ISSN 1998-0663

№2(08)—2009

www.bi-network.ru/jbi/.

БИЗНЕС-ИНФОРМАТИКА

МЕЖДИСЦИПЛИНАРНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ ГУ-ВШЭ

BUSINESS INFORMATICS

Учредитель:
Государственный университет—Высшая школа экономики

Главный редактор
Никитин В.В.

Заместители главного редактора
Горбунов А.Р.
Ульянов М.В.

Редакционная коллегия
Абдульраб А. (Франция)
Авдошин С.М.
Алескеров Ф.Т.
Белов В.В.
Вирин Ф.Ю.
Грибов А.Ю.
Громов А.И.
Гюнтер Х. (Германия)
Каменнова М.С.
Калягин В.А.
Козырев О.Р.
Кузнецов С.О.
Мальцева С.В.
Миркин Б.Г. (Великобритания)
Моттль В.В.
Мулазани М. (Италия)
Пальчунов Д.Е.
Силантьев А.Ю.
Таратухин В.В.
Терзани С. (Италия)

В ЭТОМ НОМЕРЕ:

Криптография vs. Криптоанализ

Технологии мобильной рекламы

Маркетинг: комбинированная стратегия

Навстречу ИММОД-2009

БИЗНЕС-
ИНФОРМАТИКА

http://www.bi-network.ru/jbi/ **№2(08)-2009**

СОДЕРЖАНИЕ

Программная инженерия

С.М. Авдошин, А.А. Савельева
Криптоанализ и криптография:
история противостояния.....3

Анализ данных и интеллектуальные системы

И.Ю. Каширин, С.Ю. Семченков
Интерактивная аналитическая обработка данных
в современных OLAP-системах 12

Д.Н. Касымова
Разработка системы показа
мобильной контекстной рекламы 20

Математические методы и алгоритмы решения задач бизнес-информатики

М.В. Ульянов, О.А. Наумова
Комбинированный и волновой алгоритмы
решения задачи упаковки:
принципы построения и особенности 27

В.В. Белов
Структурная и параметрическая идентификация
линейной МНК-модели без решения системы
нормальных уравнений 34

Моделирование и анализ бизнес-процессов

М.Ш. Левин
Комбинированная схема формирования
стратегии маркетинга 42

Моделирование и анализ бизнес-процессов

В.Н. Сидоренко
Имитационное моделирование в науке и бизнесе:
подходы, инструменты, применение 52

Информационные системы и технологии в бизнесе

А.П. Шабанов,
Подход к оценке производительных ресурсов
информационных систем 58

Annotations 64

ИММОД-2009 67

БИЗНЕС-ИНФОРМАТИКА
№2(08)–2009

Междисциплинарный
научно-практический журнал
ГУ-ВШЭ

Подписной индекс издания
в каталоге агентства
«Роспечать» –72315

Главный редактор
Никитин В.В.

Учредитель:
Государственный университет-
Высшая школа экономики

Выходит 4 раза в год.

Редакция:

Зам. главного редактора
Горбунов А.Р.
Зам. главного редактора
Ульянов М.В.

Научный редактор
Лычкина Н.Н.

Технический редактор
Осипов В.И.

Дизайн обложки
Борисова С.Н.

Компьютерная вёрстка
Волков А.А.

Администратор веб-сайта
Проценко Д.С.
http://www.bi-network.ru/jbi/

Адрес редакции:
105679, г. Москва,
ул. Кирпичная, д. 33.
Тел.+7 (095) 771-32-38,
e-mail: bijournal@hse.ru

За точность приведённых сведений
и содержание данных,
не подлежащих открытой
публикации,
несут ответственность авторы

При перепечатке
ссылка на журнал
«Бизнес-информатика» обязательна

Тираж 500 экз.

Журнал отпечатан в типографии
ООО «ГЕО-ТЭК»
г. Красноармейск Московской обл.
Тел: (495) 993-1623

© Государственный университет-
Высшая школа экономики

КРИПТОАНАЛИЗ И КРИПТОГРАФИЯ: ИСТОРИЯ ПРОТИВОСТОЯНИЯ

С.М. Авдошин,
кандидат технических наук, доцент, руководитель отделения программной инженерии, заведующий кафедрой «Управление разработкой программного обеспечения» Государственного университета – Высшей школы экономики,
Адрес: 105118, Москва, Кирпичная 33/5, офис 532, Авдошин С.М.,
Тел: +7(495) 771-32-38 + 5151. E-mail: savdoshin@hse.ru
А.А. Савельева,
преподаватель кафедры «Управление разработкой программного обеспечения» Государственного университета – Высшей школы экономики,
E-mail: asavelieva@hse.ru

В статье рассмотрены основные этапы развития криптологии – науки, объединяющей криптографию и криптоанализ и ставшей в эпоху компьютеризации одной из наиболее активно развивающихся областей знаний. Показано, каким образом достижения в области взлома шифров влияют на прогресс в области криптографии, а успехи криптографов становятся катализатором для криптоаналитических исследований.

Ключевые слова: Криптоанализ, стойкость шифра, взлом, криптография, квантовые вычисления.

«Все любят разгадывать других, но никто не любит быть разгаданным», – эти слова французского писателя-моралиста Ларошфуко как нельзя лучше отражают сущность современной криптологии как соревнования методов криптографии и криптоанализа. Появление новых криптографических алгоритмов приводит к разработке методов их взлома. Результатом возникновения каждого нового метода криптоанализа является пересмотр оценок безопасности шифров, что, в свою очередь, влечет необходимость создания более стойких шифров. Таким образом, исторические этапы развития криптографии и криптоанализа неразрывно связаны.

Попытка криптоанализа называется *атакой*. Результаты криптоанализа могут варьироваться по степени практической применимости. Так, криптограф Л. Кнудсен предлагает следующую классификацию успешных исходов криптоанализа блочных шифров в зависимости от объёма и качества секретной информации, которую удалось получить:

- *полный взлом* – криптоаналитик извлекает секретный ключ;
- глобальная дедукция – криптоаналитик разрабатывает функциональный эквивалент исследуемого алгоритма, позволяющий зашифровывать и расшифровывать информацию без знания ключа;
- частичная дедукция – криптоаналитику удаётся расшифровать или зашифровать некоторые сообщения;
- информационная дедукция – криптоаналитик получает некоторую информацию об от крытом тексте или ключе.

Однако взлом шифра совсем не обязательно подразумевает обнаружение способа, применимого на практике для восстановления открытого текста по перехваченному зашифрованному сообщению. В научной криптологии другие правила. Шифр считается взломанным, если в системе обнаружено слабое место, которое может быть использовано для более эффективного взлома, чем *метод полного перебора ключей* («brute-force approach»). Допустим, для дешифрования текста методом полного перебора требуется перебрать 2^{128} возможных ключей; тогда изобретение способа, требующего для дешифрования 2^{110} операций по подбору ключа, будет считаться взломом. Такие способы могут требовать нереалистично больших

объёмов подобранного открытого текста или памяти ЭВМ. Под взломом понимается лишь подтверждение наличия уязвимости криптоалгоритма, свидетельствующее о том, что свойства надёжности шифра не соответствуют заявленным характеристикам. Как правило, криптоанализ начинается с попыток взлома упрощённой модификации алгоритма, после чего результаты распространяются на полноценную версию: прежде чем браться за взлом, например, 16-раундовой версии DES, естественно для начала попытаться взломать шифр с меньшим количеством раундов, чем указано в его спецификации (например, 8-раундовую версию шифра).

Два последних десятилетия ознаменовались резким ростом числа открытых работ по криптологии, а криптоанализ становится одной из наиболее активно развивающихся областей исследований. Появился целый арсенал математических методов, представляющих интерес для криптоаналитика. Кроме того, повышение производительности вычислительной техники сделало возможными такие типы атак, которые раньше были неосуществимы.

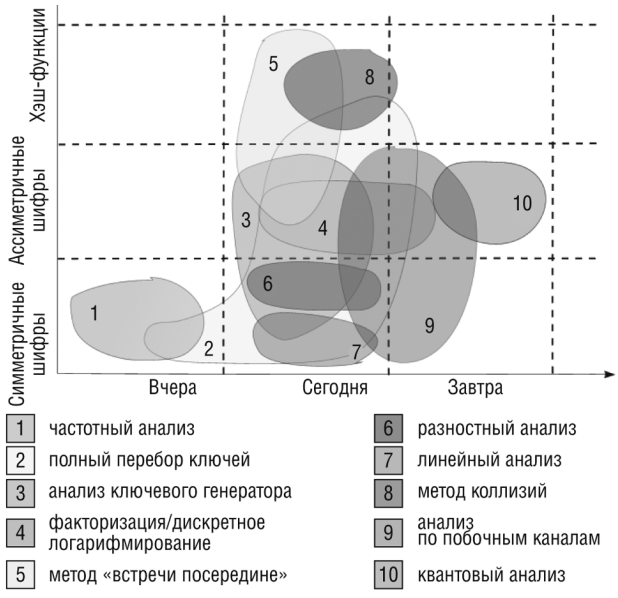


Рис. 1. Методы криптоанализа

На *рис.1* методы криптоанализа систематизированы по хронологии их появления и применимости для взлома различных категорий криптосистем. Горизонтальная ось разделена на временные промежутки: в область «вчера» попали атаки, которые успешно применялись для взлома шифров в прошлом; «сегодня» – методы криптоанализа, представляющие угрозу для широко используемых в настоящее время криптосистем; «завтра» – эффективно применяемые уже сегодня методы, значение которых в будущем может возрасти, а также методы, которые пока не оказали серьезного влияния на криптологию, однако со временем могут привести прорывам во взломе шифров. На вертикальной оси обозначены области применения методов криптоанализа: для взлома криптосистем с секретным ключом, открытым ключом или хеш-функций.

Частотный анализ

На протяжении веков дешифрованию криптограмм помогает частотный анализ появления отдельных символов и их сочетаний. Частотный криптоанализ использует статистические и лингвистические методы для получения дополнительной информации о ключе, а аналитические методы предполагают математическое изучение алгоритма шифрования. Вероятности появления отдельных букв в тексте сильно различаются. Для русского языка, например, буква «о» появляется в 45 раз чаще буквы «ф» и в 30 раз чаще буквы «э». Анализируя достаточно длинный текст, зашифрованный методом замены, можно по частотам появления символов произвести обратную замену и восстановить исходный текст. Кроме того, порядок букв в словах и фразах естественного языка подчиняется определенным статистическим закономерностям. Частотный анализ учитывает частоту появления различных буквосочетаний: например, пара стоящих рядом букв «ся» в русском языке более вероятна, чем «цы», а «оь» не встречается никогда. Для большинства естественных языков такая статистика документирована. Сегодня эти принципы широко применяются программах по подбору паролей. Если программа перебора вначале подбирает наиболее вероятные пароли, а менее вероятные оставляет на потом, то перебор сокращается в десятки и сотни раз.

Как было отмечено выше, каждый новый метод криптоанализа добавляет новые требования к алгоритмам шифрования. Так, частотный метод, в котором по распределению символов в шифртексте выдвигаются гипотезы о ключе шифрования, породил требование равномерного распределения символов в шифртексте.

Метод полного перебора

С появлением высокопроизводительной вычислительной техники у криптоаналитиков появилась возможность вскрывать шифры методом перебора ключей. В процессе криптоанализа приходится

перебирать миллиард ключей со скоростью тысяча ключей в секунду.

При осуществлении попытки атаки на основе только шифртекста криптоаналитику требуется анализировать выходные данные алгоритма и проверять их «осмысленность». В случае, когда в качестве объекта шифрования выступает графический файл или программа, задача определения «осмысленности» выходных данных становится очень трудной. Если известно, что открытый текст представляет собой предложение на естественном языке, проанализировать результат и опознать успешный исход дешифрования сравнительно несложно, тем более что криптоаналитик зачастую располагает некоторой априорной информацией о содержании сообщения. Задачу выделения осмысленного текста, т.е. определения факта правильной дешифрации, решают при помощи ЭВМ с использованием теоретических положений, разработанных в конце XIX века петербургским математиком А.А. Марковым, – *цепей Маркова*.

Атаки с использованием известного или подобранного открытого текста встречаются чаще, чем можно подумать. В среде криптоаналитиков нельзя назвать неслыханными факты добычи открытого текста шифрованного сообщения или подкупа лица, которое должно будет зашифровать избранное сообщение. Предположим, злоумышленнику известна одна или несколько пар (x, y). Пусть для простоты для любой пары (x, y) существует единственный ключ k, удовлетворяющий соотношению $E_k(x) = y$. Примем проверку одного варианта ключа $k \in K$ за одну операцию. Тогда полный перебор ключей потребует $|K|$ операций, где $|K|$ – число элементов в множестве. Если в качестве оценки трудоёмкости метода взять математическое ожидание

Можно подумать, что с ростом мощности компьютеров разрядность ключа, достаточная для обеспечения безопасности информации против атаки методом полного перебора, будет неограниченно расти. Однако это не так. Существуют фундаментальные ограничения вычислительной мощности, наложенные структурой вселенной: например, скорость передачи любого сигнала не может превышать скорость распространения света в вакууме, а количество атомов во Вселенной (из которых, в конечном счете, состоят компьютеры) огромно, но конечно. Так, например, в [5] описаны два фундаментальных ограничения:

1. *Предел, основанный на выделяемой Солнцем энергии.* Все вычисления потребляют энергию. Согласно принципам классической термодинамики и статистической механики, потребление энергии при осуществлении необратимого преобразования (вычисления) имеет порядок $k \cdot T$, где T – температура окружающей среды (по абсолютной шкале Кельвина), а k – постоянная Больцмана (равная $1{,}4 \cdot 10^{-23}$ Дж/°К). Мощность излучения Солнца составляет приблизительно $3{,}86 \cdot 10^{26}$ Вт; таким образом, за весь свой предполагаемый период существования – 10 млрд. лет, или $3 \cdot 10^{17}$ секунд – Солнце выделит около 10^{44} Дж). Предположим, температура окружающей среды $T = 10^{-6}$ градусов, тогда энергозатраты на одну операцию составляют $1{,}4 \cdot 10^{-29}$ Дж. Значит, количество вычислительных операций, которые можно осуществить с использованием всей выделяемой солнцем энергии, равно выделяемой мощности, поделенной на количество энергии, необходимой для осуществления одной операции, т.е. всего 10^{73} операций. Такое количество операций потребовалось бы на взлом ключа из 73 десятичных цифр (или около 250 бит) методом прямого перебора при грубом предположении, что для проверки одного значения ключа необходима всего одна операция (на самом деле – сотни операций). Для справки, количество атомов солнечной системы – около 10^{60}.

2. Предел, основанный на массе Земли. Масса Земли составляет порядка $6 \cdot 10^{24}$ кг. Масса протона – $1{,}6 \cdot 10^{-27}$ кг, т.е. Земля содержит приблизительно $4 \cdot 10^{51}$ протонов. Сопоставим каждому протону отдельный компьютер и примем за скорость выполнения операции на таком компьютере время, за которое луч света проходит расстояние, равное диаметру этого протона

$$\frac{10^{-15}\ \text{м}}{3 \cdot 10^{10}\ \text{м/с}}\,\text{с} = \frac{1}{3} \cdot 10^{-25}.$$

Таким образом, каждый компьютер может выполнять $3 \cdot 10^{25}$ операций в секунду. Если все эти компьютеры будут работать параллельно, их суммарное быстродействие составит $4 \cdot 10^{51} \cdot 3 \cdot 10^{25}$ операций в секунду, т.е. 10^{77} операций в секунду. Возраст Вселенной приблизительно 10 млрд. лет, т.е. $3 \cdot 10^{17}$ секунд. За весь период существования Вселенной такие гипотетические компьютеры размером с протон смогли бы выполнить $3 \cdot 10^{94}$ операций. При описанных в п. 1 предположений относительно количества операций, необходимых на проверку значения ключа, такое количество операций позволит взломать ключ длиной 95 десятичных цифр, или 320 бит.

Таким образом, минимальный размер ключа, необходимого для защиты информации от атак злоумышленника, будет расти по мере повышения быстродействия компьютеров; тем не менее, приведённые выше вычисления показывают, что существует возможность выбрать такую длину ключа, что атаку методом полного перебора будет осуществить в принципе невозможно, вне зависимости от повышения вычислительной мощности компьютеров или успехов в области классической теории алгоритмов.

случайной величины, соответствующей числу опробований до момента обнаружения использованного ключа, то мы получим $|K|/2$ операций. Кроме того, алгоритм полного перебора допускает распараллеливание, что позволяет значительно ускорить нахождение ключа.

Атака по ключам

Одной из причин ненадёжности криптосистем является использование слабых ключей. Фундаментальное допущение криптоанализа, впервые сформулированное О. Кирхгоффом, состоит в том, что секретность сообщения всецело зависит от ключа, т.е. весь механизм шифрования, кроме значения ключа, известен противнику (секретность алгоритма не является большим препятствием: для определения типа программно реализованного криптографического алгоритма требуется лишь несколько дней инженерного анализа исполняемого кода). Слабый ключ – это ключ, не обеспечивающий достаточного уровня защиты или использующий в шифровании закономерности, которые могут быть взломаны. Обычно считается, что алгоритм шифрования должен по возможности не иметь слабых ключей. Если это невозможно, то количество слабых ключей должно быть минимальным, чтобы уменьшить вероятность случайного выбора одного из них. Тем не менее, все слабые ключи должны быть заранее известны, чтобы их можно было отбраковать в процессе создания ключа.

Генераторы случайных чисел – ещё одно место, в котором часто ломаются криптографические системы. Это означает, что, если для генерации ключей используется криптографический слабый алгоритм, независимо от используемого шифра вся система будет нестойкой. Качественный ключ, предназначенный для использования в рамках симметричной криптосистемы, представляет собой случайный двоичный набор. Если требуется ключ разрядностью n, в процессе его генерации с одинаковой вероятностью должен получаться любой из $2''$ возможных вариантов. Генерация ключей для асимметричных криптосистем – процедура более сложная, т.к. ключи, применяемые в таких системах,

Известно два направления в организации параллельного вычисления ключа методом полного перебора [2].

1. *Построение конвейера.* Предположим, цель злоумышленника заключается в осуществлении атаки на основе открытого текста. Тогда ему необходимо последовательно проверять истинность соотношения $E_k(x) = y$ для всевозможных значений k при известной паре (x, y). Один шаг алгоритма можно представить в виде детерминированной цепочки простейших операций: $O_1, O_2,..., O_N$.

Возьмём N процессоров $A_1, A_2,..., A_N$ и положим, что i-й процессор выполняет три одинаковые по времени операции:

1) приём данных от $(i–1)$-го процессора;

2) выполнение операции O_i;

3) передача данных следующему $(i+1)$-му процессору.

Тогда конвейер из N последовательно соединённых, параллельно и синхронно работающих процессоров работает со скоростью $v/3$, где v – скорость выполнения одной операции процессором.

2. *Распределенный поиск.* Множество ключей K разбивается на непересекающиеся подмножества $K_1, K_2,..., K_Q$. Система из Q машин перебирает ключи так, что i-я машина осуществляет перебор ключей из множества K_i, $i = \overline{1,Q}$. Как только одна из машин находит ключ, система прекращает работу. Сложность в изложенном подходе – организация деления ключевого множества. Если организовать поиск ключа, чтобы при очередном опробовании каждый из процессоров стартовал со случайной точки, время опробования увеличится, но схема упростится. Среднее число шагов опробования N процессорами ключей из множества K составит $|K|/N$.

Реализация такого параллелизма допускает различные решения. Самое очевидное – создание компьютерного вируса для распространения программы-взломщика в глобальной сети. Программа подключается к серверу, получает от него набор ключей для перебора и после окончания работы возвращает результат. Вирус должен использовать периоды простоя компьютера (по данным исследований, компьютер простаивает 70–90% времени) для осуществления перебора по множеству ключей, создание вируса, незаметно для пользователя устанавливающего на подключённый к сети компьютер программу, способную осуществлять дешифрование сообщения путём перебора ключей. С развитием сетей (в частности, Интернета), стало возможным эффективно использовать этот метод. Подтверждение этому – вскрытие RC5-64 (блочного шифра компании RSA, использующего 64-битный ключ). Стартовавший в 1997 г. на сайте www.distributed.net проект «распределённого взлома» (в нём на добровольной основе приняли участие более 300 тысяч пользователей), успешно завершён за пять лет (1757 дней). За это время было перебрано 85% всего пространства ключей. Такой подход применим не только для взлома шифров, но и, например, для подбора двух текстов, имеющих одинаковое значение хеш-функции.

должны обладать определёнными математическими свойствами. Например, в случае системы RSA модуль шифрования представляет собой произведение двух больших простых чисел.

Исследования компании Counterpane, президентом которой является известный криптограф Б. Шнайер, показали, что определённые генераторы случайных чисел могут быть надёжными при использовании с одной целью, но ненадёжными для другой; обобщение анализа надёжности опасно.

Метод «встречи посередине»

Другой популярный метод криптоанализа – алгоритм «встречи посередине» – поддается эффективному распараллеливанию. Например, для логарифмирования в группе порядка p при параллельной работе n процессоров, где $\underline{n \times p}$, время работы алгоритма уменьшается в n раз.

Данный метод криптоанализа основан на *«парадоксе дней рождения»*. Пусть нам нужно найти ключ k по известному открытому тексту x и криптограмме y. Если множество ключей криптоалгоритма замкнуто относительно композиции, т.е. для любых ключей k' и k'' найдется ключ k такой, что результат шифрования любого текста последовательно на k' и k'' равен результату шифрования этого же текста на k, т.е. $E_{k''}(E_{k'}, x) = E_k(x)$, то можно воспользоваться этим свойством. Поиск ключа k сведем к поиску эквивалентной ему пару ключей k', k''. Для текста x построим базу данных, содержащую случайное множество ключей k' и соответствующих криптограмм $w = E_k(x)$, и упорядочим её по криптограммам w. Объём базы данных выбираем

$$O(\sqrt{|\{k'\}|}),$$

где $|\{k'\}|$ – мощность множества ключей k'.

Затем подбираем случайным образом ключи k'' для расшифровки текстов y и результат расшифрования $v = E_{k''}(y)$ сравниваем с базой данных. Если текст v окажется равным одной из криптограмм w, то ключ $k'k''$ эквивалентен искомому ключу k.

Обозначим $\alpha = |\{k\}|$ – общее количество возможных ключей k. Временная сложность метода составляет

$$O(\sqrt{\alpha}\log\alpha).$$

Множитель $\log\alpha$ учитывает сложность сортировки. Требуемая память равна

$$O(\sqrt{\alpha}\log\alpha) \text{ бит, или } O(\sqrt{\alpha}) \text{ блоков}$$

(предполагается, что длина блока и длина ключа различаются на ограниченную константу).

Алгоритм является вероятностным. Однако существуют и детерминированный аналог этого алгоритма «giant step – baby step» с такой же сложностью, предложенный американским математиком Д. Шенксом.

Криптоанализ симметричных шифров

Наибольший прогресс в разработке методов раскрытия блочных шифров был достигнут в самом конце XX века и в основном связан с появлением в начале 90-х годов двух методов – *разностного (дифференциального) криптоанализа и линейного криптоанализа*.

Метод разностного анализа сочетает в себе обобщение идеи общей линейной структуры с применением вероятностно-статистических методов исследования. Этот метод относится к атакам по выбранному открытому тексту. Хотя Д. Коппер-смит утверждает, что разностный криптоанализ был известен команде разработчиков DES алгоритма еще в начале 70-х годов, официальной датой появления этого метода считается 1990 г., а первенство в разработке признано за израильскими математиками Э. Бихамом и А. Шамиром. Разностный анализ основан на использовании неравновероятности в распределении значений разности двух шифртекстов, полученных из пары открытых текстов, имеющих некоторую фиксированную разность. Отметим, что разностный анализ применим и для взлома хеш-функций.

Подобно разностному анализу, линейный криптоанализ является комбинированным методом, сочетающим в себе поиск линейных статаналогов для уравнений шифрования, статистический анализ имеющихся открытых и шифрованных текстов, использующий также методы согласования и перебора. Этот метод исследует статистические линейные соотношения между отдельными координатами векторов открытого текста, соответствующего шифртекста и ключа, и использует эти соотношения для определения статистическими методами отдельных координат ключевого вектора.

На сегодняшний день метод линейного криптоанализа позволил получить наиболее сильные результаты по раскрытию ряда итерационных систем блочного шифрования, в том числе и системы DES. Метод линейного криптоанализа в неявном виде

появился еще в работе С. Мёрфи в 1990 г., где он успешно применялся при анализе системы блочного шифрования FEAL. В 1992 г. М. Мацуи формализовал этот подход, а позже успешно применил его к анализу криптоалгоритма DES.

В 2001 г. на смену DES и Triple DES пришёл стандарт AES (Advanced Encryption Standard), действующий и по сей день. Шифр AES основан на алгоритме Rijndael, разработанном бельгийцами Д. Дейменом и В. Райменом.

В [3] рассматриваются вопросы устойчивости алгоритмов ГОСТ 28147-89 и AES к линейному и разностному методам криптоанализа. Дать оценку устойчивости алгоритма ГОСТ к конкретным видам криптоанализа невозможно без спецификации узлов замен, которые не зафиксированы в стандарте, но существенно влияют на качество шифра. Исследования близких по архитектуре шифров с заданными таблицами подстановок показали, что криптоанализ шифра с 16 раундами требует очень большого числа исходных данных, хотя в принципе осуществим, а при 20–24 раундах становится теоретически бесполезным. Предусмотренных ГОСТом 32-х раундов шифрования хватает с запасом, чтобы успешно противостоять указанным видам криптоанализа. Шифр Rijndael, по оценкам разработчиков, уже на четырёх раундах шифрования приобретает достаточную устойчивость к линейному и разностному методам. Согласно спецификации, в шифре предусмотрено 10–14 раундов, а теоретической границей, за которой эти виды криптоанализа теряют смысл, является рубеж в 6–8 раундов в зависимости от размера блока. Таким образом, Rijndael устойчив к указанным видам криптоанализа с определенным запасом.

Криптоанализ асимметричных шифров

Практически все используемые алгоритмы асимметричной криптографии основаны на задачах факторизации (например, известная криптосистема RSA) и дискретного логарифмирования в различных алгебраических структурах (схема электронно-цифровой подписи Эль-Гамаля). Несмотря на то, что принадлежность этих задач к классу *NP*-полных задач не доказана, на сегодняшний день не найден полиномиальный алгоритм их решения. Для криптоанализа асимметричных криптосистем можно применять универсальные методы – например, метод «встречи посередине». Однако есть и другие методы, учитывающие специфику систем с открытым ключом. Они заключаются в решении математической задачи, положенной в основу асимметричного шифра. С того момента, как У. Диффи и М. Хеллман в 1976 г. предложили концепцию криптографии с открытым ключом, задачи факторизации целых чисел и дискретного логарифмирования стали объектом пристального изучения для математиков всего мира. За последние годы в этой области наблюдался значительный прогресс. Подтверждением тому может служить следующий казус: в 1977 г. Р. Ривест заявил, что разложение на множители 125-разрядного числа потребует 40 квадриллионов лет, однако уже в 1994 г. было факторизовано число, состоящее из 129 двоичных разрядов!

Задача дискретного логарифмирования считается более сложной, чем задача факторизации. Если будет найден полиномиальный алгоритм её решения, станет возможным и разложение на множители (обратное не доказано).

Последние достижения теории вычислительной сложности показали, что общая проблема логарифмирования в конечных полях не может считаться достаточно прочным фундаментом. Наиболее эффективные на сегодняшний день алгоритмы дискретного логарифмирования имеют уже не экспоненциальную, а субэкспоненциальную временную сложность. Это алгоритмы «index-calculus», использующие факторную базу. Первый такой алгоритм был предложен Л. Адлеманом и имеет временную сложность

$$L_p\left[\frac{1}{2}; c\right].$$

При вычислении дискретного логарифма в простом поле $\mathbf{Z}_p$

$$L_N\left[\gamma; c\right] = e^{(c+o(1))(\log N)^{\gamma}(\log\log N)^{1-\gamma}},$$

где $0 < y < 1$, $c = \text{const}$, $c > 0$.

На практике алгоритм Адлемана оказался недостаточно эффективным; Д. Копперсмит, А. Одлыжко и Р. Шреппель предложили алгоритм дискретного логарифмирования COS с эвристической оценкой сложности

$$L_p\left[\frac{1}{2}; 1\right] \text{ операций.}$$

Алгоритм решета числового поля, предложенный О. Широкауэром, при работает эффективнее различных модификаций метода COS; его временная сложность составляет порядка

$L_p\left[\frac{1}{3};(64/9)^{1/3}\right]$ арифметических операций.

Ряд успешных атак на системы, основанные на сложности дискретного логарифмирования в конечных полях, привёл к тому, что стандарты ЭЦП России и США, которые были приняты в 1994 г. и базировались на схеме Эль-Гамаля, в 2001 г. были обновлены: переведены на эллиптические кривые. Схемы ЭЦП при этом остались прежними, но в качестве чисел, которыми они оперируют, теперь используются не элементы конечного поля $GF(2'')$ или $GF(p)$, а эллиптические числа – решения уравнения эллиптических кривых над указанными конечными полями. Роль операции возведения числа в степень в конечном поле в обновленных стандартах выполняет операция взятия кратной точки эллиптической кривой – «умножение» точки на целое число. Надлежащий выбор типа эллиптической кривой позволяет многократно усложнить задачу взлома схемы ЭЦП и уменьшить рабочий размер блоков данных. Старый российский стандарт ЭЦП оперирует 1024-битовыми блоками, а новый, основанный на эллиптических кривых, – 256-битовыми, и при этом обладает большей стойкостью.

Алгоритмов, выполняющих дискретной логарифмирование на эллиптических кривых в общем случае хотя бы с субэкспоненциальной сложностью, на сегодняшний день не существует. Тем не менее, известны работы И.А. Семаева, в одной из которых рассматривается метод, идейно близкий методам логарифмирования в конечном поле Адлемана. В другой работе для эллиптических кривых специального вида (накладываются некоторые условия на модуль арифметики и на мощность группы точек) Семаев указал способ сведения с полиномиальной сложностью задачи логарифмирования в группе точек эллиптической кривой к задаче логарифмирования в некотором расширении простого поля. При этом используется так называемое спаривание Вейля, после чего можно применять известные субэкспоненциальные методы. Аналогичные результаты опубликованы за рубежом.

Криптоанализ хеш-функций

Основная атака на хеш – это метод коллизий [2]. Пусть M и M' – сообщения, H – хеш-функция, а ЭЦП представляет собой некоторую функцию S от хеша сообщения: $C = S[H(M)]$. Законный обладатель пары «открытый ключ – секретный ключ» готов подписать сообщение M, но злоумышленник заинтересован в получении подписи под сообщением M'. Если M' выбрано так, что $H(M) = H(M')$, злоумышленник может предъявить пару (M', C): атака удалась. Реализовать подбор такого сообщения можно методом, основанном на упомянутом

Известно, что если считать, что дни рождения распределены равномерно, то в группе из 23 человек с вероятностью 0,5 у двух человек дни рождения совпадут. В общем виде этот парадокс формулируется так: если $a\sqrt{b}$ предметов выбираются с возвращением из некоторой совокупности размером b, то вероятность того, что два из них совпадут, равна

$$1-e^{-\frac{a^2}{2}}$$

(в описанном частном случае b = 365 – количество дней в году, $a\sqrt{b}$ = 23, т.е. a ≈ 1,204).

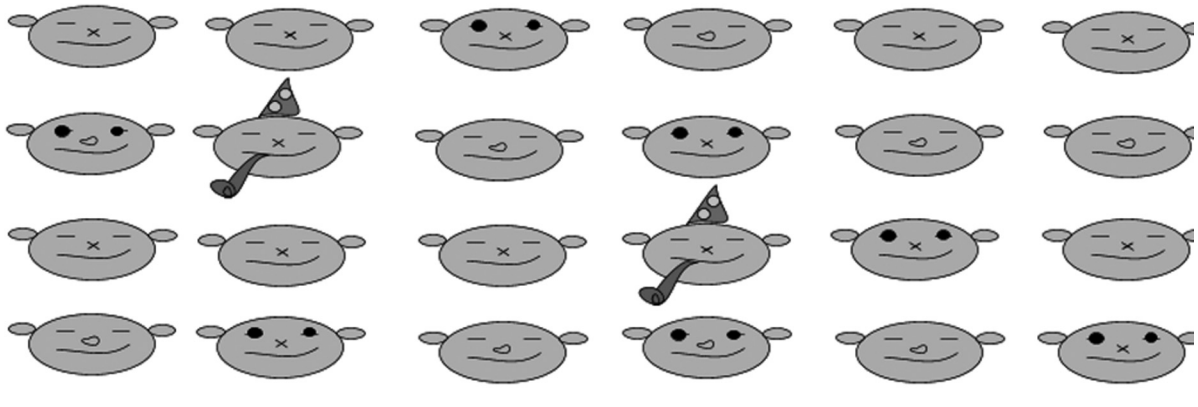


Рис. 2. Парадокс дней рождения

выше «парадоксе дней рождения». Варьируя интервалы, шрифты, формат и т.п., злоумышленник получает n пар вариантов M и M' без изменения их смысла. Сообщения $M_1,..., M_n$ отличаются слабо, а их хеш-функции – значительно, т.е. можно считать, что значения хеш-функций выбираются случайно, равновероятно и независимо друг от друга. Тогда при $n = t\sqrt{N}$ ($t > 0$ – некоторая константа, N – мощность множества всевозможных хеш-функций) вероятность того, что имеется пара сообщений M и M', для которых $H(M) = H(M')$, вычисляется по формуле

$$1 - e^{-\frac{t^2}{2}}.$$

Этот метод криптоанализа породил требования устойчивости к коллизиям для хеш-функций.

Атаки по сторонним, или побочным, каналам используют информацию, которая может быть получена с устройства шифрования и не является при этом ни открытым текстом, ни шифртекстом. Такие атаки основаны на корреляции между значениями физических параметров, измеряемых в разные моменты во время вычислений, и внутренним состоянием вычислительного устройства, имеющим отношение к секретному ключу. Этот подход менее обобщённый, но зачастую более мощный, чем классический криптоанализ.

