ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 22
2016
№ 3

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу **http://novtex.ru/IT.**
Журнал включен в систему Российского индекса научного цитирования.
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
INFORMACIONNYE TEHNOLOGII

Vol. 22
2016
No. 3

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

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"

УДК 004.272.43

А. Э. Саак, д-р техн. наук, доц., e-mail: saak@tgn.sfedu.ru,
Южный федеральный университет, г. Таганрог

Кольцевые алгоритмы диспетчеризации массивами заявок в Grid-системах

В дальнейшем развитии среды ресурсных прямоугольников, как основы теории полиномиальной диспетчеризации, определяются операции динамического интегрирования ресурсных прямоугольников с превышением и минимальным отклонением. На основе этих операций разрабатываются начально-кольцевой с превышением и начально-кольцевой с минимальным отклонением алгоритмы, адаптированные под массивы заявок кругового типа. Проводится диспетчирование и вычисляются эвристические меры ресурсных оболочек полиномиальных кольцевых алгоритмов. Сравнительный анализ показывает преимущество предлагаемых полиномиальных алгоритмов и позволяет рекомендовать к использованию в Grid-системах с централизованной структурой.

Ключевые слова: *Grid-система, централизованная структура системы диспетчирования, мультисайтный режим обслуживания, операция динамического интегрирования ресурсных прямоугольников по горизонтали с превышением, операция динамического интегрирования ресурсных прямоугольников по горизонтали с минимальным отклонением, алгоритм полиномиальной трудоемкости, неэвклидова эвристическая мера, начально-кольцевой с превышением алгоритм, начально-кольцевой с минимальным отклонением алгоритм, массив заявок кругового типа*

Введение

Grid-системы, состоящие из сайтов, содержащих параллельные системы [1—5], классифицируются по типу архитектуры на централизованные, иерархические, распределенные [1]. В централизованной структуре центральный диспетчер обладает всей информацией о вычислительных ресурсах и многопроцессорных заявках (пример системы управления ресурсами централизованной структуры KOALA приведен в работе [6]).

По способу объединения ресурсов для решения заявки различают односайтное диспетчирование и мультисайтное диспетчирование, когда многопроцессорная заявка выполняется одновременно на нескольких сайтах [1] (пример системы управления ресурсами, поддерживающей ко-аллокацию приведен в работе KOALA [7]).

Grid-системы с централизованной структурой системы диспетчирования, функционирующие в режиме мультисайтного диспетчирования, моделируются ресурсным квадрантом [8, 9].

Полагаем, что заявки требуют только процессоры и не включаем в рассмотрение другие виды ресурсов Grid-системы [10]. В настоящей работе рассматриваются вычислительные заявки [11, 12] с заранее известным временем решения [13], в которых число требуемых процессоров определяет пользователь при подаче в систему [14].

При представлении заявки пользователя для обслуживания диспетчером Grid-системы в виде ресурсного прямоугольника горизонтальное и вертикальное измерения соответственно принимаются равными числу единиц ресурса времени и процессоров, требуемому для обработки заявки [15].

Задача распределения ресурсных прямоугольников (англ. — *rectangle packing problem*) эквивалентна задаче составления расписания обслуживания многопроцессорных заявок — задаче диспетчирования (англ. — *scheduling problem*) — при условии выделения для обработки заявки процессоров с последовательными номерами [15—19] (англ. — *consecutive indexed processors* [15], *processors of consecutive addresses* [16], *contiguous processors* [17]). В работе [17] для диспетчирования, назначающего обслуживание заявки на процессоры с подряд идущими номерами, используется термин *contiguous scheduling.*

Для управления распределением вычислительно-временных ресурсов в [8, 20—23] разработана среда ресурсных прямоугольников как аппарат теории полиномиальной диспетчеризации. В среде ресурсных прямоугольников введены операции над ресурсными прямоугольниками и предложены эвристические алгоритмы распределения ресур-

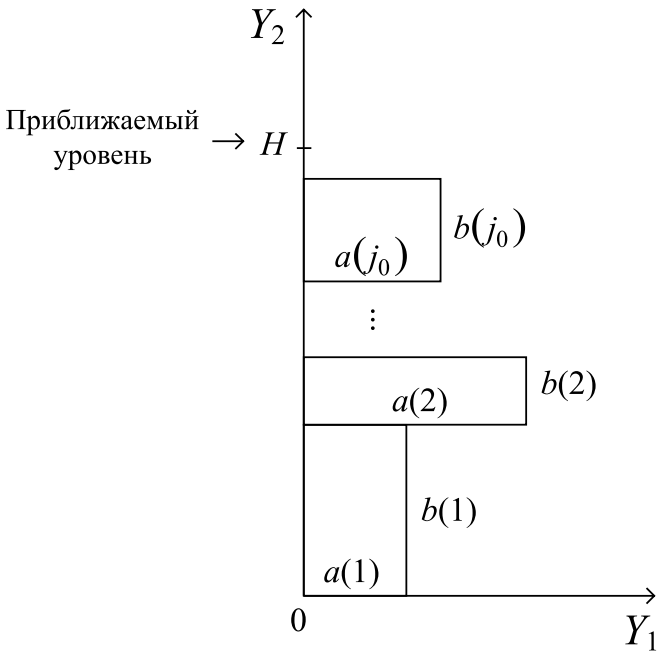


Рис. 1. Вертикальная суперпозиция граней с недостатком [8]

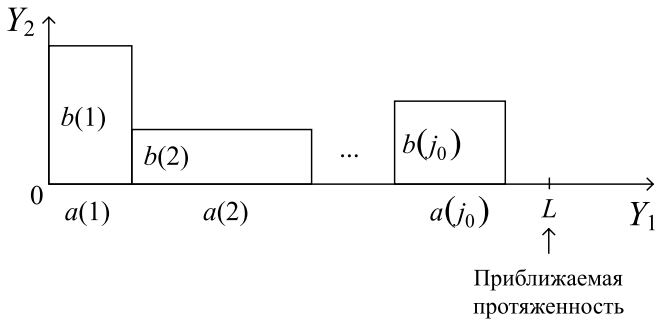


Рис. 2. Горизонтальная суперпозиция граней с недостатком [8]

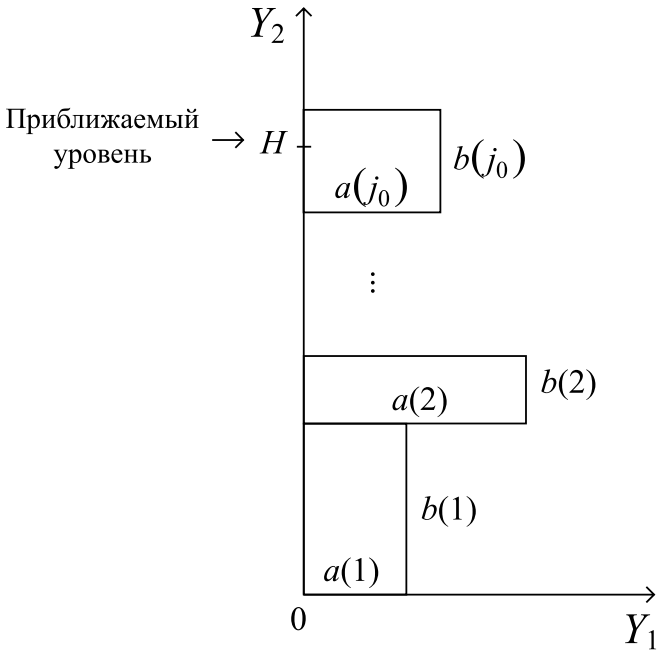


Рис. 3. Вертикальная суперпозиция граней с превышением [24]

сов, основанные на введенных операциях. Показана полиномиальная трудоемкость таких алгоритмов. В работах [8, 20—23] предложена и разработана квадратичная классификация множества заявок. Полиномиальные алгоритмы, исследуемые в [8, 20—23], адаптированы под соответствующий квадратичный тип массива заявок.

В настоящей работе вводятся новые операции в среде ресурсных прямоугольников и на их основе разрабатываются кольцевые алгоритмы, адаптированные под массивы заявок кругового типа.

Кольцевые алгоритмы обслуживания в Grid-системах

В [8] определены операции динамического интегрирования ресурсных прямоугольников по вертикали и по горизонтали, состоящие в наилучшем приближении заданного уровня с недостатком (рис. 1, 2).

В [24] определены операции динамического интегрирования ресурсных прямоугольников по вертикали с превышением (рис. 3) и с минимальным отклонением.

Символом $[a(j), b(j)]$ обозначается j-я заявка, требующая $a(j)$ единиц времени и $b(j)$ единиц процессоров. Определим следующие операции. Операция динамического интегрирования ресурсных прямоугольников по горизонтали с превышением состоит в наилучшем приближении протяженности L с избытком $\sum_{j'=1}^{j_0} a(j') = L + 0$ посредством горизонтальной суперпозиции граней $\bigcup_{j=1}^{j_0} [a(j), b(j)]$ (рис. 4).

Операция динамического интегрирования ресурсных прямоугольников по горизонтали с минимальным отклонением состоит в наилучшем приближении $\sum_{j'=1}^{j_0} a(j') = L \pm 0$ протяженности L с превышением (см. рис. 4) или недостатком (см. рис. 2)

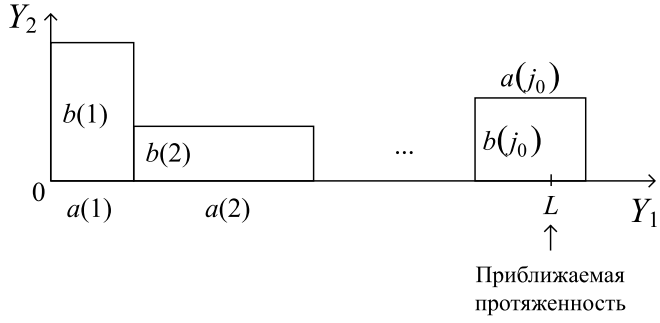


Рис. 4. Горизонтальная суперпозиция граней с превышением

посредством горизонтальной суперпозиции граней $\bigcup_{j=1}^{j_0} [a(j), b(j)]$.

Предлагаемые далее начально-кольцевой алгоритм с превышением и начально-кольцевой алгоритм с минимальным отклонением являются вариантами начально-кольцевого алгоритма с недостатком [8], также адаптированными под круговой тип массива заявок.

Начально-кольцевой алгоритм с превышением основан на операции динамического интегрирования ресурсных прямоугольников по вертикали и по горизонтали с превышением. Функционирование алгоритма аналогично приведенному в [8], с тем отличием, что на каждом шаге ресурсные прямоугольники вертикально суперпозируются до наилучшего приближения уровня оболочки с избытком или горизонтально суперпозируются до наилучшего приближения протяженности оболочки с избытком соответственно (рис. 5).

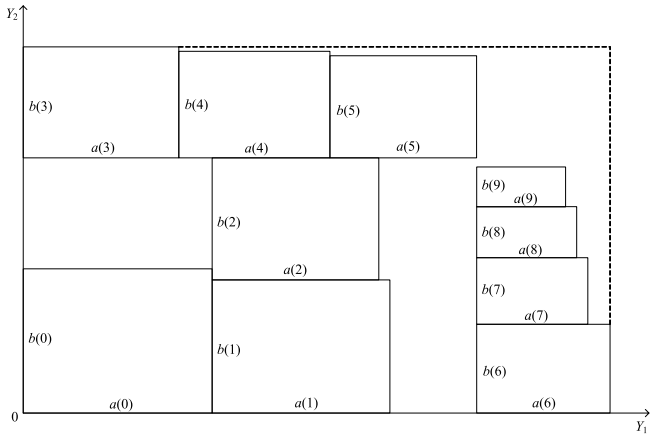


Рис. 5. Укладка начально-кольцевым алгоритмом с превышением массива заявок кругового типа

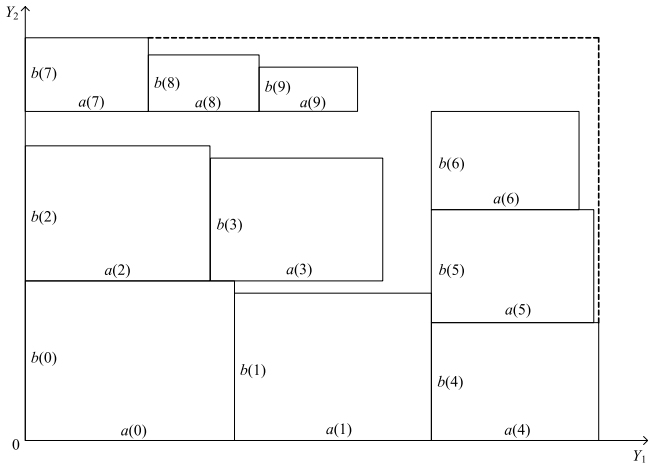


Рис. 6. Укладка начально-кольцевым алгоритмом с минимальным отклонением массива заявок кругового типа

Начально-кольцевой алгоритм с минимальным отклонением основан на операции динамического интегрирования ресурсных прямоугольников по вертикали и по горизонтали с минимальным отклонением. Функционирование алгоритма аналогично приведенному в [8], с тем отличием, что на каждом шаге ресурсные прямоугольники вертикально суперпозируются до наилучшего приближения уровня оболочки с минимальным отклонением или горизонтально суперпозируются до наилучшего приближения протяженности оболочки с минимальным отклонением соответственно (рис. 6).

