

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 24
2018
№ 11

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ОПТИМИЗАЦИЯ

- Курейчик В. М., Логунова Ю. А. Анализ перспективности применения генетического алгоритма при решении задачи коммивояжера 691
- Ульянов М. В., Фомичев М. И. Подходы к организации поискового дерева решений в методе ветвей и границ для асимметричной задачи коммивояжера 698

МОДЕЛИРОВАНИЕ

- Зак Ю. А. Принципы построения систем имитационного моделирования производственных систем 705
- Черняев А. В., Горбачев С. И., Михаленко М. А., Метелкин Е. В. Комплексное моделирование последствий аварийных разливов нефти в малых водотоках для информационной системы их прогнозирования 714

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Сорокин А. Б., Кушнарв А. П. Морфологический анализатор текста для выявления полноты информации 719

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Бурькова Е. В. Программная реализация выбора средств физической защиты объекта информатизации 725

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Лебедев Г. Н., Канушкин С. В., Кузнецова Т. И., Царегородцева М. Г. Автоматизированная компьютерная система индивидуального обучения при освоении простых и сложных навыков 731

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Кузнецова Е. В., Усадова Е. В. Управление проектными рисками в организациях сферы ИТ-услуг 741

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.

ЖУРАВЛЕВ Ю. И.,

акад. РАН, д. ф.-м. н., проф.

КУЛЕШОВ А. П.,

акад. РАН, д. т. н., проф.

ПОПКОВ Ю. С.,

акад. РАН, д. т. н., проф.

РУСАКОВ С. Г.,

чл.-корр. РАН, д. т. н., проф.

РЯБОВ Г. Г.,

чл.-корр. РАН, д. т. н., проф.

СОЙФЕР В. А.,

акад. РАН, д. т. н., проф.

СОКОЛОВ И. А.,

акад. РАН, д. т. н., проф.

СУЕТИН Н. В., д. ф.-м. н., проф.

ЧАПЛЫГИН Ю. А.,

акад. РАН, д. т. н., проф.

ШАХНОВ В. А.,

чл.-корр. РАН, д. т. н., проф.

ШОКИН Ю. И.,

акад. РАН, д. т. н., проф.

ЮСУПОВ Р. М.,

чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.

АНТОНОВ Б. И.

БАРСКИЙ А. Б., д. т. н., проф.

ВАСЕНИН В. А., д. ф.-м. н., проф.

ВАСИЛЬЕВ В. И., д. т. н., проф.

ВИШНЕКОВ А. В., д. т. н., проф.

ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.

ДОМРАЧЕВ В. Г., д. т. н., проф.

ЗАБОРОВСКИЙ В. С., д. т. н., проф.

ЗАРУБИН В. С., д. т. н., проф.

КАРПЕНКО А. П., д. ф.-м. н., проф.

КОЛИН К. К., д. т. н., проф.

КУЛАГИН В. П., д. т. н., проф.

КУРЕЙЧИК В. В., д. т. н., проф.

ЛЬВОВИЧ Я. Е., д. т. н., проф.

МАРТЫНОВ В. В., д. т. н., проф.

МИХАЙЛОВ Б. М., д. т. н., проф.

НЕЧАЕВ В. В., к. т. н., проф.

ПОЛЕШУК О. М., д. т. н., проф.

САКСОНОВ Е. А., д. т. н., проф.

СОКОЛОВ Б. В., д. т. н., проф.

ТИМОНИНА Е. Е., д. т. н., проф.

УСКОВ В. Л., к. т. н. (США)

ФОМИЧЕВ В. А., д. т. н., проф.

ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.

ЛЫСЕНКО А. В.

ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

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

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 24
2018
No. 11

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

OPTIMIZATION

- Kureichik V. M., Logunova J. A.** The Genetic Algorithm Application Prospects Analysis for the Traveling Salesman Problem Solution 691
- Ulyanov M. V., Fomichev M. I.** Approaches to Design Search Decision Tree in the Branch and Bound Method for the Asymmetric Traveling Salesman Problem . . . 698

MODELING

- Zack Yu. A.** Principles of Construction of Systems of Imitation Modeling of Production Systems 705
- Chernyaev A. V., Gorbachev S. I., Mikhaleenko M. A., Metelkin E. V.** Comprehensive Modeling of the Effects of Oil Spills in Small Streams for the Information System in their Forecasting 714

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Sorokin A. B., Kushnarev A. P.** Morphological Text Analyzer for Revealing the Completeness of Information 719

INFORMATION SECURITY

- Burkova E. V.** Program Implementation of the Selection of Physical Protection of the Object of Informatization 725

INFORMATION TECHNOLOGIES IN EDUCATION

- Lebedev G. N., Kanushkin S. V., Kuznetsova T. I., Tsaregorodtseva M. G.** Automated Computer-Based Individual Learning System for Mastering Simple and Complex Skills 731

APPLICATION INFORMATION SYSTEMS

- Kuznetsova E. V., Usadova E. V.** Project Risk Management in IT Industry Organizations 741

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vasiliev V. I., Dr. Sci. (Tech.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. V., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Martynov V. V., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Saksonov E. A., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

В. М. Курейчик, д-р техн. наук, проф., vmkureychik@sfedu.ru,

Ю. А. Логунова, аспирант, Julia1000@yandex.ru,

Федеральное государственное автономное образовательное учреждение высшего образования
"Южный федеральный университет", г. Таганрог

Анализ перспективности применения генетического алгоритма при решении задачи коммивояжера

Выполнен анализ перспективности применения генетического алгоритма (ГА) при решении актуальной в области искусственного интеллекта задачи коммивояжера. В модифицированном алгоритме применены показатели, которые характеризуют степень разнообразия популяции. Специально разработанное программное обеспечение позволило провести тестирование ГА на известной бенчмарке att-48 и показать полезность применения разработанных показателей на практике.

Ключевые слова: задача коммивояжера, генетический алгоритм, кроссинговер, оператор, популяция, хромосома

Введение

Сложные комбинаторные оптимизационные задачи имеют важное прикладное значение в различных областях человеческой деятельности: экономической, научной, производственной. Справедливо отметить, что решение оптимизационных задач является одним из основных видов научно-инженерной деятельности. Среди множества попыток разработки оптимизационных и поисковых процедур поиска решения подобных задач можно выделить отдельный класс методов оптимизации, которые имитируют реальные процессы, происходящие в природе. К ним относят:

1. Искусственные нейронные сети, которые построены по принципу функционирования сетей нервных клеток центральной нервной системы живого организма.

2. Алгоритм имитации отжига, в основе которого лежит физический процесс, происходящий при кристаллизации вещества.

3. Эволюционные алгоритмы: эволюционное программирование (Л. Дж. Фогель, 1960), генетические алгоритмы (Д. Холланд, 1975), генетическое программирование (Д. Коза, 1992) и другие методы, которые основаны на селекции

лучших элементов в популяции, в соответствии с эволюционной теорией Ч. Дарвина.

Следует отметить, что с развитием генетических и эволюционных алгоритмов в течение последних лет различия между ними постепенно нивелируются. В настоящее время используется новая парадигма исследований на основе генетических алгоритмов и ее различных модификаций [1].

В данной работе мы проанализируем генетический алгоритм для решения общеизвестной задачи комбинаторной оптимизации — задачи коммивояжера (ЗК). В неформальной постановке она заключается в отыскании кратчайшего пути (маршрута), проходящего через все заданные города по одному разу с последующим возвратом в исходный город. Данная задача проста в описании, но имеет высокую комбинаторную сложность. Она возникает в обширном классе приложений, например: поиск оптимального маршрута [2]; задача нахождения маршрута коня, проходящего через все поля шахматной доски по одному разу [3]; задача сбора монет из таксофонов [4]; задача предсказания функций протеинов [5]; задача построения изображения непрерывной линией [6] и многих других. Существует це-

льный класс методов, которые позволяют найти точное решение ЗК, однако они имеют медленную скорость сходимости и на практике для ЗК большой размерности зачастую оказываются малоэффективными. Для решения таких задач, как правило, приходится жертвовать точностью решения и применять эвристические методы. К таким методам относят генетические алгоритмы (ГА). Они обеспечивают высокое качество решений, однако оно напрямую зависит от выбора значений параметров алгоритма. В настоящее время разработано множество различных операторов ГА, комбинация которых также влияет на качество получаемых решений. В связи с этим появляется необходимость оценки перспективности применения ГА с выбранными операторами на предварительном этапе: еще до применения на реальной задаче.

Задачей данного исследования является проведение сравнительного анализа некоторых операторов и эвристик, помогающих выйти из локальных оптимумов. Поскольку наибольшее влияние на ГА оказывает оператор кроссинговера, то перед собой мы поставили задачу изучить влияние одних из лучших на сегодняшний день кроссинговеров на такой показатель, как вырождаемость популяции. Важность данного исследования состоит в том, чтобы понять, на каком этапе генетический алгоритм приводит к деградации популяции. В связи с этим возникла необходимость сравнивать хромосомы и определять степень их различия. Для сравнения хромосом в популяции будем использовать понятие из теории информации и компьютерной лингвистики — расстояние Дамерау—Левенштейна.

Новизна подхода состоит в том, чтобы использовать выведенные на основе расстояния Дамерау—Левенштейна показатели, которые могут быть рассчитаны в общем случае на любом этапе в блоке эволюционной адаптации. Это позволяет эффективно подбирать параметры генетического алгоритма и вероятность применения генетических операторов, следовательно, сократить время поиска локально-оптимального решения. **Принципиальное отличие** предложенного метода заключается в том, что авторы отошли от традиционной схемы решения ЗК с помощью ГА (практическая задача → модель → генетический алгоритм → реализация → анализ полученных результатов), а реализовали новый подход, который

позволяет с помощью показателей, характеризующих степень разнообразия особей в популяции, анализировать любой генетический алгоритм с использованием известных бенчмарков еще до начала применения его на практической задаче.

Постановка задачи и генетический алгоритм для решения ЗК

Математически классическая ЗК выглядит следующим образом: дано множество W , где w_1, w_2, \dots, w_n — города, которые должен посетить коммивояжер. И даны расстояния между каждой парой городов: $d_{ij} = d(w_i, w_j) | i, j = 1, \dots, n$. Необходимо найти минимальное значение: $\sum_{i=1}^{n-1} d_{\varphi(i), \varphi(i+1)} + d_{\varphi(n), \varphi(1)}$ среди всех $(n-1)!/2$ перестановок φ . Также в задаче накладываются определенные условия: неотрицательности $d_{ij} \geq 0$; симметричности $d_{ij} = d_{ji}$; запрет на петли $d_{ii} = \infty$, $i, j = \overline{1, n}$, и удовлетворения неравенству треугольника: $d_{ij} + d_{jp} \geq d_{ip}$, $i, j, p = \overline{1, n} | i \neq j, i \neq p, j \neq k$ [7].

В терминах теории графов моделью классической ЗК является реберно-взвешенный граф $G = (K, T)$, в котором необходимо найти гамильтонов цикл (замкнутый маршрут, который проходит все вершины графа по одному разу) минимальным значением суммарного числа расстояний. При этом вершины графа k_i , $i = 1, \dots, N$ — это города, которые необходимо посетить коммивояжеру, а ребра $t_{i,j}$, $i = 1, \dots, N$, $i \neq j$ — расстояния между городами [7].

В данной работе мы рассмотрим симметричную ЗК.

Генетические алгоритмы для решения ЗК

Генетические алгоритмы относятся к классу вероятностных вычислений, которые моделируют механизмы естественного отбора и эволюции в природе. В данных алгоритмах существуют такие понятия, как популяция, функция приспособленности, генетические операторы [8]. Фундаментальные основы ГА представлены во многих статьях и книгах, опубликованных за последние годы по эволюционной оптимизации *NP*-сложных задач из различных прикладных областей, таких как биология, микроэлектроника, криптоанализ, медицина, телекоммуникации и др. Пример

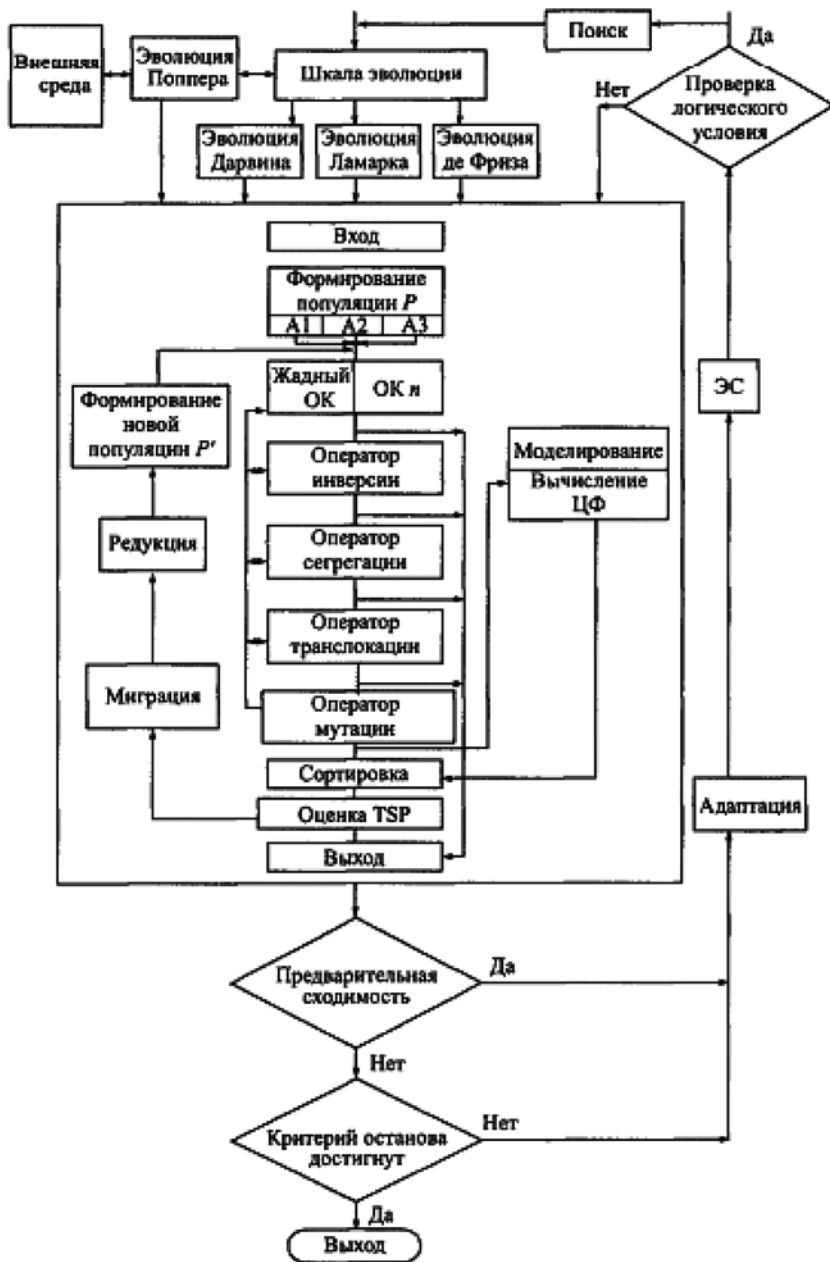


Рис. 1. Структурная схема генетического алгоритма решения задачи коммивояжера

структурной схемы работы ГА при решении ЗК приведен на рис. 1 [8].

Приведенный генетический алгоритм позволяет найти множество альтернативных решений и работает по принципу "выживает сильнейший". Каждое альтернативное решение в ГА соответствует какому-либо гамильтонову циклу (хромосоме). В первом поколении все хромосомы генерируются случайно. Далее ГА начинает формировать новую популяцию. Как правило, размер популяции постоянен на всем этапе работы алгоритма. Процесс репродукции

начинается с наиболее важного оператора — кроссинговера.

Оператор кроссинговера — это языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков [9].

Структура оператора кроссинговера в основном определяет эффективность работы всего генетического алгоритма. В настоящее время разработано множество операторов кроссинговера. По результатам исследования в работе [10] наиболее эффективными по скорости сходимости считаются следующие кроссинговеры: Genetic edge recombination crossover (ERX), Partially-mapped crossover (PMX) и Position based crossover (POS). Для изучения эффективности генетического алгоритма нами был выбран кроссинговер PMX. Также в наш сравнительный анализ включен еще один эффективный кроссинговер — Edge assembly crossover (EAX-1AB), который по сравнению с упомянутыми кроссинговерами был разработан совсем недавно — в 2006 г. японским ученым Yuichi Nagata [11].

Partially-mapped crossover (PMX) — оператор кроссинговера, который был предложен Д. Голдбергом и Р. Линглом в 1985 г. [12]. Суть данного кроссинговера заключается в том, что часть хромосомы одного родителя

второго родителя, все оставшиеся гены соответственно в хромосоме изменяются, образуя тем самым первого потомка. Проводится та же операция замены части хромосомы второго родителя на часть хромосомы 1-го и изменение оставшихся генов, формируя тем самым второго потомка.

Edge assembly crossover (EAX). Алгоритм данного кроссинговера работает следующим образом.

1. Пара родительских особей обозначаются как маршрут *A* и маршрут *B*. Граф T_{AB} получается путем наложения двух маршрутов: *A* и *B*.

2. Разделить ребра графа T_{AB} на циклы, которые представляют собой замкнутые циклы. В циклы поочередно входят ребра то из маршрута A , то из B .

3. Сформировать множество K , представляющее собой любую комбинацию из AB -циклов. Единственное ограничение: AB -циклы, сформированные из двух ребер, в построении множества K не участвуют.

4. Создается промежуточное решение: из множества K удаляются ребра, соответствующие маршруту A , и добавляются ребра из маршрута B .

5. Модифицируем промежуточное решение для того чтобы получить допустимое решение ЗК. Если два цикла связаны, то эвристически выбирается одно ребро из каждого цикла и удаляется, по такому же принципу добавляются два ребра для их соединения. На рис. 2 показан пример работы кроссингвера EAX [11].

Этот этап имеет наибольшую вычислительную сложность в данном алгоритме. Разберем этот шаг подробнее:

5-1. Пусть U_i ($i = 1, \dots, k$) — множество ребер, включенных в i -й подмаршрут, где k — число подмаршрутов в промежуточном решении.

5-2. Выбрать самый маленький подмаршрут из $|U_i|$ ($i = 1, \dots, k$), где $|U_i|$ — число ребер в U_i . Пусть U_r — выбранный подмаршрут.

5-3. Найти пару ребер, $e \in U_r$ и $e' \in U_j$ ($j \neq r$), так что он минимизирует $\{-w(e) - w(e') + w(e'') + w(e''')\}$, где e'' и e''' определяются таким образом, чтобы связывать два подмаршрута. Пусть U_s — подмаршрут, содержащий ребро e' . U_r и U_s объединяются следующим образом: $U_r := (U_r \cup U_s - \{e, e'\}) \cup \{e'', e'''\}$ и пустого U_s . $U_s := U_k$ и вычитаем 1 из k .

5-4. Если k равно 1, U_1 является допустимым маршрутом, то завершить, иначе переходим к п. 5-2.

В наших расчетах будет участвовать усовершенствованный кроссингер EAX-1AB, поскольку он от 2 до 7 раз (в зависимости от параметров) быстрее EAX, и доказано на различных бенчмарках, что его можно эффективно применять на графе до 5915 вершин [11].

Отличие EAX-1AB от EAX состоит в следующем:

1. Множество K формируется только из одного цикла AB , при этом промежуточное решение имеет тенденцию быть сходным с маршрутом A , поскольку потомки генерируются путем удаления небольшого числа ребер из маршрута A и добавления того же числа ребер из B соответственно.

2. С помощью специальной эвристики сокращена вычислительная сложность п. 5 алгоритма EAX, что несомненно является преимуществом модифицированного кроссингвера, поскольку п. 5 алгоритма имеет наибольшее влияние на вычислительную сложность всего генетического алгоритма.

На рис. 1 представлены операторы инверсии, сегрегации, транслокации и мутации. В основном они служат для увеличения разнообразия популяции. В связи с этим имеет смысл рассчитывать показатель степени деградации популяции после работы кроссингвера. Далее эту информацию полезно использовать для настройки параметров следующих операторов.

Расстояние Дамерау—Левенштейна [13, 14] в рассматриваемом генетическом алгоритме будет участвовать для определения степени

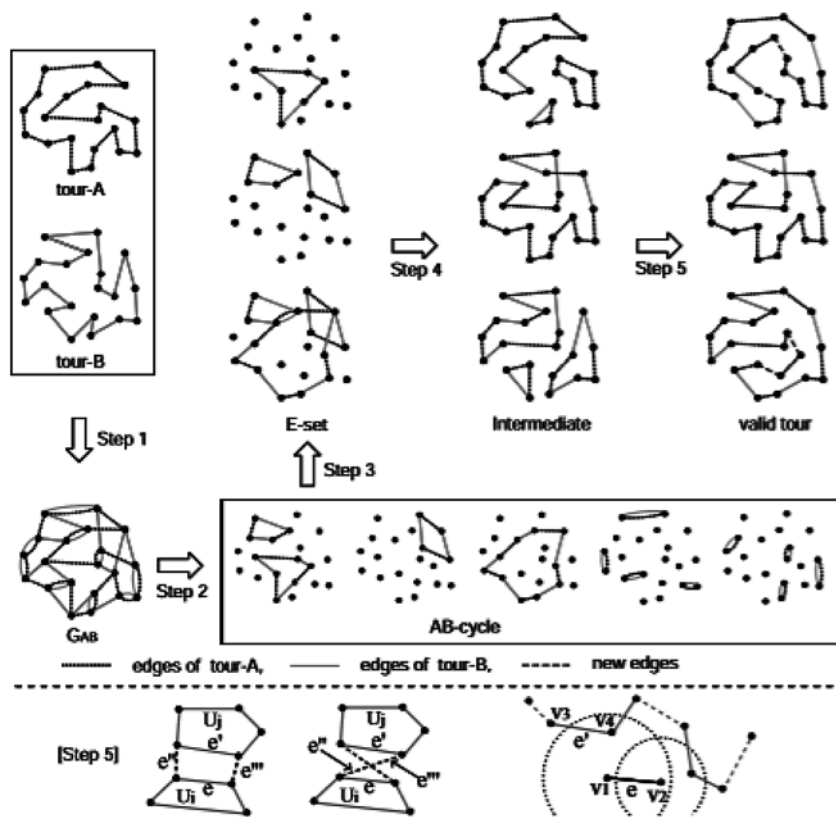


Рис. 2. Пример работы кроссингвера EAX

родства хромосом в популяциях. Данное расстояние характеризует минимальное число операций вставки, удаления, замены одного символа и транспозиции двух соседних символов, оно необходимо для того, чтобы перевести одну строку в другую. Очевидно, что чем меньше это расстояние в генетическом алгоритме между хромосомами, тем более родственными являются рассматриваемые хромосомы.

В общем случае расстояние Дамерау—Левенштейна может быть определено по рекуррентной формуле:

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i & ; j = 0, i > 0 \\ j & ; i = 0, j > 0 \\ \min(\\ \quad D(i, j - 1) + 1, \\ \quad D(i - 1, j) + 1, & ; j > 0, i > 0 \\ \quad D(i - 1, j - 1) + k(T_1[i], T_2[j]) \\ \quad) \end{cases},$$

где T_1 и T_2 — рассматриваемые строки.

Для оценки перспективности применения генетического алгоритма выведем следующие показатели:

Среднее значение различия хромосомы, соответствующей оптимальному решению в популяции, с остальными хромосомами в популяции:

$$Aver_{opt} = \sum_{j=1}^n d_{opt,j} / n,$$

где $j \neq opt$, n — размер популяции; $d_{opt,j}$ — расстояние Дамерау—Левенштейна между оптимальным решением и j -й хромосомой.

Среднее значение коэффициента различия хромосом в популяции:

$$Aver_{pop} = \left(\sum_{i=1}^n \sum_{j=1}^n d_{i,j} \right) / n,$$

где $i \neq j$, n — размер популяции; $d_{i,j}$ — расстояние Дамерау—Левенштейна между i -й и j -й хромосомами.

В общем случае тенденция уменьшения приведенных выше показателей свидетельствует о снижении разнообразия в популяции, увеличения риска попадания в локальный оптимум и, как следствие, необходимости изменения настроек ГА. Данные показатели могут рассматриваться на любом этапе эволю-

ционной адаптации. В данном исследовании в модифицированном алгоритме $Aver_{opt}$ и $Aver_{pop}$ применяют после оператора кроссинговера.

Разработка программы и экспериментальная часть

Генетический алгоритм решения задачи коммивояжера был запрограммирован на языке Java на платформе Windows.

При этом все исследования проводили на компьютере типа Intel Core i7 CPU 2.6 GHz и ОЗУ размером 16 Гбайт.

Настройки генетического алгоритма:

- 1) кроссинговеры: EAX-1, PMX;
- 2) способ выбора начальной популяции — алгоритм ближайшего соседа из каждой вершины графа;
- 3) размер начальной популяции — n , где n — число вершин;
- 4) за каждую итерацию выполняется $n/2$ кроссинговеров, родители для которых выбираются случайным образом из текущей популяции;
- 5) жадный оператор селекции;
- 6) число итераций — 50;
- 7) показатели $Aver_{opt}$ и $Aver_{pop}$ рассчитывают после оператора кроссинговера.

Эксперименты были проведены на бенчмарке att-48.

Результаты разработанной программы представлены на рис. 3, 4. В приведенных ниже графиках по оси x номер поколения, по оси y — значение коэффициента $Aver_{opt}$ или $Aver_{pop}$.

Как видно из графика рис. 3, ГА с кроссинговером PMX уже начиная с 5-й итерации,

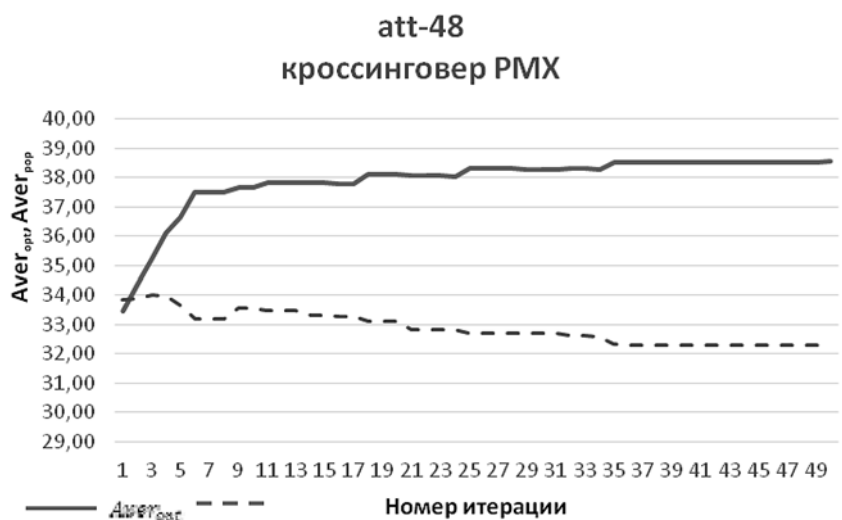


Рис. 3. Показатели $Aver_{opt}$ и $Aver_{pop}$ с кроссинговером PMX

att-48 кроссинговер EAX



Рис. 4. Показатели $Aver_{opt}$ и $Aver_{pop}$ с кроссинговером EAX

несмотря на тенденцию формирования разнообразных решений (показатель $Aver_{opt}$ растет), показывает постоянное снижение в разнообразии популяции (показатель $Aver_{pop}$ снижается).

На рис. 4 видно, что с 29-й по 36-ю итерацию алгоритм показывает рост разнообразия популяции, с 36-й по 43-ю временно происходит обратная тенденция, но затем снова наблюдается рост в разнообразии популяции.

Заключение

Данная работа относится к области искусственного интеллекта. ЗК активно используются в транспортных задачах, интеллектуальном проектировании, при решении задач поисковой оптимизации. Разработка методов решения ЗК по-прежнему остается актуальной задачей.

В результате исследования был разработан модифицированный генетический алгоритм, который показывает степень разнообразия популяции. Данная информация важна, поскольку деградация популяции может приводить к преждевременной сходимости алгоритма. Новизна подхода заключается в использовании специальных показателей: $Aver_{opt}$ и $Aver_{pop}$. Также было разработано специальное программное обеспечение. Результаты исследования на известной бенчмарке для ЗК с 48 городами (att-48) показали, что ГА с кроссинговером PMX имеет тенденцию к деградации популяции с каждым следующим поколением, начиная с 5-й итерации. Таким образом, ГА с PMX не является перспективным для запуска на больших объемах данных в исходном виде. Кроссинговер

EAX не имеет резкой тенденции к вырождаемости популяции, в связи с этим ГА с EAX не требует обязательного введения многочисленных дополнительных операторов для создания разнообразия в популяции, следовательно, он стабильно работает с низкими производительными затратами. Оптимальность (способность алгоритма выбирать "наилучший маршрут") ГА с EAX также выше, чем ГА с PMX, следовательно, можно считать ГА с EAX более перспективным.

В общем случае показатели $Aver_{opt}$ и $Aver_{pop}$ могут служить полезным критерием для оценки генетического алгоритма, а именно: степени вырождаемости популяций и дальнейшей настройки его операторов. Важно отметить, что показатели $Aver_{opt}$ и $Aver_{pop}$ также увеличивают гибкость ГА. При этом исходный алгоритм целесообразно проверять с помощью показателей $Aver_{opt}$ и $Aver_{pop}$ на небольших объемах данных перед использованием на реальной задаче в любом блоке эволюционной адаптации.

Работа выполнена за счет частичного финансирования по гранту РФФИ № 18-0700050 и ГЗ № 25537.2017/6.7.

Список литературы

1. Гладков Л. А., Курейчик В. В., Курейчик В. М., Сорочколетов П. В. Биоспирированные методы в оптимизации. М.: Физматлит, 2009. 384 с.
2. Ismail Z., Ibrahim W. R. W. Traveling Salesman Approach for Solving Petrol Distribution Using Simulated Annealing // American Journal of Applied Sciences, 2008. Vol. 5, Is. 11. P. 1543–1546.
3. McKay B. D. Knight's tours of an 8×8 chessboard // Technical Report. Department of Computer Science, Australian National University. Canberra, 1997. № TR-CS-97-03. 4 p.
4. Lenstra J., Kan A. R. Some simple applications of the travelling salesman problem // Operational Research Quarterly, 1975. Vol. 26, N. 4. Part 1. P. 717–733.
5. Johnson O., Liu J. A traveling salesman approach for predicting protein functions // Source Code for Biology and Medicine. 2006. Vol. 1, N. 3. P. 1–7.
6. Borch R., Herman A. Continuous Line Drawings via the Traveling Salesman Problem // Operations Research Letters. 2004. Vol. 32. P. 302–303.
7. Колесников А. В., Кириков И. А., Листопад С. В., Румовская С. Б., Доманицкий А. А. Решение сложных задач коммивояжера методом функциональных гибридных интеллектуальных систем / Под ред. А. В. Колесникова. М.: ИПИ РАН, 2011.
8. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. 2-е изд., испр. и доп. М.: Физматлит, 2006. 320 с.
9. Курейчик В. М. Модифицированные генетические операторы // Известия Южного Федерального Университета. 2009. № 12 (101). С. 7–10.