В последние годы количество криптографических атак, использующих слабости в реализации и размещении механизмов криптоалгоритма, резко возросло. Противник может замерять время, затрачиваемое на выполнение криптографической операции, анализировать поведение криптографического устройства при возникновении ошибок вычисления. Другой подход предполагает отслеживание энергии, потребляемой смарт-картой в процессе выполнения операций с секретным ключом (например, расшифрования или генерации подписи). Побочную информацию собрать несложно – сегодня выделено более десяти побочных каналов, в т.ч. электромагнитное излучение, ошибки в канале связи, кэш-память и световое излучение. Подробное описание перечисленных типов атак можно найти в материалах доклада А.Е. Жукова на конференции РусКрипто-2006 [4], использованных при подготовке данного раздела.

Нанотехнологии в криптоанализе

С помощью квантового компьютера можно проводить вычисления, не реализуемые на классических компьютерах. В 1994 г. П. Шор открыл так называемый «ограниченно-вероятностный» алгоритм факторизации, позволяющий разложить на множители число N за полиномиальное от размерности задачи время $O[(\log N)^3]$. Алгоритм Шора разложения чисел на множители – главное достижение в области квантовых вычислительных алгоритмов. Это не только крупный успех математики. Именно с этого момента началось усиленное финансирование работ по созданию квантовых компьютеров.

Важно отметить, что алгоритм Шора чрезвычайно прост и довольствуется гораздо более скромным аппаратным обеспечением, чем нужное для универсального квантового компьютера. Квантовое устройство для разложения на множители будет построено, вероятно, задолго до того, как весь диапазон квантовых вычислений станет технологически осуществимым. На сегодня есть конкретные результаты. Так, IBM продемонстрировала использование созданного в лабораториях компании семикубитового квантового компьютера для факторизации чисел по алгоритму Шора. Хотя решённая им задача вряд ли способна поразить воображение (компьютер верно определил, что делителями числа 15 являются числа 5 и 3), это самое сложное вычисление за всю историю квантовых компьютеров.

Заключение

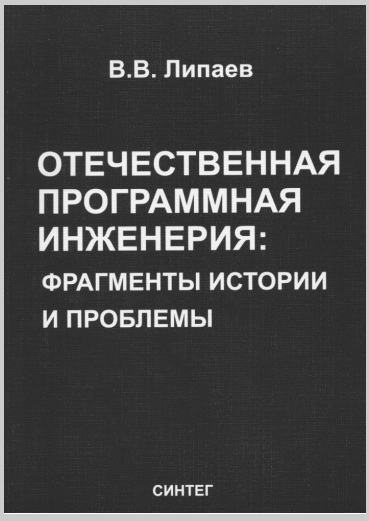
Возникает вопрос: если прогресс в области разработки новых методов взлома велик, почему мы продолжаем использовать криптосистемы, чья стойкость постоянно снижается? Ещё во времена Второй Мировой войны основоположник современной криптографии Клод Шеннон доказал существование принципиально не раскрываемых шифров – совершенно секретных систем, в которых ключ, «накладываемый» на текст, не может использоваться повторно, а его размер больше либо равен объёму текста.

Дело в том, что использование способа шифрования, получившего название *«одноразовых блокнотов»*, в большинстве случаев оказывается слишком дорогим и неоправданным. Нет смысла бороться за устойчивость системы защиты информации к взлому ниже некоторой «фоновой» вероятности, когда на другой чаше весов оказываются такие характеристики криптосистемы, как стоимость, сложность реализации и скорость доступа к зашифрованному тексту. Выбор необходимой степени защиты – это

поиск компромисса между уровнем безопасности и расходами на ее обеспечение. Таким образом, разработка новых методов криптоанализа с последующей публикацией и открытым обсуждением результатов является основным двигателем современной криптографии. ■

Литература

1. Ростовцев А.Г., Михайлова Н.В. Методы криптоанализа классических шифров // [Электронный ресурс] Сайт Санкт-Петербургского Государственного Политехнического Университета, 1998. URL: http://www.ssl.stu.neva.ru/psw/crypto/rostovtsev/cryptoanalysis.html (дата обращения: 14.04.09).
2. Грушо А.А., Применко Э.А., Тимонина Е.Е. Анализ и синтез криптоалгоритмов. Курс лекций. – Йошкар-Ола: Мар. фил. Моск. открытого соц. ун-та, 2000. – 110 с.
3. Материалы ассоциации Рускрипто: архивы научно-исследовательского семинара «Защита информации: аспекты теории и вопросы практических приложений» под рук. А.Е.Жукова и ежегодных конференций Рускрипто // [Электронный ресурс] Сайт ассоциации Рускрипто. URL: http://ruscrypto.ru/sources/ (дата обращения: 14.04.09).
4. Dam K.W., Lin H. S. Cryptography's Role in Securing the Information Society. National Academy Press. Washington, D.C. 1996.



Издательство «Синтег» выпустило новую книгу
Владимира Васильевича Липаева,
профессора кафедры управления программной инженерии ГУ-ВШЭ и главного научного сотрудника Института системного программирования РАН
«Отечественная программная инженерия: фрагменты истории и проблемы».

В монографии проанализированы этапы отечественной истории развития вычислительной техники с акцентом на методы и процессы программирования. Первая глава отражает развитие в стране автоматизации программирования в 50–60-е гг. Представлены процессы, начальные проекты отечественной вычислительной техники, развитие программирования и роль ведущих специалистов, заложивших основы в этой области. Выделены особенности развития специализированных вычислительных машин и программирования для оборонных систем реального времени. Формированию программной инженерии в 70-е гг. посвящена вторая глава. В третьей главе отражено развитие программной инженерии в 80-е гг. Изложена история развития экономики, методов и процессов программной инженерии в 70–80-е гг. Значительное внимание уделено реализации ПРОМЕТЕЙ-технологии программной инженерии для создания крупных комплексов программ реального времени оборонных систем. В четвертой главе подведены итоги развития программной инженерии и формирования ее методологии. Представлены проблемы расширения состава и совершенствования международных стандартов и инструментария программной инженерии, а также проблемы обучения методологией программной инженерии студентов и специалистов.

Книга предназначена для специалистов по вычислительной технике и программной инженерии, студентов и аспирантов, интересующихся историей развития и проблемами отечественной науки и техники в этой области.

ИНТЕРАКТИВНАЯ АНАЛИТИЧЕСКАЯ ОБРАБОТКА ДАННЫХ В СОВРЕМЕННЫХ OLAP-СИСТЕМАХ

И.Ю. Каширин,
доктор технических наук, профессор кафедры «Вычислительная и прикладная математика» Рязанского государственного радиотехнического университета,
Адрес: 390005, г. Рязань, ул. Гагарина, 59/1, кафедра «Вычислительная и прикладная математика», Каширин И.Ю. Тел: 4-912-214-667. E-mail: igor_kashirin@mail.ru
С.Ю. Семченков,
ассистент кафедры «Теоретическая и прикладная механика» Рязанского государственного радиотехнического университета,
E-mail: amfibius@mail.ru

В статье рассмотрены аспекты обработки данных с использованием многомерного анализа. Приведены основные понятия многомерной модели, рассмотрена классификация OLAP систем. Освещены вопросы совместного использования OLAP и OLTP систем, проведен сравнительный анализ существующих систем.

Ключевые слова: OLAP-системы. Многомерная модель обработки данных. Интерактивный анализ данных.

Введение

В области технологий обработки данных существует два взаимодополняющих друг друга направления развития:

- ✦ системы, ориентированные на операционную обработку данных – системы обработки данных (СОД);
- ✦ системы, ориентированные на анализ данных – системы поддержки принятия решений (СППР).

В недавнем прошлом, когда говорилось о стремительном росте числа реализаций информационных систем, прежде всего, имелись в виду системы, предназначенные для оперативной обработки данных. На первых этапах автоматизации требуется структурировать процессы повседневной рутинной обработки данных. На это ориентированы традиционные СОД. Для построения таких систем используются реляционные системы управления базами данных (РСУБД), такие как Oracle [1], Microsoft SQL Server [2] , MySQL [3], PostgreSQL [4] и другие.

Однако у такого подхода есть и недостатки, связанные с наличием нескольких источников данных. Основная проблема – необходимость согласования данных в разных источниках информации. При этом в лучшем случае проводится репликация баз данных (БД) как процесс копирования информации из одной базы в несколько других.

Необходимо учитывать следующие особенности репликации:

1) репликация – ресурсоемкий процесс, при котором большая нагрузка приходится на центральный процессор, оперативную память, сетевые каналы;

2) при большом количестве информации временные затраты на репликацию зачастую не оправдывают полученных результатов даже при достаточном количестве системных ресурсов.

С другой стороны, отсутствие согласованности информации приводит к разрозненности данных при их дальнейшей обработке. В любом случае, одни и те же данные многократно повторяются в различных СОД.

Таким образом, с ростом количества данных возникает противоречие: информация есть и её слишком много, в то же время она не структурирована, не согласована, разрознена, не всегда достоверна, её практически невозможно найти в СОД и получить. На разрешение этого противоречия нацелена технология OLAP.

OLAP (On-Line Analytical Processing – интерактивная аналитическая обработка данных) [5] – один из способов представления и анализа данных. При этом информация представляется в виде многомерного куба с возможностью произвольного манипулирования ею. Многомерные модели рассматривают данные либо как факты с соответствующими численными параметрами, либо как текстовые измерения, которые характеризуют эти факты.

Актуальность применения OLAP систем заключается в том, что они позволяют работать с данными в терминах предметной области без знания архитектуры хранения информации [6]. Кроме того, OLAP технология позволяет поддерживать различные типы бизнес-правил, использующиеся на предприятиях.

Многомерная модель для интерактивной аналитической обработки данных

Концепция OLAP-систем основана на следующих понятиях [7] (*рис. 1*).

1) Многомерный куб (n – число измерений многомерного куба) – это многомерная структура, состоящая из множества ячеек и хранящая взаимосвязанные данные, описывающие предметную область или её составляющую. Ячейка является атомарной структурой куба и при отображении располагается внутри него. Многомерный куб может создаваться и храниться как на физическом уровне представления данных (многомерные OLAP системы), так и на концептуальном уровне (виртуально) с помощью схем «звезда» или «снежинка» (реляционные OLAP системы).

2) Показатель q_j (j = 1...p, p – количество показателей) – это типизированная величина, которая является предметом анализа (например, показателями являются величина закупок, величина продаж, посещаемость сайта и т.д.). Один многомерный куб может содержать несколько показателей, причём каждому конкретному значению показателя соответствует единственная ячейка многомерного куба.

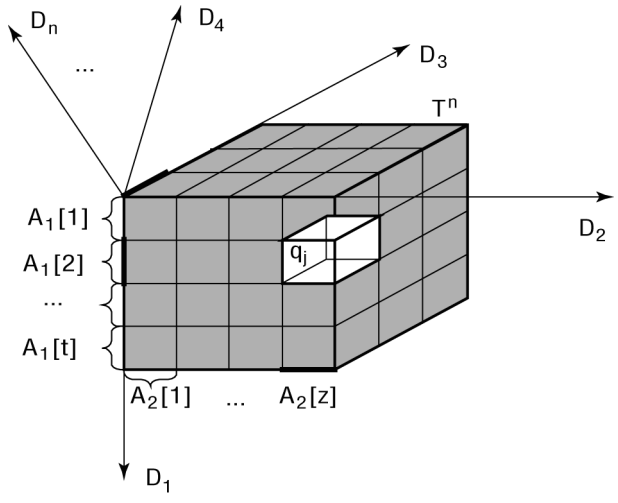


Рис. 1. Многомерная модель данных

3) Измерение D_i (i = 1...n) – это множество объектов (называемых членами измерения или значениями измерения) одного или нескольких типов, организованных в виде иерархической структуры и обеспечивающих информационный контекст типизированного показателя. Например, измерение «товар» может состоять из следующих значений: «бытовая техника», «компьютеры», «мобильные телефоны» и т.д. Значения, связанные с измерением, характеризуют какое-либо классификационное свойство сущностей предметной области. Совокупность измерений образует множество измерений $D = \{D_1, D_2,..., D_n\}$. На *рис. 1.1* $\{A_1[1], A_1[2],...A_1[t]\}$ – множество значений измерения A_1 (t – количество элементов измерения A_1), $\{A_2[1],... A_2[z]\}$ – множество значений измерения A_2 (z – количество элементов измерения A_1). Измерение принято отображать в виде ребра многомерного куба, а члены измерения – в виде точек или участков, откладываемых на ребре.

Множество всех измерений куба образует систему координат представляемого пространства данных. Значения измерений играют роль индексов, находящихся в ячейках куба. Особенностью измерений является их иерархическая структура, которая используется для агрегации и детализации значений показателей. Наиболее часто используются следующие два вида иерархий [8]:

✦ *Сбалансированные* – иерархии, в которых число уровней определено её структурой неизменно, и все ветви имеют одинаковую глубину. Для формирования сбалансированной иерархии необходимо наличие связи «один-ко-многим» между объектами

менее детального уровня по отношению к объектам более детального уровня. Например, сбалансированной является трехуровневая иерархия «направления деятельности – категории товаров – производители товаров» (*рис. 2*).

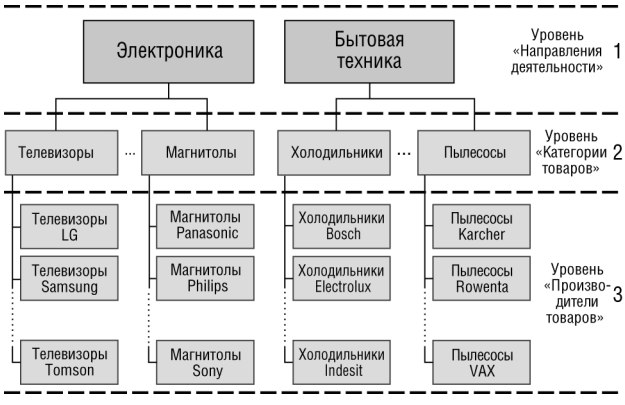


Рис. 2. Пример сбалансированной иерархии

✦ *Несбалансированные* – иерархии, в которых число уровней может быть изменено, некоторые уровни логически не одинаковы, и глубина ветвей иерархии может быть разной. Все объекты несбалансированной иерархии принадлежат одному типу. Например, несбалансированной является иерархия компаний-поставщиков (*рис. 3*). Все элементы этой иерархии имеют тип «поставщик».

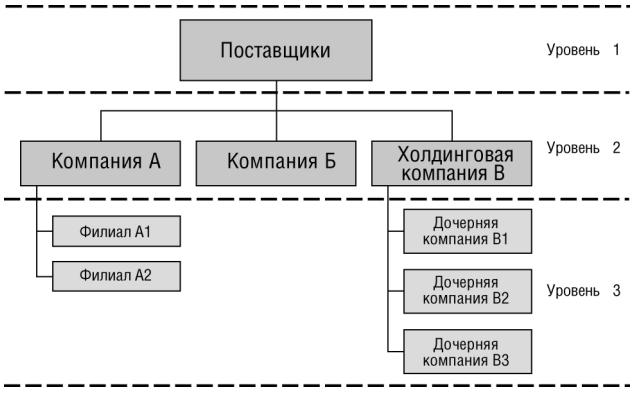


Рис. 3. Пример несбалансированной иерархии

4) Элемент измерения – уровень в иерархии значений координат измерения многомерного куба. В приведённом примере измерение содержит 3 элемента («направления деятельности», «категории товаров», «производители товаров»), каждый из них характеризуется номером уровня в иерархии.

5) Атрибут измерения. Атрибуты измерения описывают его элементы, но не образуют иерархии. Например, для каждого поставщика можно хранить адрес его электронной почты. Различные уровни измерения могут образовывать различные атрибуты. Одни атрибуты могут относиться ко всем уровням измерений, другие – лишь к определенным.

В отличие от традиционных СОД, OLAP системы должны иметь возможность работы с тремя типами данных [9]: ретроспективными, оперативными и прогнозируемыми.

Классификация OLAP систем

OLAP системы [10] можно классифицировать по следующим параметрам:

1) по способу организации данных в многомерных кубах;

2) по способу хранения на физическом уровне;

3) по месту нахождения OLAP машины, рассчитывающей многомерные кубы;

4) по степени готовности к применению для конечного пользователя.

В различных OLAP системах используются два основных варианта организации данных [11]: гиперкубическая и поликубическая модели. В гиперкубической модели все показатели должны определяться одним и тем же набором измерений. Поликубическая модель предполагает, что в БД определяется несколько гиперкубов с различной размерностью и с различными измерениями в качестве их граней.

Как исходные, так и агрегированные данные могут храниться либо в реляционных, либо в многомерных структурах. Применяются три способа хранения данных на физическом уровне [12].

1) MOLAP (Multidimensional OLAP – многомерный OLAP) – исходные и агрегированные данные хранятся в многомерных структрурах. MOLAP напрямую поддерживает многомерные представления данных с помощью многомерного механизма хранения. Скорость вычисления агрегатных значений одинакова для любого из измерений. Однако в этом случае многомерная база данных оказывается избыточной, так как многомерные данные полностью содержат исходные реляционные данные. Достоинства и недостатки MOLAP архитектуры приведены в *табл. 1.*

2) ROLAP (Relational OLAP – реляционный OLAP) – исходные данные остаются в той же реляционной базе данных, где они изначально и находились. Агрегированные данные помещают в специально созданные для их хранения служебные таблицы в той же базе данных. Серверы ROLAP наследуют возможности масштабирования и работы с транзакциями реляционных систем, однако существенные

Таблица 1

Достоинства и недостатки MOLAP систем

Достоинства	Недостатки
Напрямую поддерживается многомерное представление данных.	Низкий коэффициент использования дискового пространства, особенно в случае разреженных данных.
Многомерные запросы путём непосредственного доступа к ячейкам гипер-куба (без SQL).	Отсутствуют единые стандарты на интерфейс, языки описания и манипулирования данными.

различия между запросами в стиле OLAP и SQL являются причиной низкой производительности. Достоинства и недостатки ROLAP архитектуры приведены в *табл. 2.*

Таблица 2

Достоинства и недостатки ROLAP систем

Достоинства	Недостатки
Развитые средства администрирования; хорошая масштабируемость.	Для сложных запросов SQL не является оптимальным.
Обеспечивают высокий уровень защиты данных и разграничения прав доступа.	Неодинаковое время выполнения запроса для различных измерений.

3) HOLAP (Hybrid OLAP – гибридный OLAP) – гибридная архитектура, которая объединяет технологии ROLAP и MOLAP. HOLAP могут работать в двух режимах: вертикальной декомпозиции и горизонтальной декомпозиции [13]. В первом случае HOLAP системы применяют ROLAP подход для разреженных областей многомерного пространства (исходные данные) и MOLAP – подход для плотных областей (агрегированные данные). При горизонтальной декомпозиции применяют MOLAP-подход для оперативных данных и ROLAP-подход – для исторических.

По месту нахождения OLAP машины выделяют клиентский OLAP (DOLAP – Desktop OLAP) и серверный OLAP. В случае DOLAP сервер отправляет клиенту исходные данные, клиент выполняет расчет многомерных кубов и выдает результаты пользователю. В случае серверного OLAP сервер выполняет расчет многомерных кубов и отправляет клиенту конечный результат, клиент выдает принятый результат пользователю.

По степени готовности к применению OLAP продукты делятся на две категории [14]:

1) OLAP компонента – это инструмент разработчика. С её помощью разрабатываются клиентские OLAP программы. MOLAP компоненты являются инструментами генерации запросов к серверу. Одна из наиболее доступных компонент – Decision Cube в составе Borland C++ Builder [15].

2) Инструментальные OLAP системы – это программные продукты, предназначенные для создания аналитических приложений. Различают две категории инструментальных систем: системы для программирования и системы для быстрой настройки. Представителем первой категории является аналитическая платформа ProClarity [16]. OLAP-системы для быстрой настройки – это средства, предоставляющие визуальный интерфейс для создания приложений без программирования. В такой технологии реализованы инструменты пакета BusinessObjects [17] и аналитической платформы «Контур» [18].

Совместное использование OLAP и OLTP систем

Многомерный подход возник практически одновременно и параллельно с реляционным, но только начиная с 1993 г. к нему был проявлен всеобщий интерес. В 1993 г. появилась статья [19] одного из основоположников реляционного подхода Кодда, в которой он сформулировал 12 требований к средствам реализации OLAP и произвел анализ недостатков реляционного подхода.

Позднее эти требования были переработаны в тест FASMI (Fast Analysis of Shared Multidimensional Information – Быстрый Анализ Разделяемой Многомерной Информации), который определяет требования к продуктам OLAP [20]. Это определение состоит из пяти требований:

1) Быстрый – означает, что система должна обеспечивать выдачу большинства ответов пользователям в пределах 5 секунд;

2) Анализ – означает, что система может справляться с любым логическим и статистическим анализом, характерным для данного приложения, и обеспечивает его сохранение в виде, доступном для конечного пользователя;

3) Разделяемой – означает, что система осуществляет все требования защиты конфиденциальности и обеспечивает блокировку множественных модификаций на соответствующем уровне своевременным, безопасным способом;

4) Многомерной – ключевое требование. Система должна обеспечить многомерное концептуальное представление данных, включая полную поддержку для иерархий. Под концептуальным представлением

понимается описание данных в форме системы понятий и отношений предметной области. При этом на минимальное количество измерений и используемую технологию БД не накладывается никаких ограничений;

5) Информации – означает, что приложение должно иметь возможность обращаться к любой нужной информации, независимо от ее формата, объема и места хранения.

OLAP системы, как правило, не являются первичными источниками информации, они получают данные из внешних источников. Наиболее часто таким внешним источником является OLTP (On-Line Transactional Processing – оперативная обработка транзакций) система – система операционной обработки данных (СОД), построенная на базе реляционной СУБД. OLAP системы и OLTP системы существенно различаются. Сравнительная характеристика этих технологий по основным параметрам приведена в *табл. 3.*

Таблица 3

Сравнение OLAP и OLTP систем

Характеристика	OLTP	OLAP
Взаимодействие с пользователем	**На уровне транзакции**	**На уровне всей БД**
Время отклика	Не регламентируется	Секунды
Данные, используемые при обращении пользователя к системе	Отдельные записи	Группы записей
Характер данных (детализация)	В основном детализированные (самый низкий уровень детализации)	В основном агрегированные
Изменчивость данных	Высокая (данные обновляются с каждой транзакцией), данные обновляются маленькими порциями	Низкая (во время запроса данные обновляются редко), данные обновляются большими порциями
Источники данных	В основном внутренние (по отношению к системе)	В основном внешние (по отношению к системе)
Возраст данных	Текущие (за период от нескольких месяцев до одного года)	Исторические, текущие и прогнозируемые
Характер запросов	В основном предсказуемые	Непредсказуемые

Следует отметить, что в OLAP системах время является одним из основных критериев, по которому данные упорядочиваются в процессе обработки и представления пользователю. Вследствие этого в OLAP системе должны быть предусмотрены необходимые возможности для работы с временным измерением, такие как поддержка нестандартных календарных циклов.

Не менее важно и то, что многие критически необходимые для OLTP-систем функциональные возможности, реализуемые в РСУБД, являются избыточными для OLAP систем. Например, в OLAP системах данные обычно загружаются достаточно большими порциями из различных внешних источников (РСУБД, электронных таблиц и т.д.) [21].

Непосредственная реализация аналитической системы на основе OLTP-базы сложна. Это объясняется несколькими причинами:

1) OLAP запросы к базам данных чаще всего бывают сложными и требуют много времени. Прямой доступ к OLTP базе существенно снижает общую производительность оперативной системы, так как аналитические запросы конкурируют с транзакциями;

2) разнообразные OLTP системы неоднородны по типу используемых синтаксических соглашений и концептуальных допущений, поэтому их интеграция затруднена;

3) данные в OLTP системах часто «зашумлены», неполны и несогласованны;

4) модели данных разных OLTP систем не взаимосвязаны между собой;

5) информация за прошлые периоды теряется при обновлении OLTP-базы;

6) в OLTP-базе не хранятся данные в агрегированном, ненормализованном виде, что необходимо для оперативной аналитической обработки.

Для реализации аналитической системы необходимо наличие отдельной многомерной СУБД, интегрирующей данные из внешних источников и обрабатывающей аналитические запросы пользователей системы. С учётом этого, архитектура системы отчетности имеет следующий вид (*рис. 4*).

OLAP технология не служит заменой существующей программной инфраструктуры, она лишь дополняет ее. Пользователь по-прежнему имеет возможность обращаться к существующим РСУБД, электронным таблицам и плоским файлам. Однако

Рис. 4. Архитектура отчётной системы с применением OLAP-технологии

для выполнения аналитических запросов используется обращение к OLAP серверу.

Современные системы аналитической обработки данных

Сейчас существует множество OLAP систем от различных производителей.

Microsoft Analysis Services [22] (является составной частью СУБД Microsoft SQL Server) разработан в 1996 г. Он использует архитектуру хранения данных с применением ROLAP/MOLAP/HOLAP. В Microsoft SQL Server присутствует единая модель OLAP и реляционной отчетности. Модели анализа данных можно строить по произвольным срезам куба. При выборе одного измерения другие, связанные с ним, можно рассматривать как вложенные таблицы, причём к каждому допускается применение независимых срезов. Новая версия продукта позволяет применять OLAP-модель к глобальному кубу, делить его на группы, создавать кластеры в режиме реального времени.

OLAP Option to Oracle Database [23] является составной частью Oracle Database Enterprise Edition. OLAP Option полностью интегрирован на уровне ядра СУБД в Oracle Database Enterprise Edition. Поддерживаются как MOLAP, так и ROLAP архитектуры. В качестве языка описания и манипулирования данными используется язык OLAP DML, который может быть встроен в синтаксис языка SQL (для обработки данных в многомерной БД может быть использован язык Java). Для администрирования многомерных СУБД используется приложение Analytic Workspace Manager. Для интеграции с Microsoft Excel служит OracleBI Spreadsheet Add-In. Oracle OLAP Option поддерживает параллельное выполнение запросов и разделение многомерного куба на разделы, позволяет контролировать распределение системных ресурсов и статус долговременных операций, содержит инструменты для создания резервных копий многомерной БД и восстановления.

Palo – MOLAP сервер, в котором значения записываются непосредственно в оперативную память. Palo рассчитан прежде всего на взаимодействие с Excel, большинство задач, таких как создание измерений и кубов, выполняются с помощью плагина для Excel. Однако Palo предоставляет SDK (Software Development Kit – комплект программ для разработчика ПО) для языков C, PHP, .NET и Java). ПО Palo Eclipse Client (на основе Eclipse Framework) позволяет проектировать и просматривать измерения и кубы в платформонезависимой среде Eclipse Framework.

Mondrian – ROLAP сервер с открытым исходным кодом, написанный на Java и поддерживающий язык запросов MDX (MultiDimensional eXpressions – многомерные выражения). Кубы и измерения определяются путем создания схемы XML [24]. Архитектура Mondrian состоит из четырех уровней: уровень представления, уровень измерений, главный уровень и физический уровень. Уровень представления определяет, как будет отображаться информация, и как пользователь будет взаимодействовать с системой для определения запросов. Уровень измерений анализирует, проверяет и выполняет MDX запросы. Главный уровень поддерживает кэш с агрегированными данными. Уровень хранения – реляционная СУБД.

Сравнительные характеристики OLAP-систем приведены в *табл. 4.*

Заключение

Анализ современного состояния OLAP-технологий позволяет говорить о серьезных перспективах их развития. Многомерная обработка информации становится необходимым компонентом любого хранилища данных. В то же время, широкое разнообразие подходов к реализации таких систем и отсутствие должной унификации форматов хранения и алгоритмов обработки данных заставляет искать новые подходы к описанию и разработке OLAP-систем. Эти подходя должны использовать в своей основе математический формализм, доста-

Таблица 4

Сравнительные характеристики OLAP-систем

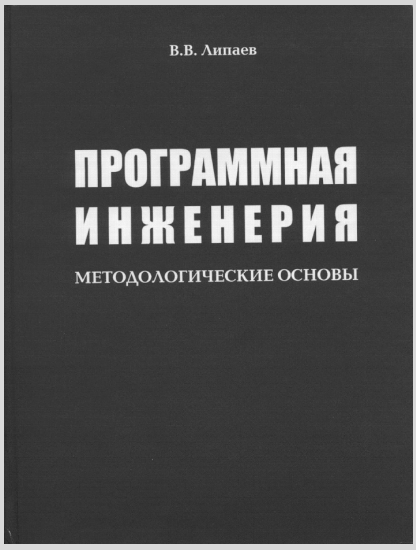
Характерис-тики	Microsoft Analysis Services	OLAP Op-tion to Oracle Database	Palo	Mondrian
Сравниваемые версии	2008 Standart Edition	11g	2.5	3.0.4
Поддержи-ваемые плат-формы	Windows XP, Windows Server 2003	Windows 2000, Linux	Windows 2000, Linux	Java SDK 1.4.2
OLAP API	XML for Analysis (XMLA)	Java API (Oracle JDBC)	NET API, PHP API, Java API, C API	XMLA, Olap4j (Java API)
Язык запросов	MDX	SQL	нет	MDX
Язык манипу-лирования дан-ными	T-SQL	OLAP DML	нет	нет
Создание отчетов на основе данных из OLAP-системы	Microsoft Re-porting Services	Oracle BI Publisher	Excel	Pentaho Report Designer
Интеграция с Microsoft Excel	Microsoft Excel Pivot Tables	OracleBI Spreadsheet Addin	Excel Addin	Pentaho Spreadshe et Services
Инструменты резервирования данных	SQL Server Management Studio	Recovery Manager, Oracle Data Pump	нет	нет

точный для описания и анализа большинства существующих систем. Новые исследования должны привести не только к выделению базовых принципов структурного OLAP-проектирования, но и дать возможность конструктивной оптимизации, а также автоматизированного проектирования систем многомерной обработки данных, обеспечивающего надежную и оперативную работу информационных хранилищ. ■

Литература

1. Oracle Database [Электронный ресурс]–URL: http://www.oracle.com/database/index.html (дата обращения: 10.11.2008).
2. SQL Server 2008 Overview, data platform, store data [Электронный ресурс] – URL: http://www.microsoft.com/sql/default.mspx (дата обращения: 10.11.2008).
3. The world's most popular open source database [Электронный ресурс] – URL: http://www.mysql.com/ (дата обращения: 10.11.2008).
4. The world's most advanced open source database [Электронный ресурс] – URL: http://www.postgresql.org (дата обращения: 10.11.2008).
5. Shoshani A. OLAP and statistical databases: similarities and differences // 16th ACM SIGACT SIGMOD SYGART Symp. On Principles of Database Systems, 1997. P. 185–196.
6. Keith Laker «OLAP Workshop 1: Basic OLAP Concepts» [Электронный ресурс] – URL: http://oracleolap.blogspot.com/ 2007/12/olap-workshop-1-basic-olap-concepts.html (дата обращения: 10.11.2008).
7. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. – Спб.: БХВ-Петербург, 2007. – 384 с.
8. Malinowski E., Zimanyi E. Hierarchies in a Multidimensional Model: From Conceptual modeling to Logical Representation// Data&Knowledge Engineering, 2006. Vol. 59. № 2. P. 348 – 377.
9. Jurgens M. Index Structures for Data Warehouses. Dissertation Fachbereich fur Mathematik und Informatik an der Freien Universitat Berlin eingereicht am 22.12.1999.
10. OLAP product reviews [Электронный ресурс]–URL: http://www.olapreport.com/ProductsIndex.htm (дата обращения: 10.11.2008).
11. Nigel Pendse. Multidimensional data structures [Электронный ресурс] – URL: http://www.olapreport.com/MDStructures.htm (дата обращения: 10.11.2008).
12. OLAP, MOLAP and ROLAP: What's next? [Электронный ресурс] – URL: http://www.itweb.co.za/office/bytes/0407060916t.htm (дата обращения: 10.11.2008).
13. O'Brien J. Next-Generation OLAP. The Future of Dimensional Analysis // The TDWI BI Executive Summit. San Diego, Aug. 18–20, 2008.
14. Чаусов В., Амириди Ю. Классификация аналитических систем. Три года спустя // Журнал «Банки и технологии» – 2002. – №6.
15. Послед Б.С. Borland C++ Builder 6. Разработка приложений баз данных. – Спб.: ДиасофтЮП, 2003. – 320 с.
16. Microsoft Office PerformancePoint Server [Электронный ресурс] – URL: http://www.microsoft.com/business/performancepoint/productinfo/previousversions.aspx.

17. Business Intelligence from Business Objects [Электронный ресурс] – URL: http://www.businessobjects.com/ (дата обращения: 10.11.2008).
18. Intersoft Lab [Электронный ресурс] – URL: http://www.iso.ru/ (дата обращения: 10.11.2008).
19. Codd E.F. Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. E.F. Codd and Associates, 1993.
20. Nigel Pendse. What is OLAP? [Электронный ресурс] – URL: http://www.olapreport.com/fasmi.htm (дата обращения: 10.11.2008).
21. Thilo Maier. A Formal Model of the ETL Process for OLAP-Based Web Usage Analysis // Proceedings of the sixth WEBKDD workshop: Webmining and Web Usage Analysis (WEBKDD'04), in conjunction with the 10th ACM SIGKDD conference (KDD'04). Seatle, Washington, USA, August 22, 2004.
22. SQL Server 2005 Analysis Services [Электронный ресурс] – URL: http://www.microsoft.com/sql/technologies/analysis/default.mspx (дата обращения: 10.11.2008).
23. Oracle OLAP [Электронный ресурс] – URL: http://www.oracle.com/technology/products/bi/olap/olap.html (дата обращения: 10.11.2008)
24. Julian Hyde. Creating interactive OLAP applications with MySQL Enterprise and Mondrian [Электронный ресурс] – URL: http://www.scribd.com/doc/2606603/Creating-Interactive-OLAP-Applications-with-MySQL-Enterprise-and-Mondrian-Presentation (дата обращения: 10.11.2008).



В рамках Инновационной образовательной программы вышла книга
В.В.Липаева
«Программная инженерия. Методологические основы»
(Допущено УМО по образованию в области менеджмента в качестве учебника для студентов ВУЗов, обучающихся по направлению «Бизнес-информатика»).

Владимир Васильевич Липаев – профессор кафедры управления разработкой программного обеспечения ГУ-ВШЭ, главный научный сотрудник Института системного программирования РАН. Около 40 лет занимается исследованиями и разработкой программного обеспечения, методов и инструментальных средств для создания управляющих программ реального времени высокого качества. Под его руководством разработаны крупные инструментальные системы программной инженерии, широко использовавшихся в оборонной промышленности и частично эксплуатируемые до настоящего времени.