Исследование начально-кольцевого алгоритма проведено в [8, 27], далее рассматриваются начально-кольцевой алгоритм с превышением, начально-кольцевой алгоритм с минимальным отклонением и проводится сравнительный анализ кольцевых алгоритмов.

Диспетчеризация кольцевыми алгоритмами массивов заявок кругового типа

Качество диспетчирования эвристических алгоритмов оценивается неэвклидовой эвристической мерой, учитывающей наряду с площадью и форму занятой ресурсной области. Вычислим эвристические меры ресурсных оболочек, получаемых при диспетчировании модельных примеров множества ресурсных квадратов, со сторонами, равными последовательным натуральным числам, начиная с единицы, полиномиальными алгоритмами: начально-кольцевым алгоритмом с превышением, начально-кольцевым алгоритмом с минимальным отклонением.

Для массива ресурсных квадратов $(k-j) \times (k-j)$, $j = 0, 1, ..., k-1$, при $k = 32$ [25, 26] соответствующие построения начально-кольцевым алгоритмом приведены на рис. 7 [27]. В центре квадрата указан размер его стороны. Эвристические меры ресурсных оболочек начально-кольцевого алгоритма для массива ресурсных квадратов вычислены в работе [8].

Для массива ресурсных квадратов $(k-j) \times (k-j)$, $j = 0, 1, ..., k-1$, при $k = 32$ соответствующие построения начально-кольцевым алгоритмом с превышением приведены на рис. 8.

Эвристические меры ресурсных оболочек начально-кольцевого алгоритма с превышением для массива ресурсных квадратов приведены в табл. 1.

Видим, что эвристические меры ресурсных оболочек начально-кольцевого алгоритма с превышением не превосходят значения $\frac{1}{2} + 0{,}29$.

Для массива ресурсных квадратов $(k-j) \times (k-j)$, $j = 0, 1, ..., k-1$, при $k = 32$ соответствующие построения начально-кольцевым алгоритмом с минимальным отклонением приведены на рис. 9.

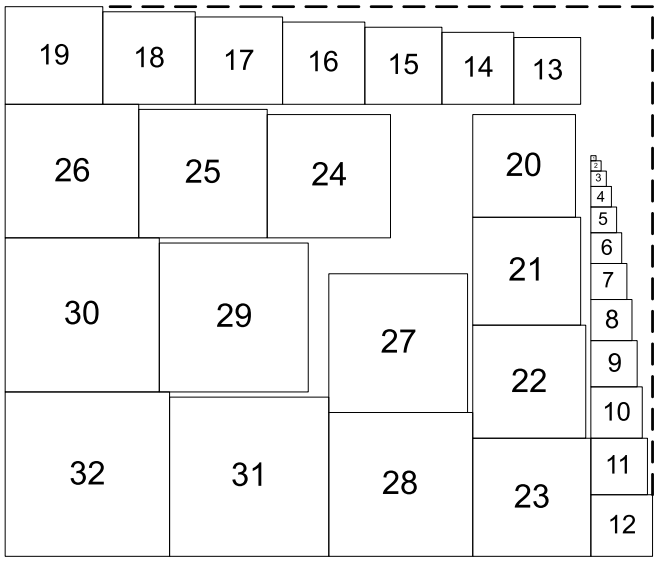


Рис. 7. Укладка начально-кольцевым алгоритмом массива ресурсных квадратов [27]

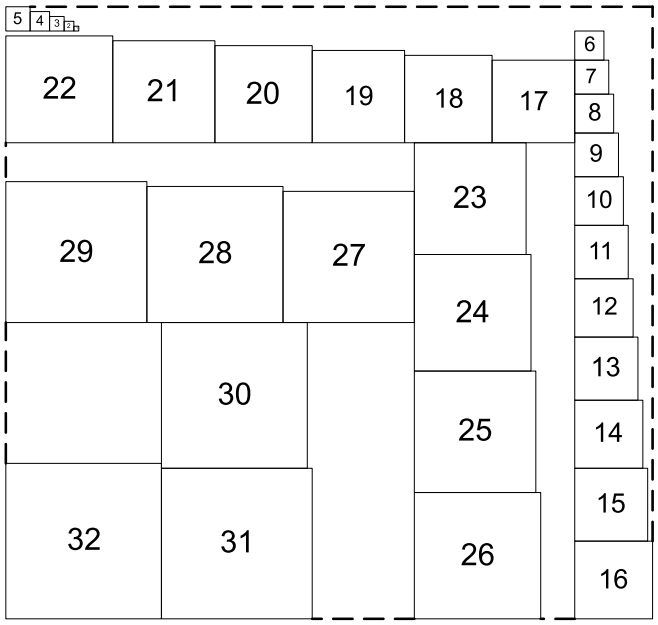


Рис. 8. Укладка начально-кольцевым алгоритмом с превышением массива ресурсных квадратов

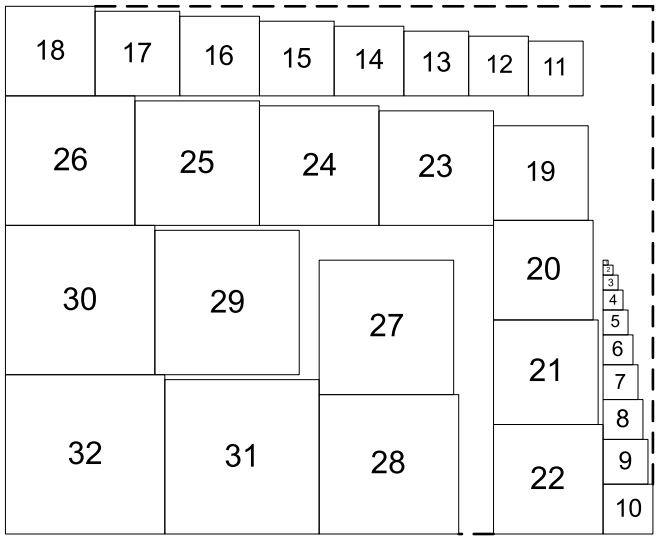


Рис. 9. Укладка начально-кольцевым алгоритмом с минимальным отклонением массива ресурсных квадратов

Таблица 1

Эвристические меры ресурсных оболочек начально-кольцевого алгоритма с превышением

k	Эвристическая мера	k	Эвристическая мера	k	Эвристическая мера
12	0,69	19	0,74	26	0,73
13	0,69	20	0,75	27	0,75
14	0,68	21	0,76	28	0,77
15	0,71	22	0,76	29	0,69
16	0,76	23	0,79	30	0,70
17	0,71	24	0,74	31	0,72
18	0,73	25	0,71	32	0,73

Таблица 2

Эвристические меры ресурсных оболочек начально-кольцевого алгоритма с минимальным отклонением

k	Эвристическая мера	k	Эвристическая мера	k	Эвристическая мера
12	0,66	19	0,65	26	0,59
13	0,63	20	0,63	27	0,60
14	0,62	21	0,64	28	0,61
15	0,61	22	0,63	29	0,61
16	0,63	23	0,63	30	0,63
17	0,65	24	0,61	31	0,64
18	0,67	25	0,60	32	0,63

Эвристические меры ресурсных оболочек начально-кольцевого алгоритма с минимальным отклонением для массива ресурсных квадратов приведены в табл. 2.

Видим, что эвристические меры ресурсных оболочек начально-кольцевого алгоритма с минимальным отклонением не превосходят значения $\frac{1}{2} + 0{,}17$.

Графики эвристической меры ресурсных оболочек начально-кольцевого, начально-кольцевого с превышением и начально-кольцевого с минимальным отклонением алгоритмов диспетчеризации массива ресурсных квадратов показаны на рис. 10.

Видим, что начально-кольцевой алгоритм с превышением имеет большую эвристическую меру по сравнению с другими анализируемыми алгоритмами. Сравнение результатов эвристических мер ресурсных оболочек начально-кольцевого с минимальным отклонением и начально-кольцевого (с недостатком) алгоритмов не позволяет отдать предпочтение какому-либо из них, так как для одних значений параметра k (например, $k = 26, 27, 28, 29$)

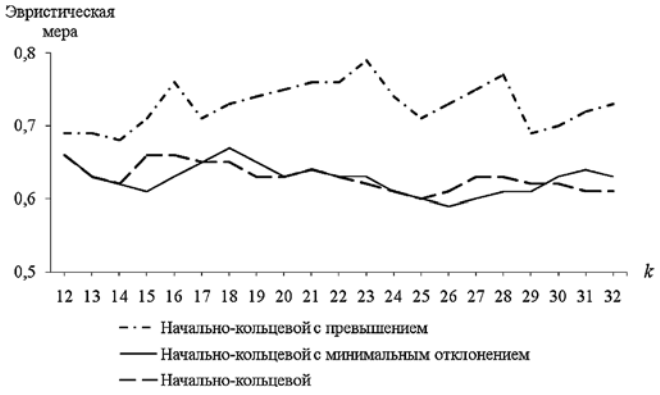


Рис. 10. Эвристические меры ресурсных оболочек полиномиальных кольцевых алгоритмов диспетчеризации массивами ресурсных квадратов

эвристическая мера лучше у предложенного в настоящей статье алгоритма, для других — у предложенного автором ранее. Проведенный анализ позволяет рекомендовать предложенный начально-кольцевой с минимальным отклонением алгоритм к использованию в Grid-системах с централизованной структурой системы диспетчирования и мультисайтным режимом обслуживания.

Заключение

В среде ресурсных прямоугольников определяются операции динамического интегрирования ресурсных прямоугольников с превышением и минимальным отклонением. На основе этих операций разрабатываются начально-кольцевой с превышением и начально-кольцевой с минимальным отклонением алгоритмы, адаптированные под массивы заявок кругового типа. Результаты экспериментов показывают определенные преимущества предложенного начально-кольцевого с минимальным отклонением полиномиального алгоритма и позволяют рекомендовать кольцевые алгоритмы к использованию в Grid-системах с централизованной структурой при обслуживании массивов заявок кругового типа.