10. Starkweather T., McDaniel S., Mathias K., Whitley C., Whitley D. A Comparison of Genetic Sequencing Operators // Proceedings on the Fourth International Conference on Genetic Algorithms, edited by Belew R. and Booker L., Los Altos, CA: Morgan Kaufmann Publishers, 1991. P. 69–76.

11. Nagata Y. Fast EAX Algorithm Considering Population Diversity for Traveling Salesman Problems // Evolutionary Computation in Combinatorial Optimization 6th European Conference, EvoCOP 2006 Budapest, Hungary, 2006. P. 171–182.

12. Goldberg D. E., Lingle R. Alleles, Loci, and the Traveling Salesman Problem // Proceedings of the First International Conference on Genetic Algorithms and Their Application / ed. by Grefenstette J. Lawrence Erlbaum Associates, Hillsdale, NJ, 1985. P. 154–159.

13. Damerou F. J. A technique for computer detection and correction of spelling errors // Communications of the ACM, ACM. 1964. Vol. 7, N. 3. P. 171–176.

14. Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии Наук СССР, 1965. Т. 163, № 4. С. 845–848.

V. M. Kureichik, Professor, CAD Department Senior Researcher, Vmkureychik@sfedu.ru,

J. A. Logunova, Graduate Student, Julia1000@yandex.ru,

Federal State-Owned Autonomy Educational Establishment of Higher Education

Southern Federal University, Taganrog, Russia

The Genetic Algorithm Application Prospects Analysis for the Traveling Salesman Problem Solution

This work belongs to the artificial intelligence field. The Traveling salesman problem (TSP) is considered in this article, which is actively used in transport tasks, intellectual design, in search optimization solving problems. The development of new solving methods for TSP is still an urgent task. It is NP-hard problem. Usually, the algorithm time complexity finding the exact solution is a factorial or exponential dependence on the input data. It is possible to allocate a separate class of optimization methods among a lot of search procedures for TSP solving, which imitates real nature processes: genetic and bio-inspired algorithms. These algorithms are used for large dimensions tasks. In this case the optimal solution cannot be obtained over the acceptable time. In this situation you need sacrifice the accuracy of the solution and find a good solution in a relatively short time. The solution quality depends greatly on choice of algorithm parameter values. The genetic algorithm can lead to the population degradation in the wrong choice of operator, and, as a result, the obtained solution is far from optimal. In this regard, we propose to use some special indicators for investigating the prospects of using the genetic algorithm. This indicators show the population diversity degree. Also they can be used to algorithm parameters adjust. Specially developed software allowed testing the GA on the well-known benchmark for TSP with 48 cities (att-48) and showed the usefulness of the developed indicators in practice.

Keywords: traveling salesman problem, genetic algorithm, crossover, population, chromosome, artificial intelligence, Dammerau-Lowenstein distance

DOI: 10.17587/it.24.691-697

References

1. Gladkov L. A., Kureichik V. V., Kureichik V. M., Sorokoletov P. V. *Bioinspirirovannye metody v optimizacii* [Bioinspired methods in optimization], Moscow, Fizmatlit, 2009, 384 p.

2. Ismail Z., Ibrahim W. R. W. Traveling Salesman Approach for Solving Petrol Distribution Using Simulated Annealing, *American Journal of Applied Sciences*, 2008, vol. 5, is. 11, pp. 1543–1546.

3. McKay B. D. *Knight's tours of an 8×8 chessboard*, Technical Report. Department of Computer Science, Australian National University. Canberra, 1997. № TR-CS-97-03. 4 p.

4. Lenstra J., Kan A. R. Some simple applications of the travelling salesman problem, *Operational Research Quarterly*, 1975, vol. 26, no. 4, part 1, pp. 717–733.

5. Johnson O., Liu J. A traveling salesman approach for predicting protein functions, *Source Code for Biology and Medicine*, 2006, vol. 1, no. 3, pp. 1–7.

6. Borch R., Herman A. Continuous Line Drawings via the Traveling Salesman Problem, *Operations Research Letters*, 2004, vol. 32, pp. 302–303.

7. Kolesnikov A. V., Kirikov I. A., Listopad S. V., Rumovskaya S. B., Domanitsky A. A. *Reshenie slozhnykh zadach kommi-voyazhera metodami funkcional'nykh gibridnykh intellektual'nykh sistem* [Solving the complex traveling salesman problems with the method of functional hybrid intelligent systems], ed. A. V. Kolesnikov, Moscow, IPI RAS, 2011.

8. Gladkov L. A., Kureichik V. V., Kureichik V. M. *Geneticheskie algoritmy* [Genetic algorithms], 2nd ed., rev. and add. Moscow, Fizmatlit, 2006, 320 p.

9. Kureichik V. M. *Modifitsirovannye geneticheskie operatory* [Modified genetic operators], *Proceedings of the Southern Federal University*, 2009, no. 12, pp. 7–10.

10. Starkweather T., McDaniel S., Mathias K., Whitley C., Whitley D. A Comparison of Genetic Sequencing Operators, *Proceedings on the Fourth International Conference on Genetic Algorithms*, edited by Belew R. and Booker L., Los Altos, CA, Morgan Kaufmann Publishers, 1991, pp. 69–76.

11. Nagata Y. Fast EAX Algorithm Considering Population Diversity for Traveling Salesman Problems, *Evolutionary Computation in Combinatorial Optimization 6th European Conference, EvoCOP 2006 Budapest, Hungary*, 2006, pp. 171–182.

12. Goldberg D. E., Lingle R. Alleles, Loci, and the Traveling Salesman Problem, *Proceedings of the First International Conference on Genetic Algorithms and Their Application*, edited by Grefenstette J., Lawrence Erlbaum Associates, Hillsdale, NJ, 1985, pp. 154–159.

13. Damerou F. J. A technique for computer detection and correction of spelling errors, *Communications of the ACM, ACM, March 1964, 7 (3)*, pp. 171–176.

14. Levenshtein V. I. *Dvoichnye kody s ispravleniem vypadenij, vstavok i zameweniy simvolov* [Binary codes with correction of fallouts, insertions and substitutions of symbols], *Reports of the Academies of Sciences of the USSR*, 1965, vol. 163.4, pp. 845–848.

М. В. Ульянов, д-р техн. наук, проф., вед. науч. сотр., проф., muljanov@mail.ru,
Институт проблем управления им. В. А. Трапезникова РАН,
Московский государственный университет им. М. В. Ломоносова,
М. И. Фомичев, магистрант, michan94@yandex.ru,
Национальный исследовательский университет "Высшая школа экономики"

Подходы к организации поискового дерева решений в методе ветвей и границ для асимметричной задачи коммивояжера¹

Повышение временной эффективности программных реализаций метода ветвей и границ для асимметричной задачи коммивояжера может быть достигнуто как за счет выбора наиболее приемлемой структуры данных, обеспечивающей эффективные по времени операции с листьями поискового дерева решений, так и за счет использования дополнительной памяти для хранения усеченных матриц в листьях поискового дерева решений. Дополнительно могут быть предложены и различные подходы к хранению и обработке матриц, соответствующих листьям поискового дерева. Такое исследование должно опираться на особенности операций с деревьями и матрицами, порожденные спецификой метода ветвей и границ. Описанию различных подходов к организации, хранению и доступу к элементам поискового дерева решений в совокупности со способами хранения матриц в листьях такого дерева и посвящена настоящая статья.

Ключевые слова: метод ветвей и границ, асимметричная задача коммивояжера, поисковое дерево решений, структуры данных, структуры хранения усеченных матриц

Введение

Достаточно широкий спектр задач, порожденных практически постановками в области информационных технологий, логистики и бизнес-информатики сводится к классической задаче коммивояжера (TSP). При этом некоторые постановки порождают асимметричную задачу коммивояжера (ATSP), например, стоимости перелетов из одного города в другой и обратно, как правило, не совпадают. При решении ATSP перед исследователями стоит задача выбора между эвристическими и точными методами решения, связанная как с размерностью исходной задачи, так и с ограничениями по допустимому времени решения. Обилие эвристических методов не означает отказа от возможности получения точных решений ATSP для небольших значений размерности. Далее объектом исследования будет точный алгоритм решения ATSP, реализующий классический метод ветвей и границ [1].

Экспериментальные данные свидетельствуют об экспоненциальном росте сложности в среднем в зависимости от размерности задачи и, следовательно, среднего времени получения

точного решения при использовании метода ветвей и границ для индивидуальной ATSP [2, 3]. Тем не менее при реализации на современных компьютерах метод позволяет получить точное решение для задач, содержащих не более 60...70 вершин, за приемлемое время, при этом очевидный интерес представляет задача повышения его временной эффективности.

Для некоторых типов задач и решающих их алгоритмов повышение временной эффективности может быть получено за счет использования дополнительного доступного объема памяти. Такая ситуация возникает в задачах, допускающих замену повторных вычислений на хранение ранее полученных промежуточных результатов. Этот подход может быть использован и для алгоритмов, реализующих метод ветвей и границ для задачи коммивояжера.

Ресурсная эффективность различных реализаций метода ветвей и границ для асимметричной задачи коммивояжера зависит в том числе и от способов организации и работы с поисковым деревом решений, порождаемым этим методом. Классическая дилемма "время—память" реализуется здесь либо в виде варианта хранения усеченных матриц в вершинах дерева, что приводит к сокращению трудоемкости при дополнительных емкостных затратах, либо как перевычисление матрицы для текущей верши-

¹Работа выполнена при поддержке Гранта РФФИ 18-07-00656.

ны, что ведет к увеличению трудоемкости при экономии памяти.

Повышение временной эффективности может быть, очевидно, достигнуто за счет хранения матриц в листьях поискового дерева решений. Такой подход требует детального исследования в целях выбора структуры данных, обеспечивающей эффективные по времени операции с листьями поискового дерева решений. Вместе с тем могут быть предложены и различные подходы к хранению и обработке матриц, соответствующих листьям поискового дерева. Для формулировки рекомендаций по выбору структуры данных и способа хранения матриц необходимо изучение особенностей операций с деревом и матрицами, порожденных спецификой метода ветвей и границ. Таким образом, предметом исследования данной статьи являются структуры данных, предназначенные для хранения поискового дерева решений, и способы хранения матриц в листьях этого дерева. Полученные при реализации этих предложений экспериментальные данные позволят дать оценки размерности задач, решаемых за приемлемое время при соответствующих затратах дополнительной памяти, и сформулировать рекомендации при выборе метода ветвей и границ как точного метода решения ATSP в практических задачах информационных технологий, логистики и бизнес-информатики.

Применение метода ветвей и границ к задаче коммивояжера

Идея применения метода ветвей и границ к ATSP, предложенная в работе [1], предполагает разделение множества допустимых решений на подмножества в целях дальнейшего сокращения перебора, это разделение называется процедурой ветвления. Под допустимым решением понимается некоторый гамильтонов цикл в полном графе, называемый туром, стоимость которого определяется суммой весов входящих в него дуг. С каждым таким подмножеством должна быть связана оценка, обеспечивающая отсечение тех подмножеств, которые заведомо не содержат оптимального решения; процедура ее построения называется процедурой вычисления границ. Тем самым метод приводит к исследованию древовидной модели пространства решений. В ATSP таким исходным множеством является множество всех туров коммивояжера, на котором минимизируется целевой функционал, задающий стоимость тура. Для постро-

ения алгоритма необходимо разработать две основные процедуры — процедуру ветвления и процедуру вычисления границ.

Авторы в работе [1] предлагают следующую процедуру ветвления. Построение поискового дерева решений начинается с корня, который будет соответствовать множеству "всех возможных туров коммивояжера", т. е. корень дерева представляет множество всех $(n - 1)!$ возможных туров в задаче с n городами. Ветви, выходящие из текущей вершины, определяются выбором дуги (для ATSP). Идея авторов алгоритма состоит в том, чтобы разделить текущее множество туров на два множества: одно — которое, весьма вероятно, содержит оптимальный тур, и другое — которое, вероятно, этого тура не содержит. Для этого предлагается специальный алгоритм выбора дуги, которая, вполне вероятно, входит в оптимальный тур. После выбора дуги текущее множество туров разделяется на два подмножества, в одно из которых входят все туры из текущего множества, содержащие выбранную дугу, т. е. проходящие через нее, а в другое — туры, не содержащие эту дугу. Отметим, что идея авторов алгоритма очень перспективна — если так организовать процесс ветвления, что на каждом шаге выбирается "правильное" ребро, то весь процесс будет завершен за n шагов.

С каждой вершиной поискового дерева связывается нижняя граница стоимости любого тура из подмножества, связанного с этой вершиной. Очевидно, что задача состоит в получении как можно более точных нижних границ. Причина этого следующая. Предположим, что уже получен конкретный тур T со стоимостью $C(T)$. Если нижняя граница, связанная с подмножеством туров, представленных некоторой другой вершиной поискового дерева, больше, чем $C(T)$, то до конца процесса поиска не нужно рассматривать эту и все следующие за ней вершины. В реализации это приводит к усечению поискового дерева решений путем отбрасывания всех листьев поискового дерева, имеющих стоимость большую, чем $C(T)$. Подробное изложение других этапов метода можно найти, например, в работах [4, 5].

Схема алгоритма метода ветвей и границ для задачи коммивояжера

В целях дальнейшего исследования приведем схему алгоритма метода ветвей и границ

для ATSP. Пусть X — текущая вершина поискового дерева, а (k, l) — дуга, по которой происходит ветвление. Обозначим Y и \bar{Y} вершины дерева, порожденные из X . Вершина Y обозначает подмножество туров из X , проходящих через дугу (k, l) , а подмножество \bar{Y} — подмножество туров, не проходящих через дугу (k, l) . Вычисленные нижние границы для подмно-

жеств Y и \bar{Y} обозначим $w(Y)$ и $w(\bar{Y})$ соответственно. Пусть z_0 — самый дешевый тур, известный алгоритму в данный момент. Мы предполагаем, что в вершинах поискового дерева решений в том или ином виде хранятся соответствующие усеченные матрицы.

Приведенная ниже схема алгоритма соответствует изложению метода в работе [4].

A(C, n) Схема алгоритма МВГ для задачи коммивояжера

(n — размерность, C — матрица стоимостей)

1. Инициализация.

2. Приведение матрицы стоимостей C — вычисление нижней границы всех туров $w(R)$.

3. Установка корня поискового дерева решений $X = R$ и $w(X)$.

While ($w(X) < z_0$)

begin

4. Выбор ребра ветвления (k, l) .

5. Процесс ветвления. Создание вершины \bar{Y} и вычисление $w(\bar{Y})$.

6. Процесс ветвления. Создание вершины Y и вычисление $w(Y)$.

If (размер матрицы стоимостей в вершине $Y = 2$)

then

begin

7. Наилучший тур в вершине Y , вычисление $w(Y)$

If ($w(Y) < z_0$)

then

begin

$z_0 = w(Y)$ (запоминаем текущий лучший тур)

end

end

8. Выбор следующей вершины поискового дерева решений и установка X

9. Чтение матрицы, соответствующей выбранной вершине X , из структуры хранения поискового дерева решений.

end (while $w(X) < z_0$)

Оптимальное (точное) решение со стоимостью z_0 найдено

End.

Структуры данных для обслуживания поискового дерева решений

Сформулируем требования, предъявляемые к структуре данных, связанные с обеспечением временной эффективности. Для этого обратим внимание на некоторые этапы приведенной выше схемы и некоторые экспериментальные результаты:

- экспериментальные результаты работ [2, 3] свидетельствуют о том, что общее число порожденных вершин растет экспоненциально с размерностью задачи и для ATSP с 50 вершинами в худшем случае может достигать сотен миллионов;
- при выборе на этапе 8 следующей текущей вершины выбирается лист текущего поиско-

вого дерева решений, имеющий минимальную границу. За все время работы алгоритма число обращений к этапу 8 линейно зависит от общего числа порожденных вершин. Содержательно это означает, что структура данных, поддерживающая хранение дерева, должна обеспечить быстрый поиск листа дерева, имеющего минимальную границу;

- кроме того, на этапе 8, помимо обеспечения быстрого поиска листа дерева решений, также необходимо удалить этот лист из структуры для обслуживания поискового дерева решений, так как на этапах 5 и 6 из этой вершины будет обеспечено ветвление, и, как результат, данный лист становится внутренним (и не активным) узлом поискового дерева решений;

- поскольку на этапе 5 необходимо добавить вершину \bar{Y} в поисковое дерево решений (при условии, что $w(\bar{Y}) < z_0$), структура данных должна поддерживать операцию вставки элемента;
- в связи с тем что нижние оценки для разных вершин поискового дерева решений могут быть равны между собой, структура данных должна иметь возможность хранить несколько одинаковых элементов (несколько одинаковых ключей);
- так как на этапе 7 может быть найден тур, который является на данный момент работы алгоритма наилучшим из известных (или даже оптимальным), то необходимо удалить из структуры данных все листья дерева решений, которые имеют оценки, большие или равные стоимости найденного тура.

Учитывая вышеизложенные особенности структуры данных, можно сделать вывод, что необходимая структура данных должна представлять такой абстрактный тип данных, как очередь с приоритетом [6].

Согласно работам [6, 7] следующие структуры данных следует рассмотреть как реализацию очереди с приоритетом:

- ▲ упорядоченный список;
- ▲ двоичная (бинарная) куча;
- ▲ самобалансирующиеся двоичные (бинарные) деревья поиска (например, красно-черные деревья и AVL-деревья).

Эти структуры данных поддерживают все необходимые операции и обладают нужными свойствами. Кроме того, сложность операции по получению минимального элемента может быть сведена к $O(1)$, если кешировать минимальный элемент. Более того, эти структуры данных имеют линейную оценку требуемой памяти $O(n)$, где n — число хранимых элементов. Однако, на этом сходство этих структур данных заканчивается. Сложность операции по удалению минимального элемента для упорядоченного списка — $O(n)$, в то время как сложность аналогичной операции для самобалансирующегося двоичного дерева и двоичной кучи — $O(\log n)$. Вместе с тем сложность операции удаления элементов, чей ключ больше заданного числа, — $O(n)$, в то время как для самобалансирующегося двоичного дерева и двоичной кучи — $O(n \log n)$. Что касается операции вставки, то в худших случаях стоимость операции вставки для упорядоченного списка

и двоичной кучи — $O(n)$, а для самобалансирующегося двоичного дерева — $O(\log n)$.

Вышеизложенные оценки сложности представлены в табл. 1.

Таблица 1

Сложность операций над структурами данных в худшем случае

Операция	Упорядоченный список	Двоичная куча	Самобалансирующееся двоичное дерево
Удаление наименьшего элемента	$O(n)$	$O(\log n)$	$O(\log n)$
Получение наименьшего элемента	$O(1)$	$O(1)$	$O(1)$
Вставка элемента	$O(n)$	$O(n)$	$O(\log n)$
Удаление элементов, чьи ключи больше, чем заданное значение	$O(n)$	$O(n \log n)$	$O(n \log n)$

Приведенные оценки сложности операций в худших случаях для рассмотренных структур данных не позволяют сделать однозначные выводы об их эффективности. Аргументированный выбор может быть сделан только на основе тщательного экспериментального исследования.

Способы хранения матриц в вершинах поискового дерева

Варианты решения в данном случае относятся к этапу 9 общей схемы. В этот момент новая текущая вершина поискового дерева X уже выбрана, и задача данного этапа — получить матрицу стоимостей, соответствующую вершине X .

Основная особенность состоит в том, что метод предполагает усечение матрицы при построении вершины, соответствующей подмножеству Y , путем исключения строки и столбца, соответствующих выбранной дуге ветвления. При этом могут быть предложены различные подходы к хранению матриц, а именно:

- подход, предполагающий прямое хранение матрицы (без усечения), при этом элементы исключенной строки и исключенного столбца помечаются специальным значением. Это приводит к увеличению (по оценке в два раза) требуемого объема памяти при возможном росте временной эффективности;

- подход, предполагающий хранение усеченных матриц, при котором необходимо вводить два специальных массива переиндексации (для строк и столбцов), элементы которых содержат "настоящие", т. е. не усеченные индексы для строк и столбцов усеченной матрицы. При этом мы сокращаем объем требуемой памяти за счет дополнительных операций, связанных с переиндексацией;
- комбинированный подход, предполагающий усеченное хранение и восстановление полной матрицы по усеченному описанию, с которой далее проводятся вычисления на этапах 4, 5 и 6 общей схемы. Этот подход, оставляя преимущество усеченного хранения (в смысле объема требуемой дополнительной памяти), позволяет сократить временные затраты при условии, что трудоемкость восстановления и обратной свертки полной матрицы окупится отсутствием двойной индексации при обработке. Очевидно, что это произойдет в случае, если объем вычислений на этапах 4, 5 и 6 общей схемы существенно превалирует над дополнительными затратами на восстановление и свертку.

Достаточно трудно сказать, какой из этих подходов является более эффективным с точки зрения времени выполнения, поскольку выбор ребра ветвления предполагает рассмотрение всех нулевых элементов приведенной матрицы [1, 4], число которых варьируется от n до $2n - 1$ в зависимости от особенностей данных. Вместе с тем временная эффективность переиндексации связана с особенностями компилятора и внутренними алгоритмами управления кеш-памятью компьютера, что не может быть аналитически точно описано.

Таким образом, решение о выборе способа хранения может быть принято только на основе анализа экспериментальных данных о временной эффективности.

Предварительные экспериментальные результаты

Для проведения экспериментального исследования был сформирован пул несимметричных матриц стоимостей. Размерность матриц от 20 до 43 с шагом один. Число различных матриц для каждой размерности: 100 000. Элементы матриц (стоимости перехода между городами) генерировались с помощью стандартного равномерного генератора псевдослучай-

ных чисел в интервале (0; 1) с точностью до шестого знака после запятой.

Все эксперименты проводили на персональном компьютере со следующими характеристиками:

- процессор Intel i7 3770K 3800 МГц;
- оперативная память Kingston KHX1600C9D3P1 16 ГБ;
- материнская плата GIGABYTE GA-Z77X-D3H.

В качестве операционной системы использовался дистрибутив Linux CentOS 7.0 с версией ядра 3.10.0-3247.4.4.el7. Для минимизации шумов операционной системы были отключены ненужные для исследования фоновые процессы (например, фаервол), а также у дистрибутива отсутствуют компоненты графического пользовательского интерфейса (управление осуществляется посредством командной строки). Более того, был отключен своппинг, чтобы скорость работы HDD не сказывалась на времени работы реализации.

Реализация всех алгоритмов осуществлялась на языке C++ с использованием библиотеки стандартных шаблонов (STL). Версия компилятора: GNU GCC 4.8.5 20150623. Замер времени работы программных реализаций алгоритмов осуществлялся посредством `chrono`. Для замера используемой памяти использовались счетчики. Наличие погрешности у чисел с плавающей точкой (согласно стандарту IEEE 754-2008) может привести к измененному процессу ветвления. Как показывает практика, хотя процесс ветвления и изменяется, и ошибка может накапливаться, это не влияет на поиск оптимального тура.

В табл. 2 (с шагом два по размерности) и на рис. 1, 2 (см. вторую сторону обложки) показаны результаты экспериментов для метода ветвей и границ с использованием двоичной кучи (ARS_{heap}), упорядоченного списка (ARS_{ord_vector}) и самобалансирующегося двоичного дерева (в качестве представителя данного класса структур данных используется красно-черное дерево [6]) (ARS_{RBT}), кроме того, для сравнения (только на рис. 1) представлены результаты работы реализации метода ветвей и границ без хранения списка вершин, при этом поиск вершины дерева решений с минимальной границей осуществляется посредством обхода дерева решений ($ARS_{traverse_tree}$).

Предварительные экспериментальные результаты показали, что за исключением двоичной кучи, использование которой дает худший

Таблица 2

Среднее затраченное время (в мкс)
для ARS_{heap} , ARS_{ord_vector} , ARS_{RBT} и $ARS_{traverse_tree}$

n	$\bar{t}_{ARS_{heap}}(n)$	$\bar{t}_{ARS_{ord_vector}}(n)$	$\bar{t}_{ARS_{RBT}}(n)$	$\bar{t}_{ARS_{traverse_tree}}(n)$
20	2228	2174	2187	2297
22	4134	4021	4004	4255
24	7243	6976	7000	7440
26	12 258	11 893	11 957	12 827
28	20 917	20 121	20 155	22 440
30	35 197	33 841	33 828	39 710
32	58 333	56 012	56 005	70 502
34	96 740	93 028	92 799	130 210
36	159 891	154 136	153 307	255 872
38	261 227	252 851	250 524	515 642
40	424 082	413 123	406 551	1 120 809
42	680 455	665 432	651 617	—
43	872 727	857 094	835 382	—

Таблица 3

Среднее затраченное время (в мкс) для ARS и $ARS_{matrices_in_leaves}$

n	$\bar{t}_{ARS}(n)$	$\bar{t}_{ARS_{matrices_in_leaves}}(n)$
20	2228	904
21	2993	1174
22	4134	1504
23	5533	1958
24	7243	2549
25	9534	3240
26	12 258	4053
27	16 266	5159
28	20 917	6432
29	27 167	8060
30	35 197	10 163
31	45 411	12 731
32	58 333	15 818
33	74 381	19 559
34	96 740	24 805
35	123 770	30 974
36	159 891	39 161
37	204 587	48 997
38	261 227	61 323
39	333 328	76 778
40	424 082	96 085
41	537 207	119 288
42	680 455	148 712
43	872 727	187 109

результат для всей рассматриваемой выборки, выбор между остальными структурами данных не оказывает существенного влияния на время работы программной реализации алгоритма. При этом упорядоченный список рабо-

тает лучше самобалансирующегося двоичного дерева до $n = 29$, а начиная с $n = 30$ — самобалансирующееся дерево показывает лучший результат. Таким образом, можно рекомендовать использовать до $n = 29$ двоичную кучу, а начиная с $n = 30$ — самобалансирующееся дерево. Это свидетельствует о том, что потребность в операции вставки элемента в очередь с приоритетом растет с размерностью задачи, а сложность этой операции у красно-черного дерева в худшем случае минимальная по сравнению с другими рассматриваемыми структурами данных.

В рамках данного предварительного экспериментального исследования, относительно хранения матриц в вершинах поискового дерева решений, мы показываем сравнение времени работы метода ветвей и границ без использования дополнительной памяти (\bar{t}_{ARS}) и с хранением усеченных матриц в листьях дерева решений ($\bar{t}_{ARS_{matrices_in_leaves}}$). Результаты предварительного экспериментального исследования представлены в табл. 3.

Экспериментальные данные показали, что хранение усеченных матриц стоимостей в листьях поискового дерева решений существенно уменьшает время расчета. Уже начиная с 43 городов, прирост в производительности по сравнению с классической реализацией составляет более чем 2 раза. Вместе с тем объем потребляемой памяти растет экспоненциально с ростом числа городов.

Заключение

В статье исследованы структуры данных, обеспечивающие эффективные по времени операции с листьями поискового дерева решений при решении асимметричной задачи коммивояжера методом ветвей и границ. Предложены различные подходы к хранению и обработке матриц, соответствующих листьям поискового дерева. Приведены предварительные экспериментальные результаты о временной эффективности соответствующих программных реализаций. Полученные результаты позволили предварительно рекомендовать в качестве решения упорядоченный список до $n = 29$ и самобалансирующееся двоичное дерево для больших размерностей для обслуживания поискового дерева решений. Что касается хранения матриц в вершинах поискового дерева решений, то предварительные результаты показали, что

хранение усеченных матриц может существенно сократить время точного решения задачи коммивояжера. На основе полученных результатов можно предварительно прогнозировать размерности задач ATSP, допускающие приемлемое время получения точного решения.

Возможные дальнейшие исследования будут направлены на проведение масштабных экспериментальных исследований в целях вероятностного прогнозирования временной эффективности точного решения индивидуальных асимметричных задач коммивояжера.

Список литературы

1. Little J. D. C., Murty K. G., Sweeney D. W., Karel C. An algorithm for the traveling salesman problem // *Operations Research*. 1963. Vol. 11. P. 972–989.

2. Ульянов М. В., Фомичев М. И., Головешкин В. А., Жукова Г. Н. Оценка параметров распределения логарифма сложности задачи коммивояжера // *Современные информационные технологии и ИТ-образование*. 2017. Т. 13, № 1. С. 19–24.

3. Головешкин В. А., Жукова Г. Н., Ульянов М. В., Фомичев М. И. Использование квантильных коэффициентов асимметрии и эксцесса для оценки сложности решения задачи коммивояжера // *International Journal of Open Information Technologies*. 2016. Т. 4, № 12. С. 7–12.

4. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов. М.: Мир, 1981. 368 с.

5. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. М.: ФИЗМАТЛИТ, 2007. 304 с.

6. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ: Пер. с англ. М.: Издательский дом "Вильямс", 2005. 1296 с.

7. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона / Пер. с англ. Ткачев Ф. В. М.: ДМК Пресс, 2010. 272 с.

M. V. Ulyanov, Leading Researcher, V. A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences, M. V. Lomonosov Moscow State University

M. I. Fomichev, Graduate Student, National Research University Higher School of Economics

Approaches to Design Search Decision Tree in the Branch and Bound Method for the Asymmetric Traveling Salesman Problem

Run time of the Branch and Bound method for solving asymmetric Traveling Salesman Problem can be reduced by using the most appropriate data structure providing time-efficient operations with leaves of the search decision tree and additional memory for storing truncated matrices in leaves of search decision tree. In additional, different approaches for storing and handling matrices, which correspond leaves of search decision tree, can be presented. Such research should be based on the features of operations with search decision tree and matrices, generated by the specificity of the Branch and Bound method. Different approaches of storing and accessing elements of search decision tree and ways to store matrices in leaves of such tree are presented at this paper.

Keywords: branch and bound method, asymmetric Traveling Salesman Problem, search decision tree, data structure, truncated matrices structure

DOI: 10.17587/it.24.698-704

References

1. Little J. D. C., Murty K. G., Sweeney D. W., Karel C. An algorithm for the traveling salesman problem, *Operations Research*, 1963, vol. 11, pp. 972–989.

2. Ulyanov M. V., Fomichev M. I., Goloveshkin V. A., Zhukova G. N. Ocenka parametrov raspredeleniya logarifma slozhnosti zadachi kommivoyazhora (Parameters estimation of log of complexity for the Traveling Salesman Problem), *Sovremennye informacionnie tehnologii i IT-obrazovanie*, 2017, vol. 13, no. 1. pp. 19–24 (in Russian).

3. Goloveshkin V. A., Zhukova G. N., Ulyanov M. V., Fomichev M. I. Ispolzovanie kvantilnix koefficientov asimmetrii i ekscessa dlya ocenki slozhnosti resheniya zadachi kommivoyazhora (The use of quantile coefficients of asymmetry and kurtosis

for estimating the complexity of solving the Traveling Salesman Problem), *International Journal of Open Information Technologies*, 2016, vol. 4, no. 12, pp. 7–12.