Учебник содержит курс лекций, отражающий методологические основы современной программной инженерии, обеспечивающей жизненный цикл (ЖЦ) сложных программных средств (ПС).

Учебник целесообразно использовать при обучении студентов старших курсов, аспирантов и менеджеров проектов при создании сложных комплексов программ на всём их ЖЦ (64 часа лекций и 32 часа семинарских занятий). Курс ориентирован также на заказчиков, менеджеров крупных проектов, аналитиков и ведущих специалистов, обеспечивающих этапы ЖЦ сложных ПС и систем, к которым предъявляются высокие требования по качеству функционирования и ограничены доступные ресурсы разработки.

РАЗРАБОТКА СИСТЕМЫ ПОКАЗА МОБИЛЬНОЙ КОНТЕКСТНОЙ РЕКЛАМЫ

Д.Н. Касымова,
программист-разработчик ООО «Мобил-2», Новосибирск,
Адрес: 630055, г. Новосибирск, ул. Мусы Джалиля, 11-819, Касымова Д.Н.,
Тел: 8-923-120-7864. E-mail: dilnara.kasymova@gmail.com

Статья посвящена развитию технологии использования сотовой связи для показа нацеленной мобильной рекламы. Рассмотрены алгоритмы подбора нацеленных рекламных сообщений абоненту сотового оператора. Проведен сравнительный анализ различных методов ускорения поиска по сходству для параметров, заданных сотовым оператором.

Ключевые слова: Мобильная реклама. Поиск по сходству. Контекстная реклама. Индексы для поиска по сходству.

Мобильная реклама – это сравнительно новое явление на российском рекламном рынке. Использование данного канала коммуникаций в рекламных кампаниях началось несколько лет назад. Многие специалисты и эксперты рекламного рынка России считают, что у рекламы в мобильных телефонах есть хорошее будущее [Крапивинский 2008: 8]. Использование технологий мобильного маркетинга в рекламных кампаниях является эффективным способом продвижения товаров и услуг на рынок. Преимущества использования мобильного маркетинга очевидны: полностью автоматизированный двусторонний канал коммуникаций, работающий в режиме 24/7; возможность персональной коммуникации с целевой аудиторией, что позволяет формировать и поддерживать клуб лояльных к бренду потребителей; доступность для целевой аудитории, вне зависимости от местонахождения.

Обычно используются следующие способы доставки рекламы абонентам сотовой сети:

- SMS (*Short Message Service* – служба коротких сообщений) – система, позволяющая посылать и принимать текстовые сообщения при помощи сотового телефона. Как правило, абонент добровольно соглашается с рассылкой SMS-сообщений «рекламно-информационного характера» в обмен на получение бонусов (например, денежное вознаграждение за каждое сообщение, которое может использоваться для оплаты услуг сотовой связи);
- MMS (*Multimedia Message Service* – служба мультимедийных сообщений) – система, позволяющая посылать и принимать мультимедийные (изображения, мелодии, видео) сообщения при помощи сотового телефона. Аналогично SMS, абонент в обмен на согласие просматривать рекламу посредством MMS получает «информационно-развлекательный бонус». Например, в МТС абоненту высылается приглашение участвовать в акции, если абонент согласен, он отправляет в ответ на приглашение пустое MMS-сообщение. После этого ежедневно в течение определенного срока он будет бесплатно получать MMS-сообщение с погодой, анекдотом, гороскопом, кратким выпуском новостей и курсом валют. Верхнюю часть каждой страницы MMS-сообщения занимает баннер с рекламой товаров и услуг;
- USSD (*Unstructured Supplementary Service Data*) – стандартный сервис в сетях GSM (глобальный цифровой стандарт для мобильной сотовой связи), позволяющий организовать интерактивное взаимодействие между абонентом сети и сервисным приложением в режиме передачи

коротких сообщений. USSD-сервис в основном предназначен для обмена сообщениями между абонентом и дополнительными сервисами, в простейшем случае, службой автоинформатора расчётного счета, тогда как SMS в основном служит для обмена короткими сообщениями между абонентами. Этот канал так же может использоваться для размещения рекламы – в списке данного меню может быть пункт меню бренда, которое составляется с учётом задач рекламной компании бренда. Например, в меню могут быть внесены такие пункты, как условия проводимой брендом промо-акции, брендированный контент, купоны на скидку и т.д.;

- WAP (*Wireless Application Protocol* – протокол беспроводного доступа) – это средство получения доступа к ресурсам интернет посредством только мобильного телефона. По сути, это технический стандарт, описывающий способ, с помощью которого информация из интернета передается на дисплей мобильного телефона. Реклама на WAP-сайтах подчиняется тем же законам, что и обычная интернет-реклама.

Мобильная реклама нацелена на людей, которые с наибольшей вероятностью купят рекламируемый товар или воспользуются предлагаемой услугой, так называемую целевую аудиторию.

Нацеливание рекламы (таргетинг) бывает следующих видов:

- географический таргетинг – показ рекламы целевой аудитории, ограниченной некоторым географическим регионом, выбранным рекламодателем;
- социально-демографический таргетинг – по возрасту, полу, доходу, должности и т.д.;
- таргетинг по времени показа (утро или вечер, будни или выходные);
- таргетинг по интересам (контекстная реклама). Показ рекламы в соответствии с некоторым контекстом.

Как видно, нацеленная мобильная реклама – это развитие ставшей уже привычной для интернет пользователей нацеленной интернет рекламы. Считается, что история возникновения контекстной интернет-рекламы начинается с 1997 г., когда Билл Гросс, основатель молодой компании Idealab, придумал продавать ссылки, показываемые одновременно с результатами запросов. Сегодня услуги по контекстной рекламе предлагают тысячи различных сайтов.

Таким образом, на рынке мобильной рекламы происходит взаимодействие сотовых операторов и сервисов контекстной рекламы. У первых есть информация, позволяющая выбрать оптимальное решение по показу рекламы, у вторых – опыт.

Преимущество сотрудничества рекламодателя с сотовым оператором – возможность четкой сегментации целевой аудитории. Количество собираемой и хранимой сотовым оператором информации огромно (пол, возраст, прописка, тарифный план, среднемесячные расходы на связь, модель телефона, частота и продолжительность зарубежных поездок, уровень дисциплинированности при оплате счетов и т.д.).

Очевидно, сотовый оператор не может продавать информацию об абоненте рекламодателю. Вместо этого, оператор может предложить свои услуги по нацеливанию рекламы – в частности свою собственную систему показа контекстной рекламы, которая будет использовать известную об абоненте информацию для выдачи рекламного сообщения, соответствующего интересам абонента.

В нашей стране в настоящее время (начиная с 2007–2008 гг.) организуются стратегические партнёрства между операторами сотовой связи и агентствами мобильной рекламы. Например, у «большой тройки» (Российские лидеры сотовой связи – МТС, Билайн, Мегафон) существуют следующие партнёрства: Билайн+BrandMobile, Мегафон+CustomLine и МТС+NMM (IMHO VI).

Данная статья посвящена разработке системы показа мобильной контекстной рекламы («BannerEngine»), которая использует известную об абоненте информацию (собранную сотовым оператором) для выдачи рекламного сообщения, соответствующего интересам абонента.

«BannerEngine» – система показа мобильной контекстной рекламы, используемая сотовым оператором. «BannerEngine» реализует следующие основные функции:

- хранение и управление баннерами (рекламными сообщениями) в системе;
- хранение и управление сервисами (сервис – именованный информационный канал) в системе;
- хранение и управление регионами (регион – набор масок MSISDN абонентов, соответствующий географическому региону) в системе;
- выдачу баннеров по запросу от сторонних приложений с учётом ограничений, наложенных на выдачу баннеров;

- ✦ учёт количества показов баннеров;
- ✦ формирование статистики показов баннеров.

В рамках данной статьи интересующим нас ограничением, наложенным на выдачу баннеров, является т.н. ограничение *«рекламные группы»*. Баннер может быть привязан к определенным рекламным группам. Привязка производится путём назначения рекламодателем баннеру списка ключевых слов. Ключевое слово (КС) – строковое представление одной из характеристик абонента или одной из тем, интересных абоненту. Например, КС «MALE», «FEMALE» – характеристики абонента. КС «CARS», «FOOTBALL», «TRAVEL» – интересы абонента. Данная возможность позволяет привязывать баннеры к целевой аудитории, на которую они рассчитаны.

Система использует информацию о пользователе сотовой сети для показа ему нацеленного рекламного сообщения при включенном ограничении «рекламные группы».

Механизм выдачи баннера с учётом рекламных групп организован в виде «черного ящика», которому на вход подаются КС конкретного абонента (список его интересов), на выходе получается список подошедших баннеров по рекламным группам. Текущий критерий соответствия – КС баннера должны оказаться подмножеством КС абонента.

Если предположить, что 1 500 000 абонентов (для сотового оператора по региону) получают в день (в течение 12 часов) 4 SMS/MMS с рекламным сообщением – алгоритм выдачи баннера должен вернуть результат $(1\,500\,000 \cdot 4)/(12 \cdot 60 \cdot 60) = 139$ раз за секунду. Однако текущая реализация алгоритма выдачи подходящего баннера позволяет выдавать только 37 решений в секунду.

Таким образом, встает задача разработать оптимальный алгоритм выбора нужного баннера с минимальными временными затратами.

Главное требование к алгоритму – высокая скорость выдачи баннеров.

Решение зависит от того, как определяется соответствие баннера абоненту. В рамках данной статьи мы рассмотрим три способа определения соответствия:

1. Наиболее подходящими считаются баннеры, имеющие максимальное количество совпавших КС абонента и баннера. Например, для абонента [$КС_1$, $КС_2$] из баннеров [$КС_1$, $КС_2$, $КС_3$], [$КС_1$, $КС_2$] и [$КС_1$] будут выбраны [$КС_1$, $КС_2$, КС3] и [$КС_1$, $КС_2$] – по два совпавших КС.

2. Подошедшими считаются все баннеры, КС которых являются подмножеством КС абонента. Например, для абонента [КС1,КС2] из баннеров [КС1,КС2,КС3], [КС1,КС2] и [КС1] будут выбраны [КС1] и [КС1,КС2].

3. Подошедшими считаются все баннеры, КС которых в точности совпадают с КС абонента.

Для каждого способа определения соответствия мы проделаем следующее:

1. Рассмотрим адекватность. Т.е. насколько выбранные по такому критерию баннеры будут соответствовать интересам абонента.

2. Рассмотрим алгоритмы выбора баннеров согласно этому критерию и проанализируем границы применения этих решений.

3. Для критериев из п.п. 1, 2 – рассмотрим для выбранных алгоритмов возможность учёта весов КС абонента. Вес КС абонента может определяться, например, количеством откликов абонентом на баннеры, содержащие это КС. В случае если заданы веса КС абонента, соответствие по п.1 будем определять суммарным весом совпавших КС абонента и баннера. В случае соответствия по п.2 наиболее подходящими будем считать баннеры-подмножества с максимальным суммарным весом.

Исходя из полученных результатов по каждому критерию, мы определим наиболее компромиссный вариант между критерием соответствия, скоростью выбора и возможностью добавления учета весов КС (учёт весов КС позволит более качественно нацеливать баннеры).

Экспериментальная часть

Для генерации искусственных данных мы руководствовались следующими ограничениями:

- ✦ количество баннеров – от 100 000 до 300 000;
- ✦ количество уникальных КС – ~125;
- ✦ количество КС на абонента/баннер – от 1 до 50;
- ✦ желаемое время поиска – от 2 до 6 мс (6 мс соответствуют выдаче 1000/6 = 166 результатов в секунду).

В наших искусственных данных КС равномерно распределены по всем объектам (т.е. каждое КС равновероятно).

Алгоритмы тестировались при следующих характеристиках:

- ✦ Intel Celeron CPU 2.40 GHz 0.98 GB of RAM;
- ✦ Java 1.6 в режиме –server -Xms512m -Xmx512m

Решение задачи выбора баннера по максимальному количеству совпавших ключевых слов

Эту задачу мы рассмотрим наиболее подробно, так как из неё можно получить решение остальных трёх задач.

1. Адекватность

На первый взгляд это достаточно адекватный критерий. Более «точные» баннеры предпочитаются более «общим»: для абонента [KC_1, KC_2, KC_3] из баннеров [KC_1, KC_2, KC_3], [KC_1, KC_2] и [KC_1] будет выбран [KC_1, KC_2, KC_3]. Но рассмотрим следующий пример: абонент = {MALE, KC_1, ... KC_N}, а среди баннеров мы имеем b = {FEMALE, KC_1, ... KC_N}. Несмотря на то, что в описании баннера присутствуют почти все интересы абонента, решение, полученное поиском максимального пересечения, не выглядит правильным. Так, например, для абонента {MALE, SPORT, SWIMMING} не исключена возможность выбрать баннер {FEMALE, SPORT, SWIMMING}, предлагающий женские купальники.

2. Алгоритмы

Задача выбора баннера поиском максимального пересечения множеств КС баннера и абонента является одной из задач поиска ближайшего соседа. Эта задача не имеет универсального решения. В каждом конкретном случае необходимо учитывать специфику данных и, исходя из этого, выбрать модель, которой будут представлены исходные данные, и определить функцию похожести или расстояния.

Для решения данной задачи наиболее подходит модель векторного пространства с функцией близости – скалярным произведением. Каждый объект (абонент, баннер) представляется в виде вектора размера N_KW, где N_KW – количество уникальных КС. В этом векторе выставляются K единиц следующим образом (K – количество КС объекта): в i-ой позиции выставляется единица, если i-ое слово встречается в объекте, иначе 0.

Следует отметить, что наша задача подходит под «проклятье размерности» [Koppen 2000: 4]. Это понятие обозначает явление, что все точные алгоритмы поиска ближайшего соседа эффективны по сравнению с «хорошим» последовательным поиском только для небольших размерностей (~ < 10) [Koppen 2000: 2]. В нашей же задаче объект (абонент) может описываться 50-ю признаками.

Были протестированы следующие алгоритмы:

- ✦ обратный индекс [Бартунов 2007: 7, Ilyinsky 2002: 3]. Обратный индекс –множество пар {KC_i ↔ постинг-лист$_i$}, где постинг-лист$_i$ – список баннеров, содержащих KC_i. Для каждого КС абонента выбирается соответствующий постинг-лист, заводится счётчик для каждого баннера, и выполняется проход по всем листам с увеличением счётчика для каждого повстречавшегося баннера на единицу. Затем выбираются баннеры с максимальным значением счётчика;
- ✦ последовательный алгоритм в рамках векторной модели, базирующийся на битовых операциях. Векторы представляются в виде набора 32-битовых чисел, над которыми выполняется операция битовое «И». Заранее строится таблица соответствия «число ↔ количество выставленных битов». Очевидно, количество выставленных битов в результате операции битового «И» будет равно размеру пересечения. Таким образом, за одну операцию сравниваются 32 признака;
- ✦ метод разбиения пространства [Yianilos 1993: 6] с использованием BK-дерева [Burkhard 1973: 1] – структуры данных, основанной на неравенстве треугольника, адаптированной к высоким размерностям (не дает при поиске экспоненциальный рост времени от количества признаков). В качестве функции расстояния между двумя объектами (баннерами b_1, b_2) по k КС использовалась d (b_1, b_2) = k – (количество совпавших КС у b1 и b2).

В *табл. 1* представлены результаты работы обратного индекса (II), последовательного алгоритма на битовых операциях (VM+SP(bit impl)) и BK-дерева.

Как видно из табл. 1, метод обратного индекса хорошо отражает проклятие размерности. Чем больше количество признаков, тем сильнее выигрывает последовательный алгоритм. Метод разбиения пространства демонстрирует линейную зависимость от количества признаков, однако показывает очень большое абсолютное значение времени поиска, что может быть связанно с недостаточно эффективной реализацией.

Таблица 1

Время поиска максимального пересечения при использовании алгоритмов II, VM+SP (bit impl) и BK-tree в зависимости от количества КС на объект и баннеров. Количество уникальных КС равно 125

Кол-во КС на объект	Кол-во баннеров	II, ms	VM+SP(bit impl), ms	BK-tree, ms
до 10	100 000	2,18	6,22	23,08
до 20	100 000	3,25	6,36	51,54
до 30	100 000	5,12	6,35	78,15
до 40	100 000	7,38	6,22	96,43
до 50	100 000	9,70	6,09	120,49
до 10	200 000	4,66	11,95	62,32
до 20	200 000	11,60	12,05	117
до 30	200 000	22,75	11,48	184,71
до 40	200 000	31,96	11,63	240,23
до 50	200 000	53,38	11,46	283,19
до 10	300 000	8,44	18,45	93,87
до 20	300 000	25,08	18,21	176,85
до 30	300 000	51,69	16,82	259,22
до 40	300 000	81,88	17,11	352,85
до 50	300 000	123,14	17,08	442,46

В *табл. 2* результаты работы метода разбиения пространства оцениваются по количеству обойдённых при поиске узлов построенного дерева – т.е. по количеству рассмотренных баннеров.

Таблица 2

Количество рассмотренных баннеров при поиске в BK-дереве в зависимости от количества КС на объект и баннеров. Количество уникальных КС равно 125

Кол-во КС на объект	Кол-во баннеров	Рассмотрено баннеров при поиске в BK-дереве
до 10	100000	32185
до 20	100000	43083
до 30	100000	37409
до 40	100000	52829
до 50	100000	43583
до 10	200000	64384
до 20	200000	89104
до 30	200000	125885
до 40	200000	61015
до 50	200000	194430
до 10	300000	192036
до 20	300000	168947
до 30	300000	237325
до 40	300000	253814
до 50	300000	184682

Алгоритм в рамках векторной модели с использованием битовых масок применим при небольшом количестве уникальных КС. В *табл. 1* приведены значения для количества уникальных КС = 125. Т.е. каждый объект представляется 125/32 + 1 = 4-мя 32-битовыми числами. При количестве КС до 32, каждый объект будет представлен одним числом. При увеличении количества уникальных КС скорость работы алгоритма будет ухудшаться.

Обратный индекс применим при условии, что есть большая база уникальных КС (не менее 100) и количество КС на баннер не превышает 10. В этом случае постинг-листы будут достаточно короткими, и их обработка займёт приемлемое время.

Метод разбиения пространства представляет тем больший интерес, чем сильнее данные (баннеры) разбиты на кластеры. Однако, чем меньше абонент будет иметь интересов, тем больше вероятность обойти все дерево полностью.

3. Возможность добавления весов

В последовательном «битовом» алгоритме и алгоритме обратного индекса есть возможность добавить работу с весами КС без потери производительности.

Для обратного индекса учёт весов означает увеличение счётчика не на единицу, а на вес КС абонента, для которого рассматривается текущий постинг-лист.

Для «битовой реализации» для учёта весов строится таблица по абоненту таким образом, чтобы по результату битового «И» можно было обратиться сразу за взвешенным значением текущего 32-битового кусочка.

При использовании метода разбиения пространства использование КС невозможно, т.к. метрическое дерево строится на основании заданной функции расстояния, которая точно в таком же виде должна использоваться и при поиске в дереве. Использование же весов КС абонента означает невозможность использовать единую функцию расстояния для всех баннеров/абонентов, т.к. каждый объект-абонент в этом случае определяет новую функцию расстояния.

Решение задачи выбора баннера поиском «баннера-подмножества»

1. Адекватность

Данный критерий исключает пример, приведенный выше, в котором выбирается баннер с «лишними» КС. Выбранный баннер будет соответствовать интересам абонента и не предложит лишнего.

2. Алгоритмы

Решение данной задачи можно получить модификацией первой (проверять количество совпавших КС на равенство количеству КС баннера), однако задача на подмножества имеет следующие преимущества:

- ✦ если задан абонент с k уникальными КС, то нет необходимости рассматривать баннеры с $\geq k + 1$ уникальными КС;
- ✦ если в баннере обнаружилось хотя бы одно «лишнее» КС, то нет смысла рассматривать этот баннер далее.

Для данного критерия мы сравнили три реализации:

1. Последовательный алгоритм, базирующийся на битовых операциях.
2. Последовательный алгоритм, базирующийся на простых числах [Clarke 1997: 2].
3. Последовательный алгоритм, базирующийся на сравнении сортированных идентификаторов КС.

Алгоритм, базирующийся на битовых операциях, был получен модификацией «битового» алгоритма, вычисляющего максимальное пересечение КС абонента и баннера. Баннеры и абоненты так же представляются в виде набора 32-битовых чисел, над которыми выполняется операция битовое «И». Если $a[i]$, $b[i]$ – 32битовые кусочки абонента и баннера, и $a[i]$ «И» $b[i]$! = $b[i]$, – то сразу ясно, что баннер не подходит абоненту по данному критерию, прекращаем его рассмотрение и переходим к следующему.

Алгоритм, базирующийся на простых числах, заключается в следующем: каждому уникальному КС ставится в соответствие простое число. И каждый баннер представляется в виде произведения своих простых чисел. Когда на вход поступает абонент, он точно так же представляется в виде произведения. Если «баннер делит абонента» без остатка, то баннер – подмножество абонента.

Алгоритм, базирующийся на сравнении сортированных идентификаторов КС, требует, чтобы идентификаторы (уникальные номера) КС баннеров/абонента были отсортированы. Для каждого идентификатора КС баннера бинарным поиском ищется такой же идентификатор среди идентификаторов КС абонента. Если поиск дал отрицательный результат, сразу переходим к следующему баннеру.

В *табл. 3* представлены результаты для всех трех реализаций: сравнение сортированных идентификаторов КС (Sorted IDs), с использованием простых чисел (Prime based), с использованием битовых масок (Bit mask), с использованием битовых масок + сортировка баннеров по количеству уникальных КС баннера (Bit mask (sorted)).

Таблица 3

Время поиска «баннера-подмножества» при использовании алгоритмов Sorted IDs, Prime Number, Bit mask и Bit mask (sorted) в зависимости от количества КС на абонента/баннер и баннеров. Количество уникальных КС равно 125

Кол-во КС на объект	Кол-во баннеров	Sorted IDs, ms	Prime Number, ms	Bit mask, ms	Bit mask (sorted), ms
до 10	100000	17,54	10,52	2,32	1,55
до 20	100000	22,35	9,85	2,18	1,45
до 30	100000	23,03	11,03	2,26	1,51
до 40	100000	23,95	11,77	2,24	1,49
до 50	100000	24,23	12,47	2,32	1,55
до 10	200000	37,91	25,47	4,64	3,09
до 20	200000	41,88	22,77	4,77	3,18
до 30	200000	45,72	27,88	3,87	2,58
до 40	200000	48,87	25,51	4,21	2,81
до 50	200000	2,07	24,12	3,83	2,55
до 10	300000	54,15	23,97	8,24	5,49
до 20	300000	57,84	36,58	6,38	4,25
до 30	300000	66,48	41,09	5,77	3,85
до 40	300000	64,79	34,25	7,17	4,78
до 50	300000	76,75	33,64	5,65	3,76

Метод с использованием битовых масок показал большую эффективность по сравнению с остальными, поэтому именно его мы улучшили сортировкой баннеров по количеству уникальных КС, что дало улучшение еще в 1,5 раза.

Как и в предыдущем случае, выбранный алгоритм зависит от количества уникальных КС. Но если для поиска максимального пересечения необходимо было обойти все 32-битовые кусочки, то в данном случае мы прекращаем рассмотрение текущего баннера, как только не выполнилось равенство ($a[i]$ & $b[i] = b[i]$). Благодаря этому в сочетании с сортировкой баннеров по количеству уникальных КС становится приемлемым использование довольно большого количества уникальных КС.

3. Возможность добавления весов

Метод битовых масок модифицируем для учета весов КС абонента точно так же как и в случае поиска максимального пересечения.

Решение задачи выбора баннера по точному совпадению ключевых слов

1. Адекватность

Выбранный баннер 100% отражает интересы абонента, однако данный критерий очень сильно ограничивает набор подходящих баннеров и есть достаточно большая вероятность вернуть пустой результат для абонента.

2. Алгоритмы

Здесь применимы все решения, описанные выше, но также на этот случай есть специфичное очень «быстрое решение» – trie-дерево [Shang 1995: 5].

Таблица 4

Время поиска точного совпадения в trie-дереве в зависимости от количества КС на абонента/баннер и баннеров. Количество уникальных КС равно 125

Кол-во КС на объект	Кол-во баннеров	Trie-дерево, ms
до 10	100000	0,01
до 20	100000	0,01
до 30	100000	0,02
до 40	100000	0,04
до 50	100000	0,17
до 10	200000	0,01
до 20	200000	0,16
до 30	200000	0,03
до 40	200000	0,03
до 50	200000	0,11
до 10	300000	0,01
до 20	300000	0,03
до 30	300000	0,01
до 40	300000	0,04
до 50	300000	0,12

Как видно из таблицы, поиск в таком дереве происходит за доли миллисекунды.

Выводы

Были проанализированы критерии соответствия баннера интересам абонента и методы выбора баннеров по этим критериям. При текущих параметрах системы (критерий соответствия «баннер-подмножество», 125 КС, 100 000–300 000 баннеров, до 25 КС на абонента/баннер) найдено удовлетворительное решение, позволяющее выдавать 1000/~2.62 = = ~384 решения в секунду и учитывать на лету веса КС абонента. Были даны рекомендации по использованию других методов при изменении соотношения параметров и критерия соответствия. ■

Литература.

1. Burkhard W. A., Keller R. M. Some approaches to best-match file searching // Communication of the ACM.-1973.-V. 16. – P. 230–236.
2. Clarke I. Testing Subsets Using Prime Numbers // Ian Clarke's Homepage/Subsets [Электронный ресурс]. – Электрон. дан. – 1997. – Режим доступа: http://www.sanity.uklinux.net/subsets.html.
3. Ilyinsky S., Kuzmin M., Melkov A., Segalovich I. An efficient method to detect duplicates of web documents with the use of inverted index // Proceedings of the eleventh Int. World Wide Web Conference (WWW). – 2002.
4. Koppen M. The curse of dimensionality // Proceedings of the fifth Online Conference on Soft Computing in Industrial Applications (WSC5). – 2000.
5. Shang H., Merrettal T. H. Tries for Approximate String Matching // IEEE Transactions on Knowledge and Data Engineering. – 1996. – V. 8.- № 4. – P. 540–547.
6. Yianilos Peter N. Data structures and algorithms for nearest neighbor search in general metric spaces // Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms. – 1993. – 25–27 Jan. – P. 311–321.
7. Бартунов О. С., Сигаев Ф. Г. Специализированные типы данных для цифровых библиотек // 9-ая Всероссийская научная конференция «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL). – 2007.
8. Крапивинский А. Кому рекламу на «сотовый»? // «Медиа-Онлайн»/Аналитика/Медиабизнес/ [Электронный ресурс]. – Электрон. дан. – 2008. – 18 авг. – Режим доступа: http://www.media-online.ru/index.php3?id=290255.

КОМБИНИРОВАННЫЙ И ВОЛНОВОЙ АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ УПАКОВКИ: ПРИНЦИПЫ ПОСТРОЕНИЯ И ОСОБЕННОСТИ

М.В. Ульянов,
доктор технических наук, профессор кафедры «Прикладная математика и моделирование систем» Московского государственного университета печати (МГУП),
Адрес: 127550, Москва, ул. Прянишникова, 2а, кафедра «Прикладная математика и моделирование систем», Ульянов М.В. Тел: 8-916-589-9404. E-mail: muljanov@mail.ru
О.А. Наумова,
студентка кафедры «Персональная электроника» Московского государственного университета приборостроения и информатики (МГУПИ),
E-mail: helga_13_07@mail.ru

В статье рассмотрены базовые алгоритмы точного решения задачи одномерной оптимальной по стоимости упаковки: рекурсивный и табличный. На основе их рационального совмещения предлагаются комбинированный и волновой алгоритмы, обладающие лучшими ресурсными характеристиками. Указаны особенности их применения и различия в ресурсных требованиях.

Ключевые слова: Задача оптимальной упаковки; комбинированный алгоритм; волновой алгоритм; ресурсная эффективность.

Введение

Начиная с конца 50-х годов XX века, когда Р. Беллман предложил и обосновал идеи метода динамического программирования, задача оптимальной по стоимости одномерной упаковки привлекает внимание программистов и ученых, занимающихся разработкой и анализом алгоритмов. Именно этой задачей открывается ряд иллюстративных примеров в классической книге Р. Беллмана и Р. Дрейфуса «Прикладные задачи динамического программирования» [1]. Повышенный интерес к этой задаче, равно как и к её многочисленным модификациям, связан как с разнообразными практическими областями применения, так и с отсутствием «быстрых» алгоритмов её решения. Постановки, сводимые к задаче оптимальной упаковки, возникают при решении целого ряда экономических задач, задач планирования бизнеса и логистики. Мощности современных компьютеров позволяют в настоящее время получать точное решение этой задачи в практически значимом сегменте длин входов, однако требование повышения временной эффективности её решения в настоящее время остаётся актуальным.

Точный классический алгоритм Беллмана для задачи одномерной упаковки может быть реализован двумя принципиально разными алгоритмами:

- ✦ табличным алгоритмом, опирающимся на вычисление и хранение векторов оптимальной упаковки для всех последовательных целочисленных значений объема;
- ✦ рекурсивным алгоритмом, непосредственно реализующим основное функциональное уравнение Белламна.

В статье рассматриваются комбинированный и волновой алгоритмы, в основу создания которых положен принцип сочетания достоинств рекурсивного и табличного алгоритмов. Предлагаемые решения направлены на повышение временной эффективности решения задачи одномерной упаковки, и, следовательно, на расширение сегмента длин входов, на котором точное решение может быть получено за реальное время.

Математическая постановка задачи одномерной упаковки

Постановка задачи: пусть задано множество

$$Y = \{y_i\}, \quad y_i = \{v_i, c_i\}, \quad i = \overline{1,n},$$

где каждый элемент y_i обладает целочисленным линейным размером – v_i, или «объёмом» в общепринятых терминах задачи упаковки, и ценовой характеристикой – c_i, которая содержательно отражает практически значимые предпочтения для загрузки объектов данного типа. Также целочисленным значением задан основной объём упаковки V. В классической постановке элементы y_i называются типами грузов. Для описания количества загружаемых в объем V элементов y_i введём в рассмотрение характеристический вектор:

$$, \qquad x = \{x_i\}, \quad i = \overline{1,n},$$

где x_i – неотрицательное целое.

Значение компонента вектора $x_i = k$ соответствует загрузке k элементов типа y_i в объём V. Таким образом, описание некоторой упаковки грузов представляет собой точку в n-мерном целочисленном пространстве E_z^n. Среди всех возможных упаковок объема V грузами из Y должна существовать по крайней мере одна упаковка, максимизирующая суммарную стоимость, что приводит к постановке задачи как задачи максимизации линейного функционала [1]:

$$P_n(\mathbf{x}) = \sum_{i=1}^{n} C_i(x_i) = \sum_{i=1}^{n} x_i \cdot c_i \to \max,$$
$$\text{при} \quad \sum_{i=1}^{n} x_i \cdot v_i \le V. \qquad (1)$$

Содержательно ограничение в (1) означает, что суммарный объём, занимаемый грузами всех типов в количествах, указанных компонентами характеристического вектора $\mathbf{x}$, не должен превышать общего объёма упаковки. Поскольку и целевая функция и ограничение линейны, а значения x_i – неотрицательные целые числа, то задача (1) является задачей линейного целочисленного программирования.

Оценку вычислительной сложности решения данной задачи методом полного перебора можно получить, если рассматривать прямоугольный параллелепипед в пространстве E_z^n, размеры сторон которого определяются типами грузов. Это приводит к верхней оценке количества точек перебора в виде [2]

$$\prod_{i=1}^{n} \left(\lfloor V/v_i \rfloor + 1 \right) \le (m+1)^n, \quad m = \max_{i=\overline{1,n}} \{ V/v_i \}.$$

При значительном разбросе значений v_i оценка по произведению значительно лучше, чем $(m + 1)^n$, но всё равно неприемлема для практически значимых постановок задачи упаковки.

Отметим, в этой связи, что рассматриваемая задача является *NP*-трудной [1], и для неё в общем виде отсутствуют сегодня полиномиальные по n точные алгоритмы решения. Это приводит к необходимости использования более эффективных точных методов решения задачи одномерной оптимальной упаковки, существенно сокращающих полный перебор, одним из которых является метод динамического программирования.

Функциональное уравнение Беллмана

Опираясь на терминологию метода динамического программирования, будем считать, что в рассматриваемой задаче распределяемым ресурсом является объём упаковки V, а максимизизация дохода, заданного линейным функционалом $P_n(\mathbf{x})$ производится путём распределения ограниченного ресурса V между грузами указанных типов. Поскольку целевой функционал $P_n(\mathbf{x})$ обладает свойством аддитивности, то метод динамического программирования применим, и основное функциональное уравнение Беллмана имеет вид [1]

$$\begin{cases} f_0(v) = 0; \\ f_m(v) = \max\limits_{x_m} \{ x_m \cdot c_m + f_{m-1}(v - x_m \cdot v_m) \}, \\ m = \overline{1,n},\ v = \overline{0,V},\ \ x_m = 0,1,\ldots,\left\lfloor \dfrac{v}{v_m} \right\rfloor. \end{cases} \qquad (2)$$

Таким образом, метод предусматривает последовательное решение одномерных задач целочисленной оптимизации с использованием информации об оптимальной упаковке объёма v предыдущими типами грузов. Решением поставленной задачи является значение $f_n(V)$ и соответствующий вектор оптимальной упаковки. Поскольку значения $f_1(v)$ могут быть элементарно вычислены на основе (2), то в дальнейшем будет рассматриваться следующее основное функциональное уравнение для задачи одномерной оптимальной упаковки, записанное в виде рекуррентного соотношения, определяющего рекурсивно заданную функцию $f_m(v)$ [2]

$$\begin{cases} f_1(v) = \left\lfloor \dfrac{v}{v_k} \right\rfloor \cdot c_1, \quad m = 1; \\ f_m(v) = \max\limits_{x_m} \{ x_m \cdot c_m + f_{m-1}(v - x_m \cdot v_m) \}, \\ m = \overline{2,n},\ x_m = 0,1,\ldots,\left\lfloor \dfrac{v}{v_m} \right\rfloor. \end{cases} \qquad (3)$$

Рекурсивный алгоритм решения задачи упаковки

Рекурсивная реализация функционального уравнения Беллмана для задачи одномерной упаковки достаточно проста. Рекурсивный алгоритм, напрямую реализующий рекуррентное соотношение (3) – алгоритм **A1**, останавливается при значении $m = 1$, когда значение функции $f_1(v)$ может быть вычислено непосредственно. Рекурсия при $m \geq 2$ выполняется для вычисления оптимальной по стоимости упаковки текущего объёма v грузами типа m совместно с грузами предыдущих $m - 1$ типов.