Список литературы

1. **Hamscher V., Schwiegelshohn U., Streit A., Yahyapour R.** Evaluation of job-scheduling strategies for grid computing. In Proceedings of the 7th International Conference on High Performance Computing, HiPC-2000 // LNCS. 2000. Vol. 1971. P. 191—202.
2. **Li M., Baker M.** The grid: core technologies. Chichester: John Wiley & Sons Ltd. 2005. 452 p.
3. **Magoulès F., Nguyen T., Yu L.** Grid resource management: toward virtual and services compliant grid computing. Boca Raton—London—New York: Taylor&Francis Group, LLC, 2009. 300 p.
4. **Magoulès F.** (ed.). Fundamentals of grid computing: theory, algorithms and technologies. Boca Raton—London—New York: Taylor&Francis Group, LLC, 2010. 298 p.
5. **Antonopoulos N., Exarchakos G., Li M., Liotta A.** (eds.). Handbook of research on p2p and grid systems for service-oriented computing: models, methodologies and applications. Hershey: IGI Global, 2 Volumes, 2010. 1342 p.
6. **Assunção M., Buyya R.** Architectural elements of resource sharing networks // In Li K., Hsu C., Yang L., Dongarra J., Zima H. (eds.), Handbook of research on scalable computing technologies. Hershey: IGI Global, 2 Volumes, 2010. Vol. 2. P. 517—550.
7. **Netto M., Buyya R.** Resource Coallocation in Grid Computing Environments // In Antonopoulos, N., Exarchakos, G., Li, M., Liotta, A. (eds.), Handbook of research on p2p and grid systems for service-oriented computing: models, methodologies and applications. Hershey: IGI Global, 2 Volumes, 2010. Vol. 1. P. 476—494.
8. **Саак А. Э.** Полиномиальные алгоритмы распределения ресурсов в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2013. № 7. Приложение. 32 с.
9. **Саак А. Э.** Управление ресурсами и заявками пользователей в Grid-системах с централизованной архитектурой // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014. Москва, 16—19 июня 2014 г. М.: Институт проблем управления им. В. А. Трапезникова РАН, 2014. С. 7489—7498.
10. **Bucur A., Epema D.** Scheduling policies for processor coallocation in multicluster systems // IEEE Transactions on Parallel and Distributed Systems. 2007. Vol. 18, N. 7. P. 958—972.
11. **Christodoulopoulos K., Sourlas V., Mpakolas I., Varvarigos E.** A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in Grid networks // Computer Communications. 2009. N 32. P. 1172—1184.
12. **Rahman M., Ranjan R., Buyya R., Benatallah B.** A taxonomy and survey on autonomic management of applications in grid computing environments // Concurrency Computat.: Pract. Exper. 2011. N. 23. P. 1990—2019.
13. **Ye D., Zhang G.** On-Line Scheduling of Parallel Jobs. In R. Kralovic and O. Sykora, ed., SIROCCO 2004, LNCS. 2004. Vol. 3104. P. 279—290.
14. **Feitelson D., Rudolph L.** Toward convergence in job schedulers for parallel supercomputers. In Job Scheduling Strategies for Parallel Processing, Feitelson D., Rudolph L. (eds.), LNCS. 1996. Vol. 1162. P. 1—26.
15. **Caramia M., Giordani S., Iovanella A.** Grid scheduling by online rectangle packing // Networks. 2004. Vol. 44, N 2. P. 106—119.
16. **Bougeret M., Dutot P.-F., Jansen K., Robenek C., Trystram D.** Approximation algorithms for multiple strip packing and scheduling parallel jobs in platforms // Discr. Math., Algorithms and Applications. 2011. Vol. 3, N 4. P. 553—586.
17. **Günther E., König F., Megow N.** Scheduling and packing malleable and parallel tasks with precedence constraints of bounded width // J. Comb. Optim. 2014. Vol. 27, Iss. 1. P. 164—181.
18. **Поспелов А. И.** Анализ одного алгоритма упаковки прямоугольников, связанного с построением расписаний для кластеров // Труды ИСП РАН. 2004. Т. 6. С. 7—12.
19. **Кузюрин Н. Н., Грушин Д. А., Фомин С. А.** Проблемы двумерной упаковки и задачи оптимизации в распределенных вычислительных системах // Труды ИСП РАН. 2014. Т. 26, Вып. 1. С. 483—502.
20. **Саак А. Э.** Локально-оптимальные ресурсные распределения // Информационные технологии. 2011. №2. С. 28—34.
21. **Саак А. Э.** Алгоритмы диспетчеризации в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2011. №11. С. 9—13.
22. **Саак А. Э.** Диспетчеризация в Grid-системах на основе однородной квадратичной типизации массивов заявок пользователей // Информационные технологии. 2012. №4. С. 32—36.
23. **Саак А. Э.** Сравнительный анализ полиномиальных алгоритмов диспетчеризации в Grid-системах // Информационные технологии. 2012. №9. С. 28—32.
24. **Саак А. Э.** Уровневые алгоритмы диспетчеризации массивами заявок кругового типа в Grid-системах // Известия ЮФУ. Технические науки. 2015. (в печати).
25. **Korf R., Moffitt M., Pollack M.** Optimal rectangle packing // Annals of Operations Research. 2010. Vol. 179, N 1. P. 261—295.
26. **Huang E., Korf R.** Optimal rectangle packing: an absolute placement approach // Journal of Artificial Intelligence Research. 2012. Vol. 46. P. 47—87.
27. **Saak A., Kureichik V., Kuliev E.** Ring algorithms for scheduling in grid systems. In Proceedings of the 4th Computer Science Online Conference, CSOC2015, Advances in intelligent systems and computing. 2015. Vol. 349. P. 201—209.

A. E. Saak, Associate Professor, e-mail: saak@tgn.sfedu.ru,
Southern Federal University

Ring Algorithms for Scheduling in Grid Systems by Sets of Tasks

Grid systems which have centralized structure of scheduling system are modeled by the resource quadrant. Multiprocessor tasks are modeled by resource rectangles. The task of the resource rectangle distribution is equivalent to the one of multiprocessor task service scheduling provided that processors with consecutive numbers are allotted for task handling. In our previous papers the resource rectangle environment were presented as an instrument of the polynomial scheduling theory for control of computational and time resource distribution. In this paper we define operations of resource rectangle dynamic integration with exceeding and minimal deviation, for the resource rectangle environment. On the base of introduced operations we develop an initial ring algorithm with exceeding and initial ring algorithm with minimal deviation which have polynomial completeness. The algorithms suggested here are adapted for use with the circular quadratic type of a multiprocessor task set. The quality estimation of the algorithms of scheduling is carried out with the use of the Non-Euclidean heuristic measure, which takes into consideration both the area and the shape of occupied resource enclosure. The values of the heuristic measure of the resource enclosures derived in the process of implementation of the initial ring algorithm with exceeding and initial ring algorithm with minimal deviation for scheduling the simulation examples of the set of resource squares with the sides equal to successive natural numbers which begin with one. The investigation shows that the initial ring algorithm with exceeding has the biggest value of the heuristic measure in comparison with the other ones. The result of the comparison of the heuristic measure values of the resource enclosures produced by the initial ring algorithm with minimal deviation and the initial ring algorithm, which was presented in the previous papers, doesn't make it possible to give preference for one of them in particular. On the base of the analysis' results we can recommend the initial ring algorithm with minimal deviation, presented here, for use in Grid systems with centralized architecture of scheduling system and the multisite mode of service.

Keywords: *Grid system, centralized architecture of scheduling system, multisite mode of service, operation of resource rectangle dynamic integration horizontally with exceeding, operation of resource rectangle dynamic integration horizontally with minimal deviation, algorithm of polynomial completeness, Non-Euclidean heuristic measure, initial ring algorithm with exceeding, initial ring algorithm with minimal deviation, circular-type set of tasks*

References

1. **Hamscher V., Schwiegelshohn U., Streit A., Yahyapour R.** Evaluation of job-schiduling strategies for grid computing. In Proceedings of the 7th International Conference on High Performance Computing, HiPC-2000, 2000. *LNCS*, 2000, vol. 1971, pp. 191—202.

2. **Li M., Baker M.** *The grid: core technologies.* Chichester: John Wiley & Sons Ltd, 2005, 452 p.

3. **Magoulès F., Nguyen T., Yu L.** *Grid resource management: toward virtual and services compliant grid computing.* Boca Raton—London—New York: Taylor & Francis Group, LLC, 2009, 300 p.

4. **Magoulès F.** (ed.). *Fundamentals of grid computing: theory, algorithms and technologies.* Boca Raton—London—New York: Taylor&Francis Group, LLC, 2010. 298 p.

5. **Antonopoulos N., Exarchakos G., Li M., Liotta A.** (eds.). *Handbook of research on p2p and grid systems for service-oriented computing: models, methodologies and applications.* Hershey: IGI Global, 2 Volumes, 2010, 1342 p.

6. **Assunção M., Buyya R.** *Architectural elements of resource sharing networks.* Li K., Hsu C, Yang L., Dongarra J., Zima H. (eds.), Handbook of research on scalable computing technologies. Hershey: IGI Global, 2 Volumes, 2010, 1086 p., vol. 2, pp. 517—550.

7. **Netto M., Buyya R.** *Resource. Co-allocation in Grid Computing Environments.* Antonopoulos, N., Exarchakos, G., Li, M., Liotta, A. (eds.), Handbook of research on p2p and grid systems for service-oriented computing: models, methodologies and applications. Hershey: IGI Global, 2 Volumes, 2010, 1342 p., vol. 1, pp. 476—494.

8. **Saak A. Eh.** Polinomialnye algoritmy raspredeleniya resursov v Grid-sistemakh na osnove kvadratichnoy tipizatsii massivov zayavok [Polynomial algorithms for resource allocation in Grid-based systems for quadratic typing, arrays applications], *Informacionnye tekhnologii* [Information Technologies], 2013, no. 7. Prilozhenie, 32 p.

9. **Saak A. Eh.** Upravleniye resursami i zayavkami polzovateley v Grid-sistemakh s centralizovannoy architektyroy [Resource and multiprocessor task management in Grid system of centralized architecture], *Trudy XII Vserossiyskogo soveshanya po problem upravleniya VSPU-2014. Moskva, 16 ijunya — 19 ijunya 2014 g.* [Proceedings of XII all-Russian conference "Control problems" RCCP'2014. Moscow, 16 June — 19 June 2014]. Moscow: Institut problem upravleniya im. V. A. Trapeznikova RAN, 2014, pp. 7489—7498.

10. **Bucur A., Epema D.** Scheduling policies for processor coallocation in multicluster systems, *IEEE Transactions on Parallel and Distributed Systems*, 2007, vol. 18, no. 7, pp. 958—972.

11. **Christodoulopoulos K., Sourlas V., Mpakolas I., Varvarigos E.** A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in Grid networks, *Computer Communications*, 2009, no. 32, pp. 1172—1184.

12. **Rahman M., Ranjan R., Buyya R., Benatallah B.** A taxonomy and survey on autonomic management of applications in grid computing environments, *Concurrency Computat.*: Pract. Exper, 2011, no. 23, pp. 1990—2019.

13. **Ye P., Zhang G.** On-Line Scheduling of Parallel Jobs. In R. Královiç and O. Sýkora, ed., SIROCCO 2004, *LNCS*, 2004, vol. 3104, pp. 279—290.

14. **Feitelson D., Rudolph L.** Toward convergence in job schedulers for parallel supercomputers. In Job Scheduling Strategies for Parallel Processing, Feitelson D., Rudolph L. (eds.), *LNCS*, 1996, vol. 1162, pp. 1—26.

15. **Caramia M., Giordani S., Iovanella A.** Grid scheduling by online rectangle packing, *Networks*, 2004, vol. 44, no. 2, pp. 106—119.

16. **Bougeret M., Dutot P.-F., Jansen K., Robenek C., Trystram D.** Approximation algorithms for multiple strip packing and scheduling parallel jobs in platforms, *Discr. Math., Algorithms and Applications*, 2011, vol. 3, no. 4, pp. 553—586.

17. **Gfinther E., Kðnig F., Megow N.** Scheduling and packing malleable and parallel tasks with precedence constraints of bounded width, *J. Comb. Optim*, 2014, vol. 27, Iss. 1, pp. 164—181.

18. **Pospelov A. I.** Analiz odnogo algoritma upakovki pryamougol'nikov, svyazannogo s postroeniem raspisanij dlya klasterov [Analysis of cluster scheduling packing algorithm], *Trudy ISP RAN* [The Proceedings of ISP RAS], Moscow, 2004, vol. 6, pp. 7—12.

19. **Kuzyurin N. N., Grushin D. A., Fomin S. A.** Problemi dvumernoi upakovki i zadachi optimizatzii v raspredelennikh vichislitelnikh sistemakh [Two-dimensional packing problems and optimization in distributed computing systems], *Trudy ISP RAN* [The Proceedings of ISP RAS], Moscow, 2014, vol. 26, Iss. 1, pp. 483—502.

20. **Saak A. Eh.** Lokalno-optimalnye resursnye raspredeleniya [Locally optimal resource allocation], *Informatsionnihe tekhnologii* [Information Technologies], 2011, no. 2, pp. 28—34.