4. Goodman S., Hidetniemi S. *Vedenie v razrabotku i analiz algoritmov* (Development and analysis of algorithms), Moscow, Mir, 1981, pp. 368 (in Russian).

5. Sigal I. H., Ivanova A. P. *Vedenie v prikladnoe discretnoe programirovanie: modeli i vichislitelnie algoritmi* (Introduction to Applied Discrete Programming: Models and Computational Algorithms), Moscow, FIZMATLIT, 2007, pp. 304 (in Russian).

6. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. *Introduction to Algorithm*, Cambridge, MA, MIT Press, 2009.

7. Wirth N. *Algorithms + Data Structures = Programs*, Upper Saddle River, NJ, Prentice Hall, 1976.

МОДЕЛИРОВАНИЕ MODELING

УДК 04.94, 658.5.012.1

DOI: 10.17587/it.24.705-713

Ю. А. Зак, д-р техн. наук, науч. консультант,
Аахен, Германия

Принципы построения систем имитационного моделирования производственных систем

Рассмотрены основные задачи, функции, принципы построения, основные компоненты и архитектура, объем необходимой информации для построения системы имитационного моделирования производства. Результаты проведенных имитационных экспериментов могут быть использованы для определения оптимальной структуры, состава технологических модулей и распределения персональных ресурсов, направлений материальных потоков, выбора стратегии, параметров и пунктов контроля незавершенного производства, размеров партии обрабатываемых изделий, а также алгоритмов управления технологическим процессом. Решения на основании этих результатов могут использоваться как на стадии оперативного и календарного планирования, так и на стадии проектирования, выбора состава и технических характеристик производственных модулей и технологической подготовки производства.

Ключевые слова: имитационное моделирование, управление материальными потоками, алгоритмы выборочного контроля, объемы информации, базы данных, графический интерфейс

Введение

Имитационное моделирование уже в течение многих лет находит широкое применение при анализе и поиске наиболее эффективных способов организации и проектирования технологических процессов, при построении оперативных и календарных планов выпуска продукции, управлении материальными потоками и маневрировании трудовыми ресурсами [1–7, 11–13].

Применение этих методов наиболее эффективно в условиях средне- и мелкосерийного производства, когда в зависимости от структуры предприятия, имеющегося портфеля и установленных сроков выполнения заказов, состояния оборудования, запасов сырья, полуфабрикатов и незавершенного производства, наличия трудовых ресурсов необходимо найти наиболее эффективный по технико-экономическим показателям календарный план работы предприятия, обеспечивающий выполнение всех ограничений на сроки поставки готовой продукции [8, 9, 11]. В этих условиях производственный процесс должен быть не только экономически эффективным, но и в значительной степени

гибким, легко перестраивать свою структуру в зависимости от требований рынка, быстро и с минимальными затратами реагировать на различные требования потребителей.

Разрабатываемая система имитационного моделирования должна быстро и эффективно определить:

— "узкое место" в технологической цепочке в данной конкретной производственной ситуации;

— мероприятия, позволяющие существенно сократить время выполнения некоторого подмножества заказов и повысить производительность производства или отдельных его технологических модулей;

— состав наиболее эффективного оборудования, необходимого для выполнения каждого заказа;

— резервы сокращения времени производства изделий и запасов незавершенного производства;

— надежность выполнения каждого из заказов в установленные сроки;

— наиболее эффективные стратегии и параметры выборочного контроля на различных стадиях производства;

— эффективное распределение во времени загрузки оборудования, трудовых и материальных ресурсов.

Другой областью применения систем имитационного моделирования являются проектирование, техническое перевооружение производства, ввод в производство новых изделий и производственных мощностей, внедрение новых более прогрессивных технологических процессов и систем автоматизации. При решении этих задач имитационное моделирование принимаемых решений позволяет наиболее эффективно выбрать состав оборудования, транспортных модулей объемов буферных накопителей, средств и стратегии контроля правильности выполнения технологических операций и обеспечения высокого качества изделия на всех стадиях его изготовления.

Все технологические процессы очень тесно взаимосвязаны друг с другом. Некоторые изменения, вносимые на определенных стадиях технологического процесса, оказывают существенное влияние на дальнейший ход производства и зачастую оказывают трудно предсказуемые воздействия. Лицу, принимающему решение, зачастую трудно комплексно и системно оценить последствия принимаемых им локальных решений на поведение всей системы в целом. В этих условиях только системно ориентированные математические или имитационные модели позволяют осуществить комплексный анализ производства. Ввиду сложности системных взаимосвязей, динамического характера, многомерности реальных производственных систем аналитические модели позволяют получить их математическое описание только в сильно упрощенном виде. Многие не учитываемые в этих моделях факторы и погрешности оказывают существенное влияние на качество принимаемого на основе этих моделей решения. В этих условиях (см., например, [6, 7]) только гибкие имитационные модели производственного процесса, учитывающие все особенности реального производства, позволяют осуществить объективный анализ состояния как отдельных локальных процессов, так и всего производства.

1. Основные задачи, функции и принципы построения системы

Использование систем имитационного моделирования в оперативном управлении сред-

не- и мелкосерийным производством должно помочь в решении следующих задач:

- выбор наиболее эффективного состава производственных модулей, буферных накопителей, направлений технологических и транспортных потоков, а также распределения трудовых ресурсов в целях выполнения всего комплекса производственных заданий в установленные сроки и с наилучшими экономическими показателями;
- быстрая реорганизация и перепланирование производства в соответствии с потребностями потребителей и состоянием производственных, материальных и трудовых ресурсов, а также незавершенного производства;
- сокращение времени выполнения заказов и объема запасов незавершенного производства;
- повышение вероятности завершения выполнения заказов в установленные договорными сроки;
- выбор правильной стратегии, а также параметров выборочного контроля незавершенного производства в целях определения наиболее эффективного компромисса между правильным определением некондиционных изделий на ранних стадиях технологического процесса, уменьшением связанных с этим экономических потерь и сокращением сроков выполнения заказов потребителей;
- повышение коэффициентов загрузки технологического оборудования;
- сокращение производственных расходов и процента брака;
- повышение квалификации персонала, принимающего решение по вопросам планирования и управления производством, на основе приобретенного ими опыта работы с имитационной моделью, лучшего понимания всех взаимосвязей в технологической цепочке.

Основные принципы построения программной системы заключаются в следующем:

- гибкость и открытость разрабатываемой системы, возможность ее легкой перестройки, расширения и включения новых программных модулей;
- модульная структура;
- возможность осуществлять моделирование как отдельных блоков, подсистем, так и всей системы в целом;
- простой графический интерфейс как на этапе построения имитационной модели объекта, так и в процессе ввода исходных данных

при решении оперативных задач планирования и управления и анализе полученных результатов;

- различные способы отражения, а также ясность в представлении и анализе как структуры моделируемого производства, так и результатов выполненного анализа и вычислений;
- наличие баз данных для хранения имитационных моделей, исходных данных и результатов выполняемых расчетов, обеспечивающих быстрый доступ и корректировку всей информации;
- возможность выполнения расчетов с учетом реального календарного времени;
- возможность вариации используемых производственных модулей, материальных и трудовых ресурсов, алгоритмов управления технологическими установками.

Имитационные модели в зависимости от постановки и содержания решаемой задачи могут рассматриваться как модели анализа текущего состояния производства, прогноза его будущего поведения, а также оптимизации ее параметров.

Имитационное моделирование предусматривает проведение одного или серии экспериментов, в процессе которых проводится анализ динамики системы в различных условиях. При этом в процессе экспериментов варьируются параметры состояния, входные данные и управляющие воздействия. Для производственных систем характерны также вариации структуры объекта, состава и использования во времени работы технологического оборудования и трудовых ресурсов. Хотя проведение таких экспериментов зачастую требует значительных вычислительных ресурсов, процесс выполнения их, а также полученные промежуточные и окончательные результаты более понятны лицу, принимающему решение (ЛПР), чем выводы, сделанные на основе аналитических моделей. Однако в результате проведения таких экспериментов не гарантировано получение наиболее эффективного решения поставленной проблемы.

Наибольшее распространение для анализа производства получили имитационные модели с дискретным временем, которые можно строить на основе следующих принципов [1–7, 12, 13]:

1) ситуационного подхода, основанного на рассмотрении новых состояний всех элементов системы в результате совершения определенных событий в каком-либо одном из ее элементов.

В качестве таких событий могут рассматриваться время завершения выполнения заказа либо необходимости останова или переналадки производственных модулей; достижение граничного уровня запасов в каком-либо из накопителей, израсходование лимита какого-то вида материальных или трудовых ресурсов и т. п.;

2) подхода, основанного на анализе каждой из транзакций, следующих последовательно друг за другом в определенные дискретные моменты времени (*transaction flow*). Транзакции — это временные системные динамические компоненты системы. Они определяют относительные временные интервалы, по завершению которых могут изменяться состояния отдельных блоков моделируемой системы, т. е. изменения объемов выпускаемой продукции, объемы запасов накопителей и состояния, т. е. активности, отдельных производственных модулей. В процессе прохождения отдельных транзакций определяются состояния всех последовательно и параллельно расположенных динамических объектов (производственных модулей) и технологических операций. Каждый производственный модуль выполняет определенные функции и характеризуется векторами входных и выходных параметров, значения которых изменяются в результате прохождения каждой из транзакций. В дискретных динамических детерминированных или стохастических имитационных моделях состояние всех блоков системы рассчитывается на каждом шаге транзакции, т. е. промежуточные состояния системы на каждом шаге расчетов служат основой и исходными данными для дальнейших шагов вычислений и могут сравниваться с результатами, полученными на реальном объекте. При прохождении транзакций каждая рабочая станция может оказаться свободной или занятой. Возникающее при этом действие в соответствии с заданным приоритетом или алгоритмом принятия решений может повлечь за собой некоторое событие или переадресовку для выполнения на другие блоки системы.

Наибольшее распространение в имитационном моделировании получил концепт, ориентированный на анализ материальных потоков (ситуационный подход и подход, ориентированный на анализе транзакций [1–6]), широкое применение находит также объектно-ориентированный подход [1, 2, 4, 5], в основе которого лежит последовательность обработки изделий.

В публикациях (см., например, [1, 2, 5–7]) описано большое число эффективных применений систем имитационного моделирования в автомобильной, электронной, химической, целлюлозно-бумажной, текстильной, пищевой промышленности и в машиностроении. Рынок программной продукции включает более 120 программных систем.

Системы имитационного моделирования, ориентированные на решение специальных задач, используют языки программирования GPSS, Modula-2, Simula, Simgen, SLAM, Pascal, C, C++, Java и позволяют пользователю при создании имитационной модели объекта создать любую структуру производственного комплекса, легко описать все материальные потоки, функции отдельных блоков, в приемлемой форме представить результаты выполненных расчетов, расширить область применения системы, введя дополнительные функции. Однако для предприятий малой и средней мощности, у которых зачастую нет высококвалифицированных специалистов по информатике и программированию, выполнение таких работ по адаптации системы не представляется возможным. Им нужны простые в эксплуатации программные системы, в которых выполнение всех этих функций должно осуществляться с помощью простого графического интерфейса.

2. Основные компоненты системы

Программная система имитационного моделирования промышленных объектов в дискретном времени должна содержать следующие компоненты [1, 3–9, 11]:

- переменные состояния, генераторы и счетчики дискретного системного времени и транзакций;
- методы и алгоритмы обнаружения и регистрации свершившихся событий (изменения состояния отдельных объектов);
- динамические структуры данных изменения состояния объектов во времени;
- библиотеку алгоритмов управления очередями;
- генераторы случайных чисел с различными функциями распределения;
- библиотеку программ интерпретации, статистического анализа и оценки полученных результатов;
- базы данных и протоколы хранения, представления и преобразования всей инфор-



Рис. 1. Основные элементы системы

мации об объекте, входных и промежуточных результатов экспериментов, а также выходной информации;

- графический интерфейс с пользователем;
- графическое и табличное представление результатов анализа, анимацию хода производственных процессов.

Основные элементы системы имитационного моделирования производства приведены на рис. 1.

3. Иерархическая структура организации производства

Технологическая схема организации производства может быть представлена в виде иерархической сети (рис. 2, см. третью сторону обложки). Каждый узел более высокого уровня этой сети может быть развернут в виде некоторой сети более низкого уровня. При этом предусматривается не только последовательное, но и параллельное соединение этих узлов, а также все возможные альтернативы направления технологических потоков. Каждый узел этой иерархической сети может определять производственный, складской, накопительный, транспортный модуль кондиционной или отбракованной продукции или пост контроля, а также отдельные технологические операции и вспомогательные операции (профилактический контроль и ремонт оборудования и т. п.). Узел более высокого уровня иерархии определяет более сложные технологические модули: группы оборудования, роботизированные посты, автоматические линии, участки и цеха производства. Каждый из этих узлов должен содержать всю необходимую для моделирования входную информацию, которая хранится

в банке данных входной информации, а также информацию, полученную в результате расчетов и содержащуюся в банке данных результатов расчетов. Отметим, что в банке данных может храниться информация не только об одном, но также и о нескольких производствах. Такое представление технологического процесса позволяет проводить имитационное моделирование на различных уровнях иерархии с разным уровнем детализации. Можно рассматривать детализованную схему расчетов, рассматривая каждую из технологических и вспомогательных операций, а также на разных уровнях укрупнения (технологические модули, участки или цеха производства). Кроме того, можно осуществлять имитационное моделирование только определенных отдельных структурных единиц различных уровней иерархии. При этом необходимо осуществить интеграцию соответствующих входных данных на основе содержащихся в системе алгоритмов расчетов. При расчетах на всех уровнях иерархии должно быть предусмотрено время переналадки оборудования и подготовительно-заключительных операций.

4. Особенности использования систем имитационного моделирования в машиностроении, электронике промышленности и приборостроении

4.1. Основные требования, предъявляемые к системе

Системы имитационного моделирования, ориентированные на широкое использование в производстве электронной аппаратуры, в машиностроении и приборостроении, должны удовлетворять следующим специфическим требованиям:

1. Современное производство электронной аппаратуры предусматривает большое число операций контроля на различных стадиях технологического процесса. Система имитационного моделирования должна предоставить пользователю широкие возможности выбора стратегии и параметров выборочного контроля, а также связанные с этими решениями оценки потребности в контрольном оборудовании, трудовых ресурсов, иметь в своем составе алгоритмы расчетов вероятности и стоимости выпуска брака и изменении времени производственного цикла.

2. Так как многие производства представляют собой объединение технологических процессов различной длительности, которые могут протекать как в течение рабочего времени, так и за его пределами (например, процессы сушки, охлаждения, разогрева изделий, затвердевания покрытий и т. п.), процесс моделирования должен выполняться в реальном календарном времени.

3. В зависимости от потребностей производства один и тот же человек может выполнять в различные моменты времени одну или сразу несколько разных технологических операций, обслуживать один или несколько производственных модулей. Поэтому в системе должны быть предусмотрены в процессе имитационных экспериментов возможности манипулирования трудовыми ресурсами.

4. Необходимо предоставить возможность пользователю проводить эксперименты как с укрупненной имитационной моделью производства, когда некоторая совокупность объектов объединяется в укрупненные блоки, так и с наиболее высокой степенью точности, с учетом каждой технологической операции и каждой единицы оборудования. При этом в качестве объекта моделирования может выбираться любая структурная единица из заданной иерархии моделей производства. Должна быть предоставлена возможность в процессе моделирования выбора различных алгоритмов и режимов работы производственного модуля, а также количества и квалификации обслуживающего персонала.

5. Все имитационные эксперименты проводятся в реальном времени. Поэтому в системе должно быть предусмотрено программное обеспечение для графического ввода и корректировки фактического календарного времени работы производства. При этом учитываются выходные и праздничные дни, число и продолжительность рабочих смен, времена перерывов и пауз, возможности введения дополнительных часов работы.

4.2. Параметры объектов имитационной модели

Основными типами объектов имитационной модели являются:

— рабочие станции, т. е. различного вида обрабатывающие модули — станки, реакторы, промышленные роботы, технологические уста-

новки, машинные комплексы, гибкие производственные модули и др.;

- транспортные модули;
- модули контроля;
- буферные емкости, склады материалов и готовой продукции, накопители заданной емкости;
- управление транзакциями.

На рис. 3 и 4 показаны взаимосвязи входных, варьируемых параметров и данных, рас-

считанных на основе выполненных имитационных экспериментов, соответственно для производственных модулей и буферных накопителей производственной системы.

При выборе в графическом режиме из соответствующего каталога некоторого объекта в другом окне выводится таблица его технико-экономических показателей, которые можно корректировать и дополнять. Предоставляются также возможности ввода в реальном времени

необходимого объема трудовых ресурсов для обслуживания каждого объекта. Алгоритмом предусмотрена возможность выбора из базы данных для каждого структурного модуля в каждый дискретный момент времени числа обслуживающего персонала, связанные с этим изменения производительности объекта и необходимость внесения изменений в ход имитационного эксперимента.

При динамическом отображении состояния производства в процессе моделирования фактически работающие производственные модули в каждый момент времени в отличие от объектов или структурных единиц, находящихся в режиме простоя или переналадки, отражаются другим цветом или в режиме мерцания.



Рис. 3. Взаимосвязи входных, варьируемых и выходных параметров производственных, транспортных и контрольных операций моделируемой системы



Рис. 4. Взаимосвязи входных, варьируемых и выходных параметров буферных накопителей производственной системы

4.3. Определение стратегии и параметров выборочного контроля на различных стадиях технологического процесса

Большое значение в системах управления производством электронной аппаратуры имеет выбор наиболее эффективной стратегии выборочного контроля. Решение вопросов, какое число выполненных ранее технологических операций на данном контрольном модуле должно быть подвержено контролю, а также какова должна быть частота выборочного контроля (контролируется ли каждое изделие, либо каждое 5, 10, 20, 50-е и т. д.), во многом определяет процент результирующего брака, вре-

мя производственного цикла, а также объемы материальных и трудовых затрат на производство. Каждая контрольная операция может либо выявить все дефекты и ошибки, допущенные на всех предыдущих стадиях изготовления изделия, либо дефекты, допущенные при выполнении технологических операций на нескольких последних производственных модулях. Ужесточение операций выборочного контроля снижает процент брака и материальных затрат на производство, однако может привести к увеличению времени выполнения заказов, уменьшить надежность выполнения производственных заказов в установленные графиком сроки и, кроме того, требует большего объема персональных ресурсов.

Правильный компромисс в решении этих вопросов может быть выбран в процессе имитационного моделирования в зависимости от конкретной производственной ситуации. На основе анализа выбранных экспертом нескольких вариантов организации выборочного контроля при выполнении заданного объема заказов по результатам имитационного моделирования выбирается наиболее приемлемый по совокупности всех технико-экономических показателей вариант. Системой ведется статистический анализ фактического процента на каждой стадии технологического процесса, а также рассчитывается на основе методов, изложенных в работе [10], процент необнаруженных некондиционных изделий после операций тестирования (в зависимости от параметров выборочного контроля). В алгоритм имитационного моделирования включены расчеты ожидаемого числа кондиционных и бракованных изделий на выходе каждого производственного и контрольного модуля на основе построенных статистических зависимостей. Реализация этой важной функции системы существенно повышает эффективность управления качеством выпускаемой продукции.

4.4. Последовательность выполняемых расчетов

В процессе имитационного моделирования осуществляется следующая последовательность расчетов:

- определение различных альтернативных вариантов структуры производственного комплекса, состава и технических характеристик используемых технологических модулей, вы-

бор алгоритмов управления оборудованием, правил и приоритетов распределения материальных потоков, выбор параметров выборочного контроля на различных стадиях производства;

- определение в зависимости от объемов заказов состояния запасов материалов, комплектующих и незавершенного производства на основании решающих правил и аналитических расчетов альтернативных размеров партий производимых изделий для структурных подразделений и отдельных производственных модулей;
- грубая оценка времени изготовления каждой партии изделий и определение "узких мест";
- формирование различных вариантов использования во времени персональных ресурсов, введение (при необходимости) дополнительных смен и сверхурочных работ;
- проведение комплекса динамических имитационных экспериментов для различных вариантов принимаемых решений, формирование банка данных полученных результатов моделирования;
- представление всех необходимых результатов работы и технико-экономических показателей производства в удобных для пользователя графическом и табличном видах, а также в динамике изменения состояния отдельных объектов;
- выполнение анализа полученных результатов моделирования и выбор наиболее эффективного решения.

5. Алгоритм имитационного моделирования

Логическая схема алгоритма имитационного моделирования приведена на рис. 5. Ниже даны пояснения функций отдельных блоков алгоритма.

1. Определение и контроль полноты представления технологических связей, структуры интегрированных в технологической схеме производственных модулей.

2. Определение относительного и абсолютного временных диапазонов функционирования технологической схемы. Разбиение всего диапазона моделирования на временные интервалы.

3. Выбор всех производственных модулей (узлов сети) на каждом уровне иерархии. Ввод необходимой входной информации о техноло-

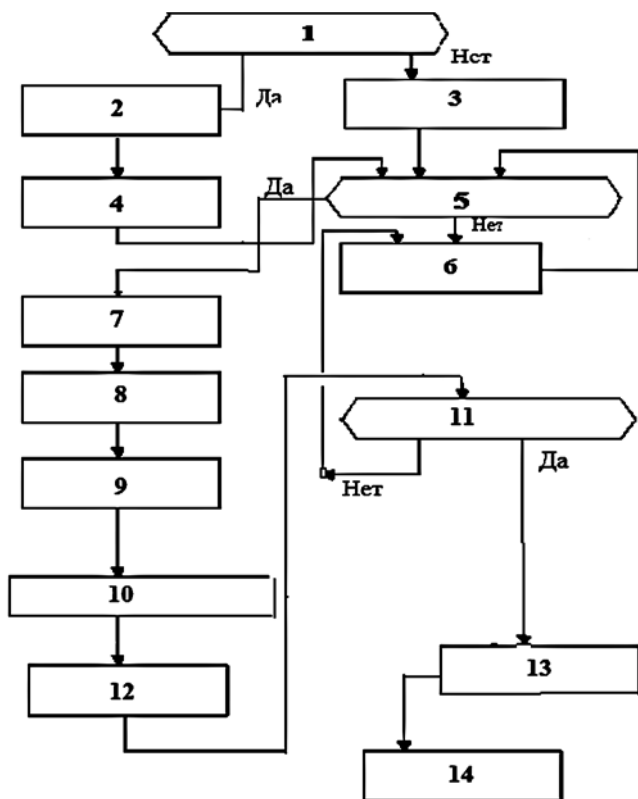


Рис. 5. Блок-схема алгоритма имитационного моделирования

гических возможностях и свойствах этих модулей из банка данных входной информации. Начало моделирования с модуля, стоящего первым в технологической цепочке.

4. Идентификация относительного номера временного интервала ($Z_Int = 0$) и номера первого стоящего в цепочке производственного модуля ($Mod_Num = 0$).

5. Определение, имеет ли рассматриваемый производственный модуль все необходимые для функционирования ресурсы. Расчет в зависимости от наличия материальных и персональных ресурсов производительности и технологических возможностей.

6. Выбор следующего стоящего в технологической цепочке производственного модуля.

7. Определение необходимого расчетного алгоритма в зависимости от выбранного типа этого производственного модуля.

8. Расчет для этого модуля к концу рассматриваемого временного интервала следующих параметров:

- произведенное (или контролируемое, транспортируемое, направленное в накопитель или в следующий модуль) число изделий;

- число отбракованных изделий, обнаруженных в процессе контроля или направленных для дальнейшей обработки;

- объем запаса в буферной емкости или на складе.

9. Определение состояния этого модуля (возможность активности в следующем временном интервале) в зависимости от наличия необходимых ресурсов и размера минимальной партии обрабатываемых изделий.

10. Ввод всех расчетных данных в банк данных результатов расчетов.

11. Проверка, является ли рассматриваемый производственный модуль последним в технологической цепочке.

12. Переход к следующему стоящему в технологической цепочке производственному модулю.

13. Пересчет результатов расчетов в режим реального времени. Ввод этих данных в банк данных результатов расчетов.

14. Завершение алгоритма имитационного моделирования для данного комплекта исходных данных.

Основные этапы внедрения системы имитационного моделирования и проведения экспериментальных расчетов на предприятии представлены в работе автора [11].

Заключение

В публикациях [8, 9, 11] описана разработанная под руководством автора система имитационного моделирования, соответствующая всем сформулированным выше принципам и требованиям. Приведены все графические интерфейсы формирования структуры моделируемого объекта, ввода календарного времени работы, необходимого объема входных данных для каждого из производственных и контрольных модулей, распределения персональных ресурсов во времени, выбора объектов и детализации выполнения имитационных экспериментов, а также определения перечня контролируемых объектов и требуемых форм представления результатов выполненных расчетов. Представлены формы графических зависимостей изменения параметров во времени в виде кривых изменения параметров в течение календарного времени суток и столбчатых диаграмм, вид которых изменяется в течение заданного диапазона времени.

Система ориентирована на проектирование и решение задач оперативно-календарного планирования для предприятий электронной промышленности и машиностроения и позволяет:

— в процессе планирования и управления производством определять наиболее эффективные партии обработки изделий, стратегии и параметры выборочного контроля на всех стадиях технологического процесса, выбирать алгоритмы управления технологическими модулями, правильно распределять во времени материальные и трудовые ресурсы;

— на стадии проектирования и ввода в производство новых изделий определить наиболее эффективную гибкую производственную структуру предприятия, состав и взаимосвязи технологических модулей, объемы буферных накопителей, требуемый объем трудовых ресурсов.

Список литературы

1. Лоу А. М., Кельтон В. Д. Имитационное моделирование, 3-е изд. СПб.: Питер, 2004. 848 с.
2. Кудрявцев Е. М. GPSS World. Основы имитационного моделирования различных систем. М.: ДМК Пресс, 2004.
3. Томашевский В. Н., Жданова Е. Г. Имитационное моделирование в среде GPSS. М.: Бестселлер, 2003. 416 с.

4. Карпов Ю. Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic. СПб: БХВ-Петербург, 2005. 400 с.

5. Кобелев Н. Б. Введение в общую теорию имитационного моделирования. Пособие для разработчиков имитационных моделей и их пользователей. М.: ООО Принт-Сервис, 2007. 128 с.

6. Kramer U., Neculau M. Simulationstechnik. Hanser Fachbuch, 1998.

7. Hopp W. J., Spearman M. L. Factory Physics-Foundations of Manufacturing Management. Chicago: Irwin-Verlag, 1996.

8. Zack Yu., Starke G. Process simulation. Mikrosystemtechnik. Production Techniques for Application specific Mikrosensors (G.Köler, T. Schroeter, G. Tshulena), Shaker Verlag, 2004. P. 291–323.

9. Starke G., Zack Yu., Friedrich A. P., Drichel E. Simulationswerkzeug bei der Herstellung von mikroelektronischen Komponenten // FB/IE Zeitschrift für Unternehmensentwicklung und Industrial Engineering. 2003. Vol. 52, N. 5. P. 204–212.

10. Беляев Ю. К. Вероятностные методы выборочного контроля. М.: Наука. Физматгиз, 1975. 438 с.

11. Zack Yu. A. Simulation systems of small and medium batch production: The functions and structure // Automation & Remote Control. Jan 2015. Vol. 76, Is. 1. P. 187–191.

12. Таха Х. А. Имитационное моделирование Гл. 18 // Введение в исследование операций = Operations Research: An Introduction. 7-е изд. М.: Вильямс, 2007. С. 697–737.

13. Строгалев В. П., Толкачева И. О. Имитационное моделирование. МГТУ им. Баумана, 2008. С. 697–737.

Yu. A. Zack, D. Sc., e-mail: yuriy_zack@hotmail.com, Aachen, Germany

Principles of Construction of Systems of Imitation Modeling of Production Systems

The main tasks, functions, principles of construction, main components and architecture, the amount of necessary information for building the system of simulation production are considered. As a result of the simulation experiments carried out, they can be used to determine the optimal structure, the composition of technological modules and the distribution of personal resources, the directions of material flows, the choice of strategies, parameters and inspection points for uncompleted production, the size of the batch of processed products, and algorithmic control of the technological process. Solutions based on these results can be used both at the stage of operational and scheduling, and at the design stage, selecting the composition and technical characteristics of the production modules and the technological preparation of the production.

Keywords: simulation modeling, material flow control, selective control algorithms, information volumes, databases, graphical interface

DOI: 10.17587/it.24.705-713

References

1. Lou A. M., Kelton V. D. *Imitacionnoje modelirovanije*, 3 izd. Sankt Petersburg, Piter, 2004, 848 p. (in Russian).
2. Kudryavzeva E. M. *GPSS World. Osnovi imitacionnogo modelirovanija raslicnich system*, Moscow, DMK Press, 2004. (in Russian).
3. Tomashevsky V. N., Zhdanova E. G. *Imitacionnoje modelirovanije v srede GPSS*, Moscow, Bestseller, 2003, 416 p. (in Russian).
4. Karpov Yu. G. *Imitacionnoje modelirovanije sistem. Vvednije v modelirovanije s AnyLogic*. Sankt Petersburg, BKHV-Peterburg, 2005, 400 p. (in Russian).
5. Koblelev N. B. *Vvednije v obshuju teoriju imitacionnogo modelirovanija. Posobije dlya rasrabotchikov imitacionnich modelej*, Moscow, Print-Servis, 2007, 128 p. (in Russian).
6. Kramer U., Neculau M. *Simulationstechnik*, Hanser Fachbuch, 1998.
7. Hopp W. J., Spearman M. L. *Factory Physics-Foundations of Manufacturing Management*, Irwin-Verlag, Chicago, 1996.

8. Zack Yu., Starke G. Process simulation. *Mikrosystemtechnik. Production Techniques for Application specific Mikrosensors* (G. Köler, T. Schroeter, G. Tshulena), Shaker Verlag, 2004, pp. 291–323.

9. Starke G., Zack Yu., Friedrich A. P., Drichel E. Simulationswerkzeug bei der Herstellung von mikroelektronischen Komponenten, *FB/IE Zeitschrift für Unternehmensentwicklung und Industrial Engineering*, 2003, vol. 52, no. 5, pp. 204–212.

10. Belyajev Yu. K. *Verojatnostnije metodi vaborochnogo kontrolya*, Moscow, Nauka, Fizmatgiz, 1975, 438 p. (in Russian).

11. Zack Yu. A. Simulation systems of small and medium batch production: The functions and structure, *Automation & Remote Control*, Jan 2015, vol. 76, is. 1, pp. 187–191.

12. Таха Х. А. Имитационное моделирование, *Vvedenije v issledovanije operazij. Operations Research: An Introduction*. 7-е изд, Moscow, Viljams, 2007, pp. 697–737 (in Russian).

