neural Networks), смоделированные по принципу поведения нейронов в человеческом мозге, частично перекрывают области применения методов случайного поиска, и также используются с целым рядом задач, включая распознавание образов, экспертные системы и машинное обучение [13].

Эволюционное программирование (Evolutionary Programming) - подход, основанный на тех же принципах, что и генетические алгоритмы, но содержащий больше эвристических зависимостей и основанный на ранжированных мутациях, применяется в ряде комбинаторных и оптимизационных задач, в задачах машинного обучения [14].

Эволюционные стратегии (Evolution Strategies) - тоже эволюционный подход, использующий при конструировании структуры индивида адаптивный уровень мутации, что позволяет приспособиться к любой переменной модели. Также применяется в области комбинаторики, экспертных системах и при машинном обучении [15].

3. Сопоставление методов случайного поиска на примере некоторых задач

До перехода к непосредственному сравне-

нию, следует отметить, что все алгоритмы случайного поиска, имеющие наибольшее практическое значение, в той или иной степени включают в себя элементы эвристики, поэтому имеют недостаток, присущий эвристическим методам: нечетко определен класс задач, для которых алгоритмы наиболее эффективны и отсутствуют четкие рекомендации по выбору параметров работы алгоритма [7]. То есть, как уже было замечено ранее, нельзя дать исчерпывающие рекомендации по применению того или иного алгоритма, так как производительность его сильно зависит от целевой функции, о которой иногда почти ничего не известно.

Поэтому в этом разделе рассмотрим несколько примеров, возникших на практике или специально сконструированных теоретически, и, в результате эмпирических тестов, сделаем вывод о целесообразности применения того или иного метода в конкретном классе задач. Необходимо еще раз подчеркнуть, что точность метода сильно зависит от удачного выбора набора параметров его работы, и если какой-то метод при эмпирическом испытании оказался значительно хуже остальных, то нужно проверить другие наборы параметров для нахождения оптимального.

Проблема «королевских дорог»

Эта проблема предложена Холландом [16] как очень сложная для оптимизации в силу множества разрывов в оптимизируемой функции. Сравнительный анализ, проведенный в [17], рассматривал метод генетических алгоритмов (GA), моделируемый отжиг (SA), эволюционное программирование (EP) и эволюционные стратегии (ES). В первую очередь сравнивалась производительность этих методов в их стандартной форме, затем производилась настройка параметров под конкретную задачу и повторное сравнение. В результате в своем стандартном варианте GA оказались на 30-40% лучше остальных методов, ES и EP в среднем действовали одинаково, и самый худший результат показал SA. После настройки параметров GA по-прежнему остался на первом месте, хотя с меньшим отрывом - 15 %, в зависимости от настройки параметров точность EP варьировалась в пределах ± 20 %, а самый большой разброс показал ES: от - 50 % до + 67 %. При должной степени настройки производитель-

ность SA оказалась вровень с EP и ES.

Задача оптимизации графа

В качестве примера NP-трудной задачи была выбрана задача об оптимизации графа, к которой сводится множество практических задач. Стохастические методы сравнивались как между собой, так и с некоторыми вариациями детерминистических методов. Проводя эксперименты с различной структурой графа, разными схемами остывания для SA, подобранными эвристиками для детерминированных методов (в основном для модификации градиентного метода) и варьируя значения параметров GA были получены следующие результаты [18]: наиболее быстрыми, как и предсказывается теорией, оказались детерминистические методы, SA показал хорошую производительность за приемлемое время, а медленнее всех работали GA. Решение, предложенное стохастическими методами, было примерно на 30 % точнее, чем у методов градиента. Более того, выяснилось, что с ростом размерности результаты GA становятся точнее, чем у SA, хотя скорость по-прежнему остается наиболее медленной. В качестве вывода для данной проблемы необходимо отметить, что для случая с небольшим количеством узлов графа лучше применять методы градиента, далее, по мере усложнения дерева графа нужно применять более общие методы, решая задачу производительности и делая выбор между скоростью и точностью в сторону того или иного алгоритма случайного поиска.

Проблема размещения. Проектирование цепей

Проблема размещения часто встречается на практике, например, при проектировании схем и цепей. В работе [19] был проведен анализ различных стохастических методов, применяемых к данной проблеме, включая GA и SA. В процессе тестирования выяснилось, что наиболее оптимальные результаты предоставляют именно эти две технологии, причем в зависимости от структуры цепи GA дают либо эквивалентный, либо превосходящий по качеству результат, чем SA. Поскольку в подобных проблемах качество результата имеет первостепенное значение по сравнению со скоростью, представляется целесообразным использование при конструировании цепей GA.