21. **Saak A. Eh.** Algoritmy dispetcherizatsii v Grid-sistemakh na osnove kvadratichnoy tipizatsii massivov zayavok [Algorithms scheduling in Grid-based systems for quadratic typing, arrays applications], *Informatsionnye tekhnologii* [Information Technologies], 2011, no. 11, pp. 9—13.

22. **Saak A. Eh.** Dispetcherizatsiya v GRID-sistemakh na osnove odnorodnoy kvadratichnoy tipizatsii massivov zayavok polzovateley [Scheduling in GRID-systems on the basis of homogeneous quadratic typing, arrays of user requests], *Infonrmtsionne tekhnologii* [Information Technologies], 2012, no. 4, pp. 32—36.

23. **Saak A. Eh.** Sravnitelnihy analiz polinomialnykh algoritmov dispetcherizatsii v GRID-sistemakh [Comparative analysis of polynomial algorithms for scheduling in GRID-systems], *Informatsionnye tekhnologii* [Information Technologies], 2012, no. 9, pp. 28—32.

24. **Saak A. Eh.** Urovnevye algoritmy dispetcherizatsii massivamy zayavok v Grid-sistemakh [Level algorithms of scheduling by circle type task sets in grid systems], *Izvestiya YuFU. Tekhnicheskie nauki'* [Izvestiya SFedU. Engineering Sciences], v pechaty [to appear].

25. **Korf R., Moffitt M., Pollack M.** Optimal rectangle packing, *Annals of Operations Research*, 2010, vol. 179, no. 1, pp. 261—295.

26. **Huang E., Korf R.** Optimal rectangle packing: an absolute placement approach, *Journal of Artificial Intelligence* Research, 2012, vol. 46, pp. 47—87.

27. **Saak A., Kureichik V., Kuliev E.** Ring algorithms for scheduling in grid systems, *Proceedings of the 4th Computer Science On-line Conference, CSOC2015, Advances in intelligent systems and computing*, 2015, vol. 349, pp. 201—209.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ
MODELING AND OPTIMIZATION

УДК 004.4’22

Д. В. Ошкало, аспирант, e-mail: dmitry.oshkalo@gmail.com,
Московский государственный технический университет им. Н. Э. Баумана, г. Москва

Разработка процессов трансформации моделей с помощью биграфов

Предложен подход к разработке процессов трансформации моделей, основанный на использовании биграфов. Показана взаимосвязь между стандартом MOF и представлением моделей в виде биграфов. Для обеспечения корректности правил трансформации применяются ограничения, накладываемые структурой биграфов, и шаблоны правил. Подход позволяет выполнять проверку свойств процессов трансформации, а также осуществлять их моделирование в условиях внесения изменений в модели.

Ключевые слова: *биграфы, моделеориентированная разработка, модели данных, трансформация моделей, правила трансформации, MOF*

Введение

С точки зрения моделеориентированного подхода артефакты процесса разработки программного обеспечения (программный код, конфигурационные файлы, документация и т. д.) представляются в виде моделей — формальных структурированных описаний на одном из языков моделирования, например *UML* [1].

В целях унификации структуры моделей, упорядочения процесса их разработки, обеспечения возможности использования в различных системах был создан стандарт *MOF* (*Meta-Object Facility*) [2], определяющий отношения между моделями и формирующий закрытую многоуровневую архитектуру моделей. Модель каждого уровня задает структурное описание — набор ограничений, накладываемых на модели уровнем ниже: каждый их элемент должен соответствовать одному из элементов модели вышестоящего (т. е. метауровня). В этом случае модель нижестоящего уровня называется экземпляром модели, расположенной уровнем выше.

Архитектура *MOF* состоит из четырех уровней (рис. 1). На верхнем уровне *M*3 находится модель *MOF*, предоставляющая универсальные средства для описания языков моделирования — метамоделей. По этой причине уровень *M*3 называется уровнем мета-метамоделей. Благодаря наличию этого уровня обеспечивается независимость от различных языков моделирования и возможность унификации работы с моделями.

На уровне *M*2 расположены метамодели, определяющие языки моделирования (такие как *UML*), которые используются для описания различных компонентов разрабатываемой системы в виде моделей уровня *M*1. На последнем уровне *M*0 расположены данные или моделируемые объекты реального мира. На рис. 1 в качестве примера приведен стек моделей, используемых при моделировании реляционного хранилища данных. Штриховыми стрелками на рисунке показаны соответствия между элементами различных уровней.

Отметим важное следствие архитектуры *MOF*. Так как все метамодели формируются согласно единой мета-метамодели, можно составить единообразное описание отображений между метамоделями, которое служит основой процедур трансформации моделей [3] — одной из ключевых активностей процесса моделеориентированной разработки, направленной на выполнение таких задач как созда-

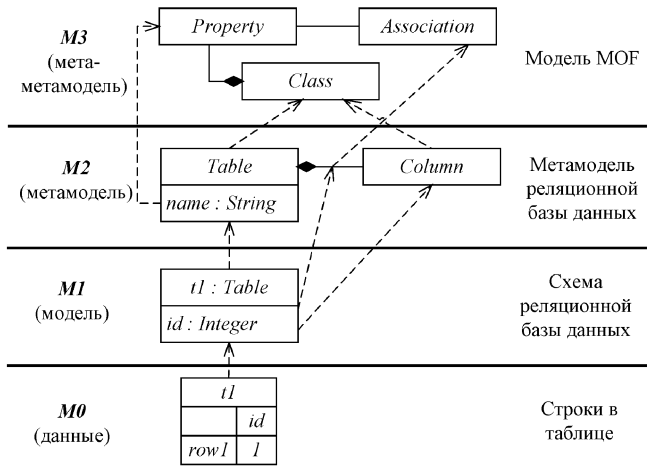


Рис. 1. Уровни архитектуры *MOF*

ние одних моделей на основе других, рефакторинг моделей, перенос изменений между моделями, автоматизированное построение различных компонентов разрабатываемой системы и т. д. Качество выполнения этих задач существенно влияет на качество всего процесса проектирования, а затрачиваемое на их решение время занимает, по разным оценкам, до 30 % времени разработки. Зачастую модели связаны друг с другом: одна из них может быть сгенерирована на основе другой или же обе они могут представлять одну и ту же информацию с разных точек зрения. В этом случае необходимо поддерживать актуальную связь между моделями в условиях внесения в них изменений в процессе разработки.

Для решения данной проблемы применяют движки трансформации моделей, использующие описанный некоторым образом набор правил трансформации и применяющие эти правила к элементам моделей в определенной последовательности (рис. 2). Таким образом, правила трансформации формулируются на уровне *M*2, архитектуры *MOF*, а процесс их применения затрагивает уровень моделей *M*1.

В движке трансформаций описан механизм применения правил к элементам в моделях, поэтому проектирование процессов трансформации сводится к разработке необходимого набора правил для конкретного сценария трансформации моделей.

В настоящее время широкое распространение получили подходы, основанные на графовых грамматиках [4—6], оперирующих моделями в терминах типизированных графов, в которых типы вершин и ребер, а также соотношения между ними регламентируются используемым языком моделирования. Популярность таких подходов обусловлена удобством практического применения, выражаемом в наглядности, возможности использования графических редакторов при проектировании правил трансформации, более прозрачных и эффективных методах контроля и отладки разработанных трансформаций. Примеры программных средств, в которых реализован этот подход, представлены в работе [5]. Выделяют две методики разработки правил [6]:

- создание в редакторе моделей для конкретного сценария трансформации, задание вручную соответствий между их элементами и отладка на тестовых примерах;

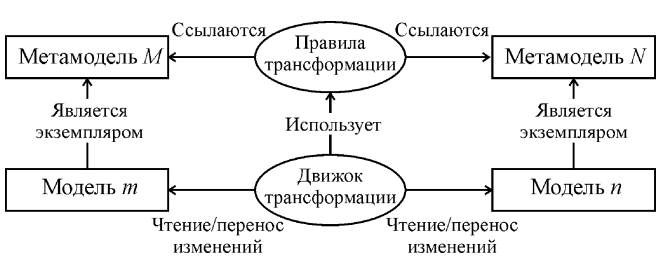


Рис. 2 Основные компоненты процесса трансформации моделей

- использование готовых примеров моделей, на основе которых автоматически будут сгенерированы правила трансформации.

Предлагаемые методики не могут считаться достаточными, поскольку не гарантируют построения корректного набора правил трансформации, не позволяют проверить свойства [3, 7] построенного набора правил и его работоспособность в условиях одновременного внесения изменений несколькими пользователями.

Для создания набора правил, удовлетворяющего этим требованиям, предлагается представить трансформацию моделей в виде системы, состоящей из нескольких параллельно взаимодействующих компонентов. Проектирование таких систем включает этап моделирования, задачей которого является анализ структуры, свойств и поведения как системы целиком, так и отдельных ее частей. Основой для формального описания и анализа в этом случае является аппарат теории процессов. Наиболее важный класс задач, для решения которых предназначена теория процессов, связан с проблемой верификации процессов, которая заключается в построении формального доказательства того, что анализируемый процесс обладает заданными свойствами [8].

Существует множество разновидностей исчислений процессов, в основе которых лежат те или иные аспекты моделируемых систем: обмен сообщениями, работа с поточными данными, мобильность и изменение конфигурации компонентов системы и т. д. Особенностью рассматриваемого случая является использование иерархических структур данных, которыми являются модели, в то время как исчисления процессов оперируют только с простыми представлениями данных. По этой причине для моделирования системы трансформации моделей предлагается использовать теорию биграфов, разработанную Робином Милнером [9—11]. Биграфы (здесь возможна путаница с двудольными графами, но автор, зная об этом, не стал менять названия) и построенные на их основе биграфовые реагирующие системы [10, 12] являются теоретическим базисом для моделирования систем, в которых в процессе взаимодействия между элементами может меняться их расположение и связи между ними. Кроме того, биграфы являются обобщающей концепцией для многих исчислений процессов, таких как π-исчисления и сети Петри [10, 13].

В данной статье рассматриваются аппарат теории биграфов, их статические и динамические компоненты, возможности представления моделей в виде биграфов и использование этого представления при проектировании процессов трансформации моделей на примере трансформации между *UML*-диаграммой классов и соответствующей ей схемой реляционного хранилища данных, а также при моделировании этих процессов и проверке их свойств.

Биграфы

Рассмотрим основные понятия, связанные со структурой биграфов, в том объеме, в котором это необходимо для представления моделей. Отметим, что устоявшегося перевода положений теории биграфов на момент написания статьи нет, поэтому используемые в статье термины приведены вместе с оригинальным написанием. Наиболее полная информация доступна в работах [9—13].

Биграф — это структура, объединяющая в себе такие особенности моделируемой системы, как взаимное расположение ее элементов и связи между ними, представленные в виде двух графов: графа расположения вершин биграфа (*place graph*) и графа связей (*link graph*). На рис. 3 изображены примеры биграфов и их компоненты по отдельности.

Граф расположения (*place graph*) представляет собой лес с именованными корнями и ребрами, описывающими иерархию вершин: вершины, располагающиеся под текущей вершиной в графе и связанные с ней ребром, являются дочерними по отношению к ней (т. е. находятся внутри нее на изображении биграфа). Например, на рис. 3, *а* вершина v_5 на графе расположения ниже вершины v_4, поэтому на биграфе v_5 внутри v_4. Корни графа расположения называют корнями (*roots*), а листья — положениями (*sites*). Первые являются участками, внутри которых находятся вершины биграфа, обозначаются штриховым четырехугольником со скругленными краями; вторые служат в качестве позиций, внутри которых могут быть расположены другие биграфы, обозначаются сплошным четырехугольником с номером.

Граф связей (*link graph*) — это гиперграф, в котором ребра могут связывать произвольное число вершин. Корни графа связей называют внешними именами биграфа, листья — внутренними. Множество имен биграфа образует связи, к которым извне могут быть подсоединены другие биграфы. Таким образом обеспечивается механизм образования новых связей между элементами.

Точки, в которых ребра соединяются с вершинами, называются портами. Каждая из вершин биграфа имеет собственный тип, определяющий число ее портов (например, на рис. 3, *а* вершины v_1, v_3, v_4 имеют тип K с двумя портами). Сигнатурой биграфа называется $\Sigma = (K, ar)$, где K — множество типов, $ar : K \to N$ — функция, для каждого типа устанавливающая число портов.