13. Strogalev V. P., Tolkacheva I. O. *Imitacionnoje modelirovanije*, MGTU im. Baumana, 2008, pp. 697–737 (in Russian).

А. В. Черняев, д-р техн. наук, проф., e-mail: alex_v_ch@mail.ru,
С. И. Горбачев, канд. техн. наук, доц., e-mail: gor-sergey1@yandex.ru,

М. А. Михаленко, аспирант, e-mail: maximabdul@mail.ru,

ФГБОУ ВО "Московский авиационный институт (национальный исследовательский университет)",

Е. В. Метелкин, д-р физ.-мат. наук, проф., e-mail: sitech47@mail.ru,
Российский государственный социологический университет

Комплексное моделирование последствий аварийных разливов нефти в малых водотоках для информационной системы их прогнозирования

Рассмотрены вопросы комплексного моделирования аварийных разливов нефти в малых реках для информационных систем прогнозирования последствий нефтяных загрязнений.

Ключевые слова: разлив нефти, математическое моделирование, экспертные системы, техногенная безопасность

Введение

Задача моделирования последствий аварийных разливов нефти несомненно является весьма важной и актуальной. Для нашей страны актуальным является и решение этой задачи для водных объектов, в частности, областей малых водотоков. Создание информационных систем прогнозирования чрезвычайных ситуаций требует развития, согласования и адаптации моделей для их успешной интеграции в эти системы.

В российской нормативно-технической документации представлен ряд методик, позволяющих проводить расчеты распространения и трансформации аварийных нефтяных разливов в условиях малых водотоков. Так, в справочнике [1] описан алгоритм расчета массы нефти, впитавшейся в землю, массы испарившихся летучих низкомолекулярных углеводородов нефти, массы растворенной или эмульсифицированной нефти, загрязняющей водоток. В методике [2] показан алгоритм расчета следующих параметров: массы разлившихся нефтепродуктов, площади разлива и массы испарившихся углеводородов в зависимости от времени с момента аварии. Эта же методика позволяет определять объем нефти, осевший на береговой поверхности русла водотока.

Достаточно полный анализ имеющихся моделей приведен в диссертации А. А. Павлова [3]. Дальнейшее развитие предложенных методов и моделей определяется, в первую очередь, их интеграцией в комплексные интеллектуальные информационные системы прогнозиро-

вания последствий чрезвычайных ситуаций, в частности, при создании систем, основанных на знаниях (СОЗ). Сложность реального построения таких систем не подлежит сомнению. Одним из перспективных подходов является применение в качестве базового компонента модели виртуальной экспертной системы (ВЭС). Такие системы отличает открытость, масштабируемость и гибкость, что определяет перспективность их применения при создании интеллектуальных информационных систем прогнозирования последствий аварийных разливов нефти.

Целью настоящей работы являлось проведение анализа структуры процессов комплексного моделирования последствий аварийных разливов нефти в малых водотоках при построении информационных систем их прогнозирования.

Структура комплексной модели

Предъявляемые к математической модели распространения нефтяного загрязнения базовые требования хорошо известны [4, 5]. Модель должна позволять:

- давать прогнозные оценки распространения нефтяных загрязнений;
- определять местоположения возможных источников загрязнения путем проведения анализа движения загрязнения;
- давать прогнозную оценку процессов преобразования загрязнения;

- оценивать количество нефти, осаждающейся на береговую поверхность.

Использование физических моделей процессов распространения и трансформации нефтяного разлива [6, 7], а также осаждения нефти на береговую растительность [8] позволяет дополнительно оценивать следующие параметры:

- площадь разлива от времени;
- объем испарившейся, диспергированной, эмульсифицированной, осевшей на дно и на береговую поверхность нефти, а также водно-нефтяной эмульсии от времени;
- промежутки времени, через которые нефтяное загрязнение достигнет заданных створов водотока;
- промежуток времени, по истечению которого нефтяное пятно пройдет заданный створ.

К процессам преобразования нефтяного разлива, происходящим в начальные моменты времени, можно отнести: растекание, перенос под действием течения водотока, испарение, эмульгирование, диспергирование, изменение состава, плотности и вязкости нефтепродуктов, осаждение на донную и береговую поверхности, осаждение на взвешенных частицах. На более поздних стадиях происходят преобразования, связанные с фото- и термохимическими изменениями, бактериологическим разложением, поглощением и утилизацией различными речными организмами. Соответственно, для расчета всех этих многочисленных параметров применяются соответствующие модели и алгоритмы. Они отличаются большим многообразием и нередко сложностью реализации.

Задача интеграции моделей в СОЗ требует выполнения дополнительных требований к системам моделей, в частности, их масштабируемости, открытости, гибкости, совместимости между собой и с другими информационными системами в сфере обеспечения техногенной безопасности.

Укрупненная структура формирования СОЗ для информационной поддержки интеллектуальных систем анализа и прогнозирования последствий аварийных разливов показана на рис. 1. Входными данными в этом случае являются массив исходных технических данных, экологические стандарты и

требования, а также нормативы безопасности. База моделей формируется по результатам анализа существующих моделей и осуществления новых этапов и процессов моделирования. В качестве моделей могут использоваться различные модели процессов и сценарные модели анализа последствий аварийных разливов. В последнее время часто используются имитационные модели, в том числе и полученные с помощью методов искусственного интеллекта.

Важным компонентом системы является прикладная геоинформационная система (ГИС), с помощью которой проводится привязка моделей к данной географической области и конкретное моделирование области распространения пятна загрязнения [9, 10]. Особенно важное значение приобретает этот компонент при анализе загрязнений в области малых водотоков, что определяется сложным характером течений в этих областях и трансформацией нефтепродуктов при их переносе в области водотоков с береговых областей.

Следует отметить сложности верификации моделей последствий разливов нефти, так как в этом случае не имеется достаточно полных и достоверных экспериментальных данных о реальных последствиях аварийных разливов в малых водотоках [11].

Система согласования предназначена для согласования "машинных" данных и знаний, а также знаний, получаемых от экспертов. В резуль-

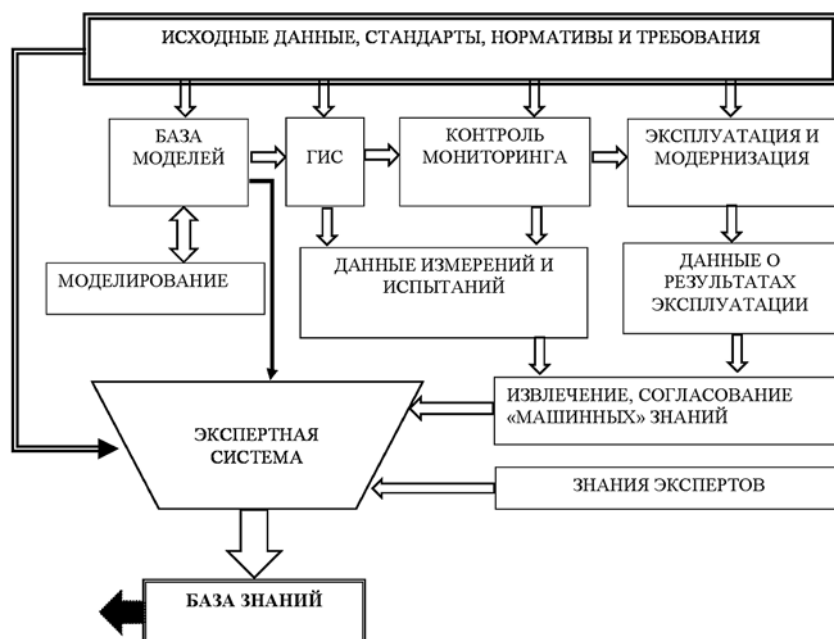


Рис. 1. Структура системы формирования знаний при моделировании аварийных разливов нефти

тате использования методов отбора знаний от экспертов получают некий набор информации, который должен быть верифицирован и согласован с имеющимися данными и "машинными" знаниями. Система согласования как раз и выполняет данную функцию. Ее работа основана на следующем принципе: полученная от эксперта информация сравнивается с информацией в базе знаний. В случае различия используются методы нечеткой логики для устранения противоречия, а именно, вычисления с использованием факторов (коэффициентов) уверенности, предназначенных для измерения степени релевантности данной информации.

Постоянная поддержка целостности и устойчивости системы, ее модернизация, устранение неопределенности и избыточности данных и т. д. являются самостоятельными и сложными научно-техническими задачами. Особенно важной представляется проблема согласования получаемых знаний, их верификация, а также мониторинг общего функционирования системы и формирование блоков вывода и протоколирования, а также различных интерфейсов.

Рассмотрим подробнее процесс построения экспертной системы. Обобщая и развивая результаты, полученные нами ранее, можно предложить в качестве базовой структуры подобной системы модель виртуальной экспертной системы (ЭС).

В этом случае, в зависимости от характера задачи компоненты виртуальной ЭС выбирают наиболее подходящий набор методов и алгоритмов функционирования ЭС. Здесь в качестве аналогов логических правил вывода обычных ЭС используются встроенные динамические ЭС, таким образом, всю архитектуру ЭС можно рассматривать как своего рода рекурсивную. Рекурсивная реализация метода не требует специальных машин параллельной обработки информации, поскольку параллельность может быть изначально заложена в сам алгоритм метода. Каждая параллель или независимая система обладает автономным информационным обеспечением. Клонирование информационного обеспечения происходит для обеспечения параллельности и получения нескольких независимых вариантов оптимальных решений. При этом каждая параллель или независимая система обладает автономным информационным обеспечением. При этом достаточно сложной и неоднозначной задачей представляется выбор конкретных методов со-

гласования и реализации динамических ЭС. Эта задача решается созданием системы управления знаниями. Традиционно в системах, базирующихся на знаниях, различают несколько базовых компонентов, основными из которых являются хранилище данных, механизм получения решений и интерфейс. Такая структура хорошо описывает ЭС, которые до настоящего времени на практике являются основным видом СОЗ. По мере развития информационных технологий становится ясно, что ЭС в их традиционном понимании могут применяться в сфере техногенной безопасности лишь для решения ограниченного круга задач. В то же время они могут быть эффективными компонентами сложных комплексных, в том числе и распределенных, СОЗ. Модельная структура подобной системы представлена на рис. 2, где не показаны интерфейсы и вспомогательные прикладные подсистемы.

Правая часть здесь представляет достаточно традиционную подсистему формирования экспертных знаний. С ее помощью формируется система экспертных знаний. Левая половина показывает подсистему извлечения и формирования знаний на основе массивов имеющихся данных и результатов моделирования. Необходимость использования в СОЗ громадных объемов разнородных данных вызывает необходимость интеграции в такие структуры подсистем, позволяющих извлекать из данных некоторые "машинные" знания, на основе которых создаются базы знаний (а точнее, базы моделей). Выбор алгоритмов и структур при реализации подобных подсистем является не-

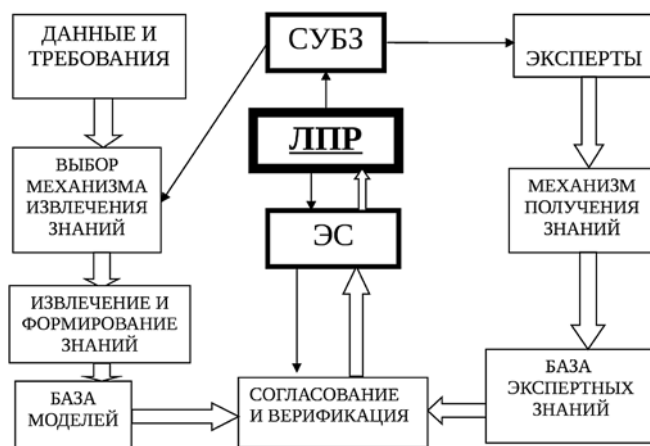


Рис. 2. Структура системы управления знаниями при моделировании аварийных разливов нефти (⇒ — информационная связь; → — управляющая связь)

тривиальной задачей, решение которой требует развития соответствующих методов, в том числе и с применением искусственного интеллекта. Ясно, что для согласования экспертных и "машинных" знаний требуются соответствующие методы и алгоритмы, выбор которых невозможен без участия человека (на схеме традиционно обозначенного как лицо, принимающее решение — ЛПР). В то же время эффективный выбор подобных средств затруднен без применения компьютеризированных структур информационной поддержки процессов принятия решений. Эти средства могут быть на практике весьма многообразны. Однако, по нашему мнению, их развитие должно идти по пути создания прикладных ЭС. В анализируемых системах поддержки процессов принятия решений ЭС выполняет следующие основные функции:

- прогнозирование;
- согласование с нормативной документацией;
- протоколирование;
- оценка рисков;
- контроль устойчивости системы;
- выдача рекомендаций.

Если же ЭС используется для согласования "машинных" и экспертных знаний, то ее структура намного сложнее. В зависимости от характера формируемых знаний и имеющихся требований и ограничений ЭС выбирает наиболее релевантный набор методов и алгоритмов для извлечения знаний, их структурирования и формализации, верификации и согласования. Основные особенности структуры ЭС в этом случае следующие:

- в качестве правил вывода (для обычных ЭС) используются встроенные динамические ЭС, т.е. всю архитектуру ЭС в целом можно рассматривать как рекурсивную;
- при выборе комплекса правил, методов, алгоритмов и прикладных программ используются как результаты экспертной оценки, так и различные методы автоматизированного принятия решений, что и позволяет определять ЭС как виртуальную.

Наиболее сложной задачей является выбор из некоторого множества нужного набора правил, методов обработки и алгоритмов. Это очень сложная задача, не имеющая однозначного решения. Одним из возможных способов частичного решения данной задачи представляется использование метода динамических приоритетов. В этом случае определяется список событий, инициирующих пересчет систе-

мы динамических приоритетов процедур, прямые и косвенные активации процедур.

Заключение

1. Показано, что разнообразие происходящих при разливах нефти в малых водотоках процессов и моделей для их анализа требует формирования и развития баз знаний для их интеграции в информационные системы прогнозирования последствий нефтяных загрязнений.

2. Проведен структурно-функциональный анализ системы информационной поддержки процессов прогнозирования последствий разливов нефти из трубопроводов, позволивший предложить и обосновать структуру подобной системы. Эта структура предполагает получение и согласование экспертных и "машинных" знаний. Последние формируются путем интеллектуального анализа исходных данных, их первичной обработки и дополнения результатами моделирования, вычислительных экспериментов с последующим извлечением и формированием знаний с помощью специальной ЭС.

3. Предложена обобщенная модель интеллектуальной системы информационной поддержки процессов мониторинга и анализа рисков аварийных разливов нефти, в основе которой лежит архитектура, основанная на виртуальной ЭС, что позволяет обеспечить необходимую гибкость логической структуры интеллектуальной системы, ее адаптивность и масштабируемость.

Список литературы

1. Мерициди И. А., Иванов В. Н., Прохоров А. Н. и др. Техника и технологии локализации и ликвидации аварийных разливов нефти и нефтепродуктов: Справочник. СПб.: НПО "Профессионал", 2008. 824 с.
2. Ларионов В. И. Моделирование аварийных разливов нефти на суше с применением ГИС-технологий. Методика. Уфа: МНТЦ "БЭСТС", 2004. С. 11—14.
3. Павлов А. А. Моделирование распространения аварийных разливов нефти по участкам водотоков малых рек. Дисс. ... канд. техн. наук: 03.02.08: защищена 24.04.12: утв. 30.08.12. М., 2012. 144 с.
4. Tsahalis D. T. Contingency planning for oil spills: River-spill a River Simulation Model // Proc. of the 1979 Oil Spill Conf, Amer. Petrol. Inst., Wash, D. C., 1979.
5. Дмитренко В. П., Роговицкий Ю. М. Моделирование процессов мониторинга аварийных разливов нефти из трубопроводов // Информационные технологии моделирования и управления. 2007. № 5. С. 588—593.
6. Kristensen F. T. Management of oil spill risks in Arctic waters // J. of Marine Environmental Engineering. 1994. Vol. 1. P. 131—159.

7. **Brovchenko I. A., Kushan F. V., Maderich V. M., Shlihtun M., Yushchenko S., Zheleznyak M.** The modelling system for simulation of the oil spills in the Black Sea // *Proc. 3 EUROGOOS Conf.* 2003. P. 56–62.

8. **Павлов А. А., Черняев А. В.** Моделирование процессов осаждения нефтяных загрязнений на береговую поверхность малых рек // *Информационные технологии.* 2009. № 11. С. 37–40.

9. **Осипов Д. Е.** Ввод и обновление пространственной информации в Мосгоргеотресте // *Информ. бюл. ГИС-Ассоц.* 2000. № 1 (23). С. 48–49.

10. **Черняев А. В., Павлов А. А.** Географическая информационная система малого речного бассейна и особенности ее формирования в задачах моделирования нефтяных загрязнений // *Информационные технологии.* 2014. № 2. С. 60–64.

11. **Reed M., Johansen O., Brandvik P. J., Daling P., Lewis A., Fiocco R., Mackay D., Prentki R.** Oil spill modeling towards the close of the 20th century: overview of the state-of-the-art // *Spill Science Technology Bulletin.* 1999. N. 5. P. 1–3.

A. V. Chernyaev, D. Sc., Prof., e-mail: alex_v_ch@mail.ru,
S. I. Gorbachev, Ph. D, Associate Professor, e-mail: gor-sergey1@yandex.ru,
M. A. Mikhalenko, Graduate Student, e-mail: maximabdul@mail.ru,
Moscow Aviation Institute (National Research University),
E. V. Metelkin, D. Sc., Prof., e-mail: sitech47@mail.ru,
Russian State Social University

Comprehensive Modeling of the Effects of Oil Spills in Small Streams for the Information System in their Forecasting

Paper studies the matters of comprehensive modelling of oil spills in small rivers for the information systems intended for the forecasting of oil pollution consequences.

Keywords: oil spill, mathematical modelling, expert systems, techno genic Safety

DOI: 10.17587/it.24.714-718

References

1. **Mericiđi I. A., Ivanov V. N., Prohorov A. N.** i dr. *Tekhnika i tehnologij lokalizacii i likvidacii avarijnyh razlivov nefti i nefteproduktov: Spravochnik* (Technology and technologies of localization and liquidation of emergency oil spills and oil products), SPb., NPO "Professional", 2008, 824 p. (in Russian).

2. **Larionov V. I.** *Modelirovanie avarijnyh razlivov nefti na sushe s primeneniem GIS-tehnologij. Metodika* (Modeling of emergency oil spills on land using GIS of technologies), Ufa, MNTC "BEHSTS", 2004, pp. 11–14 (in Russian).

3. **Pavlov A. A.** *Modelirovanie rasprostraneniya avarijnyh razlivov nefti po uchastkam vodotokov malyh rek* (Modeling the distribution of oil spills across sections of small river flows), Diss. kand. tekhn. Nauk, M., 2012, 144 p. (in Russian).

4. **Tsahalıs D. T.** Contingency planning for oil spills: River-spill a River Simulation Model, *Proc. of the 1979 Oil Spill Conf.*, Amer. Petrol. Inst., Wash, D. C., 1979.

5. **Dmitrenko V. P., Rogovickij Yu. M.** Modelirovanie processov monitoringa avarijnyh razlivov nefti iz truboprovodov (Modeling of oil spill monitoring processes from pipelines), *Information technologies of modeling and management*, 2007, no. 5, pp. 588–593 (in Russian).

6. **Kristensen F. T.** Management of oil spill risks in Arctic waters, *J. of Marine Environmental Engeneering*, 1994, vol. 1, pp. 131–159.

7. **Brovchenko I. A., Kushan F. V., Maderich V. M., Shlihtun M., Yushchenko S., Zheleznyak M.** The modelling system for simulation of the oil spills in the Black Sea, *Proc. 3 EUROGOOS Conf.*, 2003, pp. 56–62.

8. **Chernyaev A. V., Pavlov A. A.** *Modelirovanie processov osazhdeniya neftyanyh zagryaznenij na beregovuyu poverhnost' malyh rek. Informacionnye tekhnologii* (Geographic information system of the small river basin and peculiarities of its formation in the problems of modeling of oil contamination. Information technology) Moscow, 2009, no. 11, pp. 37–40 (in Russian).

9. **Osipov D. E.** Vvod i obnovlenie prostranstvennoj informacii v Mosgorgeotreste (Introduction and updating of spatial information in Mosgorgeotreste), *Inform. byul. GIS-Assoc.*, 2000, no. 1 (23), pp. 48–49 (in Russian).

10. **Черняев А. В., Павлов А. А.** Geopaficheskaya infopmacionnaya sistema malogo pechnogo bassejna i osobennosti ee fopmipovaniya v zadachah modelipovaniya neftyanyh zagpyaznenij (Geographic information system of the small river basin and peculiarities of its formation in the problems of modeling of oil contamination), *Informacionnye Tekhnologii*, 2014, no. 2, pp. 60–64 (in Russian).

11. **Reed M., Johansen O., Brandvik P. J., Daling P., Lewis A., Fiocco R., Mackay D., Prentki R.** Oil spill modeling towards the close of the 20th century: overview of the state-of-the-art, *Spill Science Technology Bulletin*, 1999, no. 5, pp. 1–3.

А. Б. Сорокин, канд. техн. наук, доц., e-mail: ab_sorokin@mail.ru,
А. П. Кушнарев, магистрант, e-mail: brainzeater@gmail.com,
Московский технологический университет (МИРЭА)

Морфологический анализатор текста для выявления полноты информации

Рассматриваются вопросы применения технологии автоматической частеречной разметки русскоязычных текстов, представленных в цифровом виде, в целях определения переизбытка или недостатка информации в тексте, выявления и построения концепта. Основное внимание уделяется этапу морфологического анализа как одному из наиболее сложных этапов в анализе текста в силу специфических особенностей морфологии русского языка, связанных с неоднозначностью соответствия слов той или иной части речи.

Ключевые слова: концептуальная структура, компьютерная лингвистика, частеречная разметка, автоматическая обработка документов, обработка текстов на естественном языке, алгоритм Портера

Введение

Развитие искусственного интеллекта и кибернетических идей привело к появлению множества научных областей, которые являются порождением часто не связанных между собой наук. Одной из таких научных областей является компьютерная лингвистика, которая позволила найти решение множества задач, с которыми прежде мог справиться только человек. Анализ текста естественного языка — одна из таких задач.

В современном мире человек вынужден существовать в бесконечном информационном потоке. В таких условиях возникает проблема зашумленности информации, определения разницы между значимой и несущественной информацией. При этом значительная часть информации представлена в виде цифровых текстовых документов. Вполне вероятно, что большинство из них обладают лишней, избыточной информацией, которая может затруднять чтение и усваивание информации из текста. Вместе с тем, текст может обладать недостаточностью для понимания информации.

Для структурирования и формализации информации необходимо выделить из нее концепт как некую логическую структуру. При этом концептуальная структура должна позволять проводить оценку полноты и адекват-

ности информации, формировать базу знаний на программном уровне. Описание такой структуры на основе ситуационного анализа представлено в предыдущих статьях [1, 2]. Очевидно, что для построения такой структуры должен быть разработан морфологический анализатор текста, который выявляет различные части речи и соотносит их с различными элементами структуры.

1. Постановка задачи для разработки морфологического анализатора

Исходя из концептуальной структуры ситуационного анализа [3] необходимо из текста выделить следующие элементы (рис. 1): субъект действия Xas ; действие Xa ; объект действия Xao ; компоненты действия $\{Xac_1, Xac_2, \dots, Xac_N\}$; взаимодействия $\{Rsc_1, Rsc_2, \dots, Rsc_N\}$; отношения $\{Rco_1, Rco_2, \dots, Rco_N\}$; наборы свойств: $\{Pas_1 \dots Pas_N\}$, $\{Pao_1 \dots Pao_N\}$, $\{Pac_{11} \dots Pac_{1N}\}$, $\{Pac_{21} \dots Pac_{2N}\}$ и $\{Pac_{N1} \dots Pac_{NN}\}$; соотношения $Rp\{Pac_{11} \dots Pac_{1N}\}$ и $\{Pao_1 \dots Pao_N\}$, $Rp\{Pac_{21} \dots Pac_{2N}\}$ и $\{Pac_{N1} \dots Pac_{NN}\}$ [1].

Выдвигается гипотеза, что каждый элемент концептуальной структуры может быть сопоставлен с определенной частью речи. При этом частью речи называется класс слов, объединенных общим грамматическим значением,

набором морфологических признаков и синтаксической ролью в предложении.

Однако сложность анализа любого элемента русского языка объясняется характерными особенностями структуры языка: неоднозначностью формализации языковых правил, законов склонения слов, произвольным порядком слов в предложении и т. д. Данные особенности не позволяют применять упрощенные алгоритмы обработки и анализа русскоязычных текстов. Однако они не являются последними в списке препятствий на пути к достижению корректного морфологического анализа [4]. Например, существует несколько точек зрения по поводу выделения причастий и деепричастий в отдельные части речи. Данная неоднозначность вызвана расхождением взглядов ученых-лингвистов относительно суффиксов,

образующих причастия и деепричастия: приверженцы первой точки зрения утверждают, что суффиксы причастий и деепричастий относятся к формообразующим, в то время как их противники придерживаются второй точки зрения, которая гласит, что эти суффиксы являются словообразующими [5].

Существуют и другие аспекты, касающиеся морфологии русского языка, по которым ученые не могут прийти к единому мнению. В данной статье рассматривается традиционная и наиболее распространенная классификация частей речи русского языка.

В русском языке принято различать междометные и немеждометные классы слов. Соответственно, междометия не используются в концептуальной конструкции в связи с их эмоциональным содержанием. Поэтому рассмотрим немеждометные классы слов,

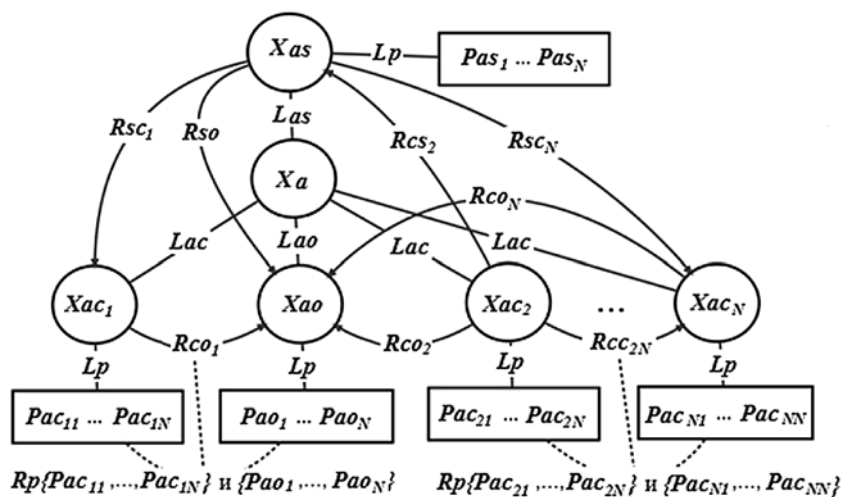


Рис. 1. Концептуальная структура на основе ситуационного анализа

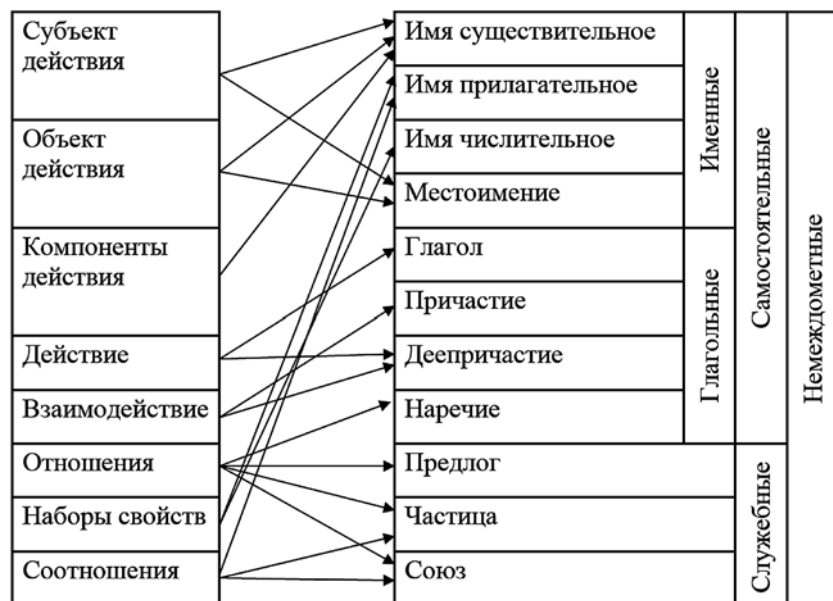


Рис. 2. Отображение элементов концептуальной структуры на части речи

которые делятся на служебные и самостоятельные. К служебным частям речи относятся предлоги, частицы и союзы. Самостоятельные части речи можно разделить на две большие группы: именные и глагольные. К именным относятся следующие части речи: имя существительное, имя прилагательное, имя числительное и местоимение. Глагольными частями речи называют глагол, причастие, деепричастие и наречие [6]. Таким образом, будут рассмотрены 11 частей речи, на которые будут отображены элементы концептуальной структуры (рис. 2).

Отображение элементов концептуальной структуры на части речи не является биективным, т. е. взаимно однозначным. Поэтому для снятия противоречий "многие к одному" и "один ко многим" предлагается использовать в разрабатываемом программном обеспечении сочетание словарного и бессловарного морфологических анализаторов.

Задачей морфологического анализа является определение по словоформе нормальной формы слова (леммы), от которой была образована данная словоформа, и набора параметров, приписанных к данной словоформе [7].

Морфологический параметр — это пара <имя параметра, значение

параметра>. Именем параметра может служить род, число, время, склонение, краткость формы прилагательного и другие признаки слов, принятые в данном языке. Значение параметра — это конкретное значение, которое может принимать данный признак. Так, например, род может быть мужским, женским, средним. При этом возможна ситуация, при которой одной словоформе может быть сопоставлено несколько параметров [8].

Важным аспектом при разработке анализатора является такое понятие, как омонимия. Омонимами называют разные по значению, но одинаковые по написанию единицы языка [5]. В терминах компьютерной лингвистики омонимию можно определить как ситуацию, при которой одной словоформе может быть приписано несколько значений для одной грамматической характеристики.

2. Выбор методов морфологического анализа

Морфологический анализ реализуется за счет двух методов: словарной и бессловарной морфологий.

Словарная морфология подразумевает наличие словаря, который содержит набор морфологических параметров для каждой словоформы, а также ее нормальную форму [4]. Соответственно, для реализации словарной морфологии требуется выделить область памяти для хранения словаря. При этом слова, отсутствующие в словаре, проанализировать не представляется возможным. Вдобавок, время поиска будет пропорционально объему базы данных и средней длине слова.