Проблема размещения. Заполнение «рюкзака»

Эта проблема (Multiknapsack problem [20]), состоящая в заполнении «рюкзака» различными предметами различного объема и веса и проблемой нахождения оптимального соотношения, тоже относится к классу NP-сложных задач и тоже часто возникает на практике во многих технических и экономических приложениях. Анализ применения различных методов, включая детерминистические методы, GA, SA, поиск с запретами (TS) и нейронные сети (NN) приведен в [21]. Анализ проводился для целого ряда различных условий и конфигураций. Были получены следующие результаты: по

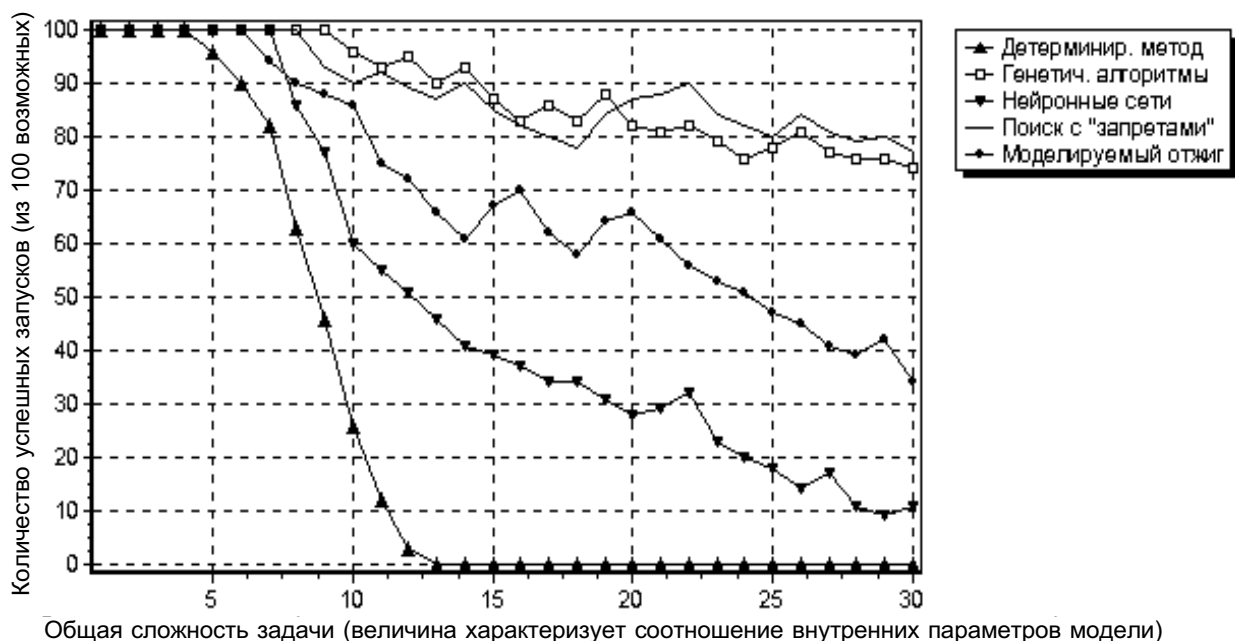


Рис. Результаты сопоставления методов поиска

мере возрастания сложности проблемы существенно ухудшаются характеристики работы детерминистических алгоритмов, поэтому, начиная с некоторой степени сложности, они не рассматриваются далее, так как не способны представить приемлемое решение в отведенное время. SA, как и в предыдущем случае, сильно зависит от выбранной схемы охлаждения и сложности задачи, хотя при удачной настройке решает до 70 % задач. Применение NN оказалось неудачным с практической точки зрения в силу слабой скорости модифицирования при изменении условий. Только около 40 % задач в лучшем случае было решено. В случае «средней» размерности проблемы GA работают быстрее TS, решая около 85 %, в то время, как TS около 80 %. По мере возрастания размерности TS имеет тенденцию опережать GA, или, по крайней мере, идти вровень.

Проблема предсказания банковского курса валют

Алгоритмы случайного поиска, в том числе GA, SA и TS, а также нейронные сети, были модифицированы для предсказания курса валют на основании существующих графиков (данные по валюте брались из сети Internet). В стандартной форме всех вышеприведенных алгоритмов лучше всех с данной задачей справились NN, предсказывавшие до 90 % результатов (для анализа был взят сравнительно гладкий период, в котором не происходило крупных кризисов). Производительность GA и TS была немного ниже - 75 - 80 % и варьировалась в зависимости от настройки параметров. По-видимому, для SA не удалось подобрать подходящую схему охлаждения, так как алгоритм предсказывал результаты только в 55 - 60 % случаев.

Алгоритм составления расписания

Алгоритмы составления расписания (Timetable algorithms) также находят широкое применение на практике: от составления расписания лекций в институте до формирования графика движения транспорта [22]. В процессе моделирования лучшие результаты показал TS, за ним, с небольшим (5 %) отрывом шел GA, за ним, с отрывом 15 %, шел SA.

Можно было бы и дальше продолжать эмпирическое сравнение различных методов случайного поиска, рассматривая все новые и

новые приложения, но на основе уже рассмотренных можно сделать некоторые выводы об особенностях применения этих методов.

4. Заключение

Как показывает практическое исследование, моделируемый отжиг является наиболее «хрупким» из рассмотренных методов, он более всего зависит от выбранных параметров моделирования, т.е. больше всех остальных нуждается в дополнительной настройке. Генетические алгоритмы, напротив, хотя при настройке они показывают лучшие (примерно на 10 - 15 %) результаты, но, даже выбирая параметры по умолчанию, можно добиться приемлемой точности.