Для моделирования сложных структур, состоящих из нескольких компонентов, вводится понятие внешних и внутренних интерфейсов, формируемых графами расположения и связей. Та часть интерфейса биграфа на рис. 3, *а*, которая формируется графом расположения, обозначается следующим образом: G^P: 0 → 3, где 0 — количество положений (часть внутреннего интерфейса); 3 — число корней (часть внешнего интерфейса). Для биграфа на рис. 3, *б*: $H^P : 3 \to 2$. Аналогично часть интерфейса биграфов на рис. 3, *а* и 3, *б*, формируемая графом связей, будет иметь следующий вид: $G^L : \varnothing \to \{xy\}$ и $H^L : \{xy\} \to \varnothing$. Сам же биграф при этом характеризуется совокупностью его внутренних и внешних интерфейсов:

$$G = \langle G^P,\ G^L\rangle : \varepsilon \to \langle 3,\ \{xy\}\rangle, \text{ где } \varepsilon = \langle 0,\ \varnothing\rangle;$$

$$H = \langle H^P,\ H^L\rangle : \langle 3,\ \{xy\}\rangle \to \langle 2,\ \varnothing\rangle.$$

Рассмотренных понятий достаточно, чтобы привести формальное определение биграфа. Биграф с сигнатурой Σ представляется как $G = (V, E, P, ctrl, prnt, link) : \langle m, X\rangle \to \langle n, Y\rangle$, где m и n — число положений и корней соответственно; X, Y — множества внутренних и внешних имен; V — множество вершин; E — множество ребер; $P = \{(v, i) | i \in ar(\mathrm{ctrl}(v))\}$ — множество портов, где v — вершина графа; i — номер порта; $\mathrm{ctrl} : V \to K$ — функция, ставящая в соответствие каждой вершине ее тип; $prtn : V \cup m \to V \cup n$ — представление графа расположений; $link : P \cup X \to E \cup Y$ — представление графа связей; $\langle m, X\rangle$ — внутренний интерфейс; $\langle n, Y\rangle$ — внешний интерфейс. Для биграфа, изображенного на рис. 3, *а*:

$$\Sigma = (\{K, L\}, \{(K, 2), (L, 1)\});$$

$$V = \{v_1, v_3, v_4, v_5\} E = \{e_1, e_2\};$$

$$P = \{p(v_1, 1), p(v_1, 2), p(v_3, 1), p(v_3, 2), p(v_4, 1), p(v_4, 2), p(v_5, 1)\};$$

$$ctrl = \{(v_1, K), (v_3, K), (v_4, K), (v_5, L)\};$$

$$prnt = \{(v_1, 0), (v_3, 1), (v_4, 2), (v_5, v_4)\};$$

$$link = \begin{Bmatrix} (p(v_1, 1), x), (p(v_1, 2), e_1), (p(v_3, 1), e_1), \\ (p(v_3, 2), e_2), (p(v_4, 1)e_2), \\ (p(v_4, 2), y), (p(v_5, 1), e_2); \end{Bmatrix}$$

$$m = 0,\ n = 3,\ X = \varnothing,\ Y = \{x, y\}.$$

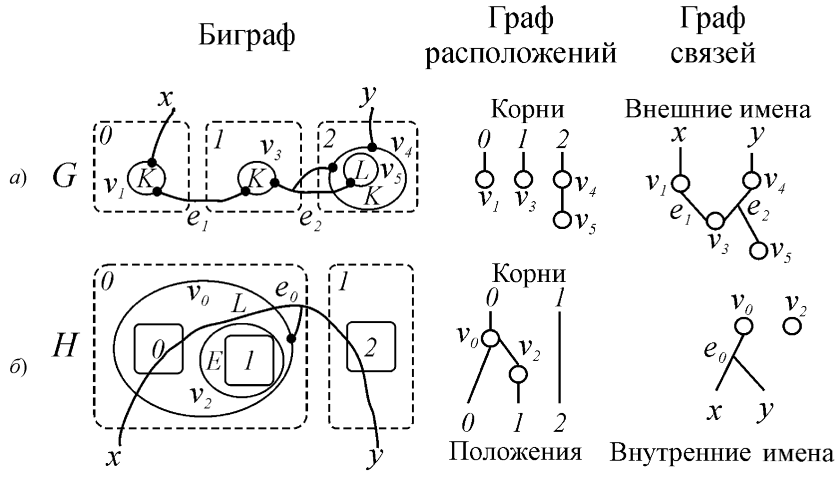


Рис. 3. Биграфы и их компоненты

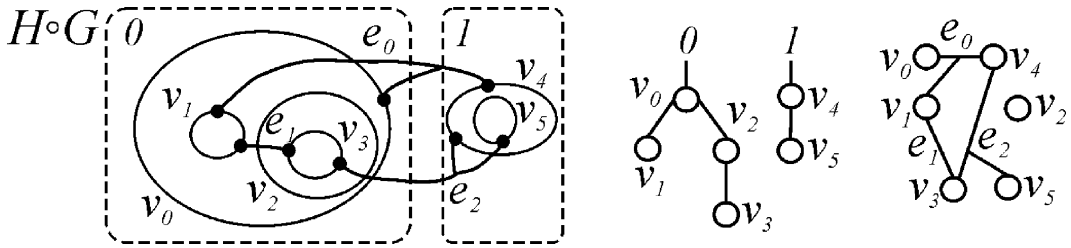


Рис. 4. Результат композиции биграфов

Используя понятие интерфейсов, определим операцию композиции биграфов. Пусть даны два биграфа $G: \langle l, X\rangle \to \langle m, Y\rangle$ и $H: \langle m, Y\rangle \to \langle n, Z\rangle$. Их композиция $H \circ G$ получается путем подстановки корней G со всей внутренней структурой на место соответствующих (по номерам) положений в H, а также соединения связей внешнего интерфейса G с соответствующими (по именам) связями внутреннего интерфейса H. Результат композиции биграфов рис. 3 изображен на рис. 4 (типы вершин не показаны).

Операция композиции биграфов позволяет не только составлять сложные системы из более простых, но и также, используя определение интерфейсов, задавать ограничения, накладываемые на биграфы, описывающие компоненты моделируемой системы.

Представление моделей в виде биграфов

Поскольку трансформация моделей охватывает уровни *M*2 и *M*1 архитектуры *MOF*, для ее описания с помощью биграфов необходимо установить соответствие между биграфами и уровнями *MOF* таким образом, чтобы биграфовое представление моделей и метамоделей сохраняло отношение между этими уровнями, накладываемое *MOF*. Кроме того, чтобы описывать трансформации моделей независимо от используемых языков моделирования, нужно установить связь между аппаратом биграфов и средствами описания языков моделирования, предоставляемыми уровнем *M*3. Для этого достаточно рассмотреть ту часть диаграммы *MOF* [3], которая накладывает ограничения на структуру метамоделей (рис. 5).

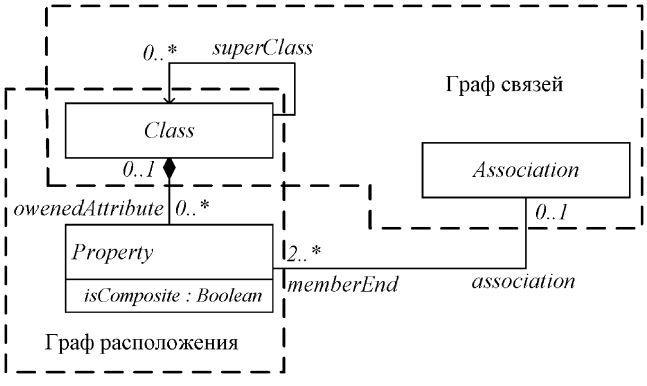


Рис. 5. Связь биграфов с уровнем *M*3 архитектуры *MOF*

Метамодели представляют собой множество элементов *Class*, содержащих набор полей *Property* и связываемых с другими объектами посредством элементов *Property* и *Association*. *Class* и *Property* связаны отношением композиции, в то время как элемент *Association* не может существовать самостоятельно и служит только для представления связей между элементами *Class*. Согласно семантике композиции, связанные объекты находятся в отношении "целое—часть". Такая структура может быть представлена в виде графа расположения, вершинами которого являются элементы *Class* и *Property*.

Возможны три вида отношений между объектами *Class*: наследование с помощью связи *superClass*; композиция, моделируемая объектами *Property* со свойством *is Composite* в значении *true* и объектом *Association*, а также аналогичным образом построенная ассоциация, но свойство *is Composite* объектов *Property* в этом случае имеет значение *false*. Ассоциация может быть *N*-арной. Очевидно, что все перечисленные виды связей между объектами могут быть представлены в виде гиперграфа, вершинами которого являются объекты метамодели, а ребрами — связи между ними.

Таким образом, изображенная на рис. 5 диаграмма может быть полностью описана с помощью компонентов биграфа. Это означает, что биграфы могут выступать в качестве универсального средства метамоделирования. В этом случае, как следует из рис. 1 и 5, существует однозначное соответствие между компонентами биграфа и элементами уровня *M*2 стандарта *MOF*, при котором сигнатура биграфа и множество классов метамодели *MOF* совпадают, отношение композиции, определенное на элементах метамодели, соответствует графу расположений, а ассоциации между элементами — графу связей биграфа метамодели. Благодаря этим соотношениям можно задать как правила перехода от метамодели *MOF* к биграфовому представлению, так и правила обратного перехода от биграфов к *MOF*. Рассмотрим преобразование метамодели *MOF* в ее биграфовое представление на примере метамодели реляционной базы данных на рис. 6 (поля классов не показаны).

Для получения биграфового представления моделей сформулируем заданные метамоделью огра-

ничения в терминах биграфов путем выполнения следующей последовательности действий:

1. Составить сигнатуру биграфов моделей. Множество K типов элементов моделей будет совпадать с множеством классов на диаграмме:

$$K = \{Schema, Table, Column, PrimaryKey, ForeignKey\}.$$

Число портов определяется количеством отношений ассоциации, в которых задействован класс. Так как элементы класса *Column* не могут одновременно быть связаны ассоциацией с *PrimaryKey* и *ForeignKey* (хотя на диаграмме этого ограничения не показано), типу *Column* соответствует один порт. Таким образом, имеем

$$\Sigma_{MM} = \left(\{K\}, \begin{Bmatrix}(Schema, 0), (Table, 0), (Column, 1), (PrimaryKey, 2),\\ (ForeignKey, 2)\end{Bmatrix}\right).$$

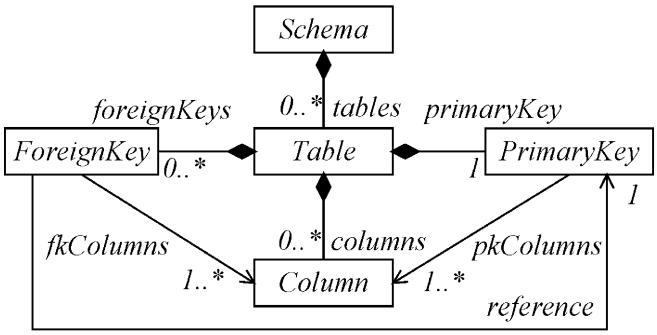


Рис. 6. Метамодель реляционной базы данных

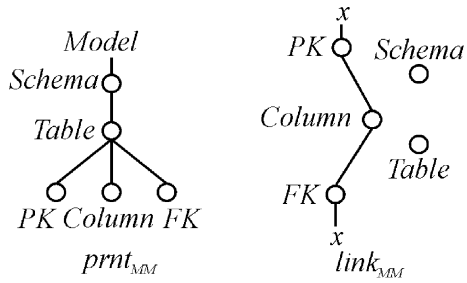


Рис. 7. Представление метамодели реляционной базы данных в виде биграфа

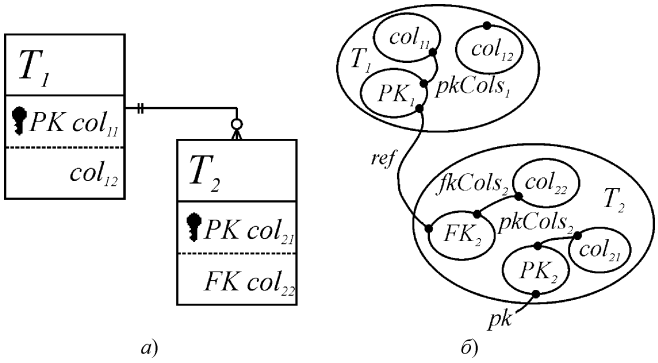


Рис. 8. Представление модели из двух таблиц со связью по внешнему ключу в виде биграфа

2. С учетом отношения композиции составить граф расположения метамодели $prnt_{MM}$, задать значение единственного корня, которым будет элемент, обозначающий модель целиком. Если элементы метамодели связаны отношением композиции, то соответствующие им вершины графа расположения будут соединены ребром.