Однако при всех недостатках словарного метода снятие омонимии может быть достигнуто только путем составления дополнительного словаря омонимов, в котором будут содержаться наиболее часто употребляемые словоформы, обладающие свойством омонимии [9]. При этом уменьшение неоднозначности произойдет при построении концептуальной структуры и генерации прямого логического вывода в программном комплексе (ПК) "Оформитель + Решатель" [1]. Напомним, что в данном комплексе в случае обнаружения каких-либо синтаксических и семантических ошибок в концептуальных структурах пользователю выдаются соответствующие сообщения. Таким образом, ПК "Оформитель + Решатель" используется как семантико-синтаксический анализатор, что упрощает

устранение конфликтов. Происходит анализ не только неоднозначного слова, но и окружающего контекста. Соответственно, подобным образом решается задача неоднозначного отображения элементов концептуальной структуры на части речи.

Необходимо заметить, что слова могут группироваться по парадигмам. Парадигма — это правила, согласно которым можно получить все формы слов в лексеме (набор словоформ одного слова) для данного стема (неизменяемая часть слова). Бессловарные морфологии хранят только парадигмы слов. При этом в парадигме в качестве постфикса (единица слова, располагающаяся в слове после корня) может храниться только окончание. Также в бессловарной морфологии может храниться набор приставок и суффиксов с привязанной к ним морфологической информацией [4].

Существенным плюсом бессловарных морфологий является то, что они могут предсказывать морфологические характеристики практически любого слова, если его парадигма изменения попадает под одну из хранимых. Бессловарные морфологические анализаторы позволяют сэкономить расходы оперативной памяти. Скорость анализа выше, чем у словарных морфологий, а объем хранимых баз значительно сокращается.

К сожалению, такой подход не лишен недостатков, главным из которых является высокий процент ошибок. Так, например, слово "кровать" за счет своего постфикса может быть идентифицировано как глагол в начальной форме. Поэтому были разработаны специальные алгоритмы выявления постфиксов в словах в определенной последовательности, что позволило снизить процент ошибок.

Среди бессловарных методов морфологии выделяют системы, основанные на стемминге (процесс приведения словоформы к неизменяемой форме). В случае стемминга нередко отбрасывается вся морфологическая информация, а в качестве нормальной формы берется неизменяемая псевдооснова, называемая стем. Так, для слова "стена" стемом будет являться строка "стен". Именно эта основа и используется в дальнейшем для идентификации слова во всех его формах.

Стеммер Портера является одним из современных вариантов реализации бессловарной морфологии в чистом виде. В данной реализации применяется алгоритм, предложенный Мартином Портером [10].

3. Программная реализация частеречной разметки текста на основе алгоритма Портера

Под частеречной разметкой текстов (POS-tagging — Part-Of-Speech Tagging) понимается этап автоматического определения частей речи слов в тексте, представленном в цифровом виде. В процессе анализа и обработки текстов присвоение каждому слову соответствующей части речи является наиболее существенной частью морфологического анализа.

Алгоритм Портера не использует при работе базу основ слов, но функционирует за счет последовательного отсечения окончаний и суффиксов согласно определенным алгоритмом правилам [11].

Для полного понимания работы алгоритма введем некоторые определения:

1. Гласные буквы — а, е, и, о, у, ы, э, ю, я. Буква "е" считается равнозначной букве "е".

2. RV — область слова после первой гласной буквы. Она может быть пустой, если гласные в слове отсутствуют.

3. R1 — область слова после первого сочетания "гласная—согласная".

4. R2 — область R1 после первого сочетания "гласная—согласная"

Например, в слове "противоестественном": RV = "тивоестественном"; R1 = "ивоестественном"; R2 = "оестественном".

Среди существующих суффиксов и окончаний выделяются множества, разбитые по группам: PERFECTIVE GERUND (деепричастие), ADJECTIVE (имя прилагательное), PARTICIPLE (причастие), REFLEXIVE (возвратность), VERB (глагол), NOUN (имя существительное), SUPERLATIVE (превосходная степень), DERIVATIONAL (словообразовательные), ADJECTIVAL (PARTICIPLE + ADJECTIVE) (свойства причастия и имени прилагательного).

При поиске окончания из всех возможных выбирается наиболее длинное. Все проверки проводятся над областью RV. Буквы перед RV не участвуют в проверках вообще.

Шаг 1. Найти окончание PERFECTIVE GERUND. Если оно существует — удалить его и завершить этот шаг. Иначе удалить окончание REFLEXIVE (если оно существует). Далее удалить, если существуют, в следующем порядке окончания: ADJECTIVAL, VERB, NOUN. Как только одно из них найдено — шаг завершается.

Шаг 2. Если слово оканчивается на "-и", то удалить "-и"

Шаг 3. Если в R2 находится окончание DERIVATIONAL, то удалить его.

Шаг 4. Возможен один из трех вариантов:

1. Если слово оканчивается на "-нн", то удалить последнюю букву.

2. Если слово оканчивается на SUPERLATIVE, то удалить его и снова удалить последнюю букву, если слово заканчивается на "-нн".

3. Если слово оканчивается на "-ь", то удалить его.

Данный алгоритм используется для нахождения неизменяемой части слова — стема. Однако на его основе с определенной долей погрешности можно выделить пять частей речи: Имя существительное, Имя прилагательное, Глагол, Причастие и Деепричастие, причем для некоторых слов можно получить дополнительные морфологические признаки, такие как "превосходная степень" или "возвратность".

Разработка программного обеспечения (ПО) "Анализатор" для частеречной разметки текстов проводилась в программной среде IntelliJ IDEA. Данный выбор обусловлен тем, что IntelliJ IDEA — интегрированная среда разработки программного обеспечения на многих языках программирования, в частности Java, и является одним из наиболее мощных редакторов исходного кода за счет множества встроенных функций, таких как: умное автодополнение, инструменты для анализа качества кода, удобная навигация, расширенные возможности рефакторинга и форматирования [12].

На основе работы ПО "Анализатор" выдается отчет, где текст разбит на слова и для каждого слова определяется не только часть речи, но и элемент концептуальной структуры. Например, слово: Мастер <Имя существительное> — [Субъект Объект Компонент].

Таким образом, алгоритм Портера успешно применен для решения задачи частеречной разметки русского языка, однако его реализация для русского языка применима только для пяти частей речи.

Для проведения оценки точности работы ПО "Анализатор" и для составления словарей небольшого объема, на которых частично будет основываться анализ, необходимо выбрать словарь, содержащий информацию о словоформах русского языка. Подобные словари называются корпусами и используются при работе словарных морфологических анализаторов.

Корпус — подобранная и обработанная по определенным правилам совокупность текстов, используемых в качестве базы для исследования языка. Они используются для стати-

стического анализа и проверки статистических гипотез, подтверждения лингвистических правил в данном языке [5].

К наиболее популярным корпусам русского языка относятся "Открытый корпус", на основе которого был проведен анализ частоты употреблений словоформ каждой части речи [13] (см. таблицу).

Частота употреблений словоформ каждой части речи в словаре

Часть речи	Число употреблений (словоформ)
Имя существительное	1 006 853
Причастие	899 250
Имя прилагательное	680 557
Глагол	412 235
Деепричастие	68 237
Наречие	4057
Междометие	305
Имя числительное	236
Союз	193
Местоимение	180
Предлог	139
Частица	133

Исходя из результатов проведенного анализа можно заключить, что наибольшим числом словоформ обладают пять частей речи: Имя существительное, Причастие, Имя прилагательное, Глагол, Деепричастие. Именно эти части речи и позволяют классифицировать алгоритм Мартина Портера.

Остальные семь частей речи — Наречие, Междометие, Имя числительное, Союз, Местоимение, Предлог и Частица — целесообразно хранить в форме словаря, так как данные части речи не обладают достаточным числом характерных окончаний, которые позволили бы классифицировать их с достаточной точностью. Следует отметить, что суммарно данные семь частей речи (5243) составляют лишь 0,17 % от общего числа словоформ в словаре (3 072 375). Хранение словаря, содержащего упомянутые семь частей речи, требует всего лишь 578 Кбайт. Данный факт значительно уменьшает объем, занимаемый программным обеспечением. При этом напомним, что междометия в концептуальных структурах не используются.

Заключение

Для определения точности распознавания частей речи проведен анализ сравнения результатов работы разработанного программного обеспечения с результатами работы словар-

ного морфологического анализатора. Каждое совпадение хотя бы с одной из интерпретаций принадлежности слова к той или иной части речи с множеством интерпретаций из словаря помечалось как успешное определение части речи. Если же выявленная в результате анализа часть речи не совпадала ни с одной из тех, что приписывались слову словарем, то такое событие отмечалось как неудачное определение части речи. На основании проведенного сравнения процент точности разработанного проекта, в сравнении со словарем, составил 85,5 %.

Также одним из возможных применений разработанного алгоритма может быть включение его в алгоритм семантико-синтаксического анализа SemSyn [14—16]. Входной язык этого алгоритма составляют вопросы многих видов, команды и утверждения (описания ситуаций) на русском языке. Входные предложения могут включать причастные обороты и придаточные определительные предложения. Алгоритм SemSyn является композицией алгоритмов BuildMatr и BuildSem. Первый алгоритм строит семантико-синтаксическое представление входного текста в виде некоторой строково-числовой матрицы, а второй алгоритм строит по матрице семантическое представление входного текста. Основные процедуры алгоритма BuildMatr запускаются в зависимости от части речи, к которой относится анализируемая лексическая единица. При этом главную роль играют процедуры, обрабатывающие глаголы, причастия, существительные и прилагательные. А это именно те части речи, где хорошо проявил себя представленный в данной статье алгоритм.

Список литературы

1. Сорокин А. Б., Смольянинова В. А. Концептуальное проектирование экспертных систем // Информационные технологии. 2017. Т. 23, № 9. С. 634—641.
2. Сорокин А. Б., Лобанов Д. А. Концептуальное проектирование интеллектуальных систем // Информационные технологии. 2018. Т. 24, № 1. С. 3—10.
3. Болотова Л. С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях. М.: Финансы и статистика. 2012. 663 с.
4. Боярский К. К. Введение в компьютерную лингвистику: учеб. пособие. СПб.: НИУ ИТМО, 2013. 72 с.
5. Большакова Е. И. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие. М.: МИЭМ, 2011. 272 с.
6. Соснина Е. П. Введение в прикладную лингвистику: учеб. пособие. Ульяновск: УлГТУ, 2012. 110 с.
7. Кронгауз М. А. Семантика: Учебник. М.: Издательский центр "Академия", 2005. 352 с.

8. **Коваль С. А.** Лингвистические проблемы компьютерной морфологии. СПб.: Изд-во С.-Петербург. ун-та, 2005. 151 с.
9. **Гребенева Ю. Н.** Словарь омонимов и омоформ русского языка. М.: Айрис-пресс, 2011. 351 с.
10. **Porter M. F.** An algorithm for suffix stripping // *Program*. 1980, vol. 14, no. 3 (14). P. 130–137.
11. **Willett P.** The Porter stemming algorithm: then and now // *Program: Electronic Library and Information Systems*. 2006, vol. 40, no. 3 (40). P. 219–223.
12. **Ефимов А.** IntelliJ IDEA. Профессиональное программирование на Java. СПб.: БХВ-Петербург. 2005. 800 с.
13. **Открытый корпус.** URL: <http://opencorpora.org/>, свободный.
14. **Фомичев В. А.** Новый метод преобразования естественно-языковых тестов в семантические представления // *Информационные технологии*. 2005. Т. 11, № 10. С. 25–35.
15. **Фомичев В. А.** Формализация проектирования лингвистических процессоров. М.: МАКС Пресс, 2005. 368 с.
16. **Фомичев В. А., Разоренов А. А.** Значение теории К-представлений для исследований по автоматическому выявлению семантических ролей // *Информационные технологии*. 2015. Т. 21. № 6. С. 403–411.

A. B. Sorokin, PhD in Technique, Associate Professor, e-mail: ab_sorokin@mail.ru,
A. P. Kushnarev, master student, e-mail: brainzeater@gmail.com,
 Moscow Technological University (MIREA)

Morphological Text Analyzer for Revealing the Completeness of Information

This article discusses the use of technology of automatic part-of-speech tagging of Russian-language texts, presented in digital form, in order to determine the excess or lack of information in the text, the identification and construction of the concept. The main attention is paid to the stage of morphological analysis, as one of the most difficult stages in the analysis of the text, due to the specific features of the morphology of the Russian language, associated with the ambiguity of matching words of a particular part of the speech. Improving the accuracy of the analysis of Russian-language texts is achieved by identifying new patterns among the five parts of speech and by adding new inflectional endings to the existing ones in Porter algorithm. Removal of homonymy is achieved by creating an additional dictionary of homonyms, which will contain the most commonly used word forms that have the property of homonymy. Identification of excessive and insufficient information for understanding the text occurs in the process of constructing a conceptual structure and generating direct logical inference in the software package "Designer + Solver".

Keywords: conceptual structure, computational linguistics, part-of-speech tagging, POS tagging, automatic processing of documents, processing of texts in natural language, Porter stemming algorithm

DOI: 10.17587/it.24.719-724

References

1. **Sorokin A. B., Smol'janinova V. A.** Konceptual'noe proektirovanie jekspertnyh sistem (Conceptual design of expert systems), *Informacionnye Tehnologii*, 2017, vol.23, no. 9, pp. 634–641 (in Russian).
2. **Sorokin A. B., Lobanov D. A.** Konceptual'noe proektirovanie intellektual'nyh sistem (Conceptual design of intelligent systems), *Informacionnye Tehnologii*, 2018, vol. 24, no.1, pp. 3–10 (in Russian).
3. **Bolotova L. S.** *Sistemy iskusstvennogo intellekta: modeli i tehnologii, osnovannye na znanijah* (Systems of artificial intelligence: models and technologies based on knowledge), Moscow, Finansy i Statistika, 2012, 663 p. (in Russian).
4. **Bojarskij K. K.** *Vvedenie v komp'yuternuju lingvistiku* (Introduction to computer linguistics), SPb, Publishing house of NIU ITMO, 2013, 72 p. (in Russian).
5. **Bol'shakova E. I.** *Avtomaticheskaja obrabotka tekstov na estestvennom jazyke i komp'yuternaja lingvistika* (Automatic processing of texts in natural language and computer linguistics), Moscow, MIJeM, 2011, 272 p. (in Russian).
6. **Sosnina E. P.** *Vvedenie v prikladnuju lingvistiku* (Introduction to Applied Linguistics), Ul'janovsk, UIGTU, 2012, 110 p. (in Russian).
7. **Krongauz M. A.** *Semantika* (Semantics), Moscow, Izdatel'skij centr "Akademija", 2005, 352 p. (in Russian).
8. **Koval' S. A.** *Lingvisticheskie problemy komp'yuternoj morfologii* (Linguistic Problems of Computer Morphology), SPb., Publishing house of S.-Peterburg. un-ta, 2005, 151 p. (in Russian).
9. **Grebeneva Ju. N.** *Slovar' omonimov i omoforn russkogo jazyka* (Dictionary of homonyms and omoforn of the Russian language), Moscow, Ajris-press, 2011, 351 p. (in Russian).
10. **Porter M. F.** An algorithm for suffix stripping, *Program*, 1980, vol. 14, no. 3, pp. 130–137.
11. **Willett P.** The Porter stemming algorithm: then and now, *Program*, Electronic Library and Information Systems, 2006, vol. 40, no. 3, pp. 219–223.
12. **Efimov A.** IntelliJ IDEA. Professional'noe programirovanie na Java (Professional programming in Java), SPb., BHV-Peterburg, 2005, 800 p. (in Russian).
13. **Otkrytyj korpus** (Open case) (materialy sajta), available at: <http://opencorpora.org/>, svobodnyj.
14. **Fomichev V. A.** Novyj metod preobrazovaniya estestvenno-yazykovykh tekstov v semanticheskie predstavleniya (A new method for converting natural language tests into semantic representations), *Informacionnye Tehnologii*, 2005, vol. 11, no. 10, pp. 25–35 (in Russian).
15. **Fomichev V. A.** Formalizaciya proektirovaniya lingvisticheskix processov (Formalization of the design of linguistic processors), Moscow, MAKS Press, 2005, 368 p. (in Russian).
16. **Fomichev V. A., Razorenov A. A.** Znachenie teorii K-predstavlenij dlya issledovanij po avtomaticheskomu vyavleniyu semanticheskix rolej (The importance of the theory of K-representations for research on the automatic detection of semantic roles), *Informacionnye Tehnologii*, 2015, vol. 21, no. 6, pp. 403–411 (in Russian).

Е. В. Бурькова, канд. пед. наук, доц., e-mail: tulpan63@bk.ru,
Оренбургский государственный университет

Программная реализация выбора средств физической защиты объекта информатизации

Рассмотрена задача автоматизированного выбора средств системы физической защиты объекта информатизации. Проведен анализ методов, применяемых для решения данной задачи; проанализированы этапы выбора средств физической защиты, представлена функциональная модель и архитектура программной реализации, разработана программа, реализующая автоматизированный выбор средств защиты.

Ключевые слова: система физической защиты, выбор средств, программная реализация

Введение

Высокий уровень информатизации современного общества обуславливает создание информационной среды на различных объектах как экономической, так и социально-культурной сферы. Практически все предприятия, организации, частные фирмы представляют собой объекты информатизации, обладающие совокупностью информационных ресурсов, средств и систем обработки информации, используемых в соответствии с заданной информационной технологией, а также средствами их обеспечения. Несанкционированный доступ на объект информатизации может привести к краже либо модификации конфиденциальной информации, краже или порче технических средств хранения, обработки и передачи информации, а также материальных и финансовых средств, и в конечном итоге нанести ущерб объекту, привести к материальным потерям, потере репутации организации и другим ущербам. Угрозы физической безопасности объекта, связанные с проникновением нарушителя и совершением им несанкционированных действий, приводящих к нарушению технологических процессов объекта и всякого рода потерям (людским, материальным, финансовым, информационным), обостряются и модифицируются. В связи с этим проблема создания надежной системы физической защиты объекта

информатизации, решающей задачи обнаружения и предотвращения несанкционированного доступа, является актуальной.

Анализ методов выбора

Система физической защиты — это совокупность инженерно-технических, сигнализационных, программно-аппаратных и иных средств для защиты объектов от хищений, диверсий и других неправомерных действий, позволяющая на заданном уровне осуществлять создание трудностей, ограничение возможностей для нарушителя и увеличение времени проникновения его на объект охраны.

В работах [1, 2] нами были проанализированы основные этапы, выполняемые при проектировании системы физической защиты объекта информатизации. Необходимо выполнить следующую последовательность этапов.

1. Провести обследование объекта.
2. Установить категорию объекта по уровню важности.
3. Составить список требований руководящих документов к физической защите объекта выявленной категории важности.
4. Составить модель угроз и провести оценку рисков физической безопасности.
5. Установить критерии выбора средств физической защиты.

6. Осуществить выбор средств физической защиты.

Необходимо учитывать, что система физической защиты периодически должна подвергаться процедуре проверки эффективности ее функционирования. Это означает, что возникает необходимость проводить оценку уровня защищенности объекта информатизации для выбора рационального состава средств физической защиты, позволяющих снизить риски от потенциальных угроз. Актуальной задачей является разработка программной реализации всех перечисленных выше этапов, что позволит сократить временные затраты и наиболее оптимально выбрать средства физической защиты.

В задаче проектирования системы физической защиты самым сложным и плохо формализуемым этапом, предусматривающим процесс обработки экспертных знаний, является выбор средств физической защиты. Для реализации этой задачи применяют различные методы. Анализ публикаций ученых [3, 5–8, 10], проводивших исследование в данной области, показал, что наиболее часто применимы методы, показанные на рис. 1: это метод анализа иерархий; методы нечеткой логики; логико-вероятностный метод; генетический алгоритм. Рассмотрим кратко эти методы в контексте их применения для решения задачи выбора средств физической защиты.



Рис. 1. Методы выбора средств физической защиты

Генетический алгоритм. Генетические алгоритмы имеют некоторые преимущества перед другими методами, применяемыми при построении систем поддержки принятия решений. Один из таких алгоритмов был рассмотрен в работе [3] и применен авторами при разработке программы проектирования систем физической защиты GenalgSfz. Эффективность работы этих алгоритмов зависит от выбранных параметров: вероятности кроссинговера; вероятности мутаций; числа точек деления хромосом при кроссинговере и др. Чаще всего поиск лучших значений параметров проводится экспериментально. Дополнительная

сложность при определении наилучших значений параметров генетического алгоритма вызывается тем, что необходимо определить не постоянные значения, а правила изменения параметров в процессе поиска решения.

Логико-вероятностный метод. Применение логико-вероятностного подхода применяется для оценки физической защиты объектов и позволяет количественно оценить уровень защищенности путем анализа эффективности средств физической защиты. Данный подход является актуальным, так как определение уровня защищенности и степени рисков позволяет устранить недостатки систем физической защиты, функционирующих на объекте [4]. Этот метод был использован при построении программы анализа физической защищенности объекта *SecurityTips*, главной задачей которой является вычисление степени риска охраняемых зон объекта и выявление наименее защищенной зоны для ее дальнейшей модернизации.

Метод анализа иерархий. При использовании этого метода проводятся парные сравнения средств физической защиты и их способность нейтрализовать актуальные угрозы безопасности. Эксперты определяют вероятность реализации угроз, размер ущерба от реализации каждой угрозы, затем вычисляется риск от каждой угрозы. Риск от угрозы равен нулю, если данное средство защиты нейтрализует угрозу. Для определения относительной вероятности реализации угроз, показателей эффективности средств защиты и коэффициентов затрат на них используют матрицы парных сравнений. Данный метод в применении к решению задачи проектирования систем физической защиты был рассмотрен в работе [6].

Методы нечеткой логики. Эти методы представляют собой математический аппарат работы с объектами, не имеющими жестких, однозначно задаваемых границ. Данная теория позволяет формально описывать нестрогие, нечеткие понятия, формализовано описывать и обрабатывать информацию в случае физической неопределенности и представлять, обрабатывать информацию в случае лингвистической неопределенности. Теория нечетких множеств хорошо согласуется с условиями моделирования систем физической защиты и дает лучшие результаты по сравнению с получаемыми при классических алгоритмах. В качестве множеств рассматриваются следующие: множество элементарных зон защиты объекта; множество

угроз; множество оценок важности зон; множество средств физической защиты; множество оценок эффективности средств защиты; множество оценок стоимости. Между основными множествами устанавливаются соответствия. В работе [10] было описано применение метода нечеткой логики и авторы использовали данный метод при разработке программы проектирования состава систем физической защиты *FuzzyConclusion*.

Каждый из рассмотренных методов имеет определенные достоинства и недостатки. Для реализации задачи выбора средств физической защиты необходимо выполнить представленную выше последовательность этапов, при разработке программы был применен логико-вероятностный метод, его формализованное описание было приведено в работе [4].

Функциональная модель и архитектура программы выбора средств физической защиты

Для автоматизации приведенных этапов процесса выбора средств физической защиты была разработана функциональная модель в нотации IDEF0, представленная на рис. 2. Входными данными являются:

- данные для определения категории защищаемого объекта (функционально-отраслевая принадлежность, категория важности, категория защищаемой информации, показатели технической укреплённости и т. д.);
- помеховая обстановка на объекте (наличие ЛЭП, железнодорожных линий, автомо-

бильной трассы и других помех, влияющих на работу охранных извещателей);

- перечень защищаемых ресурсов, оценка их значимости и ценности;
- установленные средства защиты (для учета возможности интеграции новых средств защиты с имеющимися на объекте);
- угрозы физической безопасности.

На основе определения актуальных угроз безопасности строится матрица соответствия угроз и средств защиты для их нейтрализации.

Выходными данными является спецификация средств физической защиты.

Для автоматизации приведенных этапов процесса выбора средств физической защиты была разработана программа, архитектура которой представлена на рис. 3.

Разработанная архитектура состоит из следующих подсистем:

- анализа защищаемого объекта;
- формирования результатов анализа объекта;
- формирования требований к средствам физической защиты;
- выбора средств физической защиты;
- формирования спецификации выбранных средств физической защиты.

Подсистема анализа защищаемого объекта состоит из модулей анализа свойств защищаемого объекта, таких как функционально-отраслевая принадлежность, анализ деятельности объекта, учет числа присутствующего персонала и посетителей, помеховая обстановка вокруг объекта, характеристики техниче-

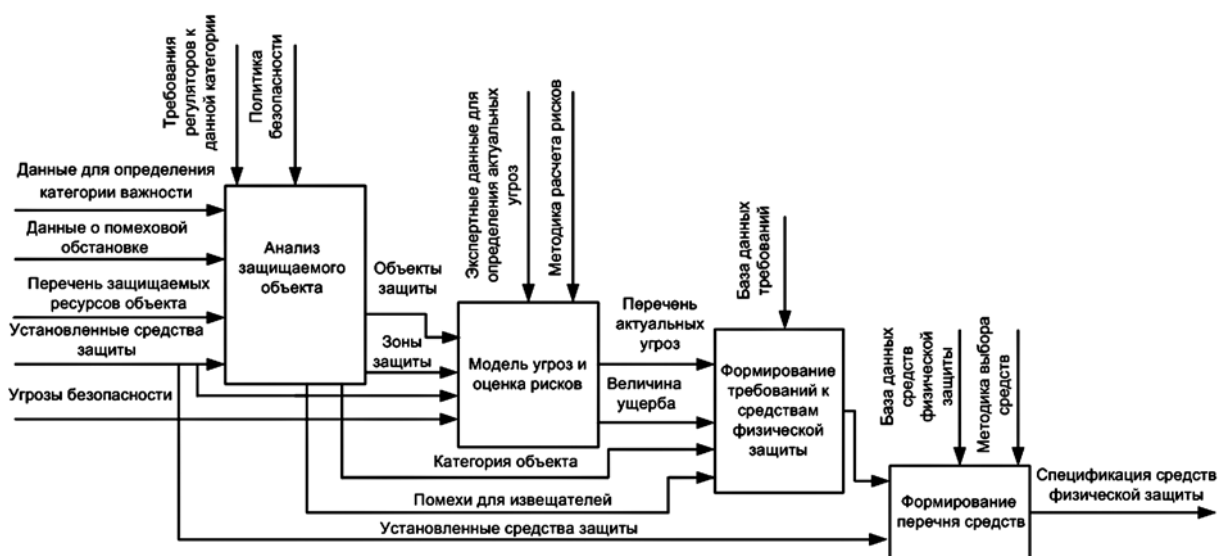


Рис. 2. Функциональная модель процесса выбора средств физической защиты

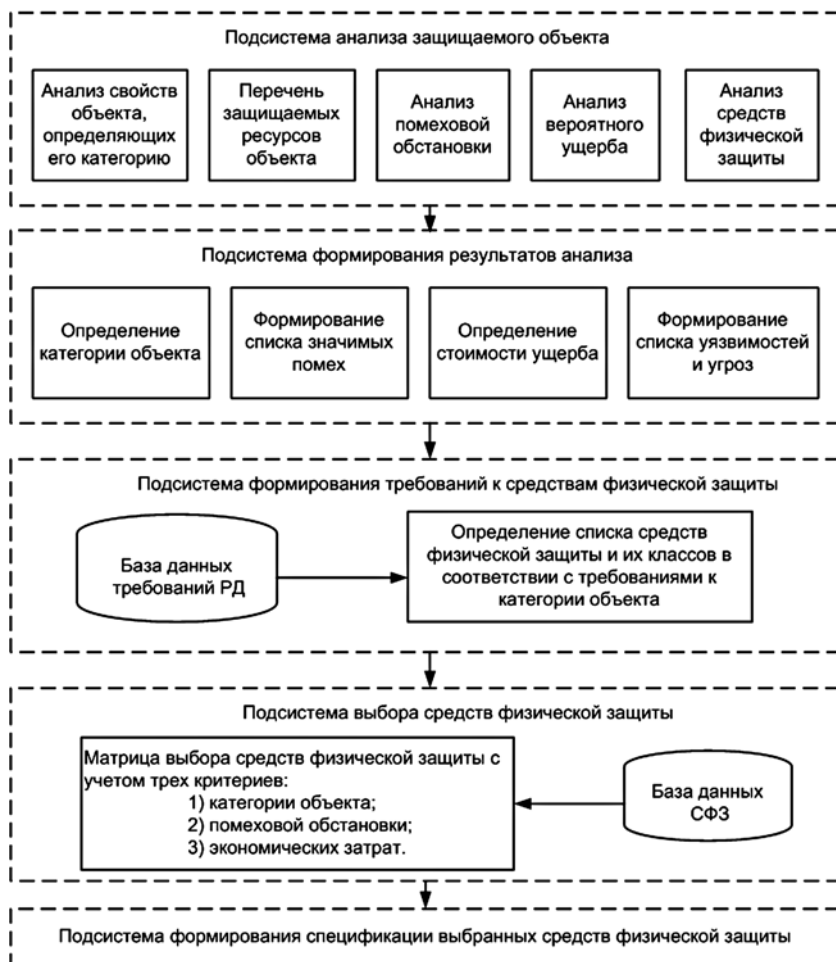


Рис. 3. Архитектура программы

ской укрепленности объекта, учет вероятного ущерба от реализации угроз, учет имеющихся средств физической защиты на объекте.

Подсистема формирования результатов анализа предназначена для получения следующей результирующей информации на основе проведенного обследования объекта: категория по уровню важности; список защищаемых ресурсов; стоимость возможного ущерба; список уязвимостей.

Подсистема формирования требований к средствам физической защиты осуществляет выбор из руководящих документов списка требований, соответствующих категории защищаемого объекта.

Подсистема выбора средств физической защиты осуществляет выбор на основе трех основных критериев: категории объекта; помеховой обстановки; предполагаемых экономических затрат на реализацию системы физической защиты.

Пользовательский интерфейс программы представлен на рис. 4.