Рассмотрим пример решения задачи одномерной упаковки с использованием данного алгоритма – нас интересует порождённое дерево рекурсии. Пусть необходимо решить задачу для трёх типов грузов при общем объёме упаковки $V = 10$. Информация об объёмах v_i и стоимостях c_i для типов грузов приведена в *табл. 1.*

Таблица 1

Описание типов грузов

i	v_i	c_i
1	2	3
2	3	5
3	4	7

Решением поставленной задачи будет значение функции Беллмана $f_3(10)$, при этом алгоритм порождает дерево рекурсии, показанное на *рис. 1.*

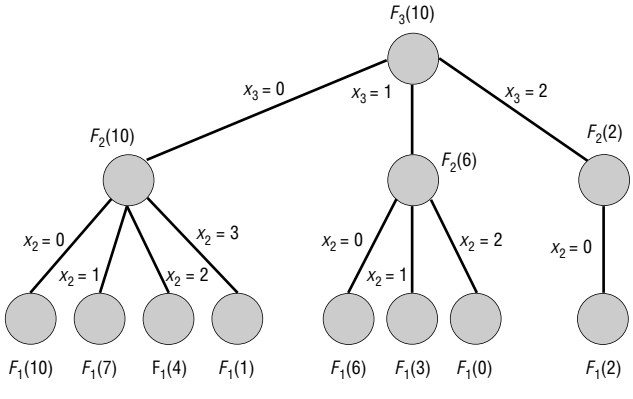


Рис. 1. Дерево рекурсии, порождаемое рекурсивным алгоритмом упаковки.

Характерным недостатком данного алгоритма является повторное вычисление значений функции $f_m(v)$ в различных фрагментах дерева рекурсии. При этом чем дерево шире, а глубина рекурсии больше, тем выше вероятность повторных вычислений. В общем случае трудоёмкость этого алгоритма асимптотически определяется биномиальными коэффициентами при специальной параметризации [1], что приводит ряда входов к значительному времени получения решения.

Табличный алгоритм решения задачи упаковки

Рассмотрим другой алгоритм точного решения этой задачи методом динамического программирования – табличный алгоритм **A2**, позволяющий получить оптимальное решение не только для заданного объёма V, но и набор оптимальных решений для всех промежуточных дискретных значений этого объёма. Обозначим вектор оптимальной упаковки для объёма v элементами y_i, $i = \overline{1,m}$, $m \leq n$, через $\mathbf{x}_v^m$, а через $f_m(v)$ стоимость оптимальной упаковки в этом объёме. В силу принципа оптимальности [1]

$$f_m(v) = \max_{x_m} P_m(\mathbf{x}),$$

а порядок предъявления элементов не является существенным. Результатом решения задачи является набор оптимальных значений целевой функции $f_n(v)$ и соответствующих векторов оптимальной упаковки $\mathbf{x}_v^n$ для всех объемов v от 0 до V исходными элементами (грузами различных типов) из множества Y. Отметим, что поскольку табличный способ позволяет получить оптимальные решения для всех промежуточных объемов упаковки, то эту информацию можно использовать, например, для исследования чувствительности целевой функции $f_n(v)$ по изменению объёма упаковки. Заметим также, что рекурсивный вызов в функциональном уравнении (3) в табличном алгоритме есть не что иное, как обращение к предыдущей таблице оптимальной упаковки для определённого значения объёма.

Пример формирования таблицы оптимальной упаковки для исходных данных, приведённых в *табл. 1*, представлен в *табл. 2.*

Очевидным недостатком данного алгоритма являются «пустые» вычисления в тех строках таблицы, которые не будут задействованы в будущем. Таким образом, алгоритм выполняет лишние вычисления, объём которых может быть значительным. Явная зависимость трудоёмкости данного алгоритма от объёма упаковки приводит к тому, что для больших значений объём лишних вычислений возрастает.

Таблица 2

Таблица оптимальной упаковки

v	x_1	$f_1(v)$
1	0	0
2	1	3
3	1	3
4	2	6
5	2	6
6	3	9
7	3	9
8	4	12
9	4	12
10	5	15

v	x_1	x_2	$f_2(v)$
1	0	0	0
2	1	0	3
3	0	1	5
4	2	0	6
5	1	1	8
6	0	2	10
7	2	1	11
8	1	2	13
9	0	3	15
10	2	2	16

v	x_1	x_2	x_3	$f_3(v)$
1	0	0	0	0
2	1	0	0	3
3	0	1	0	5
4	0	0	1	7
5	1	1	0	8
6	0	2	0	10
7	0	1	1	12
8	0	0	2	14
9	0	3	0	15
10	0	2	1	17

Комбинированный алгоритм

Совместное рассмотрение недостатков рекурсивного и табличного алгоритмов позволяет сделать вывод о том, что в крайних случаях, например, при значительном объёме упаковки и больших значениях объёмов грузов рекурсия будет существенно эффективнее табличных расчётов, а при малых объёмах грузов мы будем наблюдать противоположную ситуации. Таким образом, мы можем говорить, что эти два классических алгоритма являются в определённом смысле комплементарными – они дополняют друг друга, обладая противоположным поведением функции трудоёмкости [3]. Эта особенность позволяет создать комбинированный алгоритм, использующий табличный расчёт и рекурсию в их рациональных диапазонах.

Комбинированное алгоритмическое решение состоит в выборе одного из двух алгоритмов, что может быть реализовано при заданных значениях n, V, k, на основе сравнения их трудоёмкостей [3]. При определённых условиях, зависящих от исходных данных, более рациональным является построение комбинированного алгоритма, основной идеей которого является предвычисление оптимальных упаковок для некоторого числа типов грузов табличным алгоритмом и дальнейшее решение задачи рекурсивным алгоритмом с сокращённым числом типов грузов. Каков оптимальный по трудоемкости порог, т.е. оптимальное число типов грузов обрабатываемых табличным алгоритмом? Ответ на этот вопрос можно получить на основе следующих рассуждений.

Предположим, что табличным алгоритмом находится оптимальное решение для упаковки первыми (в порядке предъявления) m типами грузов, причем $m \geq 1$, при этом для остальных $n-m$ типов задача решается рекурсивным алгоритмом. В листьях порождённого дерева рекурсии на уровне $n-m+1$ происходит копирование вектора оптимальной упаковки, полученного табличным алгоритмом. Это приводит к необходимости модификации рекурсивного алгоритма. Этот модифицированный алгоритм, обозначим его через **A1M**, отличается от алгоритма **A1** фрагментом останова рекурсии, в котором происходит копирование строки результатов табличного алгоритма.

Параметризуем исходные данные числом грузов среднего объёма, размещаемых в общем объёме упаковки. Обозначим этот параметр через k. Поскольку на основе исходных данных значения (n, V, k) известны, то можно построить функцию $g(m)$, значением которой является совокупная трудоёмкость комбинированного алгоритма, при этом предполагается, что значения n, V, k, определяемые входом, фиксированы:

$$g(m) = f_{A1M}(n-m+1,\ k) + f_{A2}(m,\ V,\ k).$$

Оптимальное значение может быть найдено как

$$m^* = \arg\min_{1 \leq m \leq n-1} g(m). \qquad (4)$$

Отметим, что значение $m^* = m^*(n,\ V,\ k)$. Поскольку значение m является целым числом, то m^* может быть найдено простым перебором значений функции $g(m)$ при $1 \leq m \leq n-1$. Тогда трудоёмкость комбинированного алгоритма, назовём его **AC**, составит

$$f_{AC}(n,V,k,m^*) = g(m^*) = \\ = f_{A1M}(n-m^*+1,\ k) + f_{A2}(m^*,\ V,\ k).$$

В общем случае возникает задача построения комбинированного алгоритмического решения – рационального выбора одного из трёх алгоритмов на основе анализа текущего входа. Мы сравниваем между собой рекурсивный, табличный и комбинированный алгоритмы – **A1, A2, AC**. Критерием выбора является минимум значения функций трудоёмкости. Понимая под аргументом оптимизации индекс алгоритма, и фиксируя значения (n, V, k), определённые текущим входом, можно записать

$$A_{opt}:\ f_{A_{opt}}(n,V,k) = \\ = \arg\min_{A1,A2,AC} \left\{ f_{A1}(n,k),\ f_{A2}(n,V,k),\ f_{AC}(n,V,k,m^*) \right\}, \qquad (5)$$

при этом возможны три следующих случая, возникновение которых определяется текущим входом алгоритма:

1. Случай 1 –

$$f_{A_{opt}}(n,V,k) = f_{A1}(n,k).$$

В этом случае рекурсивный алгоритм **A1** обеспечивает минимальную трудоёмкость, и комбинации с табличным алгоритмом не требуется.

2. Случай 2 –

$$f_{A_{opt}}(n,V,k) = f_{A2}(n,V,$$

В этом случае табличный алгоритм **A2** имеет минимальную трудоёмкость для данного входа, и комбинации с рекурсивным алгоритмом не требуется.

3. Случай 3 –

$$f_{A_{opt}}(n,V,k) = f_{AC}(n,V,k,m^*), \quad 1 \le m^* \le n-1.$$

В этом случае рациональным является применение комбинированного алгоритма **AC** с порогом переключения равным m^*. При этом на первом этапе табличный алгоритм получает таблицу оптимальной упаковки заданного объёма грузами типов $1,\overline{m^*}$, а рекурсивный алгоритм на втором этапе, порождает дерево рекурсии от груза с номером n до груза с номером m^*, останавливает рекурсию на уровне $n-m^*+1$, копирует результаты табличного алгоритма в свою структуру данных, и цепочкой рекурсивных возвратов получает вектор оптимальной упаковки и значение целевого функционала для исходной задачи.

Программная реализация комбинированного алгоритма содержит управляющий фрагмент, который по параметрам текущего входа решаемой задачи (n, V, k) определяет значение оптимального порога переключения m^* по формуле (4) и запускает табличный, а затем рекурсивный алгоритмы, – это реализация статически-адаптивного подхода к созданию комбинированных алгоритмов.

Совокупно программная реализация, представляющая собой комбинированное алгоритмическое решение будет содержать табличный, рекурсивный и комбинированный алгоритмы, а также управляющий модуль, который выбирает в соответствии с формулой (5) алгоритм, наиболее рациональный для текущего входа.

Дополнительно отметим, что хотя окончательный результат, получаемый как табличным, так и рекурсивным алгоритмами, не зависит от порядка предъявления типов грузов, а теоретическая формула нивелирует порядок предъявления типов грузов, тем не менее, порожденный фрагмент дерева рекурсии в комбинированном алгоритме будет меньше (в смысле общего числа вершин), если исходные типы грузов будут первоначально отсортированы по убыванию их объёмов. Такой приём позволяет несколько сократить ширину уровней дерева рекурсии, порожденного рекурсивным алгоритмом, и получить лучшие временные характеристики для рекурсивного и комбинированного алгоритмов.

Волновой алгоритм

В основе волнового алгоритма также лежит идея как эффективного сочетания достоинств обоих классических решений, так и их комплементарности. Главное отличие волнового алгоритма от комбинированного – это отсутствие предвычисления порога переключения. Ни до, ни во время поиска решения мы не анализируем функции трудоёмкости табличного и рекурсивного алгоритмов. Вместо этого, после расчёта упаковки очередным грузом, значение некоторого параметра определяет, каким из алгоритмов лучше обрабатывать последующий тип грузов.

Так как табличный алгоритм на каждом шаге вычисляет оптимальную упаковку для всех дискретных значений объёма от 0 до V, то его разумно использовать, когда следующий шаг рекурсии будет порождать значительное число листьев с различными значениями объёма упаковки. К тому же эффективность рекурсии при обработке каждого последующего груза в общем случае падает, т.к. возрастает число вершин, имеющих одинаковый аргумент целевой функции. Исходя из этого, на первом этапе волнового алгоритма управление передаётся рекурсии. Но в отличие от комбинированного и рекурсивного алгоритма в чистом виде, здесь используется иная структура данных. Мы «разворачиваем» рекурсию в таблицу хранения рекурсивных обращений. Что это означает? Порождение очередной вершины дерева рекурсии обрабатывается как создание строки таблицы с номером, соответствующим значению оставшегося объёма упаковки, и записью дополнительной информации, позволяющей организовать быстрые вычисления на последнем этапе волны обратных расчётов, эквивалентном цепочке рекурсивных возвратов. Для следующего типа груза создаётся новая таблица, при этом рекурсивные вызовы, порождающие цепочку таблиц, будут продолжаться до передачи управления табличному алгоритму. Порог переключения между алгоритмами

определяется динамически, исходя из условия рационального продолжения развёртывания рекурсии. После обработки очередного уровня рекурсии, т.е. создания таблицы для соответствующего типа груза, подсчитывается число заполненных строк текущей структуры данных – таблицы хранения рекурсивных обращений. Если число обращений достигло значения $(V+1)/2$, то мы предполагаем, что на следующем шаге число рекурсивных вызовов будет больше, чем число строк таблицы, и, следовательно, управление передаётся табличному алгоритму. Структура данных табличного алгоритма идентична структуре таблиц таблицы хранения рекурсивных обращений, что позволяет при обратном ходе волны непосредственно копировать результаты табличного алгоритма в соответствующую структуру. Схема волнового алгоритма показана на *рис. 2.*

Первый этап алгоритма – волна порождения цепочки таблиц, эквивалентных рекурсии по типам грузов. Первый этап продолжается до достижения рационального порога переключения на табличный алгоритм, т.е. до достижения половины заполненных строк таблицы хранения рекурсивных обращений. Второй этап – классический табличный алгоритм выполняется для оставшихся типов грузов. В точке переключения выполняется копирование из структуры данных табличного алгоритма в таблицу хранения рекурсивных обращений. Третий, заключительный этап – волна обратных расчётов, формирующая на последнем шаге решение поставленной задачи упаковки в первой порождённой таблице рекурсии.

Как и в случае комбинированного алгоритма, степень использования рекурсивного и табличного алгоритмов зависит от входных данных с той лишь разницей, что для всех входов волнового алгоритма обязательно будет построена таблица хранения рекурсивных обращений хотя бы для первого типа грузов. Таким образом, остаётся два варианта поведения волнового алгоритма:

- после обработки груза с номером заполненными оказываются более половины строк таблицы хранения рекурсивных обращений, и управление передаётся табличному алгоритму;
- число заполненных строк не достигает значения даже при рассмотрении предпоследнего груза. В этом случае табличный алгоритм не участвует в поиске решения.

Сравнение комбинированного и волнового алгоритмов

Рациональное совмещение рекурсивного и табличного алгоритмов решения задачи одномерной упаковки – общий принцип построения волнового и комбинированного алгоритмов. Этот принцип во многом обусловливает и схожесть структур этих алгоритмов. Так какой же из методов предпочтительнее использовать для получения быстрого решения? Ответ на этот вопрос не является вполне очевидным и зависит от аппаратных возможностей.

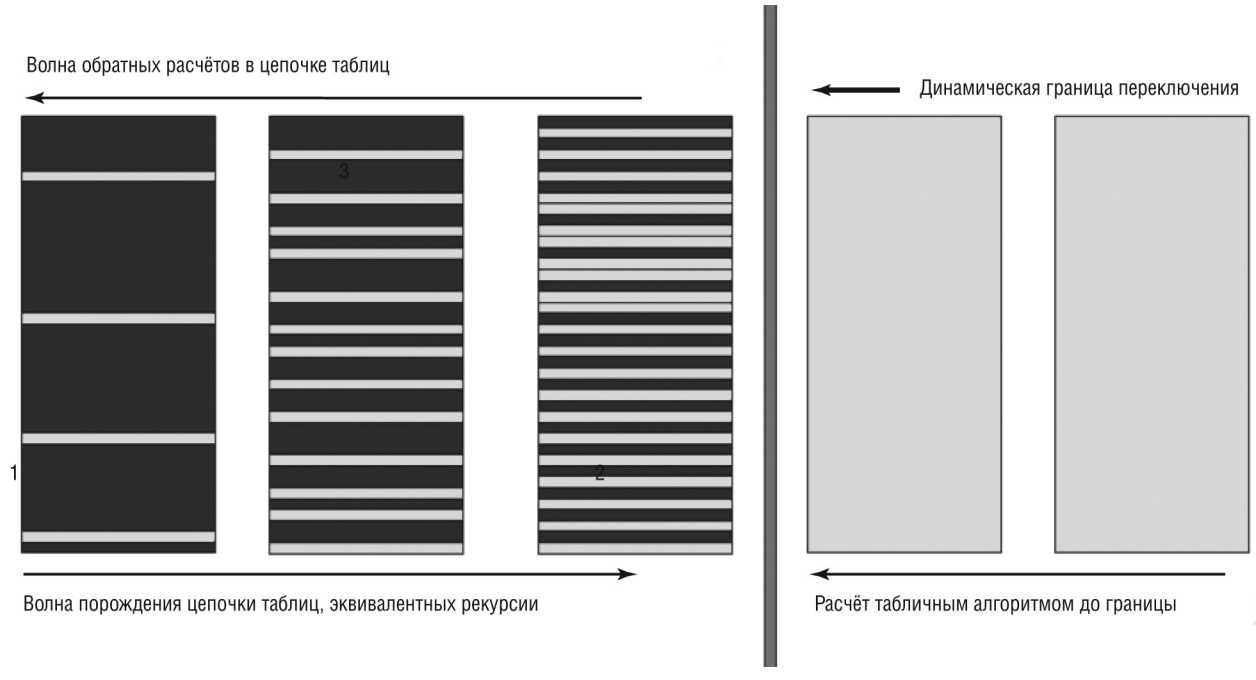


Рис. 2. Схема волнового алгоритма упаковки

Если определяющим фактором является время получения точного решения, то выбор за волновым алгоритмом. Он обеспечивает более быстрое решение за счёт:

- ✦ отсутствия модуля предвычисления порога переключения;
- ✦ отсутствия повторных расчётов целевой функции с одинаковым аргументом для вершин дерева рекурсии разных уровней;
- ✦ непосредственного копирования результатов работы табличного алгоритма в таблицу хранения рекурсивных обращений, благодаря идентичности структур.

Как известно, за всё приходится платить. В случае волнового алгоритма расплачиваться за временную эффективность приходится дополнительными затратами оперативной памяти на хранение полных ссылок рекурсивных вызовов. Волновой алгоритм демонстрирует классическую дилемму ресурсной эффективности – улучшение временной эффективности за счёт ухудшения ёмкостных характеристик.

Заключение

Таким образом, в статье предложены два алгоритма решения задачи одномерной оптимальной по стоимости упаковки, основанные на рациональном совмещении классических алгоритмов её решения – рекурсивном и табличном.

Комбинированный алгоритм использует результаты теоретического анализа трудоёмкости для предвычисления оптимального порога переключения между рекурсивным и табличным алгоритмами на основе данных конкретного входа.

Волновой алгоритм не выполняет статического расчёта порога переключения. Порог в данном алгоритме определяется динамически на основе подсчёта числа заполненных строк специальной таблицы хранения рекурсивных обращений. За счёт затрат оперативной памяти предлагаемый волновой алгоритм имеет лучшую временную эффективность.

Предложенные алгоритмы могут быть использованы при решении практических постановок задач бизнес-информатики, сводимых к задаче оптимальной одномерной упаковки. ■

Литература

1. Беллман Р., Дрейфус Р. Прикладные задачи динамического программирования: Пер. с англ. – М.: Наука, 1965, – 457 с.
2. Головешкин В. А., Ульянов М. В. Теория рекурсии для программистов. – М.: Издательство «Наука ФИЗМАТЛИТ», 2006. – 296 с.
3. Ульянов М. В., Гурин Ф. Е., Исаков А. С., Бударагин В. Е. Сравнительный анализ табличного и рекурсивного алгоритмов точного решения задачи одномерной упаковки // Exponenta Pro Математика в приложениях. 2004. №2(6). С. 64–70.

Издательство «Техносфера» пополнило серию «Мир программирования» новой книгой ***Виктора Александровича Сердюка,*** *преподавателя кафедры управления разработкой программного обеспечения ГУ-ВШЭ и генерального директора ЗАО «ДиалогНаука»* ***«Новое в защите от взлома корпоративных систем».***

СТРУКТУРНАЯ И ПАРАМЕТРИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ЛИНЕЙНОЙ МНК-МОДЕЛИ БЕЗ РЕШЕНИЯ СИСТЕМЫ НОРМАЛЬНЫХ УРАВНЕНИЙ

В.В. Белов,
доктор технических наук, профессор кафедры «Вычислительная и прикладная математика» Рязанского государственного радиотехнического университета, Адрес: 390005, Рязань, ул. Гагарина, 59/1, кафедра «Вычислительная и прикладная математика» Белов В.В. Тел: 8-903-835-54-98. E-mail: compvv@mail.ryazan.ru

Предлагается способ вычисления значений линейной МНК-модели, в частности MLR (Multiple Linear Regression), без оценивания её параметров. Предлагаемый способ может использоваться в алгоритмах поиска наилучшего в некотором смысле линейного описания процесса, представленного дискретными значениями (временным рядом). Предлагается рекуррентная схема вычисления параметров МНК-модели, альтернативная решению системы нормальных уравнений Гаусса.

Ключевые слова: Линейная МНК-модель. Последовательное введение переменных. Вычисление значений. Оценка параметров. Рекурсивная схема.

Предварительные замечания

Результаты, приведённые в данной статье, получены в процессе выполнения работ по заказам Министерства труда и социального развития России, а также Федеральной службы государственной статистики. Главные задачи работ состояли в получении прогнозных значений для заданных групп показателей производственного травматизма в РФ и занятости населения в экономике страны. Необходимые статистические данные были предоставлены Федеральной службой государственной статистики. Обусловливалось обязательное построение варианта модели в виде линейной множественной регрессии (MLR-описания) с регрессорами, входящими в сценарные условия прогнозирования. Допускалось использование альтернативных методов моделирования для дополнительной оценки надежности полученного прогноза. Поскольку крайне желательно находить MLR-описание с наименьшим количеством параметров (это, в частности, способствует минимизации дисперсии коэффициентов модели), в практике реальных приложений регрессионного анализа часто приходится решать задачу структурной идентификации линейной модели – поиска «наилучшей» в некотором смысле регрессии. В процессе решения этой задачи происходит последовательное добавление регрессоров в модель в разных сочетаниях до достижения требуемой погрешности аппроксимации.

Длительные упражнения в решении задачи последовательного синтеза наилучшей линейной модели и оценки качества частичных описаний привели к возникновению идеи вычисления векторов значений MLR-описания без предварительного оценивания его параметров, т.е. без осуществления операции параметрической идентификации. Такой приём позволяет ускорить процесс поиска: новый регрессор добавляется без повторного решения системы нормальных уравнений Гаусса. Кроме того, происходит, хотя и несущественное, повышение точности вычислений. Платой за эти преимущества являются ухудшение пространственных характеристик программного приложения – увеличиваются затраты на память.

Сопутствующим результатом явился альтернативный алгоритм параметрической идентификации – вычисления параметров линейной модели без решения системы нормальных уравнений. Однако он представляет, видимо, чисто теоретический интерес – как средство описания вариативности процедур оценки параметров линейных описаний. В плане утилитарности вряд ли можно указать условия целесообразности его применения.

История вопроса

Задача расширения линейной модели, параметры которой оцениваются методом наименьших квадратов, впервые рассмотрена в [1]. Однако В.М. Кохран ограничился рассмотрением случая добавления в модель одной переменной. На случай нескольких переменных подход В.М. Кохрана распространил М.Г. Квинауил [2]. Указанные результаты описаны в [3] и доступны на русском языке в переводе [4]. Общим началом подходов В.М. Кохрана и М.Г. Квинауила является то, что они предлагают схему добавления новых регрессоров, меняющую коэффициенты предшествующих описаний: добавление нового слагаемого $(a_p \cdot x_p)$ в MLR-описание сопровождается пересчетом всех коэффициентов $a_1, a_2, \ldots, a_{p-1}$ предыдущего варианта модели.

Последовательное расширение описания, предлагаемое в настоящей статье, отличается тем, что параметры модели не оцениваются вовсе:

1) коэффициент нового добавляемого в модель регрессора не вычисляется;

2) значения коэффициентов «старых» регрессоров не пересчитываются.

Предлагаемый способ целесообразен в тех случаях, когда конкретика значений коэффициентов не важна. Это, прежде всего, как уже указывалось, – задачи поиска наилучших линейных моделей с минимальных количеством параметров.

Классическое решение задачи последовательного введения регрессоров

При введении дополнительных регрессоров по Кохрану и Квинауилу предполагается, что подобрана модель регрессии

$$\mathrm{E}(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}, \quad \mathrm{D}(\mathbf{Y}) = \sigma^2 \mathbf{I}_n,$$

где $\mathbf{Y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$ – временной ряд, т.е. вектор последовательных равноотстоящих по времени значений некоторого процесса (результативного признака); $\mathbf{X}$ – матрица регрессоров (аргументов или факторных признаков) размером $n \times p$ ранга p; $\boldsymbol{\beta}$ – вектор коэффициентов регрессии, состоящий из p элементов; $\mathbf{I}_n$ – единичная матрица порядка n; σ – дисперсия значений временного ряда $\mathbf{Y}$. Оценка $\hat{\boldsymbol{\beta}}$ вектора $\boldsymbol{\beta}$ находится методом наименьших квадратов: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$.

Рассматривается задача добавления новых регрессоров таким образом, чтобы новая модель имела вид:

$$G: \ \mathbf{E}(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta}_G + \mathbf{Z}\boldsymbol{\gamma}_G = (\mathbf{X} \quad \mathbf{Z})\begin{pmatrix} \boldsymbol{\beta}_G \\ \boldsymbol{\gamma}_G \end{pmatrix} = \mathbf{W}\boldsymbol{\delta}_G,$$

где буква G идентифицирует новую модель;

$\mathbf{Z}$ – матрица дополнительных регрессоров размером $n \times t$ ранга t;

$\boldsymbol{\beta}_G$ – вектор новых коэффициентов при «старых» регрессорах, объединённых в матрицу $\mathbf{X}$;

$\boldsymbol{\gamma}_G$ – вектор коэффициентов при новых регрессорах, объединённых в матрицу $\mathbf{Z}$, состоящий из t элементов;

$\mathbf{W} = (\mathbf{X}\ \mathbf{Z})$ – матрица объединённых регрессоров, т.е. матрица всех регрессоров модели G, имеющая размер $n \times (p + t)$ и ранг $p + t$;

$\boldsymbol{\delta}_G = (\boldsymbol{\beta}_G\ \boldsymbol{\gamma}_G)^{\mathrm{T}}$ – вектор всех коэффициентов модели G, состоящий из $p + t$ элементов.

Указанные предположения означают, что имеет место стандартная ситуация: длина n временного ряда $\mathbf{Y}$ больше числа параметров модели, т.е. $n > p + t$ и все столбцы матрицы W (все регрессоры) линейно не зависимы.

Естественно, вычислить оценку $\hat{\boldsymbol{\delta}}_G$ вектора $\boldsymbol{\delta}_G$ и её дисперсионную матрицу $\mathrm{D}(\hat{\boldsymbol{\delta}})$ можно следующим:

$$\hat{\boldsymbol{\delta}}_G = (\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{Y}. \quad \mathrm{D}(\hat{\boldsymbol{\delta}}) = \sigma^2(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}.$$

Однако можно сократить объём вычислений, используя результат обращения матрицы $(\mathbf{X}^{\mathrm{T}}\mathbf{X})$, полученный ранее при вычислении оценки $\hat{\boldsymbol{\beta}}$ вектора коэффициентов $\boldsymbol{\beta}$.

Суть метода Кохрана-Квинауила выражается следующей теоремой [4, с. 69].

Теорема Себера

Пусть $\mathbf{R} = \mathbf{I}_n - \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$,

$$\mathbf{R}_G = \mathbf{I}_n - \mathbf{W}(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{W}^{\mathrm{T}},$$

$$\mathbf{L} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Z},$$

$$\mathbf{M} = [\mathbf{Z}^{\mathrm{T}}\mathbf{R}\mathbf{Z}]^{-1} \text{ и } \hat{\boldsymbol{\delta}}_G = \begin{pmatrix} \hat{\boldsymbol{\beta}}_G \\ \hat{\boldsymbol{\gamma}}_G \end{pmatrix}.$$

Тогда

(i) $\hat{\boldsymbol{\beta}}_G = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}(\mathbf{Y} - \mathbf{Z}\hat{\gamma}_G) = \hat{\boldsymbol{\beta}} - \mathbf{L}\hat{\gamma}_G$.

(ii) $\hat{\gamma}_G = (\mathbf{Z}^{\mathrm{T}}\mathbf{R}\mathbf{Z})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{R}\mathbf{Y}$.

(iii) $\mathbf{Y}^{\mathrm{T}}\mathbf{R}_G\mathbf{Y} = (\mathbf{Y} - \mathbf{Z}\hat{\gamma}_G)^{\mathrm{T}}\mathbf{R}(\mathbf{Y} - \mathbf{Z}\hat{\gamma}_G)$.

(iv) $\mathbf{Y}^{\mathrm{T}}\mathbf{R}_G\mathbf{Y} = \mathbf{Y}^{\mathrm{T}}\mathbf{R}\mathbf{Y} - \hat{\gamma}_G\mathbf{Z}\mathbf{R}\mathbf{Y}$

(v) $\mathrm{D}(\hat{\boldsymbol{\delta}}_G) = \sigma^2\begin{pmatrix} (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} + \mathbf{L}\mathbf{M}\mathbf{L}^{\mathrm{T}} & -\mathbf{L}\mathbf{M} \\ -\mathbf{M}\mathbf{L}^{\mathrm{T}} & \mathbf{M} \end{pmatrix}$.

Замечания:

1) пункт (ii) теоремы определяет алгоритм вычисления коэффициентов при новых переменных, добавляемых в модель; если вместо нескольких переменных добавить одну переменную, то матрица **Z** трансформируется в вектор **z**, одновременно матрицы $\mathbf{Z}^{\mathrm{T}}\mathbf{R}\mathbf{Z}$ и $\mathbf{Z}^{\mathrm{T}}\mathbf{R}\mathbf{Y}$ трансформируются в скаляры, формула коэффициента при добавляемой переменной принимает вид:

$$\hat{\gamma}_G = \frac{\mathbf{z}^{\mathrm{T}}\mathbf{R}\mathbf{Y}}{\mathbf{z}^{\mathrm{T}}\mathbf{R}\mathbf{z}},$$

и это уже не вектор, а скаляр;

2) пункт (i) определяет алгоритм коррекции вектора $\hat{\boldsymbol{\beta}}$ коэффициентов при переменных, уже входивших в модель; заметим, что он предполагает использование вектора $\hat{\boldsymbol{\gamma}}_G$ коэффициентов при добавляемых переменных, определяемого в следующем пункте, и, главное, – формула для матрицы **L** предполагает вычисление обратной матрицы $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$;

3) семантически матрица $\mathbf{L} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Z}$ представляет собой совокупность векторов модельных значений – j-й столбец этой матрицы представляет собой результат объяснения j-й добавляемой переменной всеми предыдущими переменными;

4) пункты (iii) и (iv) определяют две эквивалентные по значениям формулы вычисления остаточной суммы квадратов $\mathbf{Y}^{\mathrm{T}}\mathbf{R}_G\mathbf{Y}$ для итоговой модели через матрицы **R** и **Z**, формируемые старыми и новыми переменными.

Предлагаемое решение задачи последовательного введения регрессоров без параметрической идентификации

Теорема 1

Пусть $\mathbf{y} = (y_1, y_2,..., y_n)^{\mathrm{T}}$ – вектор вещественных чисел, условно называемый результативным признаком;

$$\mathbf{x}_1 = (x_{1,1}, x_{2,1},..., x_{n,1})^{\mathrm{T}},$$
$$\mathbf{x}_2 = (x_{1,2}, x_{2,2},..., x_{n,2})^{\mathrm{T}},$$
$$\mathbf{x}_{\mathrm{p}} = (x_{1,\mathrm{p}}, x_{2,\mathrm{p}},..., x_{n,\mathrm{p}})^{\mathrm{T}} -$$

векторы вещественных чисел, условно называемые объясняющими переменными или факторными признаками; $\mathbf{J}n = [1]_1^n$ – вектор единиц, состоящий из n элементов; $\mathbf{I}_n$ – единичная матрица размером $n \times n$, $\mathbf{X} = [\mathbf{x}_0\, \mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_p]$ – матрица объясняющих переменных, причём $\mathbf{x}_0 = \mathbf{J}_n$ и $\mathrm{rank}(\mathbf{X}) = p + 1$; $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2,..., \hat{y}_n)^{\mathrm{T}}$, – вектор значений линейной модели зависимости результативного признака от объясняющих переменных, т.е. $\hat{\mathbf{y}} = \mathbf{X}\cdot\mathbf{a}$, где $\mathbf{a} = (a_0, a_1,..., a_p)$ – вектор параметров (коэффициентов) модели, вычисляемых методом наименьших квадратов: $\mathbf{a} = (\mathbf{X}^{\mathrm{T}}\cdot\mathbf{X})^{-1}\cdot\mathbf{X}^{\mathrm{T}}\cdot\mathbf{y}$; $\mathbf{P} = \mathbf{X}\cdot(\mathbf{X}^{\mathrm{T}}\cdot\mathbf{X})\cdot\mathbf{X}^{\mathrm{T}}$ – матрица, порождающая значения модели $\hat{\mathbf{y}} = \mathbf{P}\cdot\mathbf{y}$.