3. Рассмотреть отношения ассоциации между классами и составить граф связей $link_{MM}$. Так как компоненты биграфов представляют собой неориентированные графы, направление ассоциации в расчет не принимается, но, если ассоциация используется для связи объектов с различным расположением (*ForeignKey* может ссылаться на *PrimaryKey* другой таблицы), на графе связей она разбивается на две связи, причем началу ассоциации (*ForeignKey*) соответствует внутреннее имя, а концу (*PrimaryKey*) — внешнее имя.

Для метамодели на рис. 6 получим графы расположения и связей, изображенные на рис. 7 (сокращения: *FK* — *ForeignKey*, *PK* — *PrimaryKey*).

Биграф модели

$$G_M = (V_M, E_M, P_M, ctrl_M, prnt_M, link_M): \langle m_M, X_M\rangle \to \to \langle n_M, Y_M\rangle$$

должен удовлетворять следующим условиям, накладываемым полученным представлением метамодели.

1. Соблюдение числа портов вершины каждого типа. Для этого необходимо совпадение сигнатур:

$$\Sigma_M = \Sigma_{MM}.$$

2. Корректность расположения вершин в графе расположения модели $prnt_M$: две вершины графа расположения модели соединены ребром, если соединены ребром вершины, соответствующие их типам в графе расположения метамодели:

$$\forall (v, u) \in V_M : (ctrl_M(v), ctrl_M(u)) \in prnt_{MM} \Rightarrow \Rightarrow (v, u) \in prnt_M.$$

3. Корректность связей между элементами: если существует связь между вершинами модели с определенными портами, то в метамодели между типами этих вершин также должна существовать связь с теми же портами:

$$\forall (p \in P, q \in Q, e \in E_M) : (p, e) \in link_M \wedge (q, e) \in \in link_M \Rightarrow \exists (e' \in E_{MM}) : ((ctrl_M(v), i), e') \in \in link_{MM} \wedge ((ctrl_M(u), j), e') \in link_{MM},$$

где

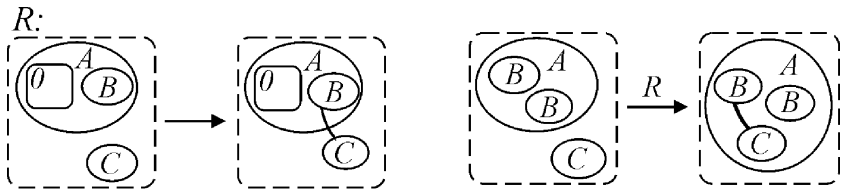
$$P = \{(v, i) | v \in V_M \wedge i \in ar(ctrl_M(v))\};$$

$$Q = \{(u, j) | u \in V_M \wedge j \in ar(ctrl_M(u)) \wedge (u \neq v \vee j \neq i)\}.$$

На рис. 8, *а* изображен пример реляционной модели, состоящей из двух таблиц, имеющих связь по внешнему ключу, а также на рис. 8, *б* показано соответствующее представление этой структуры в виде биграфа.

Моделирование поведения системы

Биграфы в том виде, в котором они рассмотрены выше, служат для описания состояния системы в некоторый момент времени. Для моделирования поведения системы (т. е. ее переходов из одного состояния в другое) используются правила реакции (*reaction rules*), схожие с правилами вывода в графовых грамматиках, но содержащие в левой (*redex*) и правой (*reactum*) частях биграфы. Процесс применения правила осуществляется путем поиска фрагмента, входящего в левую часть правила и замены его фрагментом в правой части.



а) *б*)

Рис. 9. Правило реакции (*а*) и результат его однократного применения (*б*)

На рис. 9, *а* представлено правило реакции R, устанавливающее связь между элементом C и элементом B, вложенным в элемент A, при этом в A возможно наличие других элементов. На рис. 9, *б* показан результат применения этого правила к биграфу, состоящему из элемента C, элемента A и двух вложенных в него элементов B. При этом выбор элемента B, который после применения правила будет соединен с элементом C, является недетерминированным.

В случае с трансформациями моделей набор правил реакции будет в себя включать собственно правила трансформации и возможные изменения, вносимые пользователями.

Биграф с сигнатурой Σ и набор правил реакции образуют биграфовую реагирующую систему (*bigraphical reactive system*) $Bg(\Sigma, \Re)$, где $\Re$ — набор правил реакции вида $R = (r, r')$.

Анализ поведения моделируемой системы осуществляется путем проверки желаемых свойств, выраженных с помощью биграфов. Например, сохраняет ли набор правил реакции определенное взаиморасположение элементов и характер связей между ними, или не приводит ли набор правил к каким-либо нежелательным состояниям системы. В целях более выразительного описания свойств системы можно использовать, например, возможности временной модальной логики. В этом случае каждое свойство системы будет представлено в виде некоторой формулы f. Для проверки свойств необходимо, анализируя набор правил реакции, выделить конечное множество возможных состояний системы и отношение переходов между ними, т. е. преобразовать биграфовую реактивную систему в модель Крипке:

$$Bg(\Sigma, \Re) \to (S, R, L),$$

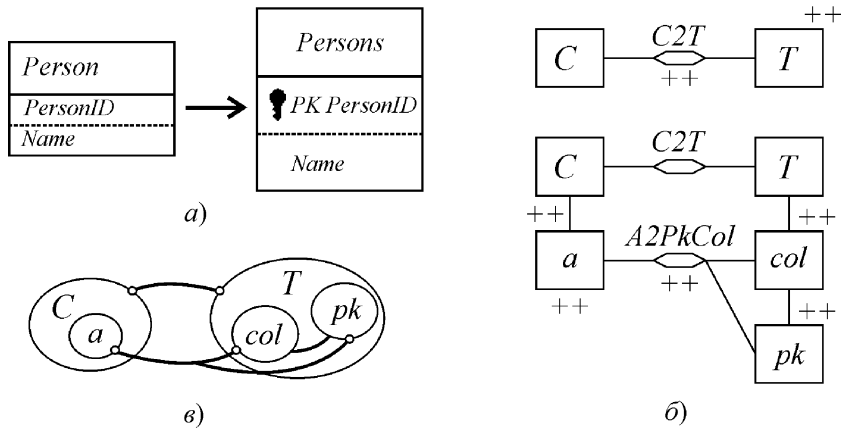
где S — множество состояний системы; $R \subseteq S \times S$ — отношение переходов между состояниями; $L : S \to 2^{AP}$ — функция, ставящая в соответствие каждому состоянию системы подмножество истинных в нем атомарных высказываний множества AP, выраженных в терминах биграфов. Для решения задачи верификации требуется обнаружить множество состояний системы, в которых выполнится f. В качестве метода решения этой задачи возможно использование алгоритмов проверки на моделях [14].

Кроме того, правила реакции позволяют описывать взаимодействие между элементами системы, работающими параллельно. В этом случае благодаря моделированию можно выяснить, корректно ли осуществляется доступ к общему ресурсу, не возникают ли конфликты при работе нескольких пользователей и т. д.

Спецификация правил трансформации

Рассмотрим процесс создания правил трансформации моделей на примере трансформации объектов диаграммы классов *UML* в объекты схемы реляционной базы данных. В этом случае каждый класс трансформируется в таблицу, а его атрибуты — в поля этой таблицы (рис. 10, *а*).

На рис. 10, *б* изображены два правила трансформации в нотации *TGG* (*Triple Graph Grammars*) [6], необходимые для преобразования объектов классов в таблицы. Правила состоят из трех частей: слева и справа расположены элементы моделей (C — класс, T — таблица, a — атрибут класса, col — столбец таблицы, pk — первичный ключ), а посередине —



а) *в*) *б*)

Рис. 10. Трансформация между объектами *UML*-диаграммы классов и реляционной схемы (*а*), правила трансформации в нотации *TGG* (*б*), правило трансформации, составленной с помощью биграфа (*в*)

связующие элементы (*C2T* — связь между классом и таблицей; *A2PkCol* — связь между атрибутом класса и первичным ключом таблицы). Знаком "++" обозначены элементы, которые созданы в результате применения правила. Элементы, не отмеченные этим знаком, должны существовать на момент применения правил, т. е. составляют условие его применения.

Механизм применения правила следующий: в графе модели выполняется поиск элементов из левой части правила. Если они найдены и условие применения правила выполняется, создаются элементы в правой части правила, помеченные знаком "++". Верхнее правило для каждого класса создает таблицу, а нижнее правило предназначено для создания у таблицы, полученной в результате применения первого правила, первичного ключа. Результатом применения каждого из правил является, помимо создания новых элементов, установление связей между элементами исходной модели классов и новыми элементами реляционной модели. Благодаря этой связи обеспечивается корректная последовательность обхода элементов, обработка изменений в моделях (например, при удалении какого-либо элемента удаляются также и все элементы, имеющие с ним связь).

Аналогичный способ описания правил трансформации можно предложить, используя биграфовое представление моделей (рис. 10, *в*). Правило состоит из двух частей, элементы которых связаны друг с другом с помощью гиперграфа. Наличие иерархической связи между элементами позволяет составить всего одно правило вместо двух в нотации *TGG*. В этом случае правило сначала создаст элементы для родительского элемента, а после этого — для дочерних элементов. Для описания связей между элементами каждому типу элементов добавляется дополнительный порт, который будет использован для установления связи с элементами в другой модели (изображены в виде белых кружков на рис. 10, *в*).

Как было отмечено ранее, стандарт *MOF* позволяет единообразно задавать отображения между метамоделями, что нашло применение в описании трансформаций моделей и отражено в стандарте *QVT* (*Query/View/Transformation*) [15]. Анализ связи между положениями данного стандарта и представлением правил трансформации в виде биграфов является довольно обширным и выходит за рамки статьи, но так как предлагаемый подход во многом схож с трансформациями моделей с помощью *TGG*, можно ознакомиться с работой [16].

Используя биграфы, можно создавать шаблоны правил. При составлении правил для каждой структуры в модели, подлежащей трансформации, нужно задать аналогичную структуру в терминах той метамодели, в экземпляр которой она преобразуется. Так как данный процесс осуществляется разработчиком вручную, возможны ошибки, которые могут привести к некорректному результату трансформации. Чтобы избежать такого рода ошибок, предлагается использовать биграфы и определенную на них операцию композиции в целях создания структур, имеющих желаемые свойства и связи с другими объектами, которые можно считать корректными на основании проведенного анализа метамоделей. Поясним данное предложение на конкретном примере.

На рис. 11 изображен шаблон для правила трансформации двух классов, связанных между собой отношением ассоциации, причем до этого момента отработали правила трансформации (аналогичные изображенным на рис. 10), согласно которым каждому из этих классов соответствует таблица, а их атрибутам — столбцы в этой таблице (порты на рисунке не показаны; точки внутри классов — анонимные элементы [13], использующиеся для связей между элементами, которые, в отличие от обычных связей, не приводят к удалению элементов, связанных с удаляемым).