По кругу проблем, к которым применяются эти методы, видно, что наиболее узкое применение (разумеется, в рамках ограниченной в начале, сложной, многопараметричной и т.д. области) имеют детерминистические алгоритмы, различные методы градиента и т.п. Далее идет моделируемый отжиг, затем в некоторых случаях применим поиск с запретами и, наконец, генетические алгоритмы применяются с очень широким кругом задач. Эта иерархия и определяет последовательность при выборе конкретного метода для решения практической задачи. То есть, прежде всего, необходимо попробовать применить детерминистические методы и, если они применимы, то велика вероятность того, что результат будет получен и быстрее и точнее. Если же они неприменимы, то надо переходить к следующему классу алгоритмов. Нужно помнить, что работа алгоритмов случайного поиска существенно зависит от подбора управляющих параметров, поэтому после выбора алгоритма стоит произвести некоторые исследования и попробовать различные схемы управления.

Одним из наиболее многообещающих подходов к решению оптимизационных задач в настоящее время является синтез методов случайного поиска с детерминированными алгоритмами. Это направление и должно стать темой дальнейших исследований.

СПИСОК ЛИТЕРАТУРЫ

1. Таха Х. Введение в исследование операций: в

- 2-х книгах, кн. 2, перев. с англ., М.: Мир, 1985.
2. Фишер Ф.Н. Проблемы идентификации в эконометрии. - М.: Финансы и статистика, 1978.
 3. Forrest S. and Mayer-Kress G. Genetic algorithms, nonlinear dynamical systems, and models of international security. In L. Davis, editor, Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.
 4. Davis L. Applying adaptive algorithms to epistatic domains. In 9th Int. Joint Conference on AI, 1985.
 5. Расстригин Л.А. Статистические методы поиска. М.: Наука, 1968.
 6. Goldberg. E. Genetic Algorithms in search, optimization and machine learning. Addison-Wesley, 1989.
 7. Жиглявский А.А., Жилинскас А.Г. Методы поиска глобального экстремума, М.: Наука, 1991.
 8. Rutenbar R.A. Simulated annealing algorithms: An overview. IEEE Circuits and Devices Magazine, January 1989.
 9. Wolpert D.H. and Macready W.G. No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation, vol. 1. no. 1, 1997
 10. Ивахненко А.Г. Системы эвристической самоорганизации в технической кибернетике. Киев: Техника, 1971.
 11. Зайцева Е.Н., Станкевич Ю.А. Некоторые современные методы решения оптимизационных задач, Материалы Второй международной конференции «Новые информационные технологии в образовании», 1996.
 12. Zade L. Fuzzy Sets. Information and Control, 1973.
 13. Harp S.A. and Samad T. Genetic synthesis of neural network architecture. In L. Davis, editor, Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.
 14. Fogel D.B. Applying evolutionary programming to selected traveling salesman problem, Cybernetics and Systems 24(1), 1993.
 15. Back T., Hoffmeister F. and Schwefel H.-P. A Survey of Evolution Strategies, in Proceedings of the 4th International Conference on Genetic Algorithms (ICCA IV), ed. R.K. Belew and L.B. Booker, Morgan Kaufman Publishers, Inc., San Diego, 1991.
 16. Mitchell M., Forrest S., and Holland J. The Royal Road for Genetic Algorithms: Fitness Landscapes and GA performance, in Proceedings of the First European Conference on Artificial Life, MIT Press, Cambridge, 1992.
 17. Keane A.J. A brief comparison of some evolutionary optimization methods, in Modern Heuristic Search Methods, ed. V. Rayward-Smith, I. Osmat, C. Reeves and G.D. Smith, J. Wiley, 1996
 18. Rintala T. Empirical comparison of stochastic algorithms, in Proceedings of the Second Nordic Workshop on Genetic Algorithms and their Applications, University of Vaasa, 1996.
 19. Manikas T.W., Cain G.T. Genetic Algorithms vs. Simulated Annealing: A Comparison of Approaches for Solving the Circuit Partitioning Problem, Technical Report No. 96-101, University of Pittsburgh, Dept. of Electric Engineering, 1996.
 20. Martello S., Toth P. Knapsack Problems: Algorithms and Computer Implementations. Wiley-Interscience, Chichester, UK, 1996.
 21. Battiti R., Tecchiolli G. Local Search with Memory: Benchmarking RTS, Operations Research Spectrum, 1995.
 22. Colorni A., Dorigo M. and Maniezzo V. A Genetic Algorithm to Solve the Timetable Problem. Technical Report No. 90-060, Politecnico di Milano, Italy, 1990.

COMPARISON OF DIFFERENT RANDOM SEARCH AND OPTIMIZATION TECHNIQUES

© 1999 I.A. Minakov

Institute for the Control of Complex Systems of Russian Academy of Sciences, Samara

The paper considers methods of random search and optimization, which are widely applied recently. After brief review of modern techniques of random search and optimization, the empirical analysis of different approaches to their usage on the basis of some examples from real life is presented. Practical recommendations for expedient applying of different methods depending on particular task are given on the basis of the analysis.