Тогда:

1) проекционная матрица $\mathbf{P} = \mathbf{X}\cdot(\mathbf{X}^{\mathrm{T}}\cdot\mathbf{X})^{-1}\cdot\mathbf{X}^{\mathrm{T}}$, порождающая значения $\hat{\mathbf{y}}$ линейной модели, построенной с использованием факторных признаков $x_1, x_2,..., x_p$, эквивалентной по значениям описанию, параметры которого оцениваются методом наименьших квадратов, может быть получена по рекуррентной схеме: $\mathbf{P} = \mathbf{P}_p$, где $\mathbf{P}_p$ – финальное значение в последовательности рекуррентных вычислений

$$\mathbf{P}_j = \mathbf{P}_{j-1} + \frac{\mathbf{R}_{j-1}\cdot\mathbf{x}_j\cdot\mathbf{x}_j^{\mathrm{T}}\cdot\mathbf{R}_{j-1}}{\mathbf{x}_j^{\mathrm{T}}\cdot\mathbf{R}_{j-1}\cdot\mathrm{x}_j}, \quad j = \overline{1, p};$$
$$\mathbf{R}_{j-1} = \mathbf{I}_n - \mathbf{P}_{j-1};$$
$$\mathbf{P}_0 = \frac{\mathbf{J}_n\cdot\mathbf{J}_n^{\mathrm{T}}}{\mathbf{J}_n^{\mathrm{T}}\cdot\mathbf{J}_n} = \frac{\mathbf{J}_n\cdot\mathbf{J}_n^{\mathrm{T}}}{n};$$

2) остаточная сумма квадратов $RSS = \mathbf{y}^{\mathrm{T}}\cdot(\mathbf{I}_n - \mathbf{P}_p)\cdot\mathbf{y}$.

Доказательство

Запишем формулу (3.37) [4, с. 72], описывающую значения модели с ортогональной структурой, для частного случая, когда матрица **Z** вырождается в вектор $\mathbf{x}_j$ в новых обозначениях:

$$\hat{\mathbf{y}}^{[j]} = \mathbf{X}_{j-1}\cdot\mathbf{a}_{j-1}^{[j-1]} + \mathbf{R}_{j-1}\cdot\mathbf{x}_j\cdot a_j^{[j]}. \qquad (*)$$

В соответствии с пунктом (ii) теоремы Себера (*см. выше*) для рассматриваемого частного случая и используемых обозначений имеем:

$$a_j^{[j]} = \frac{\mathbf{x}_j^{\mathrm{T}}\cdot\mathbf{R}_{j-1}\cdot\mathbf{y}}{\mathbf{x}_j^{\mathrm{T}}\cdot\mathbf{R}_{j-1}\cdot\mathbf{x}_j}.$$

Кроме того, учтём, что

$$\mathbf{X}_{j-1}\cdot\mathbf{a}_{j-1}^{[j-1]} = \hat{\mathbf{y}}^{[j-1]}.$$

Подставляя указанные выражения в (*), получим:

$$\hat{\mathbf{y}}^{[j]} = \hat{\mathbf{y}}^{[j-1]} + \mathbf{R}_{j-1}\cdot\mathbf{x}_j\cdot\frac{\mathbf{x}_j^{\mathrm{T}}\cdot\mathbf{R}_{j-1}\cdot\mathbf{y}}{\mathbf{x}_j^{\mathrm{T}}\cdot\mathbf{R}_{j-1}\cdot\mathbf{x}_j}.$$

Далее с учётом того, что

$$\hat{\mathbf{y}}^{[j]} = \mathbf{P}_j \cdot \mathbf{y} \quad \text{и} \quad \hat{\mathbf{y}}^{[j-1]} = \mathbf{P}_{j-1} \cdot \mathbf{y}$$

имеем

$$\mathbf{P}_j \cdot \mathbf{y} = \mathbf{P}_{j-1} \cdot \mathbf{y} + \frac{\mathbf{R}_{j-1} \cdot \mathbf{x}_j \cdot \mathbf{x}_j^{\mathrm{T}} \cdot \mathbf{R}_{j-1}}{\mathbf{x}_j^{\mathrm{T}} \cdot \mathbf{R}_{j-1} \cdot \mathbf{x}_j} \cdot \mathbf{y}.$$

Откуда и следует, что матрица $\mathbf{P}_j$ может быть вычислена рекуррентно:

$$\mathbf{P}_j = \mathbf{P}_{j-1} + \frac{\mathbf{R}_{j-1} \cdot \mathbf{x}_j \cdot \mathbf{x}_j^{\mathrm{T}} \cdot \mathbf{R}_{j-1}}{\mathbf{x}_j^{\mathrm{T}} \cdot \mathbf{R}_{j-1} \cdot \mathbf{x}_j}.$$

Терминальная ветвь рекурсии для случая $j = 1$ определяется значением $\mathbf{P}_0$. По определению

$$\mathbf{P}_0 = \mathbf{x}_0 \cdot (\mathbf{x}_0^{\mathrm{T}} \cdot \mathbf{x}_0)^{-1} \cdot \mathbf{x}_0^{\mathrm{T}}.$$

Поскольку $\mathbf{x}_0 = \mathbf{J}_n$, а произведение $\mathbf{J}_n^{\mathrm{T}} \cdot \mathbf{J}_n$ скалярно и равно n, имеем:

$$\mathbf{P}_0 = \frac{\mathbf{J}_n \cdot \mathbf{J}_n^{\mathrm{T}}}{n}.$$

Остаточная сумма квадратов RSS по определению равна $\mathbf{y}^{\mathrm{T}} \cdot (\mathbf{I}_n - \mathbf{P}) \cdot \mathbf{y}$. Равенство $\mathbf{P} = \mathbf{P}_p$ определяет возможность вычисления остаточной суммы по формуле $RSS = \mathbf{y}^{\mathrm{T}} \cdot (\mathbf{I}_n - \mathbf{P}_p) \cdot \mathbf{y}$.

Алгоритм вычислений

В соответствии с указанной теоремой алгоритм вычисления вектора значений $\hat{\mathbf{y}}$ линейной модели без операции её параметрической идентификации имеет вид:

3) вспомогательные величины:

(1) $\mathbf{J}n = [1]_1^n$ – вектор единиц, состоящий из n элементов;

(2) $\mathbf{I}_n = \mathrm{identity}(n)$ – единичная матрица размером $n \times n$;

4) начальное значение:

$$\mathbf{P}_0 = \frac{\mathbf{J}_n \cdot \mathbf{J}_n^{\mathrm{T}}}{\mathbf{J}_n^{\mathrm{T}} \cdot \mathbf{J}_n} = \frac{\mathbf{J}_n \cdot \mathbf{J}_n^{\mathrm{T}}}{n} -$$

квадратная матрица размером $n \times n$, все элементы которой равны $1/n$, – начальное значение матрицы, порождающей линейную модель, построенную по первому столбцу $\mathbf{x}_0 = \mathbf{J}_n$ матрицы объясняющих переменных $\mathbf{X}p$;

5) рекуррентные вычисления для $j = \overline{1, p}$:

(1) $\mathbf{R}_{j-1} = \mathbf{I}_n - \mathbf{P}_{j-1}$ – вспомогательная матрица для упрощения записи формул;

(2) если $\mathbf{R}_{j-1} \cdot \mathbf{x}_j \neq [0]_1^n$, где $[0]_1^n$ – вектор, состоящий из одних нулей, то вычисляется

$$\mathbf{P}_j = \mathbf{P}_{j-1} + \frac{\mathbf{R}_{j-1} \cdot \mathbf{x}_j \cdot \mathbf{x}_j^{\mathrm{T}} \cdot \mathbf{R}_{j-1}}{\mathbf{x}_j^{\mathrm{T}} \cdot \mathbf{R}_{j-1} \cdot x_j} -$$

очередное значение порождающей матрицы $\mathbf{P}$;

(3) иначе, если $\mathbf{R}_{j-1} \cdot \mathbf{x}_j = [0]_1^n$, то переменная x_j исключается из списка потенциальных аргументов модели и осуществляется переход к следующему значению j;

6) вычисляется вектор $\hat{\mathbf{y}}(\mathbf{X}) = \mathbf{P}_p \cdot \mathbf{y}$ значений модели, построенной по матрице объясняющих переменных $\mathbf{X} = [\mathbf{x}_0\ \mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_p]$.

Замечания:

1) вычисление значений модели по указанному алгоритму не требует выполнения операции обращения матрицы;

2) на каждом шаге рекуррентных вычислений могут быть получены значения частных описаний, построенных по части объясняющих переменных: $\mathbf{y}^{[j]} = \hat{\mathbf{y}}(\mathbf{x}_0\ \mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_p) = \mathbf{P}_j \cdot \mathbf{y}$;

3) условие $\mathbf{R}_{j-1} \cdot \mathbf{x}_j \neq [0]_{n \times 1}$ исключает переменные, векторы значений которых коллинеарны с векторами уже включёнными в модель; произведение $\mathbf{R}_{j-1} \cdot \mathbf{x}_j$ представляет собой вектор ошибок объяснения вектора $\mathbf{x}_j$ векторами $\mathbf{x}_0\ \mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_{j-1}$.

Вычислительный эксперимент

Проверка правильности алгоритма была осуществлена путём генерации нескольких векторов случайных чисел, равномерно распределённых от нуля до десяти. Один из векторов интерпретировался как значения результативного признака $\mathbf{y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$, а остальные – как значения факторных признаков $\mathbf{x}_1 = (x_{1,1}, x_{2,1}, \ldots, x_{n,1})^{\mathrm{T}}$, $\mathbf{x}_2 = (x_{1,2}, x_{2,2}, \ldots, x_{n,2})^{\mathrm{T}}$, …, $\mathbf{x}_p = (x_{1,p}, x_{2,p}, \ldots, x_{n,p})^{\mathrm{T}}$. Матрица, порождающая значения модели находилась двумя способами – рекуррентно $\mathbf{P}_p$ и путём прямых вычислений $\mathbf{P} = \mathbf{X} \cdot (\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^{\mathrm{T}}$. Затем находились векторы значений модели $\hat{\mathbf{y}}^{[p]} = \mathbf{P}_p \cdot \mathbf{y}$ и $\hat{\mathbf{y}} = \mathbf{P}_p \cdot \mathbf{y}$.

Длины векторов, участвовавших в вычислениях выбрались равными: 5; 10; 30; 100 и 500. Количество факторных признаков: 4; 7 и 15.

Расчёты выполнялись с использованием восьми байтов для представления вещественных чисел. Результаты показали, что во всех случаях абсолютная величина разности элементов векторов $\hat{\mathbf{y}}^{[p]}$ и $\hat{\mathbf{y}}$ мала:

$$\max\left(\overrightarrow{\left|\hat{\mathbf{y}}^{[p]} - \hat{\mathbf{y}}\right|}\right) < 10^{-14},$$

где стрелка над выражением символизирует операцию векторизации, в данном случае – формирования вектора абсолютных значений разностей

соответствующих элементов матриц и векторов. Таким образом, различие результатов вычисления проекционных матриц и значений модели классическим и предлагаемым способами имеет порядок погрешности внутримашинного представления вещественных данных.

Теорема 2

Пусть $\mathbf{y} = (y_1, y_2,..., y_n)^{\mathrm{T}}$ – вектор вещественных чисел, условно называемый результативным признаком;

$$\mathbf{x}_1 = (x_{1,1}, x_{2,1},..., x_{n,1})^{\mathrm{T}},$$
$$\mathbf{x}_2 = (x_{1,2}, x_{2,2},..., x_{n,2})^{\mathrm{T}},$$
$$\mathbf{x}_p = (x_{1,p}, x_{2,p},..., x_{n,p})^{\mathrm{T}} -$$

векторы вещественных чисел, условно называемые объясняющими переменными или факторными признаками; $\mathbf{J}_n = [1]_1^n$ – вектор единиц, состоящий из n элементов; $\mathbf{I}_n$ – единичная матрица размером $n \times n$, $\mathbf{X} = [\mathbf{x}_0\ \mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_p]$ – матрица объясняющих переменных, причём $\mathbf{x}_0 = \mathbf{J}_n$ и $\mathrm{rank}(\mathbf{X}) = p + 1$; $\mathbf{a} = (a_0, a_1,..., a_p]$ – вектор параметров (коэффициентов) модели, вычисляемых методом наименьших квадратов: $\mathbf{a} = (\mathbf{X}^{\mathrm{T}}\cdot\mathbf{X})^{-1}\cdot\mathbf{X}^{\mathrm{T}}\cdot\mathbf{y}$.

Тогда вектор параметров **a** линейной модели может быть получен альтернативно по рекуррентной схеме:

$$a_p = \frac{\mathbf{x}_p^{\mathrm{T}} \cdot \mathbf{R}_{p-1} \cdot \mathbf{y}}{\mathbf{x}_p^{\mathrm{T}} \cdot \mathbf{R}_{p-1} \cdot \mathbf{x}_p};$$

$$a_k = \frac{\mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{R}_{k-1} \cdot (\mathbf{y} - \sum_{j=k+1}^{p} a_k \mathbf{x}_k)}{\mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{R}_{k-1} \cdot \mathrm{x}_k},$$

$$k = p-1, p-2,...,1;$$

$$a_0 = \frac{\mathbf{J}_n^{\mathrm{T}} \cdot (\mathbf{y} - \sum_{k=1}^{\mathrm{p}} a_k \mathbf{x}_k)}{n}.$$

Доказательство

Утверждение 1. Обозначим $\mathbf{X}_k = [\mathbf{x}_0\ \mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_k]$, $k \le p$. При этом $\mathbf{X} = [\mathbf{x}_0\ \mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_p] = \mathbf{X}_p$. Согласно уравнениям системы $\mathbf{X}^{\mathrm{T}}\cdot\mathbf{X}\cdot\mathbf{a} = \mathbf{X}^{\mathrm{T}}\cdot\mathbf{y}$ нормальных уравнений Гаусса равенство справедливо для всех $j = \overline{0; p}$, поскольку $\mathbf{X}^{\mathrm{T}}\cdot(\mathbf{y}-\mathbf{X}\cdot\mathbf{a}) = [0]_1^{p+1}$ и $\mathrm{rank}(\mathbf{X}) = p+1$. Вследствие этого равенства $\mathrm{X}_k^{\mathrm{T}}\cdot\mathrm{e} = [0]_1^{p+1}$ справедливы для всех $k = \overline{0; \mathrm{p}}$.

Утверждение 2. Если в качестве объясняемой переменной использовать одну из объясняющих переменных $\mathbf{x}_j$, где $j \le k$, с числовым коэффициентом a, то вектор $\mathbf{b} = (b_0, b_1,..., b_k)^{\mathrm{T}}$, представляющий собой решение соответствующей системы нормальных уравнений Гаусса $\mathbf{X}_k^{\mathrm{T}}\cdot\mathbf{X}k\cdot\mathbf{b} = \mathbf{X}^{\mathrm{T}}\cdot\mathbf{x}_j a$, таков: $b_j = a$; $b_i = 0$ при $0 \le i<j$ и $j<i \le k$, т.е. $b_i = 0$ при $i \ne j$.

Утверждение 3. Рассмотрим систему алгебраических уравнений

$$\mathbf{X}_k^{\mathrm{T}} \cdot \mathbf{X}_k \cdot \mathbf{b} = \mathbf{X}^{\mathrm{T}} \cdot (\mathbf{y} - \sum_{\mathrm{j}=k+1}^{p} a_j \mathbf{x}_j).$$

относительно вектора **b**. Заметим, что

$$\mathbf{y} - \sum_{j=k+1}^{p} a_j \mathbf{x}_j = \sum_{j=0}^{k} a_j \mathbf{x}_j + \mathbf{e}.$$

Рассматриваемая система принимает вид

$$\mathbf{X}_k^{\mathrm{T}} \cdot \mathbf{X}_k \cdot \mathbf{b} = \mathbf{X}^{\mathrm{T}} \cdot (\sum_{j=0}^{k} a_j \mathbf{x}_j + \mathrm{e})$$

и с учётом утверждений 1 и 2 имеем в качестве решения $\mathbf{b} = (a_0, a_1,..., a_k)$, т.е. вектор **b** является частью вектора **a** – состоит из первых $k + 1$ элементов этого вектора.

Утверждение 4. Формула для вычисления a_p определяет значение последнего (с наибольшим номером) коэффициента линейной модели согласно пункту (ii) теоремы Себера для частного случая $\mathbf{Z} = \mathbf{x}_p$.

Утверждение 5. Формула

$$\frac{\mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{R}_{k-1} \cdot (\mathbf{y} - \sum_{j=k+1}^{p} a_k \mathbf{x}_k)}{\mathbf{x}_k^{\mathrm{T}} \cdot \mathbf{R}_{k-1} \cdot \mathbf{x}_k}$$

определяет значение последнего (k-го) коэффициента линейной модели для объясняемого вектора

$$\mathbf{y} - \sum_{j=k+1}^{p} a_j \mathbf{x}_j.$$

Согласно утверждению 3 этот коэффициент совпадает по значению с искомым коэффициентом a_k.

Утверждение 6. Заметим предварительно, что среднее значение $\bar{v}$ произвольного вектора **v** равно

$$\bar{v} = \frac{\mathbf{J}_n^{\mathrm{T}} \cdot \mathbf{v}}{\mathbf{J}_n^{\mathrm{T}} \cdot \mathbf{J}_n}.$$

При этом $\mathbf{J}_n^{\mathrm{T}}\cdot\mathbf{J}_n = n$. Формула для вычисления a_0 получается как результат объяснения остатка

$$\mathbf{y} - \sum_{j=1}^{p} a_j \mathbf{x}_j$$

вектора **y** вектором $\mathbf{x}_0 = \mathbf{J}$, т.е. это среднее значение вектора

$$\mathbf{y} - \sum_{j=1}^{p} a_j \mathbf{x}_j.$$

Пример практического использования предлагаемых решений

Предлагаемый способ последовательного введения регрессоров без параметрической идентификации был использован, в частности, для решения задачи прогноза значений группы показателей

производственного травматизма, занятости населения, реально располагаемых денежных доходов и заработной платы в РФ. В качестве исходных данных использованы временные ряды Государственного комитета по статистике РФ.

Прогнозируемые показатели для упрощения ссылок пронумерованы следующим образом:

$Т_1$ – «Число пострадавших с утратой трудоспособности на 1 рабочий день и более (в том числе со смертельным исходом), тыс. чел.»;

$Т_2$ – «Число пострадавших со смертельным исходом, чел.»;

$Т_3$ – «Число пострадавших с утратой трудоспособности на 1 рабочий день и более (в том числе со смертельным исходом), в расчете на 1000 работающих»;

$Т_4$ – «Число пострадавших со смертельным исходом в расчете на 1000 работающих»;

$Т_5$ – «Число человеко-дней нетрудоспособности на 1 рабочий день и более, временная нетрудоспособность которых закончилась в отчетном году, в расчете на 1000 человек».

$Э_1$ – «Численность занятого в экономике населения, млн чел.»;

$Э_2$ – «Общая численность безработных, млн чел.»;

$Э_3$ – «Реально располагаемые денежные доходы населения в % к предыдущему году»;

$Э_4$ – «Номинальная начисленная среднемесячная заработная плата на 1 работника, руб.».

Показатели $Т_1$ – $Т_5$, $Э_1$, $Э_2$, $Э_4$ – являются абсолютными (не относительными) и имеют натуральное выражение; $Э_3$ – цепной годовой индекс, измеряемый в процентах.

Набор факторных переменных состоял из 12 показателей, каждый из которых в процессе предварительного корреляционного анализа учитывался с лагами 0, 1, 2 и 3 года. Таким образом, общее число потенциальных регрессоров составило 48. В результате поиска «лучших» описаний для каждой модели был определен адекватный набор регрессоров. Множество всех адекватных регрессоров финальных моделей включает:

1) ВВП – валовой внутренний продукт в сопоставимых ценах; используется в 5 моделях;

2) $ВВП_{-2}$ – ВВП с лагом в 2 года; используется в двух моделях;

3) ПотрЦен – индекс потребительских цен; используется в двух моделях;

4) ПотрЦен-2 – ПотрЦен с лагом в 2 года; используется в одной модели;

5) ИнвОК – инвестиции в основной капитал в сопоставимых ценах; используется в трех моделях;

6) $ИнвОК_{-2}$ – ИнвОК с лагом в 2 года; используется в трех моделях;

7) ОПП – объем промышленного производства в сопоставимых ценах; используется в одной модели;

8) $ОПП_{-1}$ – ОПП с лагом в 1 год; используется в одной модели;

9) ЧисЗан – численность занятого в экономике населения; используется в трех моделях;

10) $ЧисЗан_{-1}$ – ЧисЗан с лагом в 1 год; используется в одной модели;

11) $ЧисЗан_{-2}$ – ЧисЗан с лагом в 2 года; используется в одной модели.

В данных Государственного комитета по статистике РФ значения факторных показателей представлены в абсолютном выражении и в виде цепных индексов. В то же время известно, что факторные показатели целесообразно представлять в виде базисных индексов, так как проблемные показатели более коррелированны с базисными индексами, нежели с цепными. Кроме этого, базирование отображает данные в окрестность интервала [0; 100], что приводит к одному порядку значений коэффициентов регрессии.

Исходные статистические данные для показателей производственного травматизма начинаются с 1993 г., данные по социально-экономическим показателям – с 1991 г. Этот факт обусловил то, что в качестве информационной платформы для построения моделей использовались показатели социально-трудовой сферы, начиная с 1993 г. В качестве базы использованы данные за 1991 г. Базисные индексы этого года приняты равными 100 %. Расчёт базисных индексов для k-го года через цепные осуществляется с помощью соотношения:

$$J_X(k) = J_X(k-1)\cdot j_X(k)/100,$$

где $J_X(k)$, $j_X(k)$ – соответственно базисный и цепной годовой индексы фактора X.

В качестве сценарных условий для прогнозирования использованы «Основные показатели прогноза социально-экономического развития Российской Федерации до 2011 года» от Минэкономразвития РФ.

Окончательная модель для показателя $Т_1$ является четырехпараметрической с тремя факторами без лагов:

$$T_1(t) = a_0 + a_1\cdot J_{ВВП}(t) + a_2\cdot J_{ИнвОК}(t) + a_3\cdot J_{ЧисЗан}(t),$$

где t – номер года, $t \geq 1993$; $a_0 = -598{,}11$; $a_1 = -8{,}94$; $a_2 = 7{,}81$; $a_3 = 12{,}86$.

Окончательная модель для показателя $Т_2$ содержит два фактора без лагов:

$$T_2(t) = a_0 + a_1 \cdot J_{ВВП}(t) + a_2 \cdot J_{ИнвОК}(t),$$

где t – номер года, $t \geq 1993$; $a_0 = 12341{,}26$; $a_1 = -241{,}06$; $a_2 = 270{,}71$.

Окончательная модель для показателя $Т_3$ содержит также два фактора без лагов:

$$T_3(t) = a_0 + a_1 \cdot J_{ИнвОК}(t) + a_2 \cdot J_{ЧисЗан}(t),$$

где t – номер года, $t \geq 1993$; $a_0 = -15{,}04$; $a_1 = -0{,}0653$; $a_2 = 0{,}257$.

Окончательная модель для показателя $Т_4$ имеет три фактора:

$$T_4(t) = a_0 + a_1 \cdot J_{ВВП}(t) + a_2 \cdot J_{ОПП}(t) + a_3 \cdot J_{ЧисЗан}(t-1),$$

где t – номер года, $t \geq 1994$; $a_0 = 0{,}476$; $a_1 = 0{,}00776$; $a_2 = -0{,}00589$; $a_3 = 0{,}00567$. Индекс численности занятых имеет лаг в 1 год.

Для показателя $Т_5$ финальная модель содержит два фактора:

$$T_5(t) = a_0 + a_1 \cdot J_{ИнвОК}(t-2) + a_2 \cdot J_{ЧисЗан}(t),$$

где t – номер года, $t \geq 1995$; $a_0 = 12{,}24$; $a_1 = -0{,}089$; $a_2 = 0{,}212$. Индекс инвестиций в основной капитал имеет лаг в 2 года.

Для показателя $Э_1$ модель содержит два фактора без лагов:

$$Э_1(t) = a_0 + a_1 \cdot J_{ВВП}(t) + a_2 \cdot J_{ПотрЦен}(t),$$

где t – номер года, $t \geq 1993$; $a_0 = 49{,}68$; $a_1 = -0{,}2748$; $a_2 = -0{,}0006274$.

Для показателя $Э_2$ модель содержит три фактора с лагами в 2 года и 1 год:

$$Э_2(t) = a_0 + a_1 \cdot J_{ВВП}(t-2) + a_2 \cdot J_{ИнвОК}(t-2) + a_3 \cdot J_{ОПП}(t-1),$$

где t – номер года, $t \geq 1995$; $a_0 = 11{,}08$; $a_1 = 0{,}2884$; $a_2 = -0{,}2589$; $a_3 = -0{,}2466$.

Для показателя $Э_3$ модель содержит три фактора:

$$Э_3(t) = a_0 + a_1 J_{ВВП}(t) + a_2 J_{ПотрЦен}(t) + a_3 J_{ЧисЗан}(t-2),$$

где t – номер года, $t \geq 1995$; $a_0 = 401{,}84$; $a_1 = 3{,}63$; $a_2 = -0{,}00974$; $a_3 = -5{,}563$. Индекс численности занятых имеет лаг в 2 года.

Для показателя $Э_4$ модель содержит три фактора с лагами в 2 года:

$$Э_4(t) = a_0 + a_1 \cdot J_{ВВП}(t-2) + a_2 \cdot J_{ПотрЦен}(t-2) + a_3 \cdot J_{ИнвОК}(t-2),$$

где t – номер года, $t \geq 1995$; $a_0 = -2899{,}9$; $a_1 = 60{,}554$; $a_2 = 0{,}51199$; $a_3 = -26{,}595$.

Все полученные модели имеют хорошие показатели качества:

1) они значимы в целом по критерию Фишера на уровне значимости $\alpha = 0{,}01$;

2) все коэффициенты отличны от нуля по t-критериям на том же уровне значимости, таким образом, все модели являются приемлемыми по критерию минимальной значимой разности (least significant difference);

3) показатель прогностической силы по Дрейперу и Смиту [5]

$$\frac{F}{F_{k_1,k_2}^{0,01}}.$$

всех полученных описаний существенно (в два раза и более) превышает пороговый уровень 4, т.е. модели пригодны для решения задачи прогнозирования.

На основании изложенного можно утверждать, что прогнозы, получаемые по полученным описаниям имеют надежность, совпадающую с надежностью сценарных условий. Наиболее существенная качественная особенность полученных результатов прогноза состоит в следующем:

1) для показателей социально-экономического развития РФ $Э_1$, $Э_3$, $Э_4$ полученные прогнозные значения количественно и качественно близки к прогнозу Минэкономразвития России, что свидетельствует о том, что оба прогноза отражают одни и те же взаимосвязи между показателями и факторами;

2) для показателя $Э_2$ («Общая численность безработных, млн чел.»); полученные прогнозные значения существенно превышают прогноз Минэкономразвития России, что свидетельствует о том, что Минэкономразвития при прогнозировании этого показателя использовал не взаимосвязи между «первичными» и «вторичными» показателями, а некоторые экспертные рассуждения.

Выводы

Научными результатами, представленным в настоящей статье, являются рекуррентные методы вычисления:

1) проекционной матрицы **P**, порождающей вектор значений линейной модели без оценки её параметров;

2) вектора параметров линейной модели без использования операции обращения матрицы объясняющих переменных.

Прикладное значение метода вычисления проекционной матрицы состоит в том, что на его основе создан алгоритм предназначенный для использования в процедурах поиска наилучшего состава регрессоров MLR-модели в заданном множестве потенциальных аргументов $x_1, x_2,..., x_p$, ориентированной на решение задачи прогнозирования. На первом этапе этих процедур отыскиваются регрессии лучшие по *RSS* – остаточной сумме квадратов с одним, двумя, тремя и так далее *p* регрессорами. При этом не требуется параметрическая идентификация модели. Затем среди найденных моделей находится наиболее устойчивое описание, прогнозирующие способности которой наиболее существенны.

Использование предложенного алгоритма на первом этапе процедуры поиска наилучшей модели в некоторых случаях позволяет сократить время и повысить точность вычислений. Указанные эффекты достигаются за счёт увеличения используемых объёмов оперативной памяти.

Предложенный в настоящей статье алгоритм вычисления параметров линейной модели без использования операции обращения матрицы объясняющих переменных представляет, видимо, чисто теоретический интерес, однако, можно утверждать, что изложенные материалы в своей совокупности позволяют исследователям глубже проникнуть в тонкости процедур оценки параметров линейной модели и расширяют арсенал средств аналитиков-практиков. ■

Литература

1. Cochran W.G. The omission or addition an independent variable in multiple linear regression // J. R. Stat. Soc. Suppl., № 5, pp. 171–176.
2. Quenouille M.H. An application of least squares to family diet surveys // Econometrica, № 18, pp. 27–44.
3. Seber G.A.F. Linear regression analysis. Wiley: New York, London, Sydney and Toronto. 1977.
4. Себер Дж. Линейный регрессионный анализ: Пер с англ. М.: Мир, 1980. 456 с.
5. Дрейпер Н., Смит Г. Прикладной регрессионный анализ: в 2-х кн. Кн.1: Пер. с англ. М.: Финансы и статистика, 1986. 366 с.

КОМБИНИРОВАННАЯ СХЕМАФОРМИРОВАНИЯ СТРАТЕГИИ МАРКЕТИНГА

М.Ш. Левин,
кандидат технических наук, старший научный сотрудник
Института проблем передачи информации РАН
Адрес: 127994, Москва, Большой Каретный пер. д. 19, стр. 1. E-mail: mslevin@acm.org

Рассматривается комбинированная схема формирования стратегии маркетинга для групп товаров, включающая четыре этапа: (а) формирование групп похожих товаров на основе кластеризации, (б) формирование сегментов рынка на основе кластеризации возможных покупателей, (в) определение соответствия между группами товаров и сегментами рынка (на основе задачи о назначении), (г) определение конкретной стратегии маркетинга для каждой пары «группа товаров – сегмент рынка» (на основе задачи о блочном рюкзаке с учётом общего ограничения на реализацию маркетинговых стратегий для всех групп товаров). В качестве численного примера приводится формирование маркетинговой стратегии для компьютерных товаров (компьютеры, процессоры, программы и др.).

Ключевые слова: комбинированная схема маркетинга, кластеризация, задача множественного выбора, сегментация рынка.

Введение

Исследования в области построения маркетинговых стратегий (планов проводятся многие годы [1–8]. В последние годы важность задач построения маркетинговых стратегий возрастает. В значительной мере это связано со следующим [5, 9–11]:

а) усложнением типов товаров и услуг;

б) возрастанием значимости вопросов технической поддержки и сервисного обслуживания товаров и услуг;

в) многообразием типов потребителей и их взаимосвязей;

г) повышением значимости ряда системных вопросов.

На *рис. 1* приведена упрощённая схема, включающая 4 компонента: компания, товары, маркетинговая стратегия, потребитель/покупатель.

Естественно возникают потребности в решении ряда вопросов: уменьшение размерности решаемых задач, в частности, группировка типов товаров; группировка возможных типов потребителей (т.е. формирование «крупных» сегментов рынка), выбор для реализации наиболее эффективных маркетинговых операций, эффективное использование сотрудников, выполнение бюджетных ограничений на проведение маркетинговых операций. Можно указать, например, следующие основные подходы к построению маркетинговых стратегий (планов):

Рис. 1. Упрощённая схема маркетинга

1. Статистические методы [12, 13].
2. Подходы на основе моделей теории игр [14, 15].
3. Подходы на основе Марковских процессов [16–18].
4. Динамическое программирование [17, 19].
5. Использование моделей выявления знаний (например, data mining) [12, 20].
6. Подходы на основе многокритериального принятия решений [21, 22].
7. Экспериментальные подходы (experimental marketing) [15].
8. Подходы на основе теории возможностей и/или размытых множеств [21].
9. Подходы на основе специальных рекомендующих систем (recommender systems) [23].

10. Методы на основе деревьев решений (decision trees) [24].

11. Методы на основе кластеризации [10, 25].

12. Подходы на основе методов искусственного интеллекта, баз знаний (например: системы продукционных правил, case-based reasoning CBS) [1, 26–28].

13. Использование методов нейронных вычисления (neural computing) [29, 30].

14. Интерактивные процедуры на основе методов визуализации информации [31, 32].

15. Комбинированные (гибридные) подходы [21, 22, 27].

Данная статья направлена на использование составной схемы решения на основе 4-х задач, каждая из которых базируется на комбинаторной модели, включая следующие:

1) задача многокритериального ранжирования;

2) задача кластеризации;

3) задача назначения/размещения;

4) задача блочного рюкзака.

Таким образом рассматривается комбинированная схема решения формирования стратегии маркетинга для групп товаров. Данная стратегия включает 4 этапа (*рис. 2*):

а) формирование групп похожих товаров на основе кластеризации;

б) формирование сегментов рынка на основе кластеризации возможных покупателей;

в) определение соответствия между группами товаров и сегментами рынка (на основе задачи о назначении);

г) определение конкретной стратегии маркетинга для каждой пары «группа товаров – сегмент рынка» (на основе задачи о блочном рюкзаке с учётом общего ограничения на реализацию маркетинговых стратегия для всех групп товаров). В качестве базового численного примера приводится формирование маркетинговой стратегии для фирмы, занимающейся компьютерными товарами (компьютеры, процессоры, платы, программы и др.). Расчёты проводились на основе программ автора в среде MATLAB (MatWorks, Inc., http://www.matworks.com).

Особо следует рассмотреть определенную «разумность» используемой составной схемы решения. В указанном выше списке известных подходов в основном каждый подход базируется на только одном методе или модели. В некоторых работах предлагаются гибридные процедуры, например, модели многокритериального принятия решений с учетом неопределенности [21], многокритериальный выбор с использованием имитационного моделирования и экспертной системы [22], экспертные процедуры на основе Веб-технологии [27], но эти походы направлены на «целостное» решение исходной задачи. В то же время часто представляется крайне важным осуществить декомпозицию задачи (это приводит с существенному уменьшению размерности решаемых задач), организовать параллельную и многоэтапную схему решения с использованием различных моделей. Предлагаемый в данной работе подход основывается на использовании комбинаторных областей небольшой размерности (например, множество альтернатив, множество промежуточных решений, множество критериев, множество уровней оценки альтернатив) и это является удобным для специалистов и экспертов. *Рис. 3* иллюстрирует общую тенденцию перехода от достаточно простых исходных приложений, когда можно использовать одну модель, один алгоритм (т.е., простую одно-шаговую схему решения) к более сложным ситуациям, когда для анализа и формализации сложных прикладных ситуаций приходится использовать ряд взаимосвязанных моделей и составную схему решения (т.е., составную схему решения, которая может включать взаимосвязанные алгоритмы и человеко-машинные процедуры). В целом представляется важным в будущем проведение исследований для сравнения различных методов (в первую очередь упомянутых выше) к решению задач формирования стратегий маркетинга на специально разработанных численных тестовых примерах. В частности, такой подход является базовым в последние два десятилетия в области решения сложных задач оптимизации (например: Webpage with optimization benchmarks http://plato.la.asu.edu/bench.html/, a Constrained and Unconstrained testing Environment CUTEr http://cuter.rl.ac.uk/cuter-www/).