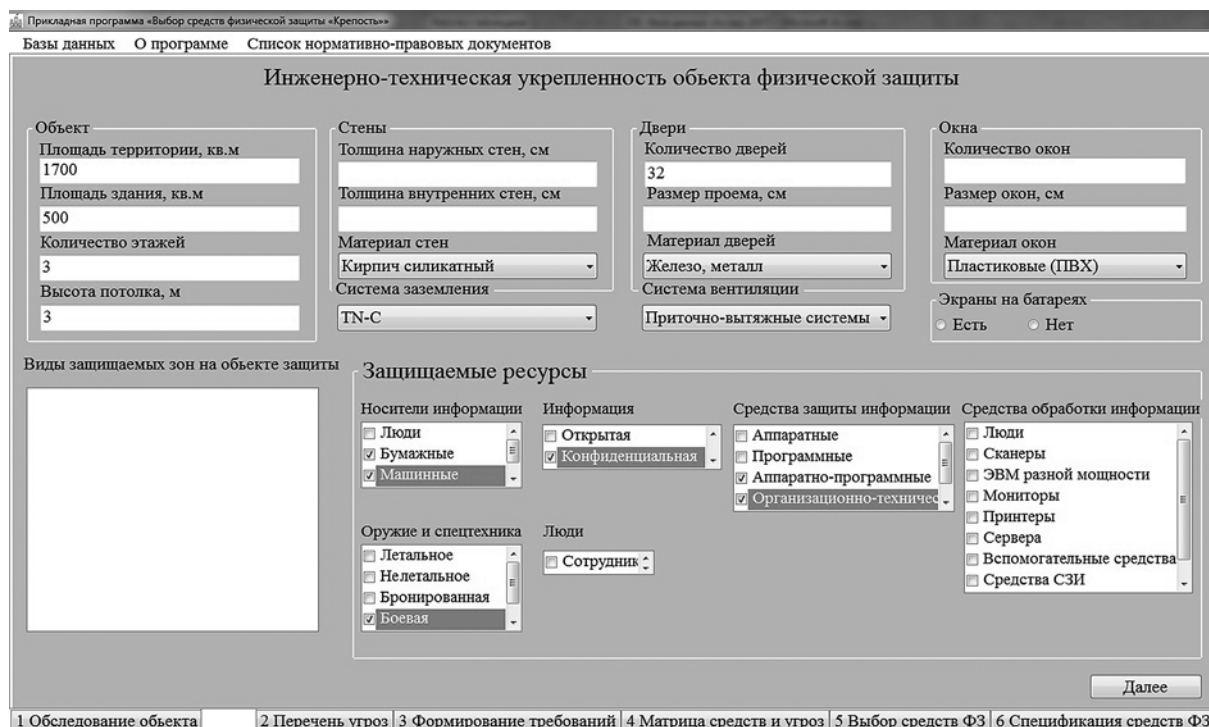


Рис. 4. Пользовательский интерфейс программы

На данном этапе программа проходит обследование инженерно-технической укрепленности объекта, вида информационных ресурсов, имеющихся средств защиты, вида защищаемых зон по уровню доступа в них и т. д.

Для каждого этапа процесса выбора реализовано отдельное окно интерфейса. Пользователь программы последовательно проходит все этапы выбора средств защиты, после прохождения каждого этапа формируется результирующий вывод. Например, после этапа обследования объекта формируется таблица результата категорирования объекта.

Результат категорирования объекта

Информативный признак категории	Категория исследуемого объекта
По функционально-отраслевой принадлежности	Объект социально-культурного назначения
По виду возможного ущерба	Людские потери, материальные, финансовые
По масштабу возможного ущерба	Региональный
По важности объекта	Объект категории Б1
По категории информации	Персональные данные, коммерческая тайна
По пожаро- и взрывоопасности	Объект категории Д
По численности персонала свыше 500 человек	Свыше 500 человек
По материальным активам свыше 500 МРОТ	Свыше 500 МРОТ

Разработанная программа предназначена для специалистов по информационной безопасности на защищаемых объектах различного назначения, а также может быть использована в учебном процессе для обучающихся по программам высшего образования по направлению подготовки 10.03.01. Информационная безопасность. Программа была протестирована на трех объектах информатизации. В результате работы программы временные затраты на выбор средств физической защиты были значительно снижены, уровень защищенности объектов соответствовал заявленным требованиям нормативных документов.

Заключение

Задача автоматизированного выбора средств физической защиты является актуальной в условиях современных угроз безопасности объектов информатизации. Автоматизация

информационных процессов позволит обеспечить качество и снизить трудоемкость выполнения данной задачи. Архитектура программы состоит из подсистем, отражающих этапы проведения выбора средств физической защиты и учитывает следующие критерии выбора: категория защищаемого объекта; помеховая обстановка на объекте; экономические затраты на систему физической защиты. Разработанная программа автоматизации выбора средств физической защиты отличается от существующих тем, что охватывает все необходимые этапы решения данной задачи и позволит определить наилучший вариант состава средств физической защиты для обеспечения высокого уровня безопасности объекта с учетом выбранных критериев оптимизации.

Список литературы

1. Бурькова Е. В. Категорирование объектов информатизации для выбора средств физической защиты // Университетский комплекс как региональный центр образования, науки и культуры: материалы Всерос. науч.-метод. конф., Оренбург, 2017. С. 3073–3076.
2. Бурькова Е. В. Структура системы поддержки принятия решений при выборе средств физической защиты // Компьютерная интеграция производства и ИПИ-технологии: материалы VIII Всерос. науч.-практ. конф., Оренбург, 2017. С. 186–189.
3. Шрейдер М. Ю., Боровский А. С., Тарасов А. Д. Проектирование систем физической защиты с помощью генетического алгоритма // Интернет-журнал "Науковедение". 2017. Т. 9, № 4. С. 67.
4. Бурькова Е. В., Гайфулина Д. А., Хакимова Э. Р. Прикладная программа оценки физической защищенности объекта на основе логико-вероятностного подхода // Материалы VI Международной науч.-практ. конф. "Информационные ресурсы и системы в экономике, науке и образовании". Пенза, 2016. С. 10–14.
5. Дураковский А. П., Петров В. Р. Применение математического аппарата при проектировании систем физической защиты // Безопасность информационных технологий. 2012. № 2. С. 80–84.
6. Костин В. Н., Шевченко С. Н., Гарнова Н. В. Проектирование систем физической защиты потенциально опасных объектов на основе развития современных информационных технологий и методов синтеза сложных систем. Оренбург: ИПК "Университет", 2014. 202 с.
7. Куделькин В. А., Янников И. М., Телегина М. В. Принципы создания интегрированных систем безопасности критически важных и потенциально опасных объектов // Интеллектуальные системы в производстве. 2017. № 1. С. 105–109.
8. Лукоянов С. В., Белов С. В. Основные требования к системам физической защиты на этапе их проектирования // Вестник Астраханского технического университета. 2010. № 2. С. 163–171.
9. Оладык В. С. Модель выбора рационального состава средств защиты в системе электронной коммерции // Вопросы кибербезопасности. 2016. № 1 (14). С. 17–22.
10. Боровский А. С., Тарасов А. Д. Использование методов нечеткой логики в задачах моделирования процессов // Информационные системы и технологии. 2010. № 3. С. 63–71.

Program Implementation of the Selection of Physical Protection of the Object of Informatization

The purpose of the article is to consider the problem of automated selection of the means of the physical protection system of the informatization object. The analysis of the methods used to solve this problem is carried out; the stages of the choice of means of physical protection are analyzed, the functional model and the architecture of the software implementation are presented, a program is developed that implements the automated selection of protection means.

Keywords: physical protection system, choice of means, software implementation

DOI: 10.17587/it.24.725-730

References

1. **Burkova E. V.** Kategorirovanie ob#ektov informatizacii dlja vybora sredstv fizicheskoj zashhity (Categorization of informatization objects for the choice of physical protection means), *Universitetskij kompleks kak regional'nyj centr obrazovanija, nauki i kul'tury: materialy Vseros. nauch.-metod. konf.* Orenburg, 2017, pp. 3073–3076 (in Russian).
2. **Burkova E. V.** Struktura sistemy podderzhki prinjatija reshenij pri vybore sredstv fizicheskoj zashhity (Structure of the decision support system when choosing the means of physical protection), *Komp'juternaja integracija proizvodstva i IPI-tehnologii: materialy VIII Vseros. nauch.-prakt. konf.* Orenburg, 2017, pp. 186–189 (in Russian).
3. **Shrejder M. Ju., Borovskij A. S., Tarasov A. D.** Proektirovanie sistem fizicheskoj zashhity s pomoshh'ju geneticheskogo algoritma (Designing physical protection systems using a genetic algorithm), *Internet-zhurnal "Naukovedenie"*. 2017, vol. 9, no. 4, pp. 67 (in Russian).
4. **Burkova E. V., Gajfulina D. A., Hakimova Je. R.** Prikladnaja programma ocenki fizicheskoj zashhishhennosti ob#ekta na osnove logiko-verojatnostnogo podhoda (Application program for assessing the physical security of an object on the basis of a logical-probabilistic approach), *Materialy VI Mezh-dunarodnoj nauch.-prakt. konf. "Informacionnye resursy i sistemy v jekonomike, nauke i obrazovanii"*. Penza, 2016, pp. 10–14 (in Russian).
5. **Durakovskij A. P., Petrov V. R.** Primenenie matematicheskogo apparata pri proektirovanii sistem fizicheskoj zashhity (Usage of Mathematical Apparatus for the Design of Physical Protection Systems), *Bezopasnost' informacionnyh tehnologij*, 2012, no. 2, pp. 80–84 (in Russian).
6. **Kostin V. N., Shevchenko S. N., Garnova N. V.** Proektirovanie sistem fizicheskoj zashhity potencial'no opasnyh ob#ektov na osnove razvitiya sovremennyh informacionnyh tehnologij i metodov sinteza slozhnyh sistem (Designing physical protection systems for potentially dangerous objects on the basis of the development of modern information technologies and methods for the synthesis of complex systems), Orenburg, IPK "Universitet", 2014, 202 p. (in Russian).
7. **Kudel'kin V. A., Jannikov I. M., Telegina M. V.** Principy sozdanija integrirovannyh sistem bezopasnosti kriticheski vazhnyh i potencial'no opasnyh ob#ektov (Principles for the creation of integrated security systems for critical and potentially hazardous facilities), *Intellektual'nye sistemy v proizvodstve*, 2017, no. 1, pp. 105–109 (in Russian).
8. **Lukojanov S. V., Belov S. V.** Osnovnye trebovanija k sistemam fizicheskoj zashhity na jetape ih proektirovanija (Basic requirements for physical protection systems at the stage of their design), *Vestnik Astrahanskogo tehničeskogo universiteta*, 2010, no. 2, pp. 163–171 (in Russian).
9. **Olad'ko V. S.** Model' vybora racional'nogo sostava sredstv zashhity v sisteme jelektronnoj kommercii (Model of the choice of the rational composition of remedies in the electronic commerce system), *Voprosy kiberbezopasnosti*, 2016, no. 1 (14), pp. 17–22 (in Russian).
10. **Borovskij A. S., Tarasov A. D.** Ispol'zovanie metodov nechetkoj logiki v zadachah modelirovanija processov (The use of fuzzy logic methods in modeling problems of processes), *Informacionnye sistemy i tehnologii*, 2010, no. 3, pp. 63–71 (in Russian).

Г. Н. Лебедев, д-р техн. наук, проф.,
Московский авиационный институт (национальный исследовательский университет),
С. В. Канушкин, канд. техн. наук, доц.,
Серпуховский филиал Военной академии ракетных войск стратегического назначения,
Т. И. Кузнецова, д-р пед. наук, проф., englishmail@mail.ru,
М. Г. Царегородцева, ст. преподаватель, marina77868@yandex.ru,
Российский химико-технологический университет им. Д. И. Менделеева

Автоматизированная компьютерная система индивидуального обучения при освоении простых и сложных навыков

Ставится задача автоматизации индивидуального компьютерного обучения простым и сложным навыкам в различных предметных областях.

Сформулировано решение задачи оптимизации обучения группе навыков за заданное время по критерию максимума суммы аддитивной и мультипликативных сверток получаемых оценок освоения каждого навыка.

Получен главный вывод о важности увеличения частоты чередования освоения навыков к концу обучения. Приведены примеры эффективности предложенного подхода в различных предметных областях.

Ключевые слова: простые и сложные навыки, автоматизация обучения, компьютерный класс, оптимизация времени обучения навыкам, динамическое программирование, частота чередования

Введение

Повышение эффективности компьютерных систем обучения является актуальной задачей. Как показывают предварительные оценки, оптимальное распределение времени освоения между простыми и сложными навыками, а также выбор последовательности их освоения и частоты их чередования позволяют снизить общее время и стоимость обучения на 10...20 %. При этом особенно важно учесть выявленные индивидуальные особенности каждого обучаемого лица как по скорости освоения, так и по степени забывания пройденного материала, чтобы дать рекомендации по времени, числу повторений и частоте чередования осваиваемых навыков.

Имеющиеся в настоящее время результаты в области компьютерного обучения относятся к следующим направлениям. В работах [1, 2] предложено различать простые и сложные навыки и считать для них неодинаковым время выполнения заданий "при регулируемом регламенте действий оператора". При этом вопрос

повышения качества обучения за счет оптимизации планов подготовки не рассматривается.

В работах [3, 4] впервые сформулирована задача оптимального планирования работы компьютерного класса при параллельности освоения простых и сложных навыков на основе динамического программирования Беллмана [5], но переходным процессам забывания и повторения с точки зрения восстановления качества подготовки внимание не уделяется.

В работах [6—8] решается задача формирования тестов или перечня заданий для освоения навыков различной трудности и даются априорная оценка сложности тестов и критерии точности тестирования в зависимости от размера выборки и числа заданий в тесте. Особое место занимают задачи формализации получения знаний не в технической области, а гуманитарной и педагогической деятельности [9—11]. В них главное место занимают критерии оценки качества приобретенных знаний.

В зарубежной литературе исследуется ряд перечисленных направлений [12—16], но в них

не учитывается в полной мере нелинейная динамика освоения и забывания навыков при их чередовании в процессе автоматизированного обучения.

Таким образом, полная динамическая модель освоения, забывания и повторения навыков различной сложности нуждается в развитии.

Целью данной работы является стремление раскрыть механизм оптимизации динамических процессов в расчете на индивидуальное обучение в компьютерном классе при следующей постановке задачи.

Постановка задачи

1. Считается, что за заданный период T обучения необходимо обеспечить освоение заданного числа n простых и m сложных навыков. Под простыми навыками понимается выполнение заданий, не требующих значительного времени, при этом вначале скорость освоения максимальна. Под сложными навыками понимается выполнение заданий, уровень освоения которых зависит от освоения предшествующих навыков, на которых базируется рассматриваемый навык.

2. Заданы две группы обучаемых лиц: сильная ($j = 1$) и слабая ($j = 2$). Для простых навыков приемлемой динамической моделью их освоения является, как показано в работе [2], экспоненциальная зависимость вида (рис. 1)

$$x_i(t_j) = 1 - e^{-a_j t_j}, \quad i = 1, \dots, n; \quad j = 1, 2, \quad (1)$$

где $j = 1$ для сильной группы обучаемых; $j = 2$ для слабой группы обучаемых, когда $a_2 < a_1$, что соответствует простому дифференциальному уравнению

$$\dot{x}_i = a_j(1 - x_i), \quad (2)$$

где $x_i(t)$ — нормированная оценка качества обучения простому i -му навыку, когда максимальный уровень освоения принят за единицу; t_j — отведенное время на освоение i -го простого навыка j -м обучаемым лицом или номером обучаемой группы; a_j — персональный показатель скорости освоения простого навыка, подлежащий идентификации.

Из рис. 1 видно, что скорость освоения вначале велика (чем меньше $1/a_j$, тем она больше),

а потом по мере обучения она падает, и дальнейшая трата времени становится неэффективной.

Для сложного навыка характерны низкая скорость освоения вначале, максимальная скорость в середине и убывание скорости при подходе к максимальному уровню, как показано на рис. 2.

Логистический характер для сложного навыка может быть представлен формулой

$$y_k(\tau_j) \approx (1 - e^{-b_j \tau_j})^2; \quad k = 1, \dots, m; \quad j = 1, 2, \quad (3)$$

где y_k — нормированная оценка качества обучения сложному k -му навыку; τ_j — отведенное время на обучение; b_j — персональный показатель скорости освоения сложного навыка, подлежащий идентификации при тестировании обучающегося.

Динамику освоения сложного навыка можно заменить системой дифференциальных уравнений, добавив промежуточную переменную Z :

$$\begin{cases} \dot{y}_k = Z_k; \\ \dot{Z}_k = 2b_j^2(1 - y_k) - 3b_j Z_k. \end{cases} \quad (4)$$

3. Динамика забывания каждого простого навыка описывается с помощью экспоненциальной модели снижения качества обучения,

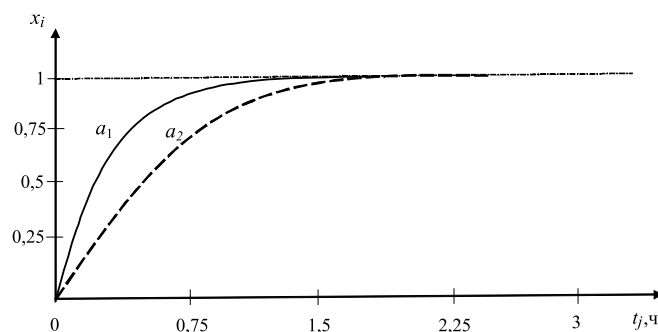


Рис. 1. График функции экспоненциальной зависимости качества обучения при $a_1 > a_2$

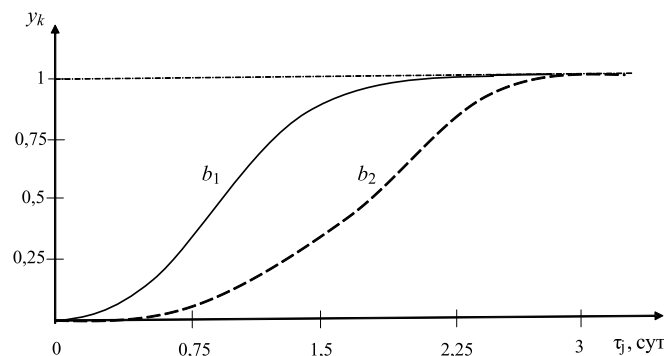


Рис. 2. График функции логистической зависимости качества обучения при $b_1 > b_2$

которой соответствует следующее простое дифференциальное уравнение:

$$\dot{x}_i = -\theta_j x_i, \quad (5)$$

где θ_j — персональный показатель скорости забывания простого навыка.

4. В итоге динамика освоения и забывания каждого простого навыка описывается с помощью дифференциальных уравнений роста и снижения качества обучения:

$$\dot{x}_i = \begin{cases} a_j(1 - x_i), & \text{при освоении простого навыка;} \\ -\theta_j x_i, & \text{при забывании простого навыка.} \end{cases} \quad (6)$$

5. Динамику забывания сложных навыков по аналогии с моделью (5) удастся описать дифференциальным уравнением вида

$$\dot{y}_k = -v_j y_k, \quad (7)$$

где v_j — персональный показатель скорости забывания сложного навыка.

Поэтому опишем динамику освоения и забывания каждого сложного навыка следующим образом:

$$\dot{z}_k = \begin{cases} Z_k; \dot{Z}_k = 2b_j^2(1 - y_k) - 3b_j Z_k - & \text{при освоении сложного навыка;} \\ -v_j y_k - & \text{при забывании сложного навыка.} \end{cases} \quad (8)$$

6. Процесс автоматизации получения оценки результатов освоения навыков при заданном регламенте проверок в данной работе не рассматривается.

7. Критерием J обучения в первом приближении является достижение максимума общего уровня освоения всех простых и сложных навыков в виде линейной свертки при заданном общем времени обучения T :

$$J_1 = \max \sum_{j=1}^2 \left[C_1 \sum_{i=1}^n x_i(t_j) + C_2 \sum_{k=1}^m y_k(\tau_j) \right], \quad (9)$$

где соблюдается ограничение по общему заданному времени обучения T :

$$\sum_{j=1}^2 \left(\sum_{i=1}^n t_{ij} + \sum_{k=1}^m \tau_{kj} \right) \leq T. \quad (10)$$

Однако при обучении не менее важным условием является недопустимость слишком низкой оценки хотя бы по одному навыку, поэтому критерий неадекватен линейной свертке, и предлагается его выбрать равным сумме линейной и мультипликативной свертки:

$$J_2 = J_1 + C_3 \prod_{i=1}^n x_i(t_j) \prod_{k=1}^m y_k(\tau_j), \quad (11)$$

где C_1, C_2, C_3 — заданные коэффициенты значимости обучения, но и в этом случае ограничение (10) должно выполняться.

В данной работе при выполнении приближенных расчетов используется более простой параметрический критерий (9).

Решение задачи обучения простым навыкам с помощью динамического программирования и метода параметрической оптимизации

Рассмотрим вначале задачу обучения двум простым навыкам. Поскольку динамика качества освоения навыков имеет дифференциальную форму, воспользуемся динамическим программированием для решения поставленной задачи.

Можно получить условие оптимальности в виде уравнения Беллмана в частных производных, решая задачу достижения заданного уровня освоения навыков за минимальное время:

$$-\frac{\partial \varepsilon}{\partial t} = \min_{j=1,2} \left\{ 1 + \frac{\partial \varepsilon}{\partial x_1} \dot{x}_1(j) + \frac{\partial \varepsilon}{\partial x_2} \dot{x}_2(j) \right\} = \min F_j(x_1, x_2), \quad (12)$$

где ε — функция Беллмана, которую можно аппроксимировать степенным полиномом второго порядка; F_j — функции риска, равенство которых определяет условие переключения с одного навыка на другой, а терминальный критерий оптимальности имеет вид

$$J_2 = C_1[x_1(\tau) + x_2(\tau)] + C_3[x_1(\tau)x_2(\tau)]. \quad (13)$$

Функцию Беллмана можно аппроксимировать степенным полиномом второго порядка:

$$\varepsilon \cong \alpha + \beta_1 x_1 + \beta_2 x_2 + 0,5\gamma_1 x_1^2 + 0,5\gamma_2 x_2^2 + \psi x_1 x_2. \quad (14)$$

Тогда функции F текущего риска равны:

$$\begin{aligned} F_1(x_1, x_2) &= (\beta_1 + \gamma_1 x_1 + \psi x_2) a_1 (1 - x_1) - \\ &- (\beta_2 + \gamma_2 x_2 + \psi x_1) \theta_2 x_2; \\ F_2(x_1, x_2) &= (\beta_2 + \gamma_2 x_2 + \psi x_1) a_2 (1 - x_2) - \\ &- (\beta_1 + \gamma_1 x_1 + \psi x_2) \theta_1 x_1. \end{aligned} \quad (15)$$

Условие оптимальности (12) используется для того, чтобы найти координаты минимального риска в различных ситуациях, используя при этом метод рабочей точки [12]. Согласно этому методу достаточно определить эти координаты как функции от искомым коэффициентов и приравнять эти координаты друг другу, как показано в работе [13].

Приведем координаты риска $C_0, C_1^+, C_2^+, C_1^-, C_2^-, C_{12}^{+-}$ в окрестности рабочей точки (x_{p1}, x_{p2}) :

$$\begin{aligned} C_0(a_1 + a_2) &= a_1 F_2(x_{p1}, x_{p2}) + a_2 F_1(x_{p1}, x_{p2}); \\ C_1^+ &= F_2(x_{p1} + \Delta x, x_{p2}); \\ C_2^+ &= F_1(x_{p1}, x_{p2} + \Delta x); \\ C_1^- &= F_1(x_{p1} - \Delta x, x_{p2}); \\ C_2^- &= F_1(x_{p1}, x_{p2} - \Delta x); \\ C_{12}^{+-} &= F_2(x_{p1} + \Delta x, x_{p2} - \Delta x). \end{aligned} \quad (16)$$

Найденные значения координат риска в виде аналитических функций от коэффициентов β_i, γ_i, ψ позволяют с помощью уравнения Беллмана (12), из которого легко получить систему пяти дифференциальных уравнений относительно этих коэффициентов, построить линию переключения от одного навыка на другой при их освоении:

$$\Delta F = F_1(x_1, x_2) - F_2(x_1, x_2). \quad (17)$$

Ниже на рис. 3 приведен результат компьютерного моделирования процесса обучения

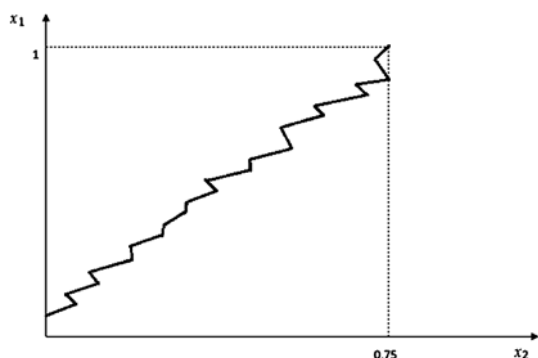


Рис. 3. График моделирования чередования двух простых навыков

двум одинаково главным простым навыкам и сама линия переключения.

Проведенные расчеты с учетом динамики чередования навыков можно упростить, если считать параметры a_j и θ_j одинаковыми, а процесс распределения общего времени между навыками рассматривать как задачу параметрической оптимизации одного занятия на одном шаге обучения. Тогда очевидно, что в случае параллельного освоения только простых навыков время на освоение каждого из них равно

$$t_1 = \frac{\Delta t}{n}; \quad \tau_1 = \frac{(n-1)\Delta t}{n}; \quad t_1 + \tau_1 = \Delta t, \quad (18)$$

где $\Delta t = T/\lambda$ — время одного занятия при заданной частоте чередования λ .

Рассмотрим теперь более сложные случаи совместного освоения сложного и простых навыков, дополнительно учитывая в терминальном критерии (9) только аддитивную составляющую.

Случай освоения одного простого и одного сложного навыков ($m = 1, n = 1$)

В этом случае можно получить более сложное условие оптимальности Беллмана с учетом двух дифференциальных уравнений для описания процесса освоения сложного навыка и промоделировать его на компьютере. Вместе с тем, если использовать в терминальном критерии (9) только первую аддитивную составляющую, то в рамках параметрической оптимизации можно записать следующее условие на одном шаге длительностью Δt :

$$\begin{aligned} J_1(\Delta t) + J_2(\Delta t) &= (1 - e^{-a_1 t}) e^{-\theta_1(\Delta t - t_1)} + \\ &+ (1 - e^{-b_1(\Delta t - \tau_1)}) e^{-v_1 \tau_1} \rightarrow \max. \end{aligned} \quad (19)$$

Аппроксимируя экспоненциальные функции степенными полиномами второго порядка, получим в первом приближении следующий ответ:

$$\begin{aligned} \frac{2t_1}{\Delta t} &= \frac{b_1^2(1 + 0,5v_1\Delta t) -}{b_1^2(1 + 0,5v_1\Delta t) - (a_1\theta_1 + b_1v_1) +} \rightarrow \\ &\rightarrow \frac{-(a_1\theta_1 + b_1v_1)}{+ 0,5(a_1^2 - b_1^2) + 0,5\Delta t(a_1^2\theta_1 - \beta_1^2v_1)}, \end{aligned} \quad (20)$$

где Δt — общее время освоения навыков; t_1 — время, отводимое на освоение простого навыка.

Из формулы (20) видно, что поскольку $a_1^2 - b_1^2 + 0,5(a_1^2\theta_1 - \beta_1^2v_1) > 0$, то время $\tau_1 = \Delta t - t_1$

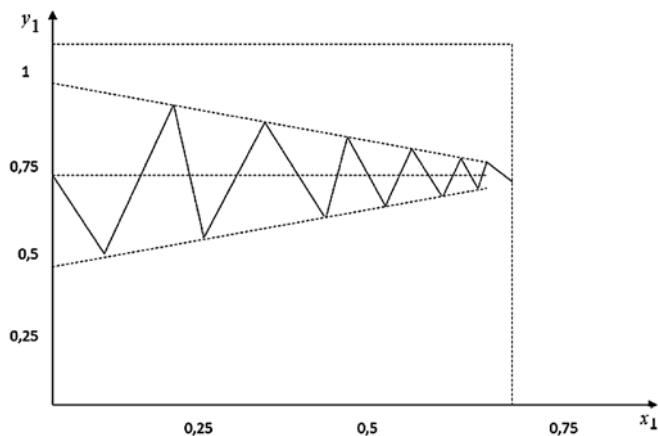


Рис. 4. График моделирования чередования простого x_1 и сложного y_1 навыков с постепенно увеличивающейся частотой

освоения сложного навыка всегда больше времени t_1 освоения простого навыка.

Следует подчеркнуть, что помимо приближенного расчета (20) результаты компьютерного моделирования в указанном случае показали, что частота чередования при освоении простого и сложного навыков постепенно при обучении возрастает, как показано на рис. 4.

При этом важно, что предельно максимальный уровень $y_1 = 0,6$ освоения сложного навыка ниже, чем уровень $x_1 = 0,8$ простого навыка.

Случай освоения n простых навыков и одного сложного навыка ($m = 1$)

В этом случае результат параметрической оптимизации имеет вид

$$J_1(\Delta t) + J_2(\Delta t) = (1 - e^{-b_1\tau_1})e^{-v_1(\Delta t - \tau_1)} + n \left(1 - e^{-\frac{a_1(\Delta t - \tau_1)}{n}} \right) e^{-\frac{\Delta t(n-1) - \tau_1}{n}\theta_1} \rightarrow \max,$$

а итогом выполнения условия экстремума является следующая формула:

$$\begin{aligned} & \frac{2\tau_1}{\Delta t} = \\ & \frac{\frac{a_1^2}{n} \left[1 - \frac{(n-1,5)}{n} \theta_1 \Delta t \right] -}{\frac{a_1^2}{n} \left[1 - \frac{(n-1,5)}{n} \theta_1 \Delta t \right] - \left[\frac{a_1 \theta_1 (2-n)}{n} + b_1 v_1 \right] -} \rightarrow (21) \\ & \frac{- \left[\frac{a_1 \theta_1 (2-n)}{n} + b_1 v_1 \right]}{- \frac{a_1 \theta_1 (n-1)}{n} + \frac{1}{2} \left(b_1^2 - \frac{a_1^2}{n} \right) + \frac{\Delta t}{2} \left(b_1^2 v_1 - \frac{a_1 \theta_1}{n} \right)}. \end{aligned}$$

Сделав некоторые упрощения, можно получить следующие оценки распределения общего времени Δt одного занятия на время τ_1 освоения сложного навыка, время t_1 освоения каждого из n простых навыков и времени t_0 повторения части простых навыков:

$$\begin{aligned} \frac{t_1}{\Delta t} &= \frac{a_1^2}{b_1^2 + n a_1^2}; \quad \frac{\tau_1}{\Delta t} = \frac{b_1^2}{b_1^2 + n a_1^2}; \\ \frac{t_0}{\Delta t} &= \frac{(1-d)e^{b_1 d \Delta t}}{e^{b_1 d \Delta t} + 10n}, \end{aligned} \quad (22)$$

где $d = \frac{a_1^2}{a_1^2 + n b_1^2}$. Например, пусть $n = 4$, $d = 0,4$; $b_1 \Delta t = 5$.

Тогда получим $t_1 = 0,14 \Delta t$; $\tau_1 = 0,4 \Delta t$; $t_0 = 0,04 \Delta t$.

Подводя итоги освоения увеличивающегося числа осваиваемых навыков, можно прийти к выводу, что **все более важным параметром обучения является оптимальная частота λ их чередования**, чтобы более длительные процессы забывания минимизировать за счет повторения навыков. Увеличение этой частоты есть следствие не линейной, а экспоненциальной и логистической зависимостей роста и падения уровней освоения навыков.

Многорежимное управление процессом обучения в компьютерном классе

Сделав предварительные выводы об оптимальном распределении времени при параллельном освоении простых и сложных навыков и о растущей частоте их чередования, можно представить следующие наиболее характерные режимы.

Режим 1 — в одном занятии за время чередуются определенные два простых навыка, в другой раз — другие два простых навыка и т. д.

Режим 2 — в одном занятии чередуются несколько простых навыков, постепенно их число увеличивается до значения n .

Режим 3 — в одном занятии чередуются один простой и один сложный навыки.