Чтобы обработать элемент *Assoc*, связанный с обоими классами, нужно подобрать структуру в терминах реляционной модели, которая бы корректно отразила функциональное и структурное назначение элемента *Assoc*. Таких структур, причем корректных, может быть несколько в зависимости от кратности отношения. Чтобы гарантировать, что в правиле будет одна из них, создадим в нижней части правила шаблон, повторяющий элемент *Assoc*. Так как каждому классу соответствует таблица, в каждой из них разместим положения (*place*) (*1* и *3* на рис. 11), в которые могут войти структуры, обеспечивающие связь между таблицами. Так как элемент *Assoc* структурно отделен от классов, дополнительное положение (*place*) (*2* на рис. 11) разместим вне таблиц, связав его с положениями

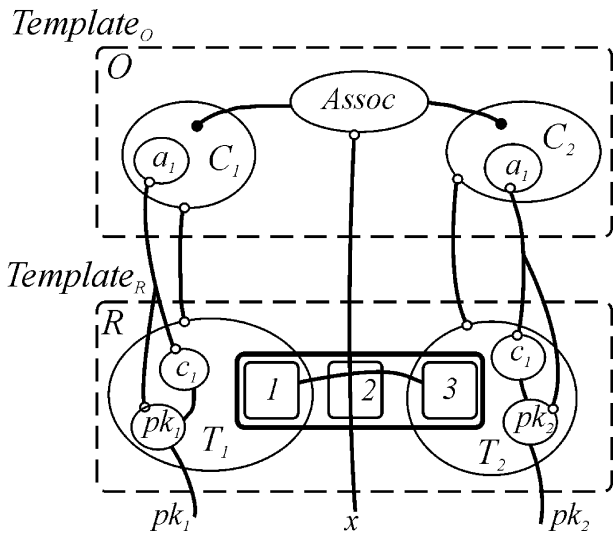


Рис. 11. Шаблон правила трансформации двух классов, связанных отношением ассоциации

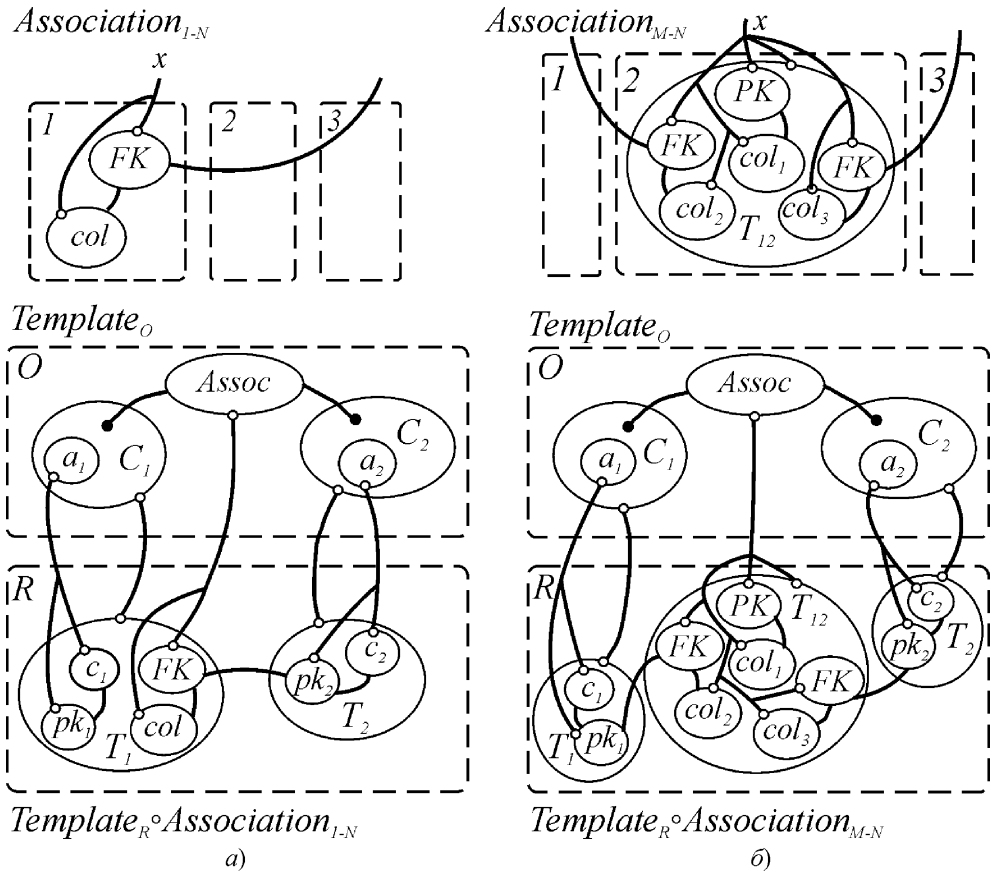


Рис. 12. Структуры, удовлетворяющие шаблону и полученные в результате правила трансформации

(*place*) внутри таблиц, а также с самим элементом *Assoc*. Таким образом, задан формат той структуры, которая в реляционном представлении будет соответствовать элементу *Assoc*. Для обеспечения связи между структурами создано внутреннее имя *x*.

На рис. 12 в верхней части представлены варианты структур, в которые будет трансформирован элемент *Assoc* (рис. 12, *а* — для отношения "один ко многим", 12, *б* — "многие ко многим"). Эти биграфы содержат элементы, которые предоставляют внутренние интерфейсы для соединения с другими элементами извне после выполнения операции композиции. В данном случае образуется связь между внешними и первичными ключами в таблицах (например, *FK* и pk_2 на рис. 12, *а*). Результатом композиции с биграфом-шаблоном, являются правила трансформации моделей, изображенные в нижней части рис. 12.

Таким образом, построенный шаблон задает ограничения для правила и служит дополнительной мерой защиты от ошибок при создании правил трансформации моделей. Заметим, что после операции композиции должна быть выполнена проверка на предмет того, является ли составленная пользователем часть правила (в данном случае реляционное представление) корректной с точки зрения метамодели, которой эта часть определяется.

Заключение

Отличительной особенностью биграфов как средства моделирования является объединение информации о взаимном расположении и связях между элементами рассматриваемой системы. Представление иерархических *UML*-подобных моделей в виде биграфов позволяет задавать ограничения на моделях, которые можно использовать в моделировании процессов трансформации.

Применение биграфов позволяет реализовать такие процедуры, как разработка правил трансформации, верификация процесса трансформации путем задания спецификации процессов в виде набора свойств и их доказательстве методом проверки на моделях. С помощью рассмотренных статических и динамических компонентов биграфов, а также перехода от стандарта *MOF* к биграфовому представлению, можно создать набор программных инструментов для работы с процессами трансформации моделей.

Таким образом, подход к проектированию процессов трансформации моделей, основанный на биграфах, является более формализованным и контролируемым в отличие от применяющихся в настоящее время методов.

Список литературы

1. **Unified** Modeling Language. URL: http://www.uml.org/ (дата обращения: 20.05.2015).
2. **Meta-Object** Facility. URL: http://www.omg.org/mof/ (дата обращения: 20.05.2015).
3. **Biehl M.** Literature Study on Model Transformations: tech. report, July 2010 / Royal Institute of Technology. Stockholm, 2010. 28 p.
4. **Schurr A.** Specification of graph translators with triple graph grammars // Proc. Of the 20th International Workshop on Graph-Theoretic Concepts in Computer Science. Herrsching, Germany, 1994. Vol. 903. P. 151—163.
5. **Hildebrandt S., Lambers L., Giese H., Reike J.** A survey of triple graph grammar tools // Bidirectional transformations. 2013. Vol. 57. P. 1—18.
6. **Kindler E., Wagner R.** Triple Graph Grammars: Concepts, Extensions, Implementations, and Application Scenarios: tech. report, June 2007. University of Padeborn. Padeborn, 2007. 79 p.
7. **Czarnecki K., Helsen S.** Feature-based survey of model transformation approaches // IBM Systems Journal — Model-driven software development. — 2006. Vol. 45. Is. 3. P. 621—645.
8. **Миронов А. М.** Теория процессов. URL: http://intsys.msu.ru/staff/mironov/processes.pdf (дата обращения: 20.05.2015).
9. **Milner R.** Bigraphical reactive systems // Proc. of 12th Conference on Concurrency Theory (CONCUR). Aalborg, 2001. Vol. 2154. P. 16—35.
10. **Milner R.** Bigraphs: a space for interaction. URL: http://www.cl.cam.ac.uk (дата обращения: 20.05.2015).
11. **Milner R.** Bigraphs: a model for mobile agents. URL: http://www.cl.cam.ac.uk (дата обращения: 20.05.2015).
12. **Grohmann D., Miculan M.** Reactive Systems over Directed Bigraphs // Proc. of 18th Conference on Concurrency Theory (CONCUR). Lisbon, Portugal, 2008. Vol. 4703. P. 380—390.
13. **Debois S., Damgaard T.** Bigraphs by Example: tech. report, October 2005. The IT University of Copenhagen. Copenhagen, 2005. 28 p.
14. **Кларк Э. М., Грамберг О., Пелед Д.** Верификация моделей программы: Model Checking / Пер. с англ. Под ред. Р. Смелянского. М.: МЦНМО, 2002. 416 с.
15. **Query/View/Transformation.** URL: http://www.omg.org/spec/QVT/1.1/ (дата обращения 20.05.2015).
16. **Greenyer J., Kindler E.** Reconcilling TGGs with QVT // Proc. of 10th International Conference on Model Driven Engineering Languages and Systems. Nashville, 2007. Vol. 4735. P. 16—30.

D. V. Oshkalo, Postgraduate Student, dmitry.oshkalo@gmail.com
Bauman Moscow State Technical University, Moscow

Bigraph-Based Development of Model Transformation Processes

Model transformation is one of the key activities of model-driven software development, which enables model and code generation, model refactoring, model synchronization, change propagation, etc. It is performed by applying a set of transformation rules to the model elements. The most practically used rule-based approaches (like relational QVT or graphical TGG) provide a formal background for specifying model transformations, but the lack of technique of developing a transformation rules' set for every concrete scenario makes the development of model transformation processes ineffective and error-prone, since there is no possibilities which allows to formulate and verify the properties of the rule set.

A using of bigraphs is proposed as a solution to build a framework for modeling model transformation processes as event-driven systems, which provides an extensible specification, that allows to define concrete model transformation rules. This approach considers models as bigraphs, which are graph-based theoretical tools that emphasize interplay between physical locality and connectivity. Bigraphs are also used as a constraint technique for building transformation rules' templates. This method allows checking the properties of the model transformations and modeling the transformation processes in multiuser environment when simultaneous model changes performed.

Keywords: *bigraphs, model-driven development, data models, model transformation, transformation rules, UML*

References

1. **Umfied** Modeling Language, available at: http://www.uml.org/ (accessed 20.05.2015).
2. **Eclipse** *Modeling Framework*, available at: http://eclipse.org/ modeling/emf/ (accessed 20.05.2015).
3. **Biehl M.** *Literature Study on Model Transformations*, Tech. report, July 2010, Royal Institute of Technology, Stockholm, 2010, 28 p.
4. **Schurr A.** Specification of graph translators with triple graph grammars, *Proc. of the 20th International Workshop on Graph-Theoretic Concepts in Computer Science, Herrsching, Germany*, 1994, vol. 903, pp. 151—163.
5. **Hildebrandt S., Lambers L., Giese H., Reike J.** A survey of triple graph grammar tools, *Bidirectional transformations*, 2013, vol. 57, pp. 1—18.
6. **Kindler E., Wagner R.** *Triple Graph Grammars: Concepts, Extensions, Implementations, and Application Scenarios.* Tech. report, June 2007, University of Padeborn, Padeborn, 2007, 79 p.
7. **Czarnecki K., Helsen S.** Feature-based survey of model transformation approaches, *IBM Systems Journal — Model-driven software development*, 2006, vol. 45, is. 3, pp. 621—645.
8. **Mironov A. M.** *Teorija processov*, available at: http://intsys.msu.ru/staff/mironov/processes.pdf (accessed 20.05.2015).
9. **Milner R.** Bigraphical reactive systems. *Proc. of 12th Conference on Concurrency Theory (CONCUR), Aalborg, Denmark*, 2001, vol. 2154, pp. 16—35.
10. **Milner R.** Bigraphs: a space for interaction, available at: http://www.cl.cam.ac.uk (accessed 20.05.2015).
11. **Milner R.** *Bigraphs: a model for mobile agents*, available at: http://www.cl.cam.ac.uk (accessed 20.05.2015).
12. **Grohmann D., Miculan M.** Reactive Systems over Directed Bigraphs. *Proc. of 18th Conference on Concurrency Theory (CONCUR), Lisbon, Portugal,* 2008, vol. 4703, pp. 380—390.
13. **Debois S., Damgaard T.** *Bigraphs by Example.* Tech. report, October 2005, The IT University of Copenhagen, Copenhagen, 2005, 28 p.
14. **Clarke E. M., Grumberg O., Peled D.** Model Checking, MIT Press Cambridge, MA., 1999, 330 p.
15. **Meta-Object** Facility, available at: http://www.omg.org/mof/ (accessed 20.05.2015).
16. **Greenyer J., Kindler E.** Reconcilling TGGs with QVT // Proc. of 10th International Conference on Model Driven Engineering Languages and Systems. Nashville, 2007. Vol. 4735. P. 16—30.