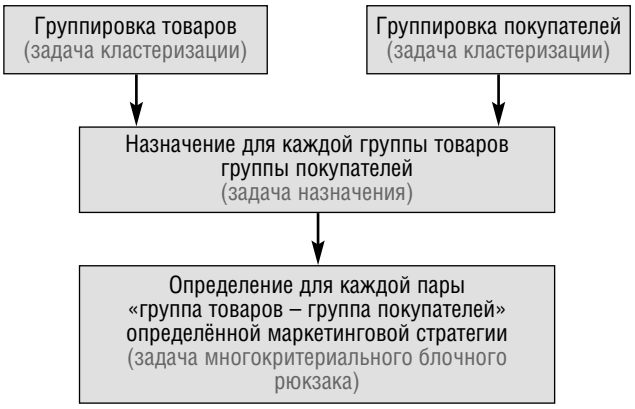


Рис. 2. Блок-схема составной стратегии

Дополнительно можно указать ряд преимуществ предлагаемой составной схемы решения, например:

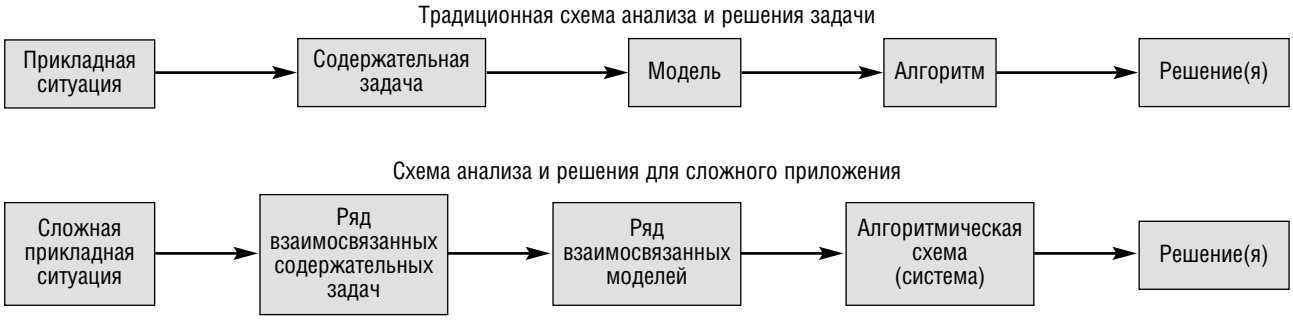


Рис. 3. Схема анализа и решения задач

1) многостадийная (последовательно-параллельная) схема решения, обеспечивающая возможность контроля/анализа промежуточных результатов на каждом этапе, включая возврат и повторное решений частей задачи;

2) независимое решение частей задачи на основе различных методов;

3) возможность использования когнитивных методов на каждом этапе/стадии процесса решения;

4) независимый анализ, оценивание, использование различных методов (различных вычислительных алгоритмов, различных экспертов) на каждом этапе/стадии решения задачи;

5) возможность интеграции аналитических, компьютерных и экспертных процедур.

Следует также отметить, что предлагаемый материал может быть успешно применен в рамках учебных процессов [33–35]. В общем случае можно выделить три базовых уровня грамотности в области информационных технологий [35, 36]:

1) базовая грамотность (умение читать, писать, считать) [37];

2) компьютерная грамотность (работа с компьютерами, написание программ, работа в Интернете и др.) [37];

3) подготовка и принятие решение, анализ и решение прикладных задач [33, 35].

Использование предлагаемой схемы направлено на 3-й уровень. Материал был частично или полностью реализован автором в ряде учебных курсов [33–35], в частности, в курсе по проектированию систем на факультете радиотехники и кибернетики МФТИ [35, 38].

Модели и составная схема решения

В данной работе использованы 4 комбинаторные задачи: многокритериальное ранжирование, кластеризация, задача о назначении и блочный рюкзак. Задача многокритериального ранжирования альтернатив является базовой задачей принятия решений и заключается в следующем. Имеется множество альтернатив $U = \{1,...,i,...,u\}$ и множество критериев для оценки альтернатив: $V = \{1,..., j,...,v\}$. Имеется матрица оценок альтернатив по критериям $\| z(i,j) \|$, где i соответствует номеру альтернативы и j соответствует номеру критерия. В задаче требуется получить частичный порядок на множестве альтернатив в виде «слоев» качества (т.е. непересекающиеся между собой и линейно-упорядоченных подмножеств исходного множества альтернатив):

$$U = \{U(1), ... ,U(k), ... ,U(m)\},$$
$$|U(k_1) \,\&\, U(k_2)| = 0 \text{ при } 1 \le k_1 < k_2 \le m.$$

При этом если $i_1 \in U(k_1)$ и $i_2 \in U(k_2)$ то i_1 предпочтительнее, чем i_2. Подмножество U(k) соответствует уровню качества k и каждый элемент этого множества $i \in U(k)$ имеет приоритет $r(i) = k$. Для решения указанной задачи используются различные методы, включая методы функций полезности [39], метод аналитических иерархий [40]. В работе для ранжирования альтернатив был использован вариант метода порогов несравнимости типа ELECTRE [41, 42].

Кластеризация (группировка) объектов является базовой научной задачей в многих областях [43–47]: разделить исходное множество элементов на группы (подмножества, кластеры) с целью минимизации «расстояния» (или близости) между элементами внутри кластеров. При этом используется следующая исходная информация:

а) параметры каждого элемента;

б) близость («расстояние») между элементами. Данная задача близка к предыдущей, но без упорядочения получаемых подмножеств. В данной работе использован один из наиболее известных алгоритмов кластеризации – иерархическая кластеризация (на основе варианта агломеративного алгоритма) [48].

Базовая простейшая задача о назначении имеет следующий вид [49]. Имеется множество элементов

n и множество позиций m. Задана матрица соответствия:

$$C = \| c_{i,j} \|, I = 1,..., n \text{ (элементы)},$$
$$j = 1,..., m \text{ (позиции)}.$$

Здесь $c_{i,j}$ представляет собой пользу от назначения элемента i на позицию j. Задача формулируется следующим образом (алгебраическая формулировка):

$$\max \Sigma_{i=1}^{n} \Sigma_{i=1}^{m} c_{ij} x_{ij}$$
$$\text{s.t. } \Sigma_{i=1}^{n} x_{ij} \le 1, \forall i = 1,..., n,$$
$$\Sigma_{j=1}^{n} x_{ij} \le 1, \forall j = 1,..., m,$$
$$x_{ij} \in \{0,1\}.$$

Приведённая задача является полиномиально разрешимой [49], но ее усложнения (например, квадратичная задача о назначении, обобщенная задача о назначении) являются NP-трудными [49–51]. В данной работе используется эвристический («жадный») алгоритм, т.е. последовательное выявление пары (i, j) с максимальным значением $c_{i,j}$.

В случае блочной задачи о рюкзаке (multiple choice problem) некое исходное множество элементов разбито на подмножества и требуется выбрать из каждого подмножества элемент (или несколько элементов) с учётом общих ресурсных ограничений и максимизации некой общей полезности [49, 52, 53]. В простейшей базовой постановке задача имеет вид:

$$\max \Sigma_{i=1}^{n} \Sigma_{j=1}^{qi} c_{ij} x_{ij},$$
$$\text{s.t. } \Sigma_{i=1}^{n} \Sigma_{j=1}^{qi} a_{ij} x_{ij} \le b,$$
$$\Sigma_{j=1}^{qi} x_{ij}, \forall i = 1,..., n,$$
$$x_{ij} \in \{0,1\}.$$

Эта задача является NP-трудной [49, 53]. При многокритериальной постановке вместо полезности каждого элемента c_{ij} используется вектор из r компонентов: $(c_{ij}^{l},..., c_{ij}^{p},..., c_{ij}^{r})$. Таким образом, проводится поиск эффективных по Парето решений на основе векторной целевой функции, и задача имеет вид [54, 55]:

$$\max \Sigma_{i=1}^{n} \Sigma_{j=1}^{qi} c_{ij}^{p} x_{ij}, \forall p = 1,..., r,$$
$$\text{s.t. } \Sigma_{i=1}^{n} \Sigma_{j=1}^{qi} a_{ij} x_{ij} \in b,$$
$$\Sigma_{j=1}^{qi} x_{ij}, \forall i = 1,..., n,$$
$$x_{ij} \in \{0,1\}$$

В данной работе используется двухэтапная процедура [54]:

1) многокритериальное ранжирование элементов;

2) последовательная упаковка элементов в рюкзак («жадная» эвристика).

Схема построения стратегии маркетинга

В данной работе рассматривается пример формирования составной маркетинговой стратегии для фирмы, занимающейся продажей и техническим обслуживанием компьютерного оборудования. Предполагается, что имеется два множества: множество товаров и множество потребителей. Кроме того, при установлении пары «товар-потребитель» рассматривается набор базовых стратегий маркетинга (например, захват рынка, сохранение рынка, создание нового рынка). Задача заключается в установлении пар «товар-потребитель» и выборе для каждой пару наилучшей базовой маркетинговой стратегии с учетом общего ресурсного ограничения. Таким образом, множество товаров соответствует множеству элементов, а множество потребителей соответствует множеству позиций. В работе используется 4-этапная схема решения (*рис. 4*):

Этап 1. Кластеризация множества товаров (элементов) для получения групп похожих товаров (уменьшение размерности рассматриваемой задачи, задача кластеризации).

Этап 2. Кластеризация множества потребителей (позиций) для получения групп похожих потребителей, т.е. сегментов рынка (уменьшение размерности рассматриваемой задачи, задача кластеризации).

Этап 3. Назначение для каждой группы товаров соответствующей группы похожих потребителей (сегмента рынка) (задача назначения).

Этап 4. Выбор для каждой пары «товар-потребитель» наилучшей стратегии маркетинга с учетом качества стратегий и общего для всей задачи ограничения по ресурсу (задача блочного рюкзака).

Пример построения стратегии маркетинга

В *табл. 1* приведены шесть типовых стратегии маркетинга [5]. Для оценивания указанных стратегий использованы следующие критерии (в скобках указан вес критерия):

а) уровень затрат на реализацию стратегии K_1 (+3);

б) уровень возможной прибыли в ближайший период времени K_2 (+5);

в) уровень возможной прибыли в перспективе K_3 (+3);

г) уровень возможной конкуренции K_4 (-2).

Оценивание альтернативных типовых стратегий проводилось на основе экспертного подхода с использованием порядковой шкалы общей [0,5]. Очевидно, что можно сгенерировать индивидуальные шкалы для каждого критерия. Кроме того, набор критериев, их веса и шкалы и оценки альтернатив по

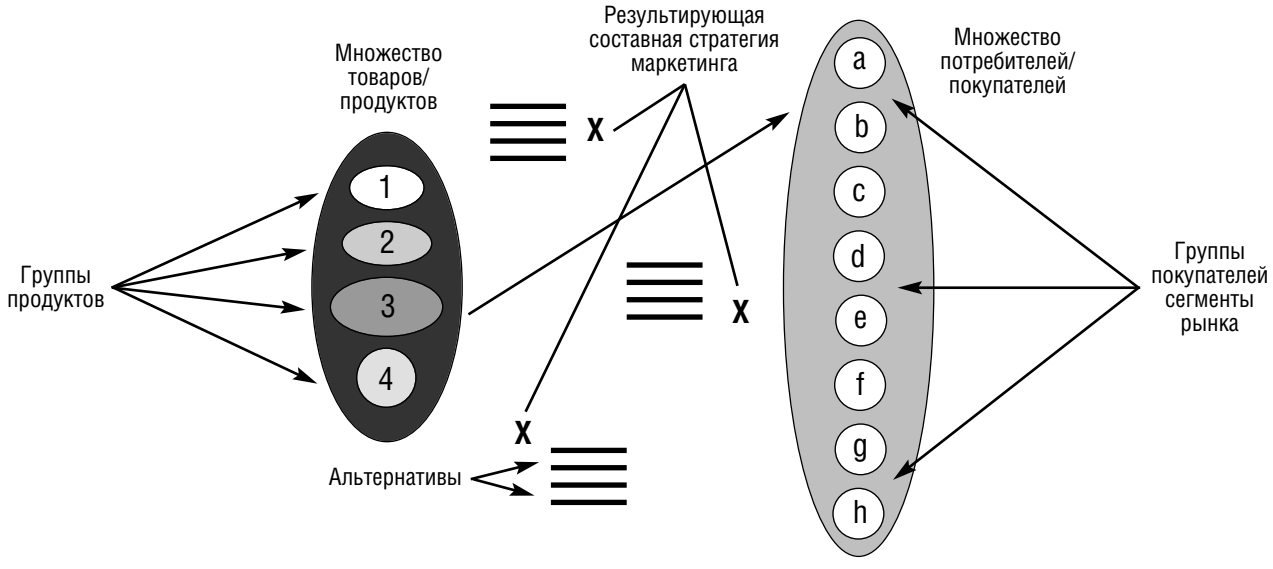


Рис. 4. Кластеризация, назначение, блочный рюкзак

критериям можно рассматривать в зависимости от конкретной пары «группа товаров – группа покупателей». В рассматриваемом примере использованы условные оценки альтернатив. Расчёт приоритетов основан на применении метода типа ELETRE [41].

Таблица 1

Базовые стратегии маркетинга и оценки

No	Стратегии	K_1	K_2	K_3	K_4	Приоритет
1	Расширение, развитие рынка (вложение в производство, рынок) M_1	4	2	4	4	2
2	Завоевание нового рынка(ов) M_2	5	2	4	5	3
3	Поддержание рынка (модификации товара, улучшение качества товара, улучшение деятельности M_3	2	4	2	3	2
4	Создание нового рынка (завоевание рынка новым товаром, создание заделов в науке M_4	4	4	5	1	1
5	Сохранение ситуации M_5	1	2	1	3	2
6	Уход с рынка M_6	1	0	0	0	2

В качестве примера товаров рассматривается компьютерная продукция типа электронного оборудования и программ (*табл. 2*). Для оценивания товаров использованы следующие параметры:

а) стоимость X_1 (+3, шкала [1,5]);

б) профессиональный уровень пользователей (профессионалы – 3, квалифицированные пользователи 2, неквалифицированные пользователи – 1) X_2 (+5, шкала [1,3]);

в) возможность организации технического обслуживания/профессиональных консультаций X_3 (+3, шкала [1,3]);

г) возможность проведения работ по усовершенствованию (upgrade) X_4 (2, шкала [0,1]).

Оценивание товаров проводилось основе экспертного подхода. Использована версия иерархического алгоритма кластеризации [48].

Таблица 2

Список товаров и их характеристик

No	Товар	X_1	X_2	X_3	X_4	Номер группы товаров
1	Персональный компьютер T_1	4	1	2	1	1
2	Сервер T_2	5	3	3	1	2
3	Ноутбук T_3	4	1	1	1	1
4	Платы T_4	2	2	0	0	3
5	Мониторы T_5	3	1	1	0	1
6	Клавиатура T_6	1	1	0	0	3
7	Коммуникационное оборудование T_7	3	2	3	1	4
8	Операционные системы T_8	2	2	3	1	4
9	Прикладные программы (редактирование и др.) T_9	3	2	1	1	5
10	Программы для бухучета T_{10}	3	2	2	1	5
11	Программы для инженерных расчетов T_{11}	5	3	2	1	2
12	Программы для математических расчетов T_{12}	3	3	1	1	5
13	Игры T_{13}	2	2	0	0	3

В качестве примера множества возможных покупателей рассматриваются типовые потребители компьютерной продукции (*табл. 3*). Для оценивания покупателей использованы следующие параметры:

а) объём возможных покупок (т.е. объём рынка) Y_1 (+3, шкала [1,4]);

б) профессиональный уровень пользователей (профессионалы – 3, квалифицированные пользователи – 2, неквалифицированные пользователи – 1) Y_2 (+5, шкала [1,3]);

в) объём требуемых консультаций и возможность организации технического обслуживания Y_3 (+3, шкала [0,3]).

Оценивание товаров проводилось на основе экспертного подхода с использованием порядковой или номинальной общей шкалы [0,1,2,3,4]. Использована версия иерархического алгоритма кластеризации [48].

После кластеризации множества исходных товаров получаются следующие группы (*табл. 2*): группа G_1: T_1, T_3, T_5; группа G_2: T_2, T_{11}; ; группа G_3: T_4, T_6, T_{13}; группа G_4: T_7, T_8; G_5: T_9, T_{10}, T_{12}. После кластеризации множества потребителей получаются следующие группы (*табл. 3*): группа R_1: P_1, P_2; группа R_2: P_3, P_4, P_7, P_{12}; ; группа R_3: P_5, P_6, P_{14}, P_{15}; группа R_4: P_8, P_{11}, P_{16}; группа R_5: P_9, P_{17}, P_{18}; группа R_6: P_{10}, P_{13}. Матрица соответствия групп товаров и групп потребителей, полученная на основе экспертного оценивания, представлена в *табл. 4.*

Таким образом получаем задачу о назначении (размерность 5*6). Задача формулируется следующим образом (алгебраическая формулировка; здесь предполагается, что для каждой группы товаров может рассматриваться до трёх сегментов рынка и для каждого сегмента рынка – до трёх групп товаров):

$$\max \Sigma_{i=1}^{n} \Sigma_{i=1}^{m} c_{ij} x_{ij},$$
$$\text{s.t. } \Sigma_{i=1}^{n} x_{ij} \le 3, \forall i = 1,..., n,$$
$$\Sigma_{j=1}^{n} x_{ij} \le 3, \forall j = 1,..., m,$$
$$x_{ij} \in \{0,1\}.$$

Результат решения этой задачи (т.е. соответствие групп товаров и групп потребителей) представлен в *табл. 5.*

Теперь для каждой из 15 полученных пар типа (G_i,R_j) (*табл. 5*) выбирается подмножество базовых типовых маркетинговых стратегий (*табл. 1*, экспертное оценивание):

$$A_1 = (G_1,R_1): M^1 = \{M_3, M_5, M_6\};$$
$$A_2 = (G_1,R_2): M^2 = \{M_1, M_2, M_4, M_5\};$$

Таблица 3

Список покупателей/потребителей, их характеристик

No	Типовой потребитель	Y_1	Y_2	Y_3	Номер группы потребителей
1	Дети P_1	2	1	0	1
2	Школьники-подростки P_2	2	2	0	1
3	Домашние хозяйки P_3	1	1	0	2
4	Студенты-экономисты P_4	1	1	1	2
5	Студенты-инженеры P_5	3	2	1	3
6	Студенты в области естественных наук (математика, физика, химия и т.п.) P_6	2	2	1	3
7	Студенты в области социальных наук (социология, история, психология и др.) P_7	1	1	0	2
8	Специалисты в области техники (инженеры) P_8	4	2	2	4
9	Специалисты в области ИТ (программирование, электроника, коммуникации) P_9	4	3	3	5
10	Специалисты в области социальных наук P_{10}	2	1	1	6
11	Специалисты в области экономики (например, бухучет) P_{11}	3	2	2	4
12	Малые фирмы P_{12}	1	1	1	2
13	Небольшие торговые фирмы (кафе, магазины) P_{13}	2	1	1	6
14	Средние фирмы P_{14}	3	2	1	3
15	Склады P_{15}	3	2	1	3
16	Большие магазины, торговые сети P_{16}	3	2	2	4
17	Крупные фирмы P_{17}	4	3	3	5
18	Банки P_{18}	4	3	3	5

Таблица 4

Матрица соответствия

	G_1	G_2	G_3	G_4	G_5
R_1	5	0	4	0	0
R_2	5	0	1	0	5
R_3	4	4	2	5	5
R_4	5	5	1	5	5
R_5	4	5	1	5	3
R_6	5	0	1	0	4

Таблица 5

Результирующее назначение

	G_1	G_2	G_3	G_4	G_5
R_1	1	0	1	0	0
R_2	1	0	1	0	1
R_3	0	1	0	1	1
R_4	1	1	0	1	0
R_5	0	1	1	1	0
R_6	0	0	0	0	1

$A_3 = (G_1, R_4)$: $M^3 = \{M_3, M_5, M_6\}$;
$A_4 = (G_2, R_3)$: $M^4 = \{M_1, M_2, M_4\}$;
$A_5 = (G_2, R_4 : M^5 = \{M_3, M_5, M_6\}$;
$A_6 = (G_2, R_5)$: $M^6 = \{M_2, M_3, M_5\}$;
$A_7 = (G_3, R_1)$: $M^7 = \{M_5, M_6\}$;
$A_8 = (G_3, R_2)$: $M^8 = \{M_5, M_6\}$;
$A_9 = (G_3, R_5)$: $M^9 = \{M_5, M_6\}$;
$A_{10} = (G_4, R_3)$: $M^{10} = \{M_1, M_2, M_4\}$;
$A_{11} = (G_4, R_4)$: $M^{11} = \{M_5, M_6\}$;
$A_{12} = (G_4, R_5)$: $M^{12} = \{M_1, M_2, M_4\}$;
$A_{13} = (G_5, R_2)$: $M^{13} = \{M_5, M_6\}$;
$A_{14} = (G_5, R_3)$: $M^{14} = \{M_1, M_3, M_5\}$;
$A_{15} = (G_5, R_6)$: $M^{15} = \{M_5, M_6\}$.

Теперь можно получить составную маркетинговую стратегию на основе задачи блочного рюкзака (приоритеты из таблицы 1 используются как полезности, оценки по критерию K_1 – как оценки затрат). Результаты приведены в *табл. 6* для 4-х вариантов ограничения: (а) ≤ 8, (б) ≤ 10, (в) ≤ 20, (г) ≤ 30.

Следует отметить, что в соответствии с задачей блочного рюкзака при недостаточном ресурсе в некоторых группах (т.е. для некоторых пар «группа товаров – сегмент рынка») маркетинговая стратегия отсутствует, т.е. маркетинговая деятельность отсутствует.

Таблица 6

Результирующие стратегии

	≤ 8	≤ 10	≤ 20	≤ 30
M^1	M_5	M_5	M_5	M_5
M^2	M_5	M_5	M_5	M_5
M^3	M_5	M_5	M_5	M_5
M^4	M_1	M_1	M_1	M_1
M^5	M_5	M_3	M_3	M_3
M^6	–	M_5	M_2	M_2
M^7	–	–	M_5	M_5
M^8	–	–	M_5	M_5
M^9	–	–	M_5	M_5
M^{10}	–	–	–	M1
M^{11}	–	–	M_5	M5
M^{12}	–	–	M_1	M_1
M^{13}	–	–	–	M_5
M^{14}	–	–	–	M_1
M^{15}	–	–	–	M_5

Использование в обучении

Описанный подход на основе составной стратегии использовался в курсе автора о проектировании систем в МФТИ (2004–2008) [35, 38] и соответствует лабораторной работе 10. При этом рассмотренные базовые задачи соответствуют другим лабораторным работам: задача многокритериального ранжирования – лабораторной работе 2, задача кластеризации – лабораторной работе 5, многокритериальная задача блочного рюкзака – лабораторной работе 6, задача о назначении – лабораторной работе 9. В качестве базового задания студенты строили стратегию маркетинга на основе численного примера. Кроме того каждый студент мог подготовить расширенный вариант работы 10 (как небольшой исследовательский проект):

а) исследовать группу товаров (включая сложные технические системы, например, самолеты);

б) рассмотреть применение подхода для других прикладных областей (например: планирование технического обслуживания, планирование тестирования систем, планирование в социально-экономических системах).

Примером такого прикладного проекта является публикация [56].

Заключение

В данной статье предложен подход к построению комбинированной стратегии маркетинга на основе четырёх моделей (многокритериальное ранжирование, кластеризация, блочный рюкзак, задача о назначении). Рассмотренный материал может быть использован в преподавании. Следует также

отметить, в статье представлена базовая составная схема решения, т.е. прототип. Очевидно, что возможно улучшение схемы по нескольким направлениям:

1. Использование более сложных комбинаторных моделей (например, вместо задачи о назначении можно рассматривать её многокритериальное обобщение и/или задачу о квадратичном назначении/обобщенную задачу о назначении [50, 57]; вместо задачи о назначении и/или задачи о блочном рюкзаке можно использовать задачи комбинаторного синтеза на основе морфологической клики [57, 58].

2. Задание «индивидуальных» ограничений в задаче о назначениях для каждого сегмента рынка и/или для каждой группы товаров.

3. Использование других алгоритмов решения возникающих комбинаторных задач.

4. Учёт неопределенности (т.е. вероятностные оценки и/или оценки на базе размытых множеств).

5. Рассмотрение динамических задач, например, построение многоэтапных маркетинговых стратегий (типа траекторий маркетинговых стратегий).

6. Использование древовидных сценариев как основу будущей маркетинговой деятельности.

7. Применение специальных исследований для получения оценок типовых маркетинговых стратегий, товаров, потребителей (вместо простых процедур экспертного оценивания).

8. Оценивание типовых маркетинговых стратегий с учётом конкретного сегмента рынка и/или конкретной группы товаров.

В заключение следует указать, что предложенная комбинированная схема представляет собой только один из первых шагов в направлении создания таких составных методов решения в области маркетинга. Очевидно, что стоит проблема проектирования комбинированных процедур (и инструментальных сред), включающих различные методы и подходы к решению задачи формирования стратегий маркетинга. Кроме того, представляется крайне важной организация специальных исследований в области оценивания качества и «стоимости» процессов решения задач (включая сравнение) в области бизнес-информатики и определения областей/ситуаций наилучшего применения методов, в том числе и для задач формирования стратегий маркетинга. ■

Литература

1. Changchien S.W., Lin M.-C. Design and implementation of a case-based reasoning system for marketing plans // Expert Systems with Applications, v. 28, n. 1, 2005, P. 43-53.

2. Dickson P.R., Ginter J.L. Market segmentation, product differentiation, and marketing strategy // J. of Marketing, v. 51, April 1987, P. 1-10.

3. Hauser J.R., Shugan S.M. Defensive Marketing Strategies // Marketing Science, v. 2, n. 4, Fall 1993, P. 319-360.

4. Jain S.C. Standardization of international marketing strategy: some research hypotheses // J. of Marketing, v. 53, n. 1, 1989, P. 70-79.

5. Kotler P., Keller K.L. Marketing Management, 13th ed., Englewood Cliffs, NJ: Prentice Hall, 2008.

6. Nash L. Direct Marketing: Strategy, Planning, Execution. NJ: McGraw-Hill, 2000.

7. Rust R.T., Lemon K.N., Zeithami V.A. Return on marketing: using customer equity to focus marketing strategy // J. of Marketing, v. 68, Jan. 2004, P. 109-127.

8. Smith W.R. Product differentiation and marketing segmentation as alternative marketing strategies // J. of Marketing, v. 4, n. 3, 1956, P. 63-65.

9. Uslay C., Morgan R.E., Sheth J.N. Peter Drucker on marketing: An exploration of five tenets // J. of the Academy of Marketing Science, v. 37, n. 1, 2009, P. 47-60.

10. Wang C.-H. Outlier identification and market segmentation using kernel-based clustering techniques // Expert Systems with Applications, v. 36, n. 2 (Part 2), 2009, P. 3744-3750.

11. Zeithami V., Bitner M.J., Gremler D. Services Marketing, 5th ed., NJ: McGraw-Hill/Irwin, 2008.

12. Drew J.H., Mani D.R., Betz A.L., Datta P. Targeting customers wit st/atistical and data mining techniques // J. of Service Research, v. 3, n. 3, 2001, P. 205-219.

13. Ratner B. Statistical Modeling and Analysis for Database Marketing. NY: Chapman & Hall/CRC, 2003.

14. Karatzas I., Shubik M., Sudderth W.D. Construction of stationary Markov equilibria in a strategic market game // Mathematics of Operations Research, v. 19, n. 4, 1994, P. 972-1006.

15. Sheu J.-J., Su Y.-H., Chu K.-T. Segmenting online game customers – The perspective of experimental marketing // Expert Systems with Applications, v. 36, n. 4, 2009, P. 8487-8495.

16. Pfeifer P., Carraway R. Modeling customer relationships as Markov chains // J. of Interactive Marketing, v. 14, n. 2, 2000, P. 43-55.
17. Tirenni G., Labbi A., Eliseeff A., Berrospi C. Efficient allocation of marketing resources using dynamic programming. Proc. of 2005 SIAM Intl. Data Mining Conf, Newport Beach, CA, 2005, P. 581-585.
18. Wicek P. Pure equilibria in a simple dynamic model of strategic market game // Mathematical Methods of Operations Research, v. 69, n. 1, 2009, P. 59-79.
19. Zufryden F.S. A dynamic programming approach for product selection and supermarket shelf-space allocation // J. of the Operational Research Society, v. 37, n. 4, 1986, P. 413-422.
20. Berry M.J.A., Linoff G.S. Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. 2nd ed., NY: J.Wiley & Sons, 2004.
21. Denguir-Rekik A., Montmain J., Mauris G. A possibilistic-valued multi-criteria decision-making support for marketing activities in e-commerce: feedback based diagnosis system // European J. of Operational Research, v. 195, n. 3, 2009, P. 876-888.
22. Li S., Li J.Z. Hybridising human judgment, AHP, simulation and a fuzzy expert system for strategy formulation under uncertainty // Expert Systems with Applications, v. 36, n. 3 (part 1), April 2009, P. 5557-5564.
23. Fleder D.M., Hosanagar K. Recommender systems and their impact on sales diversity // Proc. of the 8th ACM Conf. on Electronic Commerce, San Diego, CA, 2007, P. 192-199.
24. Abrahams A.S., Becker A.B., Sabido D., D'Souza R., Makriyiansis G., Krasnodebski M. Inducing a marketing strategy for a new pet insurance company using decision trees // Expert Systems with Applications, v. 36, n. 2 (Part 1), 2009, P. 1914-1921.
25. Chiu C. A case-based customer classification approach for direct marketing // Expert Systems with Applications, v. 22, n. 2, 2002, p. 163-168.
26. Cook R.L., Jenicke L.O. Expert systems technology applied to marketing decision making: An interdisciplinary term project // J. of Marketing Education, v. 11, n. 1, 1989, P. 45-52.
27. Li S. A Web-enabled hybrid approach to strategic marketing planning: Group Delphi+a Web-based expert system // Expert Systems with Applications, v. 29, n. 2, 2005, P. 393-400.
28. Mentzer J.T., Gandhi N. Expert systems in marketing: Guidelines for development // J. of the Academy of Marketing Science, v. 20, n. 1, 1992, P. 73-80.
29. Zahavi J., Levin N. Issues and problems in applying neural computing in target marketing // J. of Direct Marketing, v. 9, n. 3, 1995, P. 33-45.
30. Zahavi J., Levin N. Applying neural computing in target marketing // J. of Direct Marketing, v. 11, n. 4, 1997, P. 76-93.
31. Biatiatto S., Farinella G.M., Giuffrida G., Sismeiro C., Tribulato G. Using visual and text features for direct marketing on multimedia messaging services domain // Multimedia Tools and Applications, v. 42, n. 1, 2009, P. 5-30.
32. Biattiato S., Farinella G.M., Giuffrida G., Sismeiro C., Tribulato G. Exploiting visual and text features for direct marketing learning in time and space constrained domain // Pattern Analysis and Applications, 2009, P. 1-15 (in press).
33. Левин М.Ш. Типовые задачи принятия решений при подготовке бизнес планов // НТИ, сер. 1, N 10, 1995, С. 7-13.
34. Левин М.Ш. О преподавании информационных технологий // НТИ, сер. 1, N. 5, 1996, С. 14-23.
35. Левин М.Ш. О преподавании проектирования систем // Информационные технологии и вычислительные системы, N. 2, 2007, С. 89-94.
36. Левин М.Ш. О третьей грамотности // НТИ, сер. 2, N. 6, 1995, С. 20-30.
37. Ershov A.P. Aesthetics and human factor in programming // Communications of the ACM, v. 15, n. 7, 1972, P. 501-505.
38. Levin M.Sh. Student research projects in system design // Int. Conf. on Computer Supported Education CSEDU-2009, Lisbon, Vol. 2, 2009, P. 67-72.
39. Keeny R.L., Raiffa H. Decisions with Multiple Objectives: Preferences and Value Tradeoffs, J.Wiley & Sons, New York, 1976.
40. Saaty T.L. The Analytic Hierarchy Process. MacGraw-Hill, New York, 1988.
41. Левин М.Ш., Михайлов А.А. Фрагменты технологии стратификации множества объектов. Препринт. М.: ИСА РАН, 1988. 80 с.
42. Roy B. Multicriteria Methodology for Decision Aiding. Dordrecht: Kluwer, 1996.
43. Anderberg M.R. Cluster Analysis for Applications. NY: Academic Press, 1973.
44. Dorndorf U., Pesch E. Fast clustering algorithms // ORSA J. on Computing, v. 6, n. 2, 1994, P. 141-153.
45. Hartigan J.A. Clustering Algorithms. NY: J.Wiley & Sons, 1975.
46. Jain A.K., Murty M.N., Flynn P.J. Data clustering: a review // ACM Computing Surveys, v. 31, n. 3, 1999, P. 264-323.
47. Mirkin B.G. Clustering for Data Mining: A Data Recovery Approach. NJ: Chapman & Hall / CRC, 2005.
48. Levin M.Sh. Towards hierarchical clustering // Computer Science in Russia: Proceedings of 2nd Int. Conf. CSR 2007, LNCS 4649, Springer, 2007, P. 205-215.