Режим 4 — в одном занятии чередуются сложный и несколько простых навыков, вплоть до их числа, равного n .

Последовательный переход от режима 1 к последующим режимам соответствует постепенному увеличению частоты чередования навыков, а это, в свою очередь, приводит к тому, что при кратковременном повышении качества ос-

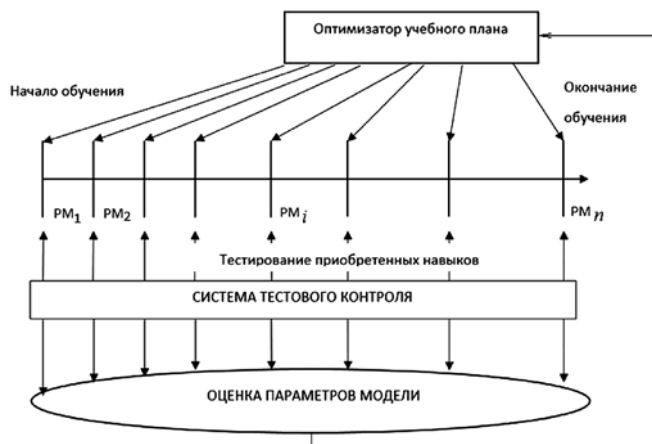


Рис. 5. Блок-схема системы компьютерного обучения:
 $PM_i, i = 1, \dots, N$, — рабочие места в компьютерном классе

ваиваемого навыка другие навыки "не успевают" потерять свои кондиции, и в конце концов процесс достижения итоговой максимальной подготовленности обучаемого лица приходит в установившееся состояние. Естественно, что эти состояния для сильной и слабой группы различны.

Схема автоматизированной системы компьютерного обучения представлена на рис. 5.

Задача оптимального управления решается следующим образом:

- с помощью первоначального задания для каждого обучаемого лица оцениваются скорости его освоения и забывания простых и сложных навыков;
- с использованием полученных оценок решается задача распределения времени параллельного обучения на освоение каждого из навыков;
- при получении итоговых оценок принимается решение о переходе на новый режим с увеличенной частотой чередования навыков в одном занятии;
- процессы получения оценок качества освоения, распределения времени между навыками и назначения частоты чередования осуществляются автоматически для каждого обучаемого лица или каждой группы с учетом их индивидуальных особенностей.

Примеры использования предложенного подхода в различных прикладных задачах обучения

В данном разделе рассмотрим процессы автоматизированного освоения навыков в трех

различных, внешне непохожих предметных областях.

В первой задаче — задаче автоматизированного обучения операторов управления сложными техническими системами (СТС) различного назначения можно выделить следующие основные функции: управление техническим компонентом, контроль технического состояния, техническое обслуживание, поиск и устранение неисправностей, проверка функционирования.

Под простыми навыками можно понимать выполнение заданий, не требующих значительного времени, определяющих деятельность специалиста по взаимодействию с техническими системами, которые отражают:

- степень овладения физико-теоретическими основами процессов функционирования систем (как минимум — определить нормальность или аномальность хода этих процессов);
- полноту знаний и умений пользоваться инструкциями и наставлениями, боевой документацией, которые регламентируют деятельность специалиста в процессе технического обслуживания систем;
- умение правильно и своевременно выполнять набор требуемых режимов и эксплуатационных процедур в процессе взаимодействия с техникой;
- знание технических и тактических возможностей систем и умение эти возможности реализовать в конкретных условиях эксплуатации техники.

Под сложными навыками понимается выполнение заданий, уровень освоения которых зависит от освоения предшествующих простых навыков, на которых базируется рассматриваемый навык.

Динамика освоения сложного навыка определяется способностью оператора к сохранению заданной эффективности работы при усложнении окружающей обстановки, с учетом следующих факторов:

- долговременная выносливость — сохранение человеком работоспособности на заданном уровне в течение определенного времени;
- устойчивость к воздействию факторов среды (температуры, влажности, давления, шума, ускорения), связанная с состоянием нервной системы оператора;
- работоспособность в экстремальных условиях, или способность принимать правильные решения при дефиците времени, в аварийных ситуациях, в условиях шумов и пр.;

- переключаемость — время "вхождения" в новую деятельность, которое определяется индивидуальными особенностями каждого человека.

Критерием успешности автоматизированного обучения является достижение максимума общего уровня освоения всех простых и сложных навыков. При оценке уровня профессиональной подготовки операторов используются следующие частные критерии:

- число правильных ответов из общего числа вопросов, заданных в процессе текущего, этапного и итогового контроля;
- коэффициент значимости знаний и практических навыков по отдельным разделам;
- точность выполнения операций (степень соответствия работы оператора при выполнении практического задания процедурам, изложенным в инструкции);
- продолжительность выполнения операций (в тех случаях, когда продолжительность выполнения отдельной операции или работы в целом имеет регламентированное значение).

Автоматизация обучения операторов на тренажерах при самоподготовке, при документировании процесса тренировок и возможность вернуться в нужную предыдущую точку в целях повторения навыков резко повышает качество подготовки операторов.

Вторую задачу — задачу автоматизации самостоятельного изучения иностранного языка в компьютерном классе можно представить в виде освоения различных видов речевой деятельности (чтения, аудирования, письменной и разговорной речи) и таких мыслительных процессов, как перевод с иностранного языка на русский и перевод с русского языка на иностранный, а также изучение таких разделов иностранного языка, как грамматика, лексикология, фонетика и др. По результатам освоения и закрепления каждого из этих умений и навыков (например, во время прохождения различных тестов) обучаемое лицо получает свою оценку, и обычным критерием общей оценки является среднее значение отдельных показателей или их общая сумма баллов.

Однако в ряде случаев эта оценка не учитывает как различную важность отдельных оценок, так и их возможную несбалансированность. Например, одну и ту же сумму баллов пяти оценок имеют показатели двух обучаемых лиц — 3, 3, 3, 3, 4 и 2, 2, 3, 4, 5, в то же время как первое обучаемое лицо явно предпо-

читительнее второго, у которого произведение показателей явно меньше, чем у первого.

Представление общей оценки в виде взвешенной суммы аддитивной и мультипликативной сверток используется в компьютерном классе без участия преподавателя при оптимальном распределении времени освоения и закрепления каждого из вышеназванных умений и навыков обучаемым лицом с учетом его индивидуальных способностей.

В свою очередь, задача оптимального управления обучением студентов решается при следующих допущениях.

1. На первом этапе каждый студент на экране своего компьютера получает первое задание по одному из видов речевой деятельности или одному из разделов иностранного языка, и через некоторое время по его ответам автоматически оценивается первоначальная скорость его освоения. Количество времени на выполнение студентами задания определяет качественную оценку скорости освоения каждого задания — обычно 3, 4 или 5 баллов.

2. Процесс изучения переключается на другой вид речевой деятельности или раздел иностранного языка, затем на третий и т. д., после чего полученные оценки скорости освоения позволяют разбить осваиваемые каждым студентом навыки на простые и сложные.

3. На втором этапе решается задача распределения оставшегося общего времени на повышение качества изучения и закрепления определенного набора умений и навыков при освоении иностранного языка. При этом единым критерием качества является сумма аддитивной и мультипликативной сверток оценок, полученных на первом этапе.

4. При получении итоговых оценок, вычисленных на компьютере автоматическим путем, учитывается неодинаковая важность изучаемых разделов иностранного языка и видов речевой деятельности.

5. Процесс выдачи подготовленных тестовых заданий студентам, оценка качества их освоения и индивидуальное распределение времени освоения простых и сложных навыков осуществляется без участия преподавателя в зависимости от индивидуальных особенностей студентов.

В настоящее время очередной задачей является формирование тестов различного уровня сложности для всех видов речевой деятельности и необходимых разделов иностранного языка и

определение частоты их чередования в начале, середине и в конце обучения студентов.

Третьей предметной областью является процесс планирования индивидуальных и общекомандных тренировок игроков при подготовке к соревнованиям по спортивным играм. В качестве простых навыков можно назвать различные элементы физической и технической подготовки, в качестве сложных — взаимодействие отдельных звеньев игроков команды в защите и нападении, отработка различных тактических действий команды в целом. При этом также важно учесть индивидуальные особенности каждого игрока, его сильные и слабые стороны, для чего в профессиональной команде уже сейчас с помощью компьютера формируется досье персональных показателей (физических параметров игрока, его игровых показателей, биометрических данных сердечной деятельности после каждого тестового упражнения и др.).

При этом в процессе подготовки к соревнованиям соблюдается главное правило — в начале тренировочного процесса частота чередования различных навыков при их освоении минимальна, затем она постепенно увеличивается.

Вершиной наибольшей эффективности этого подхода является знаменитый "поточный метод тренировок в хоккее", разработанный и приведший к выдающимся результатам величайшим в мире тренером по игровым видам спорта, заслуженным тренером СССР Анатолием Владимировичем Тарасовым.

Идея этого метода состоит в том, что по мере приближения к соревнованиям частота чередования навыков и интенсивность нагрузки достигает своего мыслимого предела — повторение всех навыков и умений происходит в непрерывном потоке без отдыха не только на каждой тренировке, но и практически в каждом упражнении этой тренировки.

Можно привести пример одной такой тренировки:

- в начале упражнения нападающая пара хоккеистов, владея шайбой, атакует ворота противника, защищаемые вратарем и противником;
- после совершения броска в случае успеха игроки нападения отправляются в один из углов площадки за воротами и выполняют физическое упражнение с "обычным" отягощением, игроки защиты должны в качестве "наказания" сделать физическое упражнение с увеличенной нагрузкой. В случае неуспеха

нападающие и защитники при выполнении физической нагрузки меняются ролями;

- сразу же после выполнения физического упражнения, не отдыхая, нападающие становятся в защиту против следующей пары, и затем выполняют нужную физическую нагрузку второй раз, после чего возвращаются к своим воротам в очередь — на "поток" для повторения этого упражнения 10...15 раз;
- в конце занятия в журнале фиксируются показатели сердечной деятельности и все ошибки каждого игрока в защите и нападении.

В итоге контролируемая общая физическая нагрузка и частота чередования навыков становятся в несколько раз выше, чем в игре, и в соответствии с принципом "тяжело в учении — легко в бою" команда сборной СССР по хоккею с шайбой на протяжении нескольких десятилетий была многократным чемпионом мира и олимпийских игр.

Заключение

1. При параллельном обучении простым и сложным навыкам предложено проводить тестирование обучаемых лиц в целях идентификации скоростей освоения и забывания этих навыков и применять экспоненциальную модель для оценки уровня подготовки обучаемой группы.

2. С помощью динамического программирования решена задача распределения времени обучения между простыми и сложными навыками с учетом индивидуальных особенностей обучаемых лиц.

3. Показано, что для каждого обучаемого лица или группы существует предельный уровень подготовки в зависимости от индивидуальных особенностей обучаемых лиц, по достижении которого дальнейшее обучение нецелесообразно.

4. При освоении сложных навыков наряду с простыми навыками частота переключения должна постепенно возрастать и достигать максимума у сильной группы обучаемых лиц.

5. Приведенные примеры оптимизации компьютерного обучения в различных предметных областях подтвердили универсальность предложенного подхода.

Список литературы

1. Канушкин С. В., Лебедев Г. Н., Чан Ван Туен, Швыдченко К. И. Подготовка операторов безотказного управле-

ния сложными подвижными объектами при регулируемом регламенте проведения проверок // Мехатроника, автоматизация, управление. 2012. № 3. С. 67–70.

2. **Лебедев Г. Н., Романов О. В., Алексеев А. Ю.** Проблемы построения компьютерных систем обучения специалистов ракетно-космических комплексов // Авиакосмическое приборостроение. 2003. № 9. С. 25–31.

3. **Лебедев Г. Н.** Интеллектуальные системы управления и их обучение с помощью методов автоматизации. М.: Изд. МАИ, 2002. 124 с.

4. **Лебедев Г. Н., Ву С. Д.** Применение динамического программирования при автоматизированном обучении операторов управления воздушным движением // Труды МАИ, № 44, 2011. // <http://elibrary.ru/item.asp?id=20354343>, ISSN:0044-4448.

5. **Беллман Р.** Динамическое программирование. М.: ИИЛ, 1960. 161 с.

6. **Бессарабов Н. А., Бондаренко А. В., Кондратенко Т. Н.** Задача выбора математической модели тестирования знаний для дихотомических данных // Вестник компьютерных и информационных технологий. 2016. № 2. С. 46–51.

7. **Бессарабов Н. А., Бондаренко А. В., Кондратенко Т. Н.** Автоматизированная система управления процессом тестирования знаний // Вестник компьютерных и информационных технологий. 2016. № 5. С. 45–51.

8. **Шукшунов В. Е., Циблиев В. В., Потоцкий С. И.** Тренажерные комплексы и тренажеры. Технологии разработки и опыт эксплуатации. М.: Машиностроение, 2005. 384 с.

9. **Нейман Ю. М., Хлебников В. А.** Введение в теорию моделирования и параметризации педагогических тестов. М.: Прометей, 2000. 168 с.

10. **Кузнецова Т. И., Царегородцева М. Г.** Комплексная оценка качества изучения иностранного языка в виде суммы аддитивной и мультипликативной сверток отдельных показателей освоения различных разделов на всех этапах обучения // Сб. трудов XXV Международной научно-технической конференции "Современные технологии в задачах управления, автоматизации и обработки информации", Алушта, 2016. М.: Технология, 2016. С. 208.

11. **Царегородцева М. Г., Кузнецова Т. И.** Постановка задачи оптимального выбора времени при самостоятельном изучении разделов иностранного языка в компьютерном классе с учетом неодинаковой сложности заданий и индивидуальных особенностей обучаемых лиц в студенческой группе // Сб. трудов XXV Международной научно-технической конференции "Современные технологии в задачах управления, автоматизации и обработки информации", Алушта, 2016. М.: Технология, 2016. С. 209.

12. **Baker F. B., Kim S. H.** Item Response Theory: Parameter Estimation Techniques. NY: CRC Press, 2004. 528 p.

13. **Kang T., Cohen A. S.** IRT Model Selection Methods for Dichotomous Items // Applied Psychological Measurement. 2007. V. 31, N. 4. P. 331–358.

14. **Claeskens G., Hjort N. L.** Model Selection and Model Averaging. NY: Cambridge University Press, 2008. 332 p.

15. **Ayala R. J.** The Theory and Practice of Item Response Theory. NY: Guilford Press, 2009. 448 p.

16. **Chalmers R. P.** Mirt: A Multidimensional Item Response Theory Package for the R Environment // Journal of Statistical Software. 2012. Vol. 48, N. 6. P. 1–29.

G. N. Lebedev, Dr. of Ph., Professor,

"Moscow aviation institute" (National Research University), 125080, Moscow, Russian Federation,

S. V. Kanushkin, Ph. D., Associate Professor,

Serpukhov branch of the HA RSF, Serpukhov, Russian Federation,

T. I. Kuznetsova, Dr. of Ph., Professor, The Head of department of foreign languages, englishmail@mail.ru,

M. G. Tsaregorodtseva, Senior lecturer of department of foreign languages, marina77868@yandex.ru,

Mendeleev University of Chemical Technology of Russia, Moscow, Russian Federation

Automated Computer-Based Individual Learning System for Mastering Simple and Complex Skills

The task is to automate individual computer training in simple and complex skills in various subject areas. A solution to the problem of optimization of training to a group of skills for a given time is formulated on the criterion of the maximum of the sum of additive and multiplicative convolutions of obtained estimations of the development of every skill. The main conclusion about the importance of increasing the frequency of alternation of mastering skills by the end of the training has been obtained. Examples of the effectiveness of the proposed approach in various subject areas are given. The task was set to automate individual computer training in simple and complex skills in various subject areas. The performance of tasks that do not require significant time is understood by simple skills, when initially the rate of mastering is maximal. Complex skills are meant as the fulfillment of tasks, the level of development of which depends on the development of the previous skills on which the skill in question is based. The dynamics of learning and forgetting skills, in so doing, is described by differential equations. The decision of a problem of optimization of training to a group of skills for a given time is formulated on the criterion of the maximum of the sum of additive and multiplicative convolutions of obtained estimations of development of every skill. With parallel learning of simple and complex skills, it is suggested to conduct testing of trainees in order to identify the rates of mastering and forgetting these skills and apply an exponential model to assess the level of training of the student group. With the help of dynamic programming, the problem of the distribution of learning time between simple and complex skills is solved, taking into account the individual characteristics of the trainees. When mastering complex skills along with simple skills, the switching frequency should gradually increase and reach a maximum for a strong group of trainees. Examples of the effectiveness of the proposed approach in various tasks are given:

automated training of management operators for complex technical systems for various purposes, automation of independent study by students of a foreign language in a computer class and computer planning of individual and team-wide training of players in preparation for competitions in sports games. The main conclusion about the importance of increasing the frequency of alternation of mastering skills by the end of the training has been obtained.

Keywords: simple and complex skills, automation of training, computer class, optimization of training time, dynamic programming, frequency of alternation

DOI: 10.17587/it.24.731-740

References

1. **Kanushkin S. V., Lebedev G. N., Chan Van Tuen, Shvydchenko K. I.** Podgotovka operatorov bezotkaznogo upravlenija slozhnymi podvizhnymi obektami pri reguliruemom reglamente provedenija proverok (Training of operators of trouble-free control of complex mobile objects under the regulated procedure for conducting inspections), *Mehatronika, Avtomatizacija, Upravlenie*, 2012, no. 3 (in Russian).
2. **Lebedev G. N., Romanov O. V., Alekseev A. Ju.** Problemy postroenija komp'juternyh system obucheniya specialistov raketno-kosmicheskikh kompleksov (Problems of building computer systems for training specialists in rocket and space complexes), *Aviakosmicheskoe Priborostroenie*, 2003, no. 9, pp. 25–31 (in Russian).
3. **Lebedev G. N.** *Intellektual'nye sistemy upravlenija i ih obuchenie s pomoshh'ju metodov avtomatizacii* (Intelligent control systems and their training using automation methods), Moscow, Publishing house of MAI, 2002, 124 p. (in Russian).
4. **Lebedev G. N., Vu Suan Dyk** Primenenie dinamicheskogo programirovanija pri avtomatizirovannom obuchenii operatorov upravlenija vozdušnym dvizheniem (The use of dynamic programming in the automated training of air traffic control operators), *Proceedings MAI*, no. 44, 2011. <http://elibrary.ru/item.asp?id=20354343> (in Russian).
5. **Bellman R.** *Dinamicheskoe programirovanie* (Dynamic programming), Moscow, IIL, 1960, 161 p. (in Russian).
6. **Bessarabov N. A., Bondarenko A. V., Kondratenko T. N.** Zadacha vybora matematicheskoj modeli testirovanija znanij dlja dihotomicheskikh dannyh (The problem of choosing a mathematical model of knowledge testing for dichotomous data), *Vestnik Komp'juternyh i Informacionnyh Tehnologij*, 2016, no. 2, pp. 46–51 (in Russian).
7. **Bessarabov N. A., Bondarenko A. V., Kondratenko T. N.** Avtomatizirovannaja sistema upravlenija processom testirovanija znanij (Automated control system for the testing of knowledge), *Vestnik komp'juternyh i informacionnyh tehnologij*, 2016, no. 5, pp. 45–51 (in Russian).
8. **Shukshunov V. E., Cibliev V. V., Potockij S. I.** *Trenazhernye komplekсы i trenazhery. Tehnologii razrabotki i opyt jekspluatacii* (Training complexes and simulators), Mashinostroenie, Moscow, Mechanical Engineering, 2005, 384 p. (in Russian).
9. **Nejman Ju. M., Hlebnikov V. A.** *Vvedenie v teoriju modelirovanija i parametrizacii pedagogicheskikh testov* (Introduction to the theory of modeling and parametrization of pedagogical tests), *Prometej*, 2000, 168 p. (in Russian).
10. **Kuznecova T. I., Tsaregorodtseva M. G.** Kompleksnaja ocenka kachestva izuchenija inostrannogo jazyka v vide summy additivnoj i mul'tiplikativnoj svertok otdel'nyh pokazatelej osvoenija razlichnyh razdelov na vseh jetapah obucheniya (Complex evaluation of the quality of learning a foreign language in the form of a sum of additive and multiplicative convolutions of separate indicators of the development of various sections at all stages of learning), *Sb. trudov HHV Mezhduнародного nauchno-tehnicheskogo seminaru "Sovremennye tehnologii v zadachah upravlenija, avtomatiki i obrabotki informacii"*, Alushta, 2016, Moscow, Technology, 2016, 208 p. (in Russian).
11. **Tsaregorodtseva M. G., Kuznecova T. I.** Postanovka zadachi optimal'nogo vybora vremeni pri samostojatel'nom izuchenii razdelov inostrannogo jazyka v komp'juternom klasse s uchetom neodinakovoj slozhnosti zadaniy i individual'nyh osobennostej obuchaemyh lic v studencheskoj gruppe (Formulation of the problem of the optimal timing for a self-study of foreign language sections in a computer class, taking into account the uneven complexity of tasks and the individual abilities of the learners in a student group), *Sb. trudov HHV Mezhduнародnoj jnauchno-tehnicheskoi konferencii "Sovremennye tehnologii v zadachah upravlenija, avtomatik i iobrabotki informacii"*, Alushta, 2016, Moscow, Technology, 2016, 209 p. (in Russian).
12. **Baker F. B., Kim S. H.** *Item Response Theory: Parameter Estimation Techniques*, NY, CRC Press, 2004, 528 p.
13. **Kang T., Cohen A. S.** IRT Model Selection Methods for Dichotomous Items, *Applied Psychological Measurement*, 2007, vol. 31, no. 4, pp. 331–358.
14. **Claeskens G., Hjort N. L.** *Model Selection and Model Averaging*, NY, Cambridge University Press, 2008, 332 p.
15. **Ayala R. J.** *The Theory and Practice of Item Response Theory*, NY, Guilford Press, 2009, 448 p.
16. **Chalmers R. P.** Mirt: A Multidimensional Item Response Theory Package for the R Environment, *Journal of Statistical Software*, 2012, vol. 48, no. 6, pp. 1–29.

Е. В. Кузнецова, канд. экон. наук, доц., доц. каф. бизнес-аналитики школы бизнес-информатики Национального исследовательского университета "Высшая школа экономики", Москва, e-mail: Ev.Kuznetsova@hse.ru,
Е. В. Усадова, бизнес-аналитик, e-mail: lena-220994@yandex.ru, ООО "Глоубайт Консалтинг", Москва

Управление проектными рисками в организациях сферы ИТ-услуг

Показано, что применение гибких подходов в управлении ИТ-проектами является конкурентным преимуществом для организаций сферы ИТ-услуг. Однако ИТ-проекты с высокой степенью неопределенности имеют высокий уровень сложности, изменчивости и риска. Авторами предложен тиражируемый подход к управлению рисками ИТ-проектов, команды которых используют методологию Scrum. Показаны возможности применения современного ПО для автоматизации процессов управления рисками. Предлагаемый подход апробирован на практике и проиллюстрирован в статье практическими примерами.

Ключевые слова: управление ИТ-проектами, управление рисками, гибкие методологии управления проектами, Scrum

Введение

Проектной деятельности в целом присуща высокая степень неопределенности, обусловленная новизной получаемого результата и/или способом его достижения. Особенно ярко это свойство проявляется в ИТ-проектах, сложных, подверженных частым изменениям, имеющих высокий уровень риска.

Вопросы управления рисками (УР) в целом достаточно широко представлены и полно исследованы в научной литературе в области управления проектами (УП), в международных и национальных стандартах УП. Ставшие традиционными подходы закреплены в стандартах [1–3]. Существуют стандарты, посвященные исключительно УР в проектной деятельности, например [4]. Несмотря на некоторые отличия в изложении вопросов риск-менеджмента, в этих источниках управление проектными рисками рассматривается исключительно в контексте проектной деятельности, вне связи с корпоративной системой управления рисками. Ограниченность такого подхода была отмечена в последней 6-й редакции стандарта РМВоК [1], где в качестве тенденций и формирующихся практик в области управления рисками проек-

та среди прочего выделена тенденция интегрированного управления рисками. Отмечено, что скоординированный подход к УР в масштабах всей организации обеспечивает согласованность и последовательность порядка управления на всех уровнях.

Интегрированное УР наиболее актуально для проектно-ориентированных организаций (ПОО), выполняющих проекты для внешних заказчиков. В данной работе авторы используют трактовку понятия ПОО, предложенную А. Д. Расселом: ПОО — это организации, основной бизнес (деятельность) которых составляют проекты. Стратегии роста в таких организациях находят свое отражение в характере, размерах, месте выполнения и роде проектов, предлагаемых компанией на рынке, а также в выборе способа обеспечения этих проектов ресурсами (внутреннее обеспечение или аутсорсинг [5]). ПОО противопоставляются проектно-зависимым организациям, операционная деятельность которых не является проектной, но их рост и развитие зависят от проектов.

Значительную часть ПОО составляют организации, работающие в сегменте ИТ-услуг. Данные услуги включают в себя системную интеграцию, ИТ-консалтинг, разработку заказно-

го программного обеспечения (ПО), установку и поддержку оборудования и ПО, ИТ-обучение и тренинги. После падения в 2014—2015 гг. рынок ИТ-услуг настроен в своих прогнозах оптимистично. Однако специалисты отмечают происходящую трансформацию модели управления ИТ-проектами. Экономия вынуждает заказчиков переходить на модели управления, позволяющие быстро достигать результата, что предполагает использование гибких (соответствующий английский термин — *Agile*) методологий УП. Учет требований ведется на основе непрерывно уточняемого документа, который обновляется регулярно, а приоритезация работ может изменяться по мере прогресса проекта. Однако большая вариативность приводит к более высокому уровню неопределенности и риска. Поэтому в УП с использованием адаптивных подходов риски необходимо рассматривать при выборе содержания каждой итерации; анализ, идентификация рисков и управление ими должны осуществляться также в ходе каждой итерации. Необходимо отметить, что УР в проектах, управление которыми осуществляется с использованием гибких методологий, практически не исследовано в научной литературе. Данные вопросы решаются практиками УП в отдельных проектах, но их опыт еще не нашел обобщения в научных публикациях.

В связи с изложенным в современных условиях создание корпоративной системы управления проектными рисками для ПОО, работающих в сфере ИТ-услуг и применяющих гибкие методологии УП, становится фактором выживания в конкурентной борьбе и одним из факторов роста на меняющемся рынке. При создании этой системы необходимо решить следующие задачи:

- 1) интегрировать процессы управления проектными рисками в общую систему управления рисками в компании и ИТ-рисками в частности;

- 2) разработать процесс управления рисками в рамках применения конкретной гибкой методологии УП;

- 3) выработать совместно с заказчиком проекта единый подход к управлению рисками в рамках конкретного проекта.

В настоящей работе предложены методы и средства решения указанного комплекса задач, проиллюстрированные на конкретном примере.

Методологические источники разработки

По мнению авторов, документами, наиболее полно описывающими риски в сфере ИТ на данный момент, являются документы, входящие в "Control Objectives for Information and Related Technologies", версии 5 — сокращенно COBIT5 [6]. COBIT5 представляет собой сборник стандартов, документов и лучших практик в области управления аудита и бизнеса в сфере ИТ. В состав COBIT5 входят два документа, имеющие непосредственное отношение к рискам: COBIT5 for Risk и Risk Scenarios Using COBIT5 for Risk, включающий в себя 111 типовых сценариев ИТ-рисков. COBIT5 основывается на идеях и моделях других стандартов и лучших практик по управлению рисками, таких как ISO 27005, ISO 31000, модель COSO, поэтому может быть использован как в рамках одной компании, так и при организации взаимодействия между заказчиками и исполнителями ИТ-проектов. Однако положения COBIT5 в части управления проектными рисками нуждаются в спецификации применительно к определенной используемой в организации методологии УП.

Наиболее популярной в настоящее время в России и в мире гибкой методологией УП является Скрим (Scrum). Более половины Agile-проектов реализуется с использованием данной методологии (рис. 1, см. третью сторону обложки) [7—9].

Данная методология с успехом применяется в ИТ-проектах и предполагает использование небольших, независимых, самоорганизующихся команд разработчиков, состоящих из Владельца продукта, Команды разработки и Scrum-мастера, а также короткие циклы выпуска (спринты), длительностью в один месяц. Основными мероприятиями Scrum являются планирование спринта, ежедневные 15-минутные встречи, обзор спринта и ретроспектива спринта. Scrum-команды создают продукт итеративно и инкрементально, максимально используя возможности для получения обратной связи [8]. Однако вопросы управления рисками в Scrum остаются непроработанными.

Еще одним документом, весьма популярным у практиков и широко используемым в качестве основы для разработки корпоративных политик, процедур, правил, инструментов и методов УП, является руководство РМВоК института PMI. В 2017 г. вышла его шестая ре-

дакция. Процессы УР детально проработаны в данном документе и используются на практике большим числом ПОО. В РМВоК 6 ed сделан определенный поворот в сторону применения гибких методологий, рассмотрены тенденции и формирующиеся практики в УП, приведены соображения для гибких/адаптивных сред. В 2017 г. PMI выпустил также руководство Agile Practice Guide [10], что делает возможным адаптацию подходов, изложенных в РМВоК, для проектов, управляемых на основе гибких методологий.

В соответствии с изложенным авторами принято решение разработать подход к управлению рисками для ИТ-проектов, управляемых на основе Scrum, с учетом рекомендаций COBIT5 и РМВоК 6 ed. В частности, для формализованного описания риска будут использованы положения COBIT5, а процессы, инструменты и методы будут основываться на РМВоК 6 ed, что позволит успешно решить поставленные выше задачи.

Средства автоматизации, используемые в Agile-проектах

Создание методологической составляющей УП и УР должно поддерживаться соответствующими информационными технологиями. Обычно в проектах, управляемых по Agile, используется совместно несколько программных продуктов, решающих различные задачи автоматизации процессов УП. Для рассматриваемой категории проектов критически важно обеспечить единое рабочее информационное пространство для участников проекта и эффективные коммуникации между ними в режиме онлайн. Ниже перечислены информационные системы (ИС), комплексное использование которых позволяет успешно автоматизировать процессы УП и УР с учетом специфики гибких методологий, и приведены примеры ИС, примененные для автоматизации управления конкретным ИТ-проектом создания корпоративного хранилища данных. Далее на примере будет показано, как данное ПО используется в процессах УР.

1. Система для верхнеуровневого календарного и ресурсного планирования — MS Project.

2. Система для детального планирования работ, отслеживания статуса выполнения задач, формирования бэклога, учета рабочего

времени, организации коммуникации команды проекта — bug-tracker Atlassian JIRA.

3. Программный продукт для хранения корпоративной базы знаний по УП и документации отдельных проектов, позволяющий совместно работать над контентом — вики-система Atlassian Confluence.

4. Программный продукт для хранения файлов кода и управления версиями кода — HG Mercurial.

5. Система, автоматизирующая процесс управления качеством продукта, — HP Quality Center.