49. Garey M.R., Johnson D.S. Computers and Intractability: A Guide to the Theory of NP-Completeness. NY: W.H. Freeman Company, 1979.
50. Cela E. The Quadratic Assignment Problem, Dordrecht: Kluwer, 1998.
51. Pardalos P.M., Wolkowicz H., Eds. Quadratic Assignment and Related Problems, Providence: AMS, 1994.
52. Kellerer H., Pferschy U., Pisinger D. Knapsack Problems, Berlin: Springer, 2004.
53. Martello S., Toth P. Knapsack Problem: Algorithms and Computer Implementation, NY: J.Wiley & Sons, 1990.
54. Левин М.Ш., Сафонов А.В. Проектирование и перепроектирование конфигурации оборудования в коммуникационной сети // Информационные технологии и вычислительные системы, N. 4, 2006, С. 63-73.
55. Poladian V., Sousa J.P., Garlan D., Schmerl B., Shaw M. Task-based adaptation for ubiquitous computing // IEEE Transactions on Systems, Man, and Cybernetics – Part C, v. 36, n. 3, 2006, P. 328-340.
56. Levin M.Ш., Merzlyakov A.O. Composite combinatorial scheme of test planning (example for microprocessor systems), IEEE Region 8 Int. Conf. Sibircon-2008, Novosibirsk, 2008, P. 291-295.
57. Levin M.Sh. Combinatorial optimization in system configuration design // Automation and Remote Control, v. 70, n. 3, 2009, P. 519-561
58. Levin M.Sh. Composite Systems Decisions. NY: Springer, 2006.

Прогрессивные программы риск-менеджмента – это ключ к успешному развитию Вашего бизнеса!

«Международный Институт Исследования Риска» обеспечивает обучение и консультирование в управлении рисками в течение более чем 9 лет.

Чтобы достичь устойчивости в развитии компании, сотрудники должны обладать современными знаниями в рамках разработанных нами обучающих программ.

Преимущества наших учебных программ:

- **комплексный подход;**
- **практическая направленность;**
- **актуальность и оперативность;**
- **профессионализм;**
- **гибкая система скидок.**

Программы предназначены для: руководителей, менеджеров высшего и среднего звена предприятий, специалистов по стратегическому планированию и управлению, а также для тех, у кого есть желание повысить свой уровень знаний и навыков в области управления рисками организаций.

Обучение в «МИИР» проводится стабильным профессорско-преподавательским составом известных государственных ВУЗов и квалифицированными преподавателями-практиками (руководителями крупных организаций) с использованием авторских методических разработок.

Контакты: 117418, г. Москва, Новочеремушкинская ул., д. 42а.
Телефон: (495) 128-91-77, 128-91-67
e-mail: marfinuk@miir.ru, www.miir.ru

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ В НАУКЕ И БИЗНЕСЕ: ПОДХОДЫ, ИНСТРУМЕНТЫ, ПРИМЕНЕНИЕ

В.Н. Сидоренко,
кандидат физико-математических наук, кандидат экономических наук, кандидат юридических наук, доцент кафедры «Экономика природопользования» экономического факультета МГУ, доцент кафедры прикладной математики Московского городского университета управления Правительства Москвы,
Адрес: 119991, Москва, Ленинские горы, 2-ой гуманитарный корпус, экономический факультет, комн. 592, Сидоренко В.Н. Тел: 8-916-650-9304. E-mail: v_sidorenko@mail.ru
А.В. Красносельский,
аспирант кафедры «Экономика природопользования» экономического факультета МГУ, a-krasnoselskiy@yandex.ru

Статья является обзором существующих подходов и программных средств в имитационном моделировании. Приводятся три общепринятые парадигмы системной динамики, дискретно-событийного и агентного моделирования. Более подробно рассматривается агентный подход, сравнительно редко используемый в России, но являющийся основой для создания эффективных систем поддержки принятия решений в бизнесе. Статья предназначена для первоначального знакомства с имитационным моделированием и описания того круга бизнес задач, где его применение может оказаться эффективным.

Ключевые слова: Имитационное моделирование, дискретно-событийное и агентное моделирование, агентный подход.

Неотъемлемой чертой любого бизнес-решения является риск, который берет на себя ответственное лицо. Если требуется принять решение, выходящее за стандартные процедуры, и которое потенциально несет в себе риски критичные для компании, возникает естественное желание узнать последствия, просчитав тот или иной сценарий развития событий. Если обычного Excel для этого не хватает, и необходимо в динамике проиграть изменения рыночной ситуации, действия конкурентов, смоделировать и оптимизировать бизнес-процессы, то можно обратиться к имитационному компьютерному моделированию с помощью специализированных программных средств.

Имитационное моделирование

Имитационное моделирование представляет собой процесс построения обобщенной компьютерной модели системы с алгоритмическим описанием основных правил ее поведения и процессов. Имитационная модель, как правило, создается для ответа на вопросы «что, если...», т.е. для исследования возможных сценариев развития системы при вариации определенных параметров. После создания модели с ней проводят многочисленные компьютерные симуляции – имитационные эксперименты. В ходе них идет итерационный процесс уточнения или отбрасывания гипотез, использовавшихся для описания системы. Такой подход позволяет

получить аналог эксперимента в экономике, социологии, экологии, при решении задач оптимизации и планирования в бизнесе.

Имитационную модель можно рассматривать как множество правил, определяющих процессы функционирования некоторой системы и ее переходов из одного временного состояния в следующее. Эти правила могут определяться любым доступным для компьютера способом – в виде блок-схем, дифференциальных уравнений, диаграмм состояний, автоматов, сетей. Имитационные модели, как правило, менее формализованы, чем аналитические, система описывается «как она есть», в терминах максимально приближенных к реальным.

В имитационном моделировании к настоящему моменту сложились три самостоятельные парадигмы – системная динамика, дискретно-событийное и агентное моделирование. Они соответствуют разным уровням абстракции при создании модели, что обуславливает применение того или иного подхода. Принято различать три уровня абстракции: высокий (стратегический), средний (тактический) и низкий (оперативный) (1). При низком уровне моделируется поведение отдельных объектов, но, в отличие от физического моделирования, используются не точные траектории и времена, а их усредненные или стохастические значения. На этом уровне принято решать задачи, связанные с пешеходным движением, транспортом, компьютерными системами. На среднем уровне абстракции обычно оперируют с расписаниями, задержками, мощностями и емкостями, физическое перемещение при этом не анализируется. Здесь абстрагируются от индивидуальных свойств объектов моделирования (людей, машин, товаров) и в основном рассматривают их потоки. Характерными задачами этого уровня являются системы массового обслуживания, модели бизнес-процессов, логистика. При высоком уровне абстракции в модели, как правило, отсутствуют индивидуальные объекты сами по себе, а оперируют лишь с их количеством и агрегированными показателями. На данном уровне моделируется проблемы рыночного равновесия, социально-экономического развития городов, экологические процессы.

Дискретно-событийное моделирование

Подходом, соответствующим низкому и среднему уровню абстракции, является дискретно-событийное (далее ДС) моделирование. Его концепцию предложил в 60-х годах прошлого века Джефри Гордон, разработав популярное и сегодня программное средство GPSS. В работе (2) он предложил использовать концепции заявок (entities), ресурсов и потоковых диаграмм (flowcharts). На рис.1 изображена типичная потоковая диаграмма, согласно которой моделируется работа call-центра. Заявки, в данном случае звонки, представляют собой некие пассивные объекты, которые перемещаются, захватывают и освобождают ресурсы согласно потоковым диаграммам – схемам, описывающим изучаемый процесс. Заявки могут представлять собой людей, товары, детали, документы, сообщения. ДС моделирование является дискретным – каждому событию соответствует определенный дискретный момент времени. Характерной чертой данного подхода является «обезличенность» заявки, от ее индивидуальных свойств абстрагируются. Считается, что все заявки обладают универсальной логикой поведения и обрабатываются по единому, заранее известному алгоритму. Ядро модели отвечает за генерацию, обработку и уничтожение заявок.

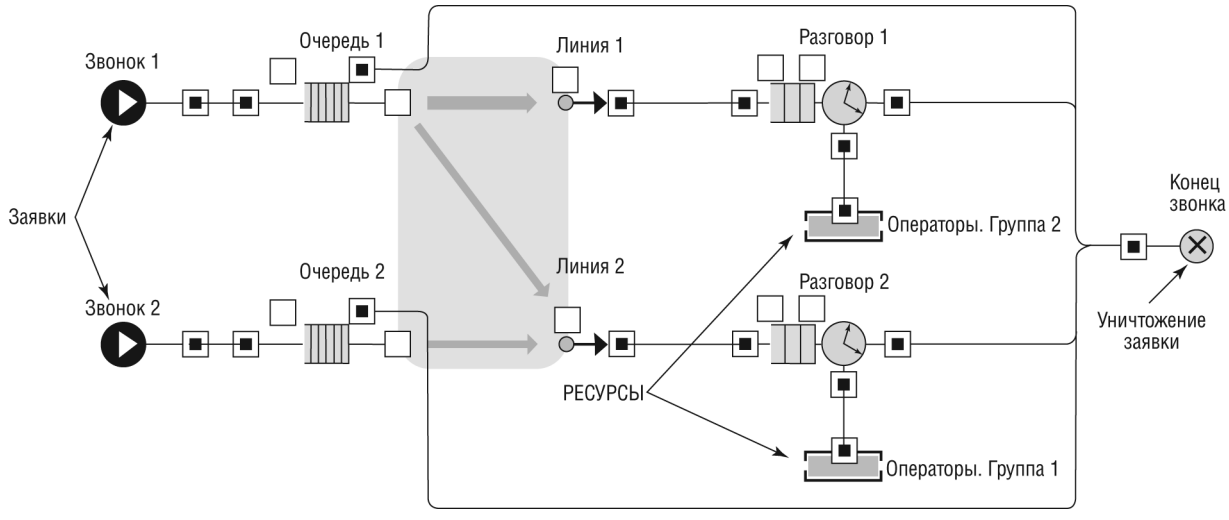


Рис. 1 Пример потоковой диаграммы «обработка звонков в call-центре»

Инструменты дискретно-событийного моделирования. Существует достаточно много программных продуктов, поддерживающих ДС подход. Среди всех средств, предназначенных для имитационного моделирования, этот класс наиболее представительный. Например, в отчёте (3) приводятся характеристики 65 различных инструментов. Это объясняется большой популярностью дискретно-событийного моделирования в таких областях как системы массового обслуживания, бизнес-процессы, производство, логистика, медицина, транспорт и других. Часть программных продуктов являются достаточно универсальными (Arena, Extend, GPSS, Witness), другие заточены под более узкий класс задач и содержат абстрактные элементы, взятые из предметной области исследования (AutoMod – транспорт, логистика, производство; MedModel – моделирование и оптимизация систем в здравоохранении; Comnet – телекоммуникации).

Стандартными характеристиками большинства продуктов являются: возможность создания модели в графическом режиме (потоковые диаграммы рисуются, используются стандартные шаблоны для отдельных блоков, параметры элементов системы задаются через подменю); интерфейсы для соединения с базами данных, средства для обработки статистики на входе и выходе модели, для ее оптимизации, для создания анимации в ходе имитационных экспериментов.

Системная динамика

Высокому уровню абстракции в имитационном моделировании соответствует подход системной динамики (далее СД), предложенный в 50-х годах прошлого века американским ученым Джемом Форрестером. При данном подходе не рассматриваются индивидуальные объекты, а лишь их количества и агрегированные показатели. Системная динамика применяется тогда, когда нет необходимости или возможности исследовать влияние отдельных объектов, а достаточно изучить поведение системы на уровне агрегированных величин. Форрестер предложил использовать для этого понятия «накопители», stocks, и «потоки» между ними, flows (4). Накопители могут относиться к различным материальным объектам, например, к людям в демографических моделях, товарам, деньгам при моделировании баланса на рынке, природным ресурсам в задачах экологии. На *рис. 2* приведена системно-динамическая диаграмма из модели распространения продукта на рынке. В ней три объекта-накопителя: «потенциальные пользователи», люди, формирующие спрос и люди, являющиеся реальными пользователями. Такие факторы как реклама, частота контактов определяют потоки, в соответствие с которыми накопители «перетекают» друг в друга. С формальной точки зрения системно-динамическая модель представляет собой систему дифференциальных (в частном случае алгебраических) уравнений, определяющую потоки между накопителями. Отличительной чертой СД является неразличимость объектов, находящихся в одном накопителе. Их невозможно индивидуализировать, приписав различные свойства, логику поведения или процесс обработки. Все взаимосвязи задаются на уровне накопителей, то есть между агрегированными величинами.

Системно-динамические модели обычно применяются при стратегическом анализе и долгосрочном планировании. Интересные примеры их использования можно найти как монографиях по системной динамике, например в (5) и (6), так и в многочисленных статьях по конкретным системно-динамическим моделям, созданным за более чем полвека развития данной парадигмы.

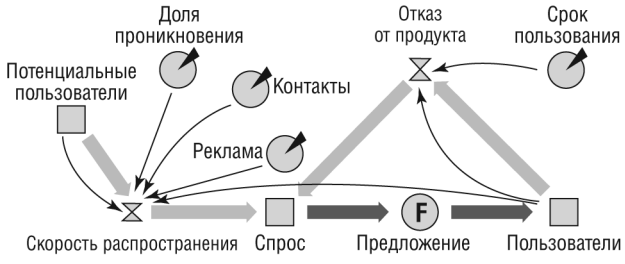


Рис. 2. Пример системно-динамической диаграммы «Проникновение продукта на рынке»

Инструменты системной динамики. Число программных продуктов, предназначенных для использования системно-динамического подхода, относительно невелико по сравнению с ДС моделированием. Наиболее распространены инструменты Vensim компании High Performance Systems (www.hps-inc.com), Powersim компании Powersim, SA (www.powersim.com), IThink компании Ventana Systems (www.vensim.com). Vensim является из них наиболее простым и недорогим, обладая при этом такими характеристиками как поддержка графического создания потоковых диаграмм, встроенные логические операторы и генераторы случайных чисел, средства для связи с базами данных, для генерации отчетов, анимации и анализа чувствительности.

Важной характеристикой инструментов Powersim и IThink являются встроенные алгоритмы частичной поддержки дискретного моделирования. Это позволяет создавать модели с разными уровнями абстракции для отдельных блоков. Более подробную информацию о характеристиках данных инструментов можно найти на сайтах компаний-разработчиков и в руководстве по Powersim (7).

Агентное моделирование

Третьей парадигмой в имитационном моделировании является агентное моделирование (далее АМ). Агент представляет собой индивидуализированный активный объект, который может обозначать человека, транспортное устройство, компанию, населенный пункт. В зависимости от того, какой объект представляет собой агент, модель может соответствовать высокому уровню абстракции (агент – компания, страна), среднему (агент – транспортная единица), низкому (агент – отдельный человек) или сочетать несколько уровней. Таким образом, данная парадигма является наиболее универсальной. Основным отличием агентного подхода от первых двух является построение модели по принципу снизу-вверх. Зависимости между агрегированными величинами не задаются исходя из знаний о реальном мире, а получаются в процессе моделирования индивидуального поведения десятков, сотен или тысяч агентов, их взаимодействия друг с другом и с объектами, моделирующими окружающую среду. Например, исследование рынка будет происходить не в понятиях совокупного спроса и предложений, а в модели будут заложены возможные реакции отдельного человека на изменение цены, его потребительские характеристики. У агентов появляется возможность «общаться» между собой, обмениваться информацией, предпочтениями, влияя, тем самым, на поведение друг друга. Модель может учитывать пространственные характеристики, взаиморасположение агентов по отношению друг к другу и объектам окружающей среды.

К преимуществам агентного подхода следует отнести: отсутствие предопределенности в поведении системы на глобальном уровне, что может привести к появлению новых гипотез о ее функционировании в ходе симуляции модели; реализм и гибкость в описании системы, возможность моделировать самые сложные нелинейные обратные связи, использовать любой необходимый уровень детализации и абстракции. В АМ отсутствуют ограничения на гетерогенность элементов модели; появляется возможность моделирования общения и обмена информацией. К потенциальным барьерам для построения агентной модели следует отнести, во-первых, наличие адекватных данных. Как правило, собрать статистику по характеристикам индивидуальных объектов сложнее, чем по агрегированным показателям. Во-вторых, придется определить логику поведения отдельного агента в терминах, доступных для обработки компьютером. Если это сложный объект, например человек, то приходится моделировать такие иррациональные вещи, как психологию поведения, выбора, привычки. С последними результатами в этой области, используемыми в агентных моделях, можно ознакомиться в статье (8). В процессе имитационных экспериментов могут возникнуть вычислительные сложности, поскольку агентные модели в среднем требуют больших аппаратных и программных мощностей для проведения симуляций, чем системная динамика или дискретно-событийное моделирование.

Инструменты агентного моделирования. Агентный подход возник в 90-х годах прошлого века изначально в университетской среде США. В связи с этим большинство инструментов предназначалось для академических и учебных целей, многие до сих пор не являются коммерческими продуктами в полной мере. Одной из наиболее популярных разработок такого типа является среда Swarm (www.swarm.org) – коллекция библиотек под язык C, созданная в институте Санта-Фе. Наиболее известными коммерческими инструментами являются среды Ascape, RePast, AnyLogic. Последний из них является разработкой российской компании XJTeknologies (www.xjtek.ru). Его конкурентным преимуществом является поддержка всех трех парадигм имитационного моделирования и возможность использования их в рамках одной модели. Также AnyLogic отличает мощное производительное ядро, позволяющее симулировать поведение миллионов агентов; богатые возможности для анимации и графического описания модели; поддержка разнообразных типов экспериментов, включая анализ чувствительности, метод Монте-Карло, встроенный оптимизатор OptQuest; возможности интеграции с базами данных, ERP – , CRM-системами; набор библиотечных объектов из областей логистики, бизнес-процессов, пешеходной динамики. Информацию по другим инструментам агентного моделирования можно найти на электронном ресурсе (9).

Применение агентного моделирования

Агентный подход является наиболее молодым и потому наименее знакомым российским специалистам. Рассмотрим примеры успешного применения агентного моделирования в науке и бизнесе. К области научного применения относятся, например, задачи эпидемиологии. Простейший пример диаграммы, согласно которой моделируется распространение заболевания, изображен на *рис.3.* Одному объекту при этом может соответствовать несколько диаграмм состояний или блок-схем. Из рисунка видно, что при агентном подходе предметом моделирования является поведение отдельного объекта, а глобальное состояние системы является следствием. В эпидемиологии агентный подход позволяет моделировать сложные социальные сети, в том числе с учетом пространственного фактора, контакты между людьми, разную восприимчивость людей и степень их иммунитета. Это позволяет добиваться хороших результатов при прогнозировании скорости и характера распространения заболевания (10).

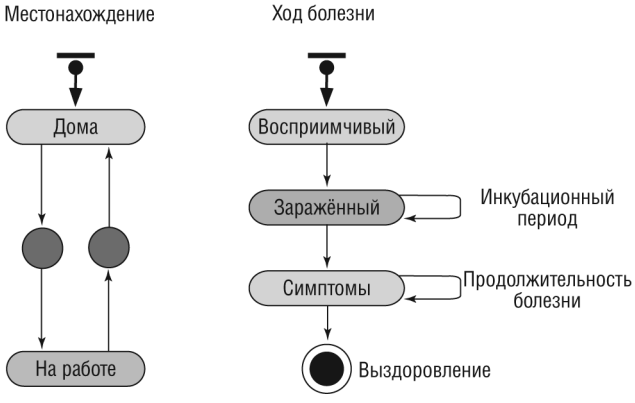


Рис. 3. Пример диаграммы состояний агента из модели «распространение гриппа»

Бизнес-проблематику, где успешно применяется агентный подход, можно разделить на 4 класса (11). К первому относятся задачи с различного рода потоками. Они могут состоять из людей (проблемы пропускной способности помещений, эвакуации), единиц транспорта (моделирование городского трафика, планировки аэропортов, вокзалов). Часто для этого применяется дискретно-событийный подход. Если же объекты моделирования слишком разнородны или необходимо учесть их пространственное взаиморасположение, то использование агентного подхода предпочтительнее. Второй класс задач относится к моделированию рынков, потребительских или финансовых. Агентный подход позволяет сделать акцент на индивидуальные предпочтения, стереотипы поведения потребителей при выборе ими продуктов и услуг. Отдельно выделяют задачи моделирования инноваций в бизнесе, их первоначального распространения на рынке. В четвертую группу относят задачи, связанные с оптимизацией организационной структуры, бизнес-процессов и снижением операционных рисков.

Примером задачи первого класса служит моделирование эвакуации при давках в местах массового скопления граждан. Поведение людей в таких ситуациях часто становится иррациональным и не поддается моделированию традиционными методами. Агентный подход зарекомендовал себя для отыскания оптимальных методов эвакуации и минимизации возможных рисков (12). Агентное моделирование часто применяется западными компаниями при проектировании парков развлечений, супермаркетов. В таких задачах оптимизируется геометрическое расположение элементов системы относительно друг друга (например, аттракционов и кафе, или касс и полок с продуктами). Целью может служить увеличение пропускной способности, сокращение времени стояния в очередях, оптимальное расположение рекламных материалов. Наглядным примером является оптимизированная модель супермаркета Sainsbury's в западном Лондоне, разработанная компанией SimStore в 1999 году (13). Собрав статистику о предпочтениях посетителей и характерных для них путях следования в магазине, разработчики использовали агентный подход в сочетании с генетическими алгоритмами для улучшения расположения элементов супермаркета. Отметим, что решать подобные задачи аналитическими или статистическими методами крайне затруднительно, и агентное моделирование является одним из немногих возможных средств поддержки принятия решений. Агентный подход хорошо применим для моделирования функционирования бирж. Результаты торгов зависят от поведения множества независимых людей с различными целевыми функциями, и их поведение логично моделировать в рамках агентного подхода. Подтверждением этому служит, например, модель Bios Group фондовой биржи NASDAQ (14). Модель измерения и минимизации операционных рисков управления активами банка Societe Generale (15) служит примером применения АМ в финансовых институтах и банках.

Выводы

Современные инструменты имитационного моделирования позволяют эффективно применять его не только в научных изысканиях, но и в качестве средства для построения систем поддержки принятия решений в бизнесе. Для достижения практически значимых результатов необходимо знать об особенностях и ограничениях каждого из трех существующих подходов. Выбор той или иной парадигмы должен обуславливаться не столько предметной областью моделирования, а необходимой степенью детализации системы и имеющимися в распоряжении данными. Выбор инструмента моделирования при этом не столь первостепенен. Большинство современных программных средств сделает процесс симуляции удовольствием для разработчика, предоставив широкие возможности для анимации и оптимизации моделируемых процессов. ■

Литература

1. Борщёв А. От системной динамики и традиционного ИМ – к практическим агентным моделям: причины, технология, инструменты. – Электрон. Дан. – Режим доступа: www.gpss.ru/paper/borshevarc.pdf, с. 3.
2. Gordon G. A General Purpose Systems Simulation Program. // McMillan NY, Proceedings of EJCC, Washington D.C., 1961, p. 87–104.
3. OR/MS Today. Simulation Software Survey. – Электрон. Дан. – Режим доступа: www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html.
4. Forrester J. Industrial Dynamics: A Major Breakthrough for Decision Makers. Harvard Business Review, 1958, vol. 36, №. 4, p. 37–66.
5. Медоуз Д.Л. Системное поведение, «мания»-структура и загрязнение окружающей среды. Свердловск: изд-во Уральского университета, 1990.
6. Сидоренко В.Н. Системная динамика. М.: ТЕИС, 1998.
7. Сидоренко В.Н. Системно-динамическое моделирование в среде POWERSIM: Справочник по интерфейсу и функциям. М.: Макс-Пресс, 2001.
8. Jager W., Janssen M.A., De Vries H.J.M., De Greef J., Vlek C.A.J. Behaviour in commons dilemmas: Homo Economicus and Homo Psychologicus in an ecological-economic model. // Ecological Economics, 2000, vol. 35, p. 357–380.
9. General software and toolkits. Agent-Based Computational Economics (ACE) and Complex Adaptive Systems (CAS). – Электрон. Дан. – Режим доступа: www.econ.iastate.edu/tesfatsi/acecode.htm.
10. Dunham J.B. An Agent-Based Spatially Explicit Epidemiological Model in MASON. // Journal of Artificial Societies and Social Simulation, 2005, vol. 9, no. 1.
11. Bonabeau E. Agent-based Modeling: Methods and Techniques for Simulating Human Systems. // Proceedings of the National Academy of Sciences of the United States of America, 1999(suppl.3), p.7280–7287.
12. Helbing D., Farkas I., Vicsek T. // Nature (London) 2000, p. 407, 487–490.
13. Venables M., Bilge U. Complex adaptive modelling at J Sainsbury: the SimStore supermarket supply chain experiment. – Электрон. Дан. – Режим доступа: www.psych.lse.ac.uk/complexity/Seminars/1998/report98mar.htm.
14. NASDAQ Stock Market. – Электрон. Дан. – Режим доступа: www.worldscibooks.com/economics/6217.html.
15. Bonabeau E. Application of Simulation to Social Sciences. // Paris: Herme`s Sciences, 2000, p. 451–461.

ПОДХОД К ОЦЕНКЕ ПРОИЗВОДИТЕЛЬНЫХ РЕСУРСОВ ИНФОРМАЦИОННЫХ СИСТЕМ

А.П. Шабанов,
Кандидат технических наук, доцент,
главный эксперт ООО «Информационные Бизнес Системы»
Адрес: 127434, Москва, Дмитровское ш., д.9 Б. E-mail: AShabanov@ibs.ru.

Рассмотрен подход к оценке производительных ресурсов, минимально необходимых для функционирования информационных систем предприятия с территориально распределёнными филиалами (офисами). При оценке используется аппарат интервалов занятости системы массового обслуживания с ожиданием. Предлагаемый подход направлен на решение проблемы минимизации людских ресурсов, производительности серверов и пропускной способности трактов передачи данных, затрачиваемых на управление информационными системами предприятия.

Ключевые слова: оценка производительных ресурсов, распределённые информационные системы, управление информационными системами предприятия, системы массового обслуживания с ожиданием.

Постановка задачи

Исследование информационных систем производится, как правило, с помощью теории управления организационными системами, теории информации, теории активных систем и теории массового обслуживания. Общие вопросы построения и функционирования организационных структур исследуются в теории управления организационными системами. Вопросы, связанные с применением информации, исследуются в теории информации. Основным методом исследования в рамках данной теории является математическое моделирование процессов хранения, преобразования и передачи информации. Вопросы, обусловленные проявлениями активности работниками, исследуются в теории активных систем. Вопросы, связанные с расчётом мощности организационных структур, исследуются в теории массового обслуживания и приложениях к ней. Основным методом исследования в рамках данной теории является математическое моделирование обслуживающих систем с потерями, с ожиданием, с приоритетами, смешанных обслуживающих систем, сетей систем обслуживания, транспортных потоков и трафика, а также оптимизационное моделирование и численное моделирование систем обслуживания. Разработка новых подходов, конкретных методов и моделей в рамках или на стыке перечисленных выше теорий часто осуществляется в ответ на требования, которые выдвигает практика человеческой деятельности на том или ином промежутке времени. В настоящее время современное человеческое общество характеризуется применением, по существу, во всех сферах деятельности информационных систем, создаваемых с помощью и на основе средств вычислительной и телекоммуникационной техники. Производительными ресурсами таких информационных систем являются:

- людские ресурсы, – персонал, эксплуатирующий средства информационных систем;
- вычислительные ресурсы, – рабочие станции и персональные компьютеры, серверное и вспомогательное оборудование;
- транспортные ресурсы, – каналы связи, тракты передачи данных и коммуникационные устройства;
- программно-прикладные ресурсы, – программы и комплексы программ, производимые для

использования в области применения информационных систем.

Исследование, результаты которого представлены в настоящей работе, относятся к проблеме минимизации производительных ресурсов информационных систем при заданных вероятностно-временных требованиях со стороны бизнеса. Целью работы является изложение подхода к решению задачи оценки производительных ресурсов информационных систем на основе аппарата систем массового обслуживания с ожиданием.

Модель представления требований к производительным ресурсам

Создание информационных систем на современном предприятии осуществляется на основе процессного подхода. В соответствии с этим подходом производятся следующие работы:

- ✦ работы по описанию деятельности субъектов предприятия. Деятельность описывается в виде бизнес процессов, для каждого из которых составляется формальное описание функций, входов и выходов, функциональных ролей субъектов, информационного обеспечения;
- ✦ проектные работы;
- ✦ работы по внедрению информационной системы в практику предприятия, с предварительным проведением автономных и комплексных испытаний, опытной эксплуатации. При этом могут корректироваться описания бизнес процессов;
- ✦ разработка регламентов с описанием действий субъектов бизнеса по использованию средств информационной системы в каждом бизнес-процессе.

При проектировании информационной системы, в первую очередь, определяется состав функций бизнес процесса, выполнение которых осуществляется с применением производительных ресурсов информационной системы. На *рис. 1* приведена диаграмма бизнес процесса и показаны функции бизнес-процесса, выполняемые с применением производительных сил информационной системы.

Далее, производится оценка производительных ресурсов информационной системы, требуемых для поддержки бизнес-процессов. Результатом такой оценки являются значения следующих показателей производительных сил:

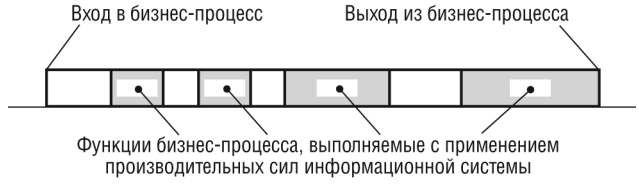


Рис. 1. Диаграмма бизнес-процесса

- ✦ требуемые производительности центрального вычислительного комплекса (ЦВК) информационной системы, других вычислительных ресурсов;
- ✦ требуемые пропускные способности трактов передачи данных между филиалами, офисами предприятия и внутри их;
- ✦ структура и состав служб эксплуатации информационной системы.

Проведение оценки производительных ресурсов информационной системы особенно необходимо для крупных предприятий, осуществляющих свою деятельность в различных районах планеты и в различных областях бизнеса. Для таких предприятий требуются высокопроизводительные вычислительные ресурсы, арендуемые или собственные транспортные ресурсы, персонал, эксплуатирующий систему и формализация условий применения производительных ресурсов (например, соглашения о предоставляемых бизнесу услугах в области информационных технологий). Важным, влияющим на необходимость проведения оценки, фактором является всё более возрастающая зависимость между мощностью (производительностью, пропускной способностью) программно-прикладных, вычислительных, транспортных ресурсов и их стоимостью. Оценка производительных ресурсов информационной системы производится исходя из вероятностно-временных требований, предъявляемых к ним бизнес процессами. Для представления требований к производительным ресурсам предварительно, исходя из накопленного в мире практического опыта, выбираются несколько вариантов структурных схем распределения ресурсов по филиалам (офисам) предприятия. При этом если это допускается условиями бизнеса, один вариант соответствует централизации вычислительных ресурсов (все операции выполняются в ЦВК). Для каждого варианта составляется модель представления требований к производительным ресурсам (*рис. 2*). В модели каждый производительный ресурс представляется в виде однородной системы массового обслуживания с ожиданием $(T_{зад}; P_{зад})_j^i$, где j – принимает следующие значения:

- ✦ j = ППР, – означает, что данный ресурс является программно-прикладным;
- ✦ j = ВР, – означает, что данный ресурс является вычислительным;
- ✦ j = ТР, – означает, что данный ресурс является транспортным;
- ✦ j = ЛР, – означает, что данный ресурс является людским;

i – принимает следующие значения:

- ✦ для j = ППР, $i \in$ 1,2,...,N1, 1УИС, 2УИС,..., N1УИС, – означает идентификационный номер программно-прикладного ресурса. УИС, – управляющая информационная система, – система используется персоналом, эксплуатирующим информационную систему предприятия;

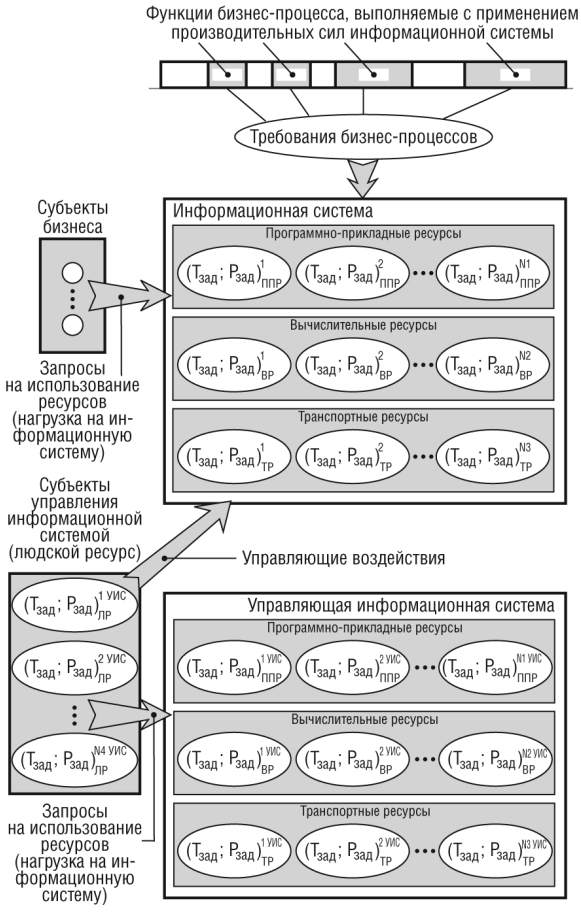


Рис. 2. Модель представления требований к производительным ресурсам

- ✦ для j = ВР, $i \in$ 1,2,...N2, – означает идентификационный номер вычислительного ресурса;
- ✦ для j = ТР, $i \in$ 1,2,...N3, – означает идентификационный номер транспортного ресурса;
- ✦ для j = ЛР, $i \in$ 1,2,...N4, – означает идентификационный номер людского ресурса (количество специалистов, выполняющих ту или иную функциональную роль по эксплуатации информационной системы);
- ✦ $T_{зад}$ и $P_{зад}$, – заданное время обработки (передачи) данных и заданная вероятность непревышения этого времени для соответствующего производительного ресурса, определяемого значениями i и j.

Подход к оценке производительных ресурсов

Оценка производительных ресурсов информационной системы для каждого варианта структурной схемы распределения ресурсов по филиалам (офисам) предприятия осуществляется следующим образом:

1. Оценка каждого программно-прикладного ресурса информационной системы предприятия производится с учётом заданных бизнесом вероятностно-временных требований, и принимаемых в качестве исходных данных параметров нагрузки (потока запросов) на использование данного ресурса. В результате определяются требования к суммарному времени облуживания вычислительных ресурсов, на которых функционирует соответствующее программное приложение (его клиентская и серверная составляющие) и тех транспортных ресурсов, через которые осуществляется передача данных (между клиентскими программами и сервером приложения). Конкретные взаимосвязанные цепи вычислительных и транспортных ресурсов, относящиеся к одному программно-прикладному ресурсу информационной системы, определяется при разработке варианта структурной схемы для распределения ресурсов. Также при разработке варианта определяются части общего времени обслуживания, приходящиеся на каждый ресурс в цепи взаимосвязанных ресурсов.