6. Программный продукт, позволяющий создавать и хранить файлы в "облаке" для обеспечения совместного доступа членов команды и стейкхолдеров к информации из любой точки мира, — Google Docs.

7. Программный продукт для организации коммуникации участников проекта в режиме реального времени, поддерживающий видеоконференцсвязь, — Skype для бизнеса.

Планирование управления рисками

Рассмотрим далее процессы УР проектов на основе РМВоК 6 ed [1] в целях выявления особенностей этих процессов при использовании гибкой методологии Scrum в контрактных ИТ-проектах.

Согласно РМВоК первым процессом УР проекта является **планирование УР**. В качестве входов данного процесса в РМВоК 6 указаны: Устав проекта, План управления проектом, Реестр заинтересованных сторон, факторы среды предприятия и активы процессов организации. Выходом процесса является План управления рисками проекта.

Как отмечалось выше, в ПОО, выполняющих контрактные проекты, необходимо интегрировать процессы УР отдельно взятого проекта в систему риск-менеджмента организации в целом. Поэтому особое внимание должно быть уделено таким активам процессов организации, как корпоративные стандарты и шаблоны документов, используемых в процессах УР. На корпоративном уровне предварительно должен быть определен уровень риск-аппетита по организации в целом и уровень допустимого совокупного риска проекта; разработан классификатор рисков проектов, матрица вероятности и влияния, шаблоны реестра рисков и карточки риска, формы отчетности. Подходы

к УР в организациях заказчика и исполнителя проекта могут заметно различаться, в связи с чем в данном процессе их необходимо привести в соответствие для применения в рамках конкретного проекта. Руководителям проекта со стороны заказчика и исполнителя проекта на данном этапе необходимо согласовать:

- порядок расчета совокупного риска проекта;
- используемые методы качественного и количественного анализа рисков;
- матрицу "вероятность-влияние";
- шаблон реестра рисков;
- шаблон карточки риска;
- порядок взаимодействия команды проекта в управлении рисками;
- шаблоны отчетности по рискам, а также назначить риск-менеджера проекта. Им может быть как руководитель проекта, так и один из членов проектной команды. Такое назначение может проводиться в случае, если проектный менеджер руководит параллельно несколькими проектами или его ресурса недостаточно, чтобы осуществлять эффективное УР.

Документами, которые позволяют на практике формализовать и закрепить достигнутые договоренности, являются Устав проекта и План управления рисками проекта. По опыту авторов, План управления рисками проекта в качестве отдельного документа в проектах, управляемых по Agile, встречается крайне редко. Поэтому необходимо дополнить Устав проекта разделом, описывающим согласованный с исполнителем и заказчиком порядок УР проекта. Таким образом, дополненный Устав будет являться выходом процесса планирования УР.

Авторами предложен шаблон карточки риска, разработанный на основе положений COBIT5, так как в соответствии с данным фреймворком риск можно описать наиболее полно по пяти компонентам:

- источник — характеризует ту часть системы, где риск берет свое начало — внутри организации/проекта или за их пределами;
- тип угрозы — характеризует то, в чем проявляется риск;
- событие — характеризует то, как проявляется угроза;
- актив — объект, вовлеченный в риск, возникновение его причин, последствий, на который риск может влиять;
- время — отличительной особенностью COBIT5 является связь рисков с временными параметрами; параметры, характеризую-

щие риск во времени — это время наступления, длительность воздействия, скорость обнаружения, задержка влияния.

Заполнение карточки риска выполняют в ходе идентификации и качественного и количественного анализа рисков. Пример заполнения карточки риска приведен в табл. 1.

Таблица 1

Пример заполнения карточки риска

№ пункта	Показатель	Значение
1.	Проекты, в которых идентифицирован риск	Создание корпоративного хранилища для заказчика XXX
2.	Идентификатор риска	R-03
3.	Название	Задержка предоставления доступа к необходимым информационным ресурсам
4.	Категория риска	Организационный
5.	Описание риска	Не предоставлен в установленное время доступ к необходимым информационным ресурсам. Для предоставления доступа к информационным ресурсам требуется согласование непосредственного руководства. Ответственный за доступы сотрудник может забыть согласовать заявки на предоставление доступа.
6.	Последствия риска	Сдвиг сроков завершения текущей части проекта. Также, отсутствие информации повышает уровень неопределенности в задачах сотрудников, вследствие чего работы могут быть выполнены с ошибками.
Компоненты риска		
7.	Тип угрозы	Случайная
8.	Источник	Внутренний Человек
9.	Событие, за которым следует риск	Неисполнение правил и требований
10.	Актив, который является причиной риска	Персонал и навыки
11.	Параметр проекта, на который риск оказывает воздействие	Сроки Стоимость Качество
12.	Время наступления риска	Критичное

№ пункта	Показатель	Значение
13.	Длительность воздействия риска на актив	Среднее
14.	Скорость обнаружения риска	Высокая
15.	Задержка влияния риска	Нет
Качественная оценка риска		
16.	Вероятность	Небольшая (0,3)
17.	Последствия	Небольшие (0,4)
18.	Оценка	0,12
Количественная оценка риска		
19.	Выводы по результатам количественной оценки	Не проводилась
Реагирование на риск		
20.	Выбранная стратегия	Снижение
21.	Меры реагирования	Регламентировать срок, в который заказчиком должен быть предоставлен доступ к информационным ресурсам
22.	Проекты, в которых риск материализовался	
23.	Оценка эффективности мер реагирования	

За процессом планирования УР следуют следующие процессы: идентификация рисков; качественный анализ рисков; количественный анализ рисков; планирование реагирования на риски; осуществление реагирования на риски и мониторинг рисков. Ниже каждый из них будет рассмотрен подробнее. В связи с высокой подверженностью Agile-проектов изменениям, данные процессы УР должны повторяться на каждой итерации проекта. На рис. 2 приведена предложенная авторами схема УР, на которой процессы УР соотнесены со встречами, определенными по методологии Scrum.

Идентификация рисков

Идентификация рисков представляет собой процесс определения того, какие риски могут

повлиять на проект, и описания их характеристик. Входом процесса являются документы проекта, План управления рисками или соответствующие ему положения Устава, активы процессов организации (например, база знаний по рискам) и факторы среды предприятия. Определение рисков носит итеративный характер, поскольку новые риски могут быть обнаружены на любой стадии жизненного цикла проекта. Идентификация рисков проводится на встрече проектной команды, которая посвящена планированию спринта. Используются методы мозгового штурма и анализа базы знаний. Для определения триггеров рисков используется метод анализа ключевых факторов. Вся Scrum-команда должна быть вовлечена в процесс идентификации рисков, чтобы ее члены могли развивать и поддерживать чувство ответственности за конкретные риски и действия по реагированию на них. Для территориально-распределенных команд подобные встречи рекомендуется проводить с помощью видеоконференции.

Выходом процесса являются частично заполненные реестр рисков и карточки идентифицированных рисков. В карточке риска (см. табл. 1) в процессе идентификации заполняются пункты 1–14.

При разработке карточки риска авторами было предложено осуществить следующие изменения относительно описания риска по COBIT5.

1. "COBIT5 for Risk" рассматривает 111 типовых сценариев рисков и возможностей. Эти сценарии объединены в 20 групп, которые определяют категорию риска. Поскольку не все выделенные категории имеют отношение к проектным рискам, то авторами было принято решение заменить такую категоризацию рисков на использующуюся в компании-исполнителе проекта для ИТ-проектов. А именно: разделять риски на технические, управленческие, организационные и внешние.

2. Вместо актива, на который влияет риск, рассматривать параметр проекта, на который оказывают влияние стоимость, сроки, качество, объем работ (предметная область).

3. В соответствии с COBIT5 принято оценивать риски по их соответствию уровню влияния на:

- обеспечение/получение выгод от ИТ, реализацию ИТ-программ и ИТ-проектов;
- эксплуатацию ИТ;
- предоставление ИТ-услуг.

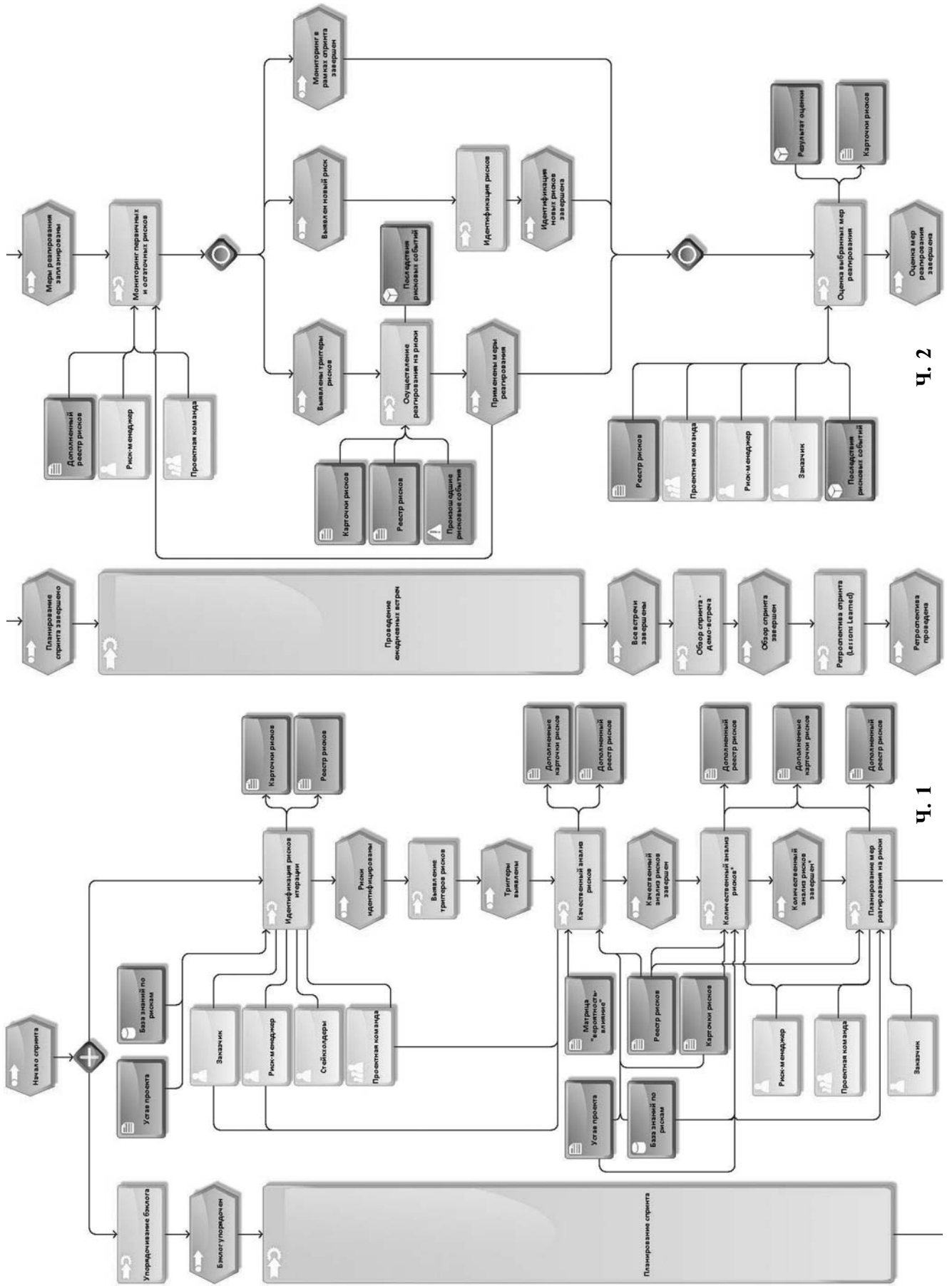


Рис. 2. Процессы УР, соотнесенные со встречами Scrum

Данную оценку было решено заменить на качественную оценку вероятности и влияния риска, так как все риски ИТ-проектов слабо влияют на эксплуатацию и предоставление ИТ-услуг и сильно влияют на реализацию ИТ-проектов. Исходя из этого оценка была бы одинакова для многих рисков ИТ-проектов.

В карточку риска также были добавлены описание мер реагирования и оценка эффективности их применения. В рассматриваемом примере карточки рисков, идентифицированных в ходе выполнения реализованных ранее ИТ-проектов, составляют базу знаний, хранящуюся в вики-системе Atlassian Confluence, что существенно сокращает время на описание идентифицированных рисков. Постепенно у проектных команд накапливается собственный опыт УР, пополняющий корпоративную базу знаний. Данные карточек рисков, идентифицированных в различных выполняемых проектах, можно использовать для анализа рисков и совершенствования УР на уровне компании.

По результатам идентификации рисков в реестре рисков проекта заполняются столбцы 1–15 (см. табл. 1). В рассматриваемом проекте реестр рисков формировали члены проектной команды в режиме онлайн с помощью совместно редактируемого Google-документа — Google Spreadsheet.

В системе MS Project были созданы таблица, содержащая реестр рисков, и настраиваемые поля, позволяющие указывать идентифицированные риски для задач проекта. Верхнеуровневый план проекта, сформированный в MS Project с использованием корпоративного шаблона, представлен на рис. 3 (см. третью сторону обложки). Для задач проекта указаны идентифицированные угрозы и возможности, а результат их оценки визуализирован с помощью графических индикаторов.

Качественный анализ рисков

Качественный анализ заключается в оценке приоритетов выявленных рисков с использованием вероятности их возникновения и воздействия на цели проекта. Вероятность возникновения и влияние рисков могут быть описаны качественными терминами, такими как очень высокие, высокие, средние, низкие и т. п. Такая оценка отражает отношение проектной команды и стейкхолдеров к идентифициро-

ванным рискам, является быстрым и эффективным средством определения приоритетов.

Входами процесса согласно РМВоК являются План управления проектом, документы проекта, факторы среды предприятия и активы процессов предприятия. Как было показано выше, в *Agile*-проектах план УР целесообразно заменить на соответствующие положения Устава проекта. Основными документами проекта, используемыми в данном процессе, являются реестр рисков и карточки идентифицированных рисков; основным активом процессов предприятия является корпоративная база знаний о завершенных проектах. Качественный анализ рисков требуется проводить в рамках каждой итерации работ по проекту в ходе встреч по планированию спринта. Для получения качественной оценки актуален опрос экспертов, используется Матрица вероятности и влияния. Итоговая оценка риска представляет собой произведение оценки вероятности возникновения на оценку последствий. Чем выше полученная риском оценка, тем выше его приоритет.

Для получения качественной оценки совокупного риска проекта авторами были экспертно определены значения весов для рисков разных категорий:

- для технических рисков — 0,5;
- для управленческих рисков — 0,2;
- для организационных рисков — 0,2;
- для внешних рисков — 0,1.

При таком подходе оценку совокупного риска проекта можно рассчитать следующим образом: просуммировать итоговые оценки рисков одинаковых типов, затем рассчитать сумму взвешенных суммарных оценок рисков разных типов. Итоговую сумму требуется поделить на число идентифицированных рисков, результатом будет общий показатель риска проекта.

Выходом процесса качественного анализа рисков являются дополненные результатами оценки карточки рисков и реестр рисков. В системе MS Project результаты оценки рисков представляются визуально с помощью графических индикаторов. Поскольку качественная оценка является субъективной, то она требует дальнейшего уточнения для наиболее приоритетных рисков.

Количественный анализ рисков

Под **количественным анализом** рисков понимается процесс численного анализа совокуп-

ного воздействия идентифицированных индивидуальных рисков проекта на цели проекта. В качестве основных методов количественного анализа рисков используют анализ чувствительности, метод деревьев решений, имитации, как правило, методом Монте-Карло.

Сложность применения количественного анализа рисков в рамках одной итерации в *Agile*-проектах обусловлена длительностью проведения и трудоемкостью перечисленных методов. Поэтому рекомендуется проводить качественный анализ рисков опционно в отношении рисков отдельных категорий с максимально высоким приоритетом по результатам количественного анализа. В рассматриваемом примере было принято решение проводить количественную оценку рисков для всех рисков, получивших качественную оценку 0,4 балла и выше.

В табл. 2 приведен сравнительный анализ перечисленных выше методов, позволяющий вы-

брать метод количественного анализа, применение которого целесообразно в *Agile*-проектах.

По мнению авторов, метод деревьев решений является наиболее приемлемым для применения в *Agile*-проектах в рамках каждой итерации. Данный метод сочетает в себе финансовый анализ и бизнес-кейсы, так как для каждого нового состояния проекта, вызванного риском, определяются финансовые выгоды или потери. Для каждого риска строится дерево, описывающее принятие решений в случае наступления рискованного события. Данный метод является наименее трудозатратным из рассмотренных выше и не требует специальной подготовки членов команды проекта для его применения.

На рис. 4 приведен пример построения дерева решений и расчет по нему показателя Ожидаемая стоимость узла вероятности (Expected Value, EV) для риска "Большое (сверх установ-

Таблица 2

Сравнительный анализ методов количественного анализа рисков

Название метода	Характеристика метода	Недостатки метода	Достоинства метода
Анализ чувствительности	Данный метод наглядно показывает влияние отдельных факторов риска на значения показателей эффективности проекта. Обычно в качестве таких показателей рассматриваются NPV, IRR или ROI. Чаще всего результаты визуализируются с помощью диаграммы "Торнадо"	Метод однокритериального анализа: рассчитывается воздействие одного из рисков факторов на один из параметров эффективности, остальные входные параметры остаются неизменными. Для проведения расчетов требуются финансовые данные, характеризующие текущее состояние проекта	Данный метод является хорошей иллюстрацией влияния отдельных факторов риска на конечный результат проекта
Анализ ожидаемой денежной стоимости (метод деревьев решений)	Предполагает формирование иных сценариев развития проекта помимо базового и анализ его эффективности при реализации этих сценариев. Разработка каждого варианта развития проекта сопровождается оценкой рисков и затрат. Визуализировать результаты применения метода можно с помощью деревьев решений	Ограничением практического использования данного метода является исходная предпосылка о том, что проект должен иметь доступное для обзора или допустимое число вариантов развития	Наглядный и часто используемый метод. Устраняет ограничение анализа чувствительности по числу факторов. Метод особенно полезен в ситуациях, когда решения, принимаемые в каждый момент времени, сильно зависят от решений, принятых ранее, и, в свою очередь, определяют сценарии дальнейшего развития событий
Моделирование (метод Монте-Карло)	В рамках применения метода используется модель, которая переводит неопределенности проекта в их потенциальное воздействие на цели проекта. Моделирование обычно выполняется с использованием метода Монте-Карло. Модель проекта вычисляется многократно, причем входные значения выбираются случайным образом для каждой итерации из вероятностных распределений этих переменных	Присутствует необходимость совершения большого числа итераций. Также в качестве недостатка можно выделить сложность восприятия полученных моделей, учитывающих большое число внешних и внутренних факторов	Одновременный учет максимально возможного числа факторов внешней среды, оценка имеет высокую точность

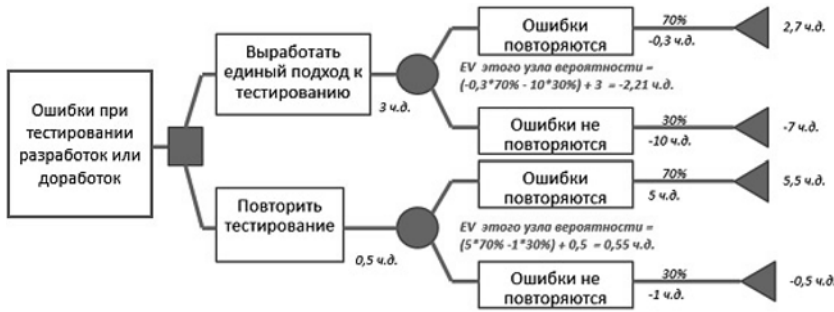


Рис. 4. Пример дерева решений

ленной нормы) число ошибок при тестировании разработок или доработок". Данный риск был качественно оценен проектной командой как высокий, в связи с чем команда проекта провела его количественный анализ. Дерево построено с использованием показателей трудозатрат (в человеко-днях), которые можно перевести в стоимость, умножив на ставку. По результатам построения дерева решений можно сделать вывод, что более выгодным вариантом решения будет разработка единого подхода к тестированию, так как при его выборе команда получит большую среднюю полезность каждой из альтернатив.

По результатам проведенного качественного анализа дополняются карточки рисков.

Планирование реагирования на риски

Планирование реагирования на риски заключается в выборе стратегии и конкретных мер реагирования на риски в соответствии с выбранной стратегией. Планирование мер

реагирования основывается на результатах качественной и количественной оценок. Планирование мер реагирования целесообразно выполнять для рисков, получивших по результатам качественного анализа оценку выше определенного уровня, например, не планировать для рисков, уровень которых оценен как низкий. Планирование реагирования на вновь идентифицированные риски также осуществ-

ляется в ходе каждой итерации на встрече по планированию спринта экспертным методом с использованием корпоративной базы знаний.

По результатам планирования реагирования на риски дополняют реестр рисков и карточки рисков: в них вносят выбранные стратегии реагирования и конкретные меры реагирования. На этом заполнение реестра рисков и карточек рисков завершается. Пример реестра рисков рассматриваемого проекта приведен в табл. 3.

Результаты анализа рисков, предложенные стратегии и меры реагирования необходимо валидировать у заказчика и других стейкхолдеров проекта. В связи с тем, что предусмотренные меры реагирования могут привести к изменению плана проекта и увеличению стоимости отдельных работ, необходимо сформировать запросы на изменения в соответствии с установленным в Уставе проекта порядком.

Для владельцев рисков в системе Atlassian JIRA могут быть созданы задачи мониторинга конкретных рисков и реагирования на риски.

Таблица 3

Пример реестра негативных рисков (угроз)

ID	Описание риска	Тип риска	Последствия риска	Триггеры риска	Владелец риска	Вероятность возникновения риска	Последствия	Оценка риска	Стратегия реагирования	Меры реагирования
1	2	3	4	5	6	7	8	9	10	11
R-01	Ошибки при осуществлении разработок или доработок	Технический	Необходимо переделать выполненные работы, что приводит к дополнительным затратам ресурсов	Несоответствие требующихся трудозатрат на разработку имеющимся трудозатратам	Иванов И. М.	Средняя (0,4)	Большие (0,6)	0,24	Снижение	Сформулировать общий для разработчиков порядок разработки

ID	Описание риска	Тип риска	Последствия риска	Триггеры риска	Владелец риска	Вероятность возникновения риска	Последствия	Оценка риска	Стратегия реагирования	Меры реагирования
1	2	3	4	5	6	7	8	9	10	11
R-02	Ошибки при тестировании разработок или доработок	Технический	Необходимо переделать выполненные работы, что приводит к дополнительным затратам ресурсов	Определение неверного подхода к тестированию, несоответствие требованиям трудозатрат на тестирование имеющимся трудозатратам	Иванов И. М.	Средняя (0,4)	Значительные (0,7)	0,28	Снижение	Разработать единый подход к тестированию
R-03	Задержка в предоставлении доступов к необходимым ресурсам	Организационный	Увеличение сроков, стоимости и качества работ проекта	Предоставление доступов не было заранее оговорено с заказчиком	Петрова О. А.	Небольшая (0,3)	Небольшие (0,4)	0,12	Снижение	Регламентировать срок, в который заказчиком должны быть предоставлены любые доступы
R-04	Неверная оценка требующихся доработок	Управленческий	Выполнены лишние доработки; требующиеся доработки не выполнены	Неправильно собраны требования с бизнес-заказчика, проблемы при планировании	Сидоров П. П.	Средняя (0,4)	Средние (0,5)	0,2	Уклонение	Регламентировать процесс сбора бизнес-требований, изменить его таким образом, чтобы исключить возможность неверной оценки
R-05	Изменение состава бизнес-заказчиков	Организационный	Изменение "курса" работ по проекту	Массовые организационные изменения в компании-заказчике	Петрова О. А.	Незначительная (0,1)	Критические (0,8)	0,08	Принятие	—
R-06	Необходимость некоторых сотрудников работать параллельно на двух проектах	Организационный	Доработки не выполнены к оговоренному сроку	В команду попали сотрудники, которые уже работают на другом проекте	Петрова О. А.	Незначительная (0,1)	Небольшие (0,4)	0,04	Уклонение	Не набирать в команду проекта сотрудников, которые не имеют возможности работать на полную ставку
R-07	Заказчик не согласовал документацию вовремя	Организационный	Сокращение времени, отведенного на разработку и тестирование	Задержка сроков согласования документации более чем на 5 дней	Петрова О. А.	Небольшая (0,3)	Значительные (0,7)	0,21	Снижение	Регламентировать срок, в который заказчиком должна согласовываться документация
R-08	Зависимость реализации доработок по проекту от сроков и технических решений по другим связанным проектам	Управленческий	Увеличение длительности проекта	Задержка сроков реализации связанных проектов более чем на две недели	Сидоров П. П.	Незначительная (0,1)	Значительные (0,7)	0,07	Уклонение	Заранее спланировать проект таким образом, чтобы большая часть работ не зависела от работ связанных проектов

Реагирование на риски, мониторинг рисков

В случае, если рисковое событие произошло или вот-вот произойдет, осуществляется **реагирование** на данный риск — применение разработанных мер реагирования в рамках выбранной стратегии. Реагирование на риски осуществляется в рамках каждого спринта.

Последним из процессов управления рисками является **мониторинг рисков** — отслеживание идентифицированных рисков, мониторинг остаточных и идентификация новых рисков, обеспечение применения планов реагирования на риски. Данный процесс позволяет извлечь максимальную выгоду от УР конкретного проекта, так как позволяет оптимизировать выбор стратегий и мер реагирования на риски в будущем.

Факты материализации рисков, применение мер реагирования, проблемы, выявленные в рамках мониторинга, обсуждаются на ежедневных встречах членов Scrum-команды. На ретроспективной встрече по окончании спринта проводится оценка эффективности выбранных стратегий и мер реагирования на риски. Результат оценки эффективности вносится в карточку риска.

Заключение

Предложенные авторами методы и средства решения задач управления рисками для проектов, управляемых на основе Scrum-методологии, прошли успешную апробацию в проекте создания корпоративного хранилища данных в российском банке. Руководитель проекта признал, что этот проект для него стал значительно более прозрачным, а значит и лучше управляе-

мым. Вовлеченность членов команды проекта со стороны исполнителя и заказчика в процессы УР сделала их проактивными и повысила эффективность мер реагирования.

По результатам апробации сделан вывод о возможности тиражирования данной разработки как в рамках ИТ-компании, реализующей описанный проект, так и в других ИТ-компаниях, выполняющих контрактные проекты для внешних заказчиков с применением методологий семейства *Agile*.

Список литературы

1. **A Gide** to the Project Management Body of Knowledge (PMBoK), Sixth ed / Project Management Institute, Inc., 14 Campus Boulevard, Newton Square, Pennsylvania 19073-32 99 USA, 2017, 756 p.
2. **ГОСТ Р ИСО 21500—2014**. Руководство по проектному менеджменту. М.: Стандарт-инфо, 2015. 45 с.
3. **ГОСТ Р 54869—2011**. Проектный менеджмент. Требования к управлению проектом. М.: Стандарт-инфо, 2012. 8 с.
4. **Practice Standard for Project Risk Management** / Project Management Institute, Inc., Newton Square, Pennsylvania 19073—3299 USA, 2009, 116 p.
5. **Арчибальд Р.** Управление высокотехнологичными программами и проектами / Пер. с англ. Мамонтова Е. В.; под ред. Баженова А. Д., Арефьева А. О. 3-е изд. М.: Компания АйТи; ДМК Пресс, 2010. 464 с.
6. **COBIT5**. A Business Framework for the Governance and Management of Enterprise IT // URL: https://www.oo2.fr/sites/default/files/document/pdf/cobit-5_res_eng_1012.pdf.
7. **Вольфсон Б.** Гибкие методологии разработки // URL: http://agilerussia.ru/methodologies/borisvolffson_ebook/
8. **Скрам-гайд** // URL: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Russian.pdf>.
9. **Version One** 12th Annual State of Agile Report // URL: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>.
10. **Agile Practice Guide** / Project Management Institute, Inc., 14 Campus Boulevard, Newton Square, Pennsylvania 19073-3299 USA, 2017. 168 p.

E. V. Kuznetsova, PhD, Associate Professor,

National Research University Higher School of Economics, Moscow, Russia, e-mail: Ev.Kuznetsova@hse.ru,

E. V. Usadova, Business analyst, Glowbyte Consulting Ltd, e-mail: lena-220994@yandex.ru

Project Risk Management in IT Industry Organizations

Nowadays any company's ability to quickly adapt to unstable market conditions is the key to the enterprise successful operation. Choosing the appropriate project management methodology and taking into account project features and management processes automaton provide the company's possibility to adapt to market volatility.

The paper shows that Agile approaches application to IT-project management provides a competitive advantage for IT services enterprises. However, high-uncertainty IT-projects have high rates of change, complexity and risk. The authors propose replicable approach to risk management for IT-projects implemented using Scrum Method. Authors have used COBIT5 regulations to create IT-project risks' formalized description. The processes, tools and methods for project risk management are based on PMBoK 6ed provisions. The risk management business process and document templates for it have been developed. The possibilities of using advanced software for risk management processes automation are shown. The proposed approach has been tested on real problems and illustrated by practical examples.

The approach has been successfully used in the real project execution.

Keywords: *IT-project Management, Risk Management, Agile approaches in project Management, Scrum*

DOI: 10/17587/it.24.741-752

References

1. **A Guide** to the Project Management Body of Knowledge (PMBOK), Sixth ed / Project Management Institute, Inc., Newton Square, Pennsylvania 19073-3299, USA, 2017, 756 p.
2. **GOST R ISO 21500—2014**. Rukovodstvo po proyekt-nomu menedzhmentu, Moscow, Standart-info. 2015, 45 p. (in Russian).
3. **GOST R 54869—2011**. Proektny'j menedzhment. Trebovaniya k upravleniyu proektom, Moscow, Standart-info, 2012, 8 p. (in Russian).
4. **Practice** Standard for Project Risk Management / Project Management Institute, Inc., 14 Campus Boulevard, Newton Square, Pennsylvania 19073-3299 USA, 2009, 116 p.
5. **Archibald R. D.** *Managing high-technology programs and projects*, 3rd ed. Moscow, Kompaniya Aj Ti; DMK Press, 2010, 464 p.
6. **COBIT5**. A Business Framework for the Governance and Management of Enterprise IT, available at: https://www.oo2.fr/sites/default/files/document/pdf/cobit-5_res_eng_1012.pdf.
7. **Vol'fson B.** *Gibkie metodologii razrabotki*, available at: http://agilerussia.ru/methodologies/borisvolfson_ebook/
8. **Skram-gajd**, available at: URL: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Russian.pdf>
9. **VersionOne** 12th Annual State of Agile Report, available at: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>
10. **Agile** Practice Guide. / Project Management Institute, Inc., Newton Square, Pennsylvania 19073-3299, USA, 2017, 168 p.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 12.09.2018. Подписано в печать 23.10.2018. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT1118. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.