2. Оценка каждого вычислительного ресурса информационной системы предприятия производится с учётом заданных бизнесом вероятностно-временных требований, а также требований ко времени облуживания в данном вычислительном ресурсе (*см. п. 1*). В результате определяется производительность этого вычислительного ресурса (количество обрабатываемых данных в единицу времени), удовлетворяющая заданным требованиям. Производительности вычислительного ресурса ставится в соответствие стоимость, определяемая исходя из цен (прайс-листов), основных производителей вычислительных ресурсов.

3. Оценка каждого транспортного ресурса информационной системы предприятия производится

с учётом заданных бизнесом вероятностно-временных требований, а также требований ко времени облуживания (передаче данных) в данном транспортном ресурсе (*см. п. 1*). В результате определяется требуемая пропускная способность этого транспортного ресурса (количество передаваемых данных в единицу времени). Пропускной способности транспортного ресурса ставится в соответствие стоимость, определяемая исходя из цен (прайс листов), основных производителей транспортных ресурсов, стоимости приобретения (организации и эксплуатации) или стоимости аренды каналов связи (трактов передачи данных).

4. Оценка каждого программно-прикладного ресурса управляющей системы производится с учётом накопленного мирового опыта по эксплуатации информационных систем предприятий и особенностей конкретного предприятия. Формализация деятельности субъектов управления и автоматизация этой деятельности осуществляется также на основе процессного подхода. При этом устанавливаются соотношения между услугами субъектов управления, с одной стороны, и бизнес процессами, с другой стороны. Базовой основой для функционирования управляющей информационной системы, как правило, являются готовые программные продукты, производимые на основе процедур реализации процессов управления [1] и документации предприятия – разработчика программ [2]. При этом учитывается опыт, накопленный в мировой практике при формализации процессов, и особенности, обусловленные проведением проектных работ [3–5].

5. Оценка каждого вычислительного и транспортного ресурса производится в соответствии с п.п. 2 и 3.

6. Оценка людских ресурсов осуществляется в соответствии положениями, изложенными в работе [6].

7. Для каждого варианта структурной схемы для распределения ресурсов выполняются п.п. 1–6.

8. Производится оценка совокупной стоимости производительных ресурсов по каждому варианту и выбирается один из них. Правило определения совокупной стоимости и правило выбора варианта для последующей реализации определяются при разработке вариантов структурной схемы для распределения ресурсов.

Математический аппарат для оценки производительных ресурсов

Оценка производительных ресурсов информационной системы для каждого варианта структурной схемы распределения ресурсов по филиалам (офисам) предприятия производится с помощью известного аппарата систем массового обслуживания с очередями [7]. В то же время, существуют определённые предпосылки к применению, по крайней мере, для оценки программно-прикладных, вычислительных и транспортных ресурсов управляющей информационной системы, малоизвестного аппарата интервалов занятости. Первые упоминания о математическом аппарате интервалов занятости систем массового обслуживания (*рис. 3*) для расчёта значений вероятностно-временных показателей технических систем связаны с исследованиями известного американского учёного Т. Саати в области вооружений во время второй мировой войны [8].

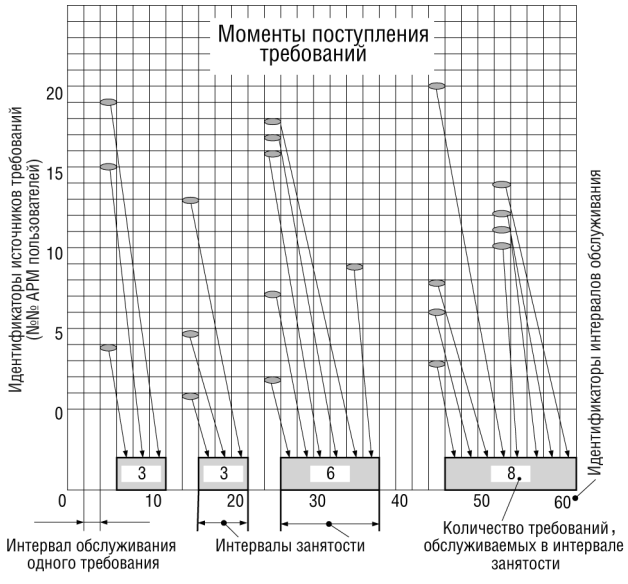


Рис. 3. Схема образования интервалов занятости

По мнению автора, в настоящее время, в связи с внедрением в информационные системы технических и программных средств автоматизации на основе заранее проводимой формализации процессов, аппарат интервалов занятости также востребован. Основанием для этого являются следующие достаточно естественные гипотезы:

- ✦ моменты поступления требований (запросов) на обслуживание в производительные ресурсы информационных систем, – случайны;
- ✦ требования группируются в интервалах занятости производительных ресурсов;
- ✦ параметры оценки обслуживания (время ожидания, время передачи, время обработки, время отклика приложения или базы данных и т.п. и допустимые вероятности не превышения этого времени) в производительных ресурсах имеют критичные значения;

- ✦ среда обеспечения деятельности субъектов бизнеса и субъектов управления информационными системами – изменчива;
- ✦ деятельность по использованию и управлению производительными ресурсами в большой степени формализована и автоматизирована;
- ✦ предметные сущности в среде обеспечении деятельности – исчисляемы (серверы, персональные компьютеры, каналы связи, программы и т. п.).

Изложенные выше гипотезы могут оказать определяющее влияние на методические стороны исследования современных информационных систем в целом, и их производительных ресурсов, в частности. На *)* приведена схема модели системы массового обслуживания с ожиданием, которая используется для определения вероятностно временных параметров производительных ресурсов.

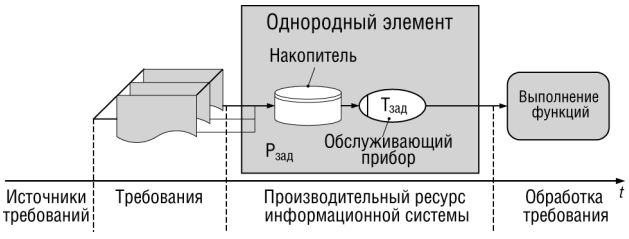


Рис. 4. Схема модели системы массового обслуживания

Применение аппарата интервалов занятости в такой системе базируется на значениях вероятностей ожидания требований внутри интервалов занятости различной длительности. Для расчёта этих значений используются математические соотношения, приведённые в работе [9]. Применение данных соотношений носит прогнозированный характер, и могут быть применены для структурирования производительных ресурсов при проведении проектных работ.

Примеры использования подхода к оценке производительных ресурсов

Приведённый в данной работе подход к оценке производительных ресурсов информационных систем был использован в проектных работах для решения задач:

- ✦ структурирования средств автоматизации управленческой деятельности [10];
- ✦ определения минимально достаточной производительности централизованного вычислительного комплекса, осуществляющего поддержку функций управления деятельностью на предприятии с пространственно разнесёнными филиалами и офисами [11];
- ✦ определения минимально достаточной пропускной способности тракта передачи информации на участке между филиалом (удалённым офисом) предприятия и центральным вычислительным комплексом [12];
- ✦ разработки архитектуры системы контроля над достаточностью ресурсов автоматизированной системы, в условиях её динамично изменяющейся во времени загрузки [13];
- ✦ построения и функционирования системы оперативного оповещения специалистов служб поддержки потребителей услуг информационных технологий и оперативно-технического управления [14];
- ✦ построения и функционирования системы управления учетными записями сотрудников предприятия (организации, учреждения) на объектно-ролевой основе и централизованного распределения прав доступа [15];
- ✦ построения и функционирования централизованной системы подготовки отчётности о качестве предоставления услуг информационных технологий [16]. ■

Литература

1. ИТ Сервис-менеджмент, введение. – Издатель: Van Haren Publishing, 2003. Русский перевод «IT Expert», 2003, 223 с.
2. BMC Atrium CMDB 2.0: Implementation Best Practics. – Издатель: BMC Software, Inc., 2006, 662 p.
3. Государственный стандарт Российской Федерации. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Стадии создания.
4. Шабанов А.П. «Подход к автоматизации деятельности организационных структур». Системы автоматизации в образовании, науке и производстве: Труды V Всероссийской научно-практической конференции/ Под общ. ред. д.т.н., проф. Кулакова С.М., д.т.н., проф. Мышляева Л.П. – Новокузнецк, СибГИУ, 2005, с. 124-128.
5. Шабанов А.П. «Подход к формализации учётно-контрольного процесса при решении проблемы минимизации ресурсов организационной структуры». Современные сложные системы управления: Сборник трудов научно-практической конференции/ Под ред. д.т.н., проф. Буркова В.Н., д.т.н., проф. Баркалова С.А. – Воронеж, ВГАСУ, 2005, с. 186.

6. Шабанов А.П., Беляков А.Г. Организационные структуры массового обслуживания – М., 2007 (Научное издание / Институт проблем управления им. В.А. Трапезникова РАН).
7. Ивченко В.И., Каштанов В.А., Коваленко И.Н. Теория массового обслуживания. – М.: Высшая школа, 1982.
8. Саати Т.Л. Элементы теории массового обслуживания и её приложения. – М.: Советское радио, 1971.
9. Шабанов А.П. Метод оценки достаточности мощности однородной организационной структуры. – Системы управления и информационные технологии, 2005, N 3(20), с. 103-106.
10. Аракелян М.А., Шабанов А.П. Методологические задачи структурирования средств автоматизации управленческой деятельности. – М., 2009 / Сборник тезисов докладов первой научно-практической конференции «Информационные бизнес системы».
11. Голяндин А.Н., Шабанов А.П. Определение производительности вычислительного комплекса автоматизированной системы с централизованным управлением.- М., 2009 / Сборник тезисов докладов первой научно-практической конференции «Информационные бизнес системы».
12. Парижский О.Ю., Шабанов А.П. Определение пропускной способности трактов передачи управляющей информации. – М., 2009 / Сборник тезисов докладов первой научно-практической конференции «Информационные бизнес системы».
13. Рузавин С.Г., Шабанов А.П. Контроль над достаточностью ресурсов информационных систем. – М., 2009 / Сборник тезисов докладов первой научно-практической конференции «Информационные бизнес системы».
14. Мартыньш А.В., Терегулов К.А. Подсистема оперативного оповещения для поддержки потребителей услуг информационных технологий. – М., 2009 / Сборник тезисов докладов первой научно-практической конференции «Информационные бизнес системы».
15. Боганов А.В. Система автоматической идентификации потребителей услуг информационных технологий. – М., 2009 / Сборник тезисов докладов первой научно-практической конференции «Информационные бизнес системы».
16. Жаворонков А.В. Интегрированная система отчётности об информационно-технологической поддержке предприятия. 8. М., 2009 / Сборник тезисов докладов первой научно-практической конференции «Информационные бизнес-системы».

Уважаемые коллеги!

Приглашаем Вас принять участие в международной конференции

ICCS'09 CONCEPTUAL STRUCTURES: LEVERAGING SEMANTIC TECHNOLOGIES

26-31 июля 2009 г.,

Москва, Государственный Университет Высшая Школа Экономики (ГУ-ВШЭ)
http://iccs09.hse.ru

Труды конференции будут опубликованы в серии Lecture Notes in Artificial Intelligence, Springer к началу конференции

Срок подачи текстов докладов – до 12 января 2009 г.

Язык конференции – английский

Участникам конференции – гражданам Союзного Государства будет предоставлена значительная скидка в уплате конференционного взноса.

Следите за информацией на сайте конференции

С.О. Кузнецов, *председатель оргкомитета ICCS'09*

✦

CRYPTANALYSIS AND CRYPTOGRAPHY: THE ETERNAL CONFRONTAION

Avdoshin Sergey Mihailovich,
State University - Higher School of Economics, Russia, Moscow.
Savelieva Alexandra Alexandrovna,
State University - Higher School of Economics, Russia, Moscow.

Annotation

In this paper, we provide a survey of the history of cryptology - the science incorporating cryptography and cryptanalysis that has become one of the most advanced knowledge areas in the computer era. We illustrate how the advent of new code-breaking techniques fosters the development of new ciphers, while cryptographers'.

Key words: Cryptanalysis, cipher security, code-breaking, cryptography, quantum computing.

✦

INTERACTIVE ANALYTICAL DATA PROCESSING IN MODERN OLAP SYSTEMS

Kashirin Igor Yurievich,
Ryazan Government University of Radio Engineering (RGURE), Russia, Ryazan.
Semchenkov Sergey Yurievich,
Ryazan Government University of Radio Engineering (RGURE), Russia, Ryazan.

Annotation

In article aspects of data processing with usage of the multidimensional analysis are considered. The main concepts of multidimensional model are described, classification OLAP of systems is considered. Questions of sharing OLAP and OLTP systems are taken up, the comparative analysis of existing systems is carried out.

Key words: OLAP systems. Multidimensional model of data processing. Interactive data analysis.

✦

DEVELOPMENT OF SYSTEM TARGETED TO CONTROL BANNER AD SHOWING IN CELLULAR NETWORK

Kasymova Dilnara Nadimullaevna,
Limited company «Mobil-2», Russia, Novosibirsk.

Annotation

This article is devoted to technology of using cellular communications in Russia for banner ad showing. Here we investigate algorithms for selection of targeted banner ad for specified telephone subscriber and provide

comparative analysis of different similarity search acceleration methods for parameters specified by cell phone company.

Key words: Mobile advertisement. Similarity search. Context advertisement. Index-based similarity search.

✦

COMBINED AND WAVE SOLUTION ALGORITHMS FOR THE PROBLEM OF THE PACKING: THE PRINCIPLES OF CONSTRUCTION AND SPECIAL FEATURE

Uljyanov Mikhail Vasiljevich,
The Moscow State University of Print (MGUP), Russia, Moscow.
Naumova Olga Anatoljevna,
The Moscow State University of Instrumental Engineering and Computer Sciences (MGUPI), Russia, Moscow.

Annotation

The article is viewing basic exact solution algorithms for the problem of one-dimensional optimum for the cost packing - recursive and table algorithms. Combined and wave algorithms are suggested since rational combination of basic algorithms allows them to have better resource characteristics. Differences in use and in the resource requirements of suggested algorithms are indicated.

Key words: Problem of optimum packing; combined algorithm; wave algorithm; resource effectiveness.

✦

STRUCTURAL AND PARAMETRICAL IDENTIFICATION OF LLSQ-MODEL WITHOUT THE SOLUTION OF THE NORMAL EQUATIONS SYSTEM

Belov Vladimir Victorovich,
The Ryazan State Radio Engineering University (RSREU), Russia, Ryazan.

Annotation

The way of calculation of values of the LLSQ-model, in particular MLR (Multiple Linear Regression), without estimate its parameters is offered. The offered way can be used in algorithms of search of the best somewhat the linear description of the process submitted by discrete values, for example, by a time series. The recursion scheme of calculation of parameters of the LLSQ-model, alternative to the solution of the Gauss normal equations system is offered.

Key words: Linear least squares mode (LLSQ-model). Consecutive addition of variables. Calculation of values. Estimate of parameters. Recursion scheme.

✦

COMBINATORIAL SCHEME FOR DESIGN OF MARKETING STRATEGY

Levin Mark Sh. (Levin M.Sh.),
Institute for Information Transmission Problems (Russian Academy of Sciences).

Annotation

Combinatorial scheme for design of marketing strategy is examined. The scheme consists of four stages: (a) clustering of good set to obtain some groups of close goods, (b) clustering of customer set to obtain some groups of close customers (i.e., market segments), (c) assignment for each group of close goods a certain group of close customers

(i.e., market segment) or several customer groups (usage of assignment problem), (d) definition of a certain marketing strategy for each pair «group of goods - group of customers» (usage of multiple choice problem while taking into account a total constraint to implementation of all selected marketing strategies). Realistic numerical example is targeted to computer goods (computers, processors, software etc.).

Key words: marketing strategies, combinatorial marketing design, clustering , multiple choice problem, marketing strategies and segmentation.

✦

IMITATION MODELING IN SCIENCE AND BUSINESS: APPROACHES, TOOLS AND APPLICATION

Sidorenko Vladimir Nikolayevich,
The Moscow State University (MGU), Russia, Moscow.
Krasnoselskiy Aleksey Valeryevich,
The Moscow State University (MGU), Russia, Moscow.

Annotation

The article is an overview of existing approaches and development tools in simulation modeling. All the three paradigms - system dynamics, discrete-event and agent modeling - are depicted. An agent-based approach is considered in more details as it is comparatively seldom used in Russia but is known as an efficient tool for developing decision support systems in business. The article has its aim to give first acquaintance with simulation modeling and to describe that circle of problems in business where its use could be efficient.

Key words: Imitation modeling, discrete-event and agent modeling, agent-based approach.

✦

THE APPROACH TO PRODUCTIVITY RESOURCES OF THE INFORMATION SYSTEMS ESTIMATION

A.P. Shabanov,
Candidate of technical sciences, associated professor, the principal expert «Informational Business Systems « Ltd.

Annotation

Approach to the estimation of the productive resources is examined, which are minimally necessary for the functioning of the information systems of enterprise with the territorially distributed branches (offices). During the estimation the apparatus of the busy periods of the system of queueing with the expectation is used. The proposed approach is directed toward the solution of the problem of the minimization of the human resources, of the productivity of servers and capacity of the circuits of the transmission of the data, spent on control of the information systems of the enterprise.

Key words: production resources assessment, distributed information systems, management of enterprise information systems, system of queueing with the expectation.

ЧЕТВЕРТАЯ ВСЕРОССИЙСКАЯ
НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ
«ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ. ТЕОРИЯ И ПРАКТИКА»

(ИММОД-2009)

21–23 октября 2009 г.

Генеральный спонсор конференции
ООО «Экс Джей Технолоджис»

www.anylogic.ru

Россия
Санкт-Петербург, 21–23 октября 2009 г.

Четвертая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности **«Имитационное моделирование. Теория и практика»** ИММОД-2009 состоится в Санкт-Петербурге (Россия) 21–23 октября 2009 года.

Цель конференции – распространение методов и средств имитационного моделирования для решения научных и практических задач, активизация творческой деятельности и укрепление научно-производственного потенциала РФ.

Организаторы и учредители конференции:
- ✦ ОАО «Центр технологии судостроения и судоремонта» (Санкт-Петербург);
- ✦ Институт информатики и автоматизации РАН (Санкт-Петербург);
- ✦ Отделение нанотехнологий и информационных технологий РАН (Москва);
- ✦ Российский национальный комитет по индустриальной и прикладной математике.

Информационную поддержку конференции осуществляют:
- ✦ ООО «Элина-Компьютер» (Казань);
- ✦ Институт информатики и автоматизации РАН (Санкт-Петербург);
- ✦ ОАО «Центр технологии судостроения и судоремонта» (Санкт-Петербург);
- ✦ компания XJ Technologies (Санкт-Петербург);
- ✦ журнал «Rational Enterprise Management» (Санкт-Петербург);
- ✦ журнал «Судостроение» (Санкт-Петербург).

Генеральный спонсор конференции – компания XJ Technologies (Санкт-Петербург).
Спонсоры конференции:
✦ Отделение нанотехнологий и информационных технологий РАН (Москва);
✦ Российский фонд фундаментальных исследований (Москва);
✦ Комитет по науке и высшей школе при Администрации Санкт-Петербурга.

Задачи конференции:
✦ развитие и обобщение теории имитационного моделирования и квалиметрии моделей;
✦ обмен опытом и обсуждение результатов исследований и практических приложений имитационного моделирования;
✦ обмен опытом применения имитационного моделирования для решения научных и практических задач;
✦ распространение опыта обучения теории и практике имитационного моделирования.

В программу конференции предполагается включать доклады, относящиеся, в первую очередь, к области моделирования систем с дискретными событиями и временем. Работы по моделированию непрерывных систем на базе дифференциальных уравнений будут приниматься к рассмотрению только в том случае, если они содержат результаты, относящиеся к общим методологическим и программно-техническим проблемам имитационного моделирования.

Научная программа конференции включает следующие тематические направления:
✦ теоретические основы и методология имитационного моделирования;
✦ методы оценивания качества моделей;
✦ методы и системы распределенного моделирования;
✦ моделирование глобальных процессов;
✦ средства автоматизации и визуализации имитационного моделирования;
✦ системная динамика (с обязательным наличием имитационной составляющей);
✦ практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования;
✦ имитационное моделирование в обучении и образовании.

Место проведения конференции – Санкт-Петербург, Россия.
Рабочий язык конференции – русский.

Размещение участников конференции и сопровождающих лиц будет организовано в гостиницах Санкт-Петербурга. Резервирование гостиниц производится в соответствии с предварительной заявкой, помещённой в извещении о намерении. Оплата проживания и других гостиничных услуг производится участником конференции.

Организационный взнос - 500 рублей.
Для аспирантов и студентов участие бесплатное.

Вам предоставляется возможность обсудить с коллегами из разных городов последние достижения российской науки в области решения задач имитационного моделирования и применения их в научных исследованиях и промышленности и других областях народного хозяйства.

Программа конференции предусматривает всестороннее обсуждение основных результатов и тенденций в области решения теоретических и прикладных проблем имитационного моделирования и определении наиболее приоритетных и перспективных направлений дальнейших исследований в данной предметной области.

В рамках перечисленных тематических направлений будут проведены пленарные заседания, организована работа секций и круглых столов. Кроме того, предусмотрена презентация участниками конференции своих программных средств, используемых при моделировании сложных объектов и процессов в различных предметных областях.

На основе представленных участниками конференции материалов будет сформирована и издана программа конференции, а также сборник докладов.

Организационный комитет надеется, что работа на конференции будет способствовать укреплению сотрудничества отечественных партнеров, а также установлению новых научных и деловых связей с коллегами за рубежом.

Мы полагаем, что эта встреча будет для Вас приятной и запомнится надолго, поскольку состоится в Санкт-Петербурге – одном из красивейших городов мира, центре науки и культуры.

Оргкомитет конференции ИММОД-2009

Приглашение к докладам

Авторы приглашаются представить материалы докладов в секретариат конференции не позднее 17 сентября 2009 года (только в этом случае гарантируется их размещение в сборнике материалов конференции). Заявки на выступление с докладом (в форме извещения о намерении) принимаются с 16 марта 2009 года. Предельный срок подачи заявок на выступление с докладом – не позднее 12 октября 2009 года.

Авторы и участники приглашаются к участию в стендовых демонстрациях на автоматизированных рабочих местах. Компьютеры для демонстрации предоставляются.

Для участия в стендовой демонстрации необходимо заполнить заявку участника стендовой демонстрации (извещение о намерении). В заявке обязательно указать необходимые аппаратные и системные требования к компьютеру. Заявки на стендовую демонстрацию принимаются с 16 марта 2009 года.

Предельный срок заявок на выступление со стендовым докладом –
не позднее 12 октября 2009 года.

Пожалуйста, направьте заявку на участие в стендовом докладе вместе с заполненной формой справочных данных об авторах доклада для их регистрации в секретариат конференции.

Материалы для публикации в сборнике, включая все необходимые графические иллюстрации, таблицы и формулы, должны быть подготовлены в виде электронного документа в Microsoft Word (с установленной автоматической проверкой правописания ? меню Сервис, команда Правописание). Нумерация иллюстраций, таблиц и формул сквозная. Подрисуночные надписи должны быть выполнены отдельно от рисунка (не должны включаться в графический рисунок). Формат страницы – А4 (210×297 мм) с полями со всех сторон не менее 25 мм. Основной текст следует набирать шрифтом Times New Roman 12. Отступ абзацев – 1.25 см. Межстрочный интервал – одинарный. Номера формул, заключенные в круглые скобки, следует помещать с правой стороны. Имеющиеся в тексте рисунки должны быть также приложены в виде отдельных файлов в формате jpg. К публикации принимаются материалы объемом до 5 страниц (пленарные доклады объемом до 10 страниц). Образец оформления помещен на сайте www.gpss.ru. При подготовке графических иллюстраций следует иметь в виду черно-белый формат оформления сборника.

Версия текстового редактора – не выше Microsoft Word 2003.

Текст материалов доклада должен включать название доклада (на русском и английском языках), аннотацию объемом до 500 знаков (на русском и английском языках), полное имя и фамилию автора (соавторов) (на русском и английском языках), почтовый адрес, номера телефона и факса, электронный адрес.

Предпочтительно представление материалов докладов электронной почтой в заархивированном виде (желательно архиватором zip). При отсутствии электронной почты – представление материалов по обычной почте либо непосредственно в секретариат конференции.

Принимаются только доклады, затрагивающие проблемы имитационного моделирования и содержащие чётко выделенные элементы существенной новизны. Каждый доклад должен быть отнесен автором к одной из следующих категорий: теоретической, педагогической, практической, стендовой.

Текст доклада должен завершаться выводами (не путать с перечислением результатов!).

Материалы докладов, включенных в программу конференции, будут опубликованы на русском языке в сборнике докладов конференции ИММОД-2009 к ее началу. Представленные доклады будут проходить научное рецензирование. Программный комитет оставляет за собой право отклонять материалы, не соответствующие профилю конференции, имеющие недостаточную научную или практическую значимость, либо выполненные с явными нарушениями требований на оформление материалов и на низком литературном уровне. Публикуемые в сборнике материалы будут помещаться в формате «как есть», литературное редактирование текстов будет выполняться только в случаях явных опечаток и описок.

Аспиранты и студенты допускаются к выступлениям после представления в Программный комитет выписки из протокола научно-технического семинара, на котором заслушивался доклад.

Компьютеры для демонстрации в ходе доклада предоставляются. Возможны выступления с компьютерами докладчиков. Весь иллюстративный материал к докладу представляется в виде компьютерных презентаций (PowerPoint) либо видеофильмов (в том числе озвученных). Компьютерные презентации (в формате pps) и видеофильмы по желанию авторов будут записаны на CD для распространения среди участников конференции. Предельный срок передачи презентаций и видеофильмов для записи на CD – первый день работы конференции (через секретариат).

Предварительная программа конференции будет размещаться на сайтах www.gpss.ru и www.crist.ru.

Пожалуйста, направьте материалы доклада вместе с заполненной формой справочных данных об авторах доклада для их регистрации в секретариат конференции по адресу:

ОАО «Центр технологии судостроения и судоремонта»
Россия, 198095, Санкт-Петербург, Промышленная ул., д.7
Плотников Александр Михайлович (руководитель секретариата)
Долматов Михаил Анатольевич (исполнительный секретарь)
E-mail: immod2009@crist.ru

Реквизиты банковского счета
для перечисления организационного взноса:

ОАО «ЦТСС», ИНН 7805482938, КПП 998050001
198095, Санк-петербург, ул.Промышленная, д.7
ОАО «Банк ВТБ Северо-Запад», филиал «Кировский объединенный»
БИК 044030791, к/с 30101810200000000791
р/с 40702810915000003966

Назначение платежа –
организационный взнос конференции
ИММОД-2009.

Оргкомитет конференции ИММОД-2009

Тематические рубрики журнала «БИЗНЕС-ИНФОРМАТИКА»

№	Специальность номенклатуры	Рубрика
1	05.13.10	Математические модели социальных и экономических систем
2	05.13.11	Программная инженерия
3	05.13.17	Анализ данных и интеллектуальные системы
4	05.13.18	Математические методы и алгоритмы решения задач бизнес-информатики
5	05.13.18	Моделирование и анализ бизнес-процессов
6	05.25.05	Информационные системы и технологии в бизнесе
7	05.25.05, 05.13.11	Электронный бизнес
8	05.25.05, 05.13.17	Интернет-технологии
Дополнительные рубрики вне номенклатуры		
9		Тематические обзоры
10		Правовые вопросы бизнес-информатики
11		Стандартизация, сертификация, качество, инновации
12		Дискуссионный клуб / Опыт бизнеса

К СВЕДЕНИЮ АВТОРОВ

Редакция просит авторов при оформлении статей и тематических обзоров придерживаться следующих правил и рекомендаций:

1. Предоставляемый авторами материал должен соответствовать рекомендуемой структуре статей журнала.

2. Статья направляется в редакцию в электронном виде (в формате MS WORD версия 2003) и в виде бумажной копии, распечатанной на одной стороне листов А4. Первая страница оригинала подписывается всеми авторами статьи.

3. Ориентировочный объём статьи, предлагаемой к публикации, – 20–25 тыс. знаков (с пробелами) или 30–35 тыс. знаков – для обзорных статей по направлениям.

4. Кегль набора – 12 пунктов с полуторным интервалом. Нумерация страниц – вверху по центру. Поля: левое – 2,5 см, верхнее, нижнее и правое – по 1,5 см.

5. При наборе выключных и строчных формул должен быть использован редактор формул MS Equation. В формульных и символических записях греческие (русские) символы, а также математические функции записываются прямыми шрифтами, переменные аргументы функций в виде английских (латинских) букв записываются наклонным начертанием (курсивом), например, «cos *a*», «sin *b*», «min», «max».

6. Формулы, таблицы и сноски (не концевые) оформляются стандартными средствами редактора MS WORD. Нумерация формул, рисунков и таблиц – сквозная, по желанию авторов допускается двойная нумерация формул с указанием структурного номера раздела статьи и – через точку – номера формулы в разделе.

7. Рисунки (графики, диаграммы и т.п.) оформляются средствами Word, Excel, Illustrator. Ссылки на рисунки в тексте обязательны и должны предшествовать позиции размещения рисунка. Допускается использование графического векторного файла в формате wmf/emf или cdr v.10. Фотографические материалы предоставляются в формате TIF или JPEG с разрешением не менее 300 dpi.

8. Библиографический список составляется в соответствии с требованиями ГОСТ. Нумерация библиографических источников – в порядке цитирования. Ссылки на иностранную литературу – на языке оригинала без сокращений.

Структура статей строится по правилам, рекомендованным журналом «Бизнес-информатика».

Плата с аспирантов за публикацию рукописей не взимается.

Журнал «БИЗНЕС-ИНФОРМАТИКА» Рекомендуемая структура статей

Журнал публикует исследовательские научные статьи, размещаемые в рубриках журнала, тематические обзоры, отражающие современное состояние проблем в области бизнес-информатики и сообщения, размещаемые в рубриках «Дискуссионный клуб» и «Опыт бизнеса».

Титульный лист рукописи начинается с указания Ф.И.О. авторов публикации с обязательным указанием учёной степени, учёного звания, должности, основного места работы и e-mail. Титульный лист должен быть подписан всеми авторами статьи.

I. Исследовательские научные статьи (для размещения в тематических рубриках)

Редколлегия рекомендует авторам после названия статьи приводить ***аннотацию***, в которой излагается краткое содержание статьи, её основные результаты и область применения. Авторам рекомендуется структурировать статью, выделяя ***введение***, содержащее описание проблемы или задачи, обзор существующих подходов или методов решения, их недостатки, и основную цель статьи; ***постановку задачи***, включающую допущения и ограничения; ***содержательную часть*** статьи, в которой предлагаемые решения должны быть аргументированы и сравниваться с существующими подходами или решениями; ***заключение***, содержащее краткое изложение новых результатов, полученных в статье и область их применения; ***библиографический список***, оформленный в соответствии с ГОСТ. Текст статьи должен содержать нумерованные ссылки на все указанные библиографические источники. Структурирование статьи и нумерация её разделов проводится по усмотрению авторов.

Возможный вариант структуры статьи:

- Ф.И.О;
- учёная степень, учёное звание, должность, основное место работы, e-mail;
- название статьи.
- аннотация;
- 1. Введение.
- 2. Постановка задачи.
- 3. Основная содержательная часть статьи.
- 4. Экспериментальные результаты (опционально).
- 5. Заключение.
- 6. Библиографический список.

II. Тематические обзоры по направлениям

Редколлегия рекомендует авторам структурировать обзор, выделяя аннотацию, содержащую тематику, краткое содержание обзора и область применения; ***введение***, в котором даётся краткий исторический обзор тематики; ***содержательную часть*** обзора с критическим анализом существующих направлений; ***заключение***, в котором отражаются перспективы развития в рамках обозреваемой тематики и наиболее интересные направления с точки зрения научных и практических разработок и методов; ***библиографический список***, оформленный в соответствии с ГОСТ.

Текст обзора должен содержать нумерованные ссылки на все указанные библиографические источники. Структурирование обзора и нумерация его разделов проводится по усмотрению авторов.

Возможный вариант структуры обзора:

- Ф.И.О;
- учёная степень, учёное звание, должность, основное место работы, e-mail;
- название обзора;
- аннотация;
- 1. Введение.
- 2. Основная содержательная часть обзора.
- 3. Заключение.
- 4. Библиографический список.

Редколлегия журнала проводит обязательное рецензирование рукописей. Статья принимается к публикации только после получения положительного заключения рецензента и одобрения на заседании редакционной коллегии журнала.

ЖУРНАЛ «БИЗНЕС-ИНФОРМАТИКА» ОСУЩЕСТВЛЯЕТ РАЗМЕЩЕНИЕ РЕКЛАМНЫХ И РЕКЛАМНО-ИНФОРМАЦИОННЫХ МАТЕРИАЛОВ

Расценки:

Обложка: 2, 3, 4 страница обложки, полноцветная печать, полоса 210×290 мм (А4) – 40 тыс. руб.

Текстовый блок, чёрно-белая печать:
- ✦ полоса – 20 тыс. руб.;
- ✦ 1/2 полосы – 15 тыс. руб.;
- ✦ 1/4 полосы – 10 тыс. руб.;
- ✦ меньший объём – 7 тыс. руб.

Вставка (4 полосы, полноцветная печать – 60 тыс. руб.).

Рекламно-информационный блок (8 полос, полноцветная печать) – 80 тыс. руб.
Рекламно-информационный блок (16 полос, полноцветная печать) – 90 тыс. руб.

Корпоративный специальный выпуск –
по договоренности.

Материалы принимаются с учётом следующих параметров:
- ✦ дообрезной формат – 215×300 мм;
- ✦ обрезной формат – 210×290 мм;
- ✦ поле набора полосной рекламы – 190×270 мм – с отступом от границ обрезного формата по 10 мм с каждой стороны;
- ✦ файл TIF, EPS, PDF – разрешение не менее 300 dpi.