УДК 004.023

А. С. Филиппова, д-р техн. наук, проф., e-mail: annamuh@mail.ru,
Башкирский государственный педагогический университет им. М. Акмуллы,
Э. И. Дяминова, канд. техн. наук, доц., e-mail: xasel@mail.ru,
Ю. И. Валиахметова, канд. техн. наук, доц., e-mail: julikas@inbox.ru,
Уфимский государственный авиационный технический университет

Метод ограниченной декомпозиции для решения комплексной задачи геометрического покрытия и раскроя

Рассматривается NP-трудная комплексная задача геометрического покрытия и раскроя. Эффективность ее решения оценивается двумя критериями: коэффициентом покрытия и коэффициентом раскроя. Предлагается метод ограниченной декомпозиции области покрытия, использующий матричное представление исходных данных. Проведен вычислительный эксперимент, направленный на исследование эффективности предложенного метода и ранее разработанного гибридного эволюционного алгоритма.

Ключевые слова: *геометрическое покрытие, раскрой, многосвязный ортогональный полигон, метод матричной декомпозиции, метод ограниченной декомпозиции*

Введение

Постановка и математическая модель рассматриваемой в статье комплексной задачи геометрического покрытия и раскроя (ГПР) разработана в рамках уфимской научной школы [1] по исследованию задач раскроя-упаковки. Сама проблема представляет собой синтез двух хорошо известных оптимизационных подзадач: задачи раскроя и задачи геометрического покрытия. Ранее на практике решение этих подзадач осуществлялось по отдельности и не рассматривалось в комплексе. Однако решение комплексной задачи ГПР является целесообразным с точки зрения ресурсосбережения, как показано в работах [2, 3]. Комплексная задача ГПР заключается в нахождении размеров покрывающих прямоугольных элементов, плана покрытия заданной многоугольной области, плана раскроя покрывающих элементов из прямоугольных листов или рулонного материала заданных размеров. При этом требуется максимизировать размеры покрывающих элементов и минимизировать расход раскраиваемого ресурса.

Каждая подзадача относится к классу NP-трудных проблем раскроя-упаковки [4]. Начало широкого научного исследования задач раскроя положила в 1951 г. книга Л. В. Канторовича и В. А. Залгаллера [5]. Различают одномерные (линейные), двумерные (прямоугольные), трехмерные задачи [6, 7]. Цель задач раскроя-упаковки разместить предметы известных размеров в заданном ресурсе: если ресурс задан в виде стержней, корзин, листов, контейнеров и т. п., то минимизировать их число; если ресурс в виде полубесконечной полосы, полубесконечного контейнера, то минимизировать занятую площадь/объем.

В настоящее время под задачами раскроя-упаковки понимается широкий класс проблем, допускающих различное прикладное толкование, например, задача составления расписания многопроцессорных систем, задача выбора ассортимента предметов, задача планировки помещений, задача обеспечения ритмичности производственного процесса, задача распределения памяти вычислительной машины и др. [6, 7]. В свою очередь математические модели каждой из них имеют общую структуру и могут быть различным образом конкретизированы. Для решения подобных задач применяются общие подходы: точные методы, простые эвристики и метаэвристики [7]. Ввиду неполиномиальной сложности точных алгоритмов решения этих задач авторами многих работ уделяется значительное внимание приближенным методам. Аналитический обзор, посвященный задачам раскроя-упаковки и методам их решения, приведен в работах [6, 7]. Современное состояние методов и алгоритмов их решения приведено в обзоре, выполненном под руководством Э. А. Мухачевой, которым дополнено третье переиздание книги Л. В. Канторовича и В. А. Залгаллера [8]. Вопросам исследования проблем раскроя-упаковки посвящаются ежегодные международные конференции в рамках научной группы ESICUP (Euro Special Interest Group on Cutting and Packing) [9].

Задачи геометрического покрытия исследуются последние десятилетия. Постановки имеют большое разнообразие, которое диктует практическая направленность [2, 10—14]. Как правило, целью решения этих задач является минимизация числа покрывающих предметов (элементов). Например, в работах [10, 14] рассмотрены случаи покрытия многоугольной области минимальным числом кругов заданного радиуса. Встречаются задачи, цель кото-

рых — максимизация покрытой области [11]. Основное отличие постановок задач раскроя-упаковки и геометрического покрытия заключается в ограничениях: в задачах геометрического покрытия необходимо размещать элементы без "просветов" между ними, а в некоторых случаях [10—12, 14] разрешается пересечение покрывающих элементов между собой и сторонами покрываемой области; в задачах раскроя-упаковки подобное перекрытие не допускается. Несмотря на различия, задачи раскроя и покрытия имеют близкую структуру математических моделей, поэтому для решения часто используют схожие методы и подходы.

В настоящее время для практического решения задач раскроя-упаковки и геометрического покрытия популярны высокоэффективные метаэвристические алгоритмы, в том числе и на основе эволюционных стратегий [12, 15, 16]. В данной статье предлагается метод решения комплексной задачи ГПР на основе эволюционной стратегии. Схема предлагаемого метода решения комплексной задачи предполагает использование ранее разработанных авторами алгоритмов решения подзадачи прямоугольного раскроя [16]. Кроме того, возможно использование других алгоритмов решения задачи прямоугольного раскроя, в том числе точных, например, в случае небольшой (до 100 предметов) размерности подзадачи раскроя [17], или однопроходных эвристических, позволяющих получать рациональные решения за полиномиальное время [18].

Для решения подобных NP-трудных задач актуальной является разработка новых подходов с использованием быстрых эвристических и метаэвристических алгоритмов. В данной статье для решения комплексной задачи ГГТР предлагается метод ограниченной декомпозиции, основанный на модификации метода матричной декомпозиции, ранее разработанного для задачи размещения ортогональных объектов на полигонах с препятствиями [19].

Постановка комплексной задачи ГПР

Введем следующие понятия и определения.

Назовем *ортогональным полигоном* односвязную область, ограниченную многоугольником, возможно невыпуклым, ребра которого либо вертикальны, либо горизонтальны.

Многосвязным ортогональным полигоном (МОП) будем называть ортогональный полигон, содержащий односвязные запретные области, являющиеся ортогональными полигонами.

Прямоугольный ресурс в задачах раскроя-упаковки и геометрического покрытия — это листы (контейнеры) прямоугольной формы заданных размеров либо полубесконечная полоса (рулон) заданной ширины.

Под *деловыми отходами* будем понимать прямоугольный ресурс нестандартных размеров. Как правило, на практике часто возникает потребность в использовании деловых отходов в целях ресурсосбережения. Деловые отходы могут быть получены при решении задач раскроя-упаковки (и/или геометрического покрытия) как неиспользованные фрагменты прямоугольного ресурса.

В статье рассматривается двухкритериальная проблема, встречающаяся, например, в строительной индустрии при покрытии пола строительным материалом (линолеумом, древесноволокнистыми плитами и др.). Впервые ее постановка приведена в работе [2]. В связи с NP-трудностью проблемы, сложностью представления и обработки исходной информации при использовании фигурных объектов авторами сделано допущение, практикуемое, например, при решении задачи фигурного раскроя [7, 8].

В случае неортогональных границ покрываемой геометрической области (рис. 1, *а*) она аппроксимируется вертикальными и горизонтальными линиями покрытия (рис. 1, *б*). Каждое препятствие, имеющее непрямоугольную форму, произвольным образом разбивается на прямоугольники, и далее под препятствиями будем понимать прямоугольные области. Таким образом, для решения задачи область покрытия задается в виде МОП (рис. 1, *в*).

Кроме размеров МОП задаются размеры прямоугольного ресурса. Необходимо найти план покрытия МОП покрывающими элементами и план раскроя прямоугольного ресурса на эти элементы, при этом максимизировать размеры покрывающих элементов и минимизировать расход (площадь или количество) раскраиваемого ресурса. Эта проблема формализуется математической моделью задачи с двумя целевыми функциями.

Для описания области покрытия МОП введем прямоугольную систему координат: оси Ox и Oy совпадают соответственно с нижней и левой боковой сторонами огибающего МОП прямоугольника (рис. 2, *а*), W, L — ширина и длина огибающего МОП прямоугольника. Тогда положение каждого прямоугольного запретного участка B_ν задается координатами (χ_ν, η_ν) его нижнего левого угла и размерами ω_ν, λ_ν, $\nu = \overline{1, \mu}$, где μ — число запретных участков.

Математическая модель комплексной задачи геометрического покрытия и раскроя. *Дано:* МОП, подлежащий покрытию, заданный совокупностью $P = \langle W, L, (\chi_\nu, \eta_\nu), (\omega_\nu, \lambda_\nu), \nu = \overline{1, \mu} \rangle$. Кроме того,

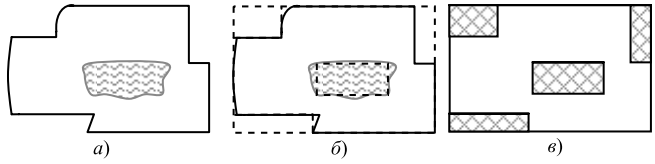


Рис. 1. Пример аппроксимации фигурной области с помощью МОП:
а — исходная область; *б* — аппроксимация области вертикальными и горизонтальными линиями; *в* — МОП

имеется неограниченное множество прямоугольных листов длины Θ и ширины h (для задач с ресурсом в виде листов), либо ширина h рулонного ресурса, подлежащего раскрою на прямоугольные элементы (рис. 2, *б*). Также задана информация о деловых отходах в виде множества $K = (k_1, k_2, ..., k_\tau)$ прямоугольных элементов с размерами $l_j^+ \times w_j^+$, $j = \overline{1, \tau}$.

Требуется: найти план покрытия МОП P (рис. 3), план раскроя прямоугольного ресурса (рис. 4) на покрывающие прямоугольные элементы и минимизировать значения функций, характеризующих эффективность решения подзадач геометрического покрытия и раскроя:

$$\min F_{cov}(T) = \sum_{i=1}^{m} (l_i + w_i);$$

$$\min F_{cut}(T) =$$

$$= \begin{cases} N^{\text{РЛ}}, \text{если материал — прямоугольные листы;} \\ L^{\text{РП}} = \max\limits_{i=1...m} (x'_i + l_i), \text{если материал —} \\ \text{полубесконечная полоса,} \end{cases}$$

где $T = \{m, w, l\}$, $w = (w_1, w_2, ..., w_m)$, $l = \{l_1, l_2, ..., l_m)$ — множество из m прямоугольных элементов с размерами $w_i \times l_i$, $i = \overline{1, m}$, образующих покрытие МОП и подлежащих раскрою из ресурса и деловых отходов; l_i, w_i — длина и ширина покрывающего прямоугольного элемента $t_i \in T$, входящего в покрытие МОП, $T = Q \cup K^+$, Q — множество покрывающих МОП прямоугольных элементов, раскроенных из стандартного ресурса; K^+ — множество покрывающих МОП прямоугольных элементов, раскроенных из деловых отходов; $N^{\text{РЛ}}$ — число использованных прямоугольных листов (в случае листового ресурса в задаче раскроя); $L^{\text{РП}}$ — длина занятой части полубесконечной полосы (в случае рулонного ресурса в задаче раскроя); x'_i, y'_i — координаты левого нижнего угла прямоугольника с номером i в плане раскроя прямоугольного ресурса.

Для формализации задачи введем дополнительно переменные z_{ij}^x, z_{ij}^y, $\tilde{z}_{iv}^x$ и $\tilde{z}_{iv}^y$, такие что:

- $z_{ij}^x = 1$ ($z_{ij}^y = 1$), если проекция прямоугольного элемента i на ось Ox (Oy) находится левее (ниже) проекции прямоугольного элемента j на ось Ox (Oy), иначе $z_{ij}^x = 0$ ($z_{ij}^y = 0$);
- $\tilde{z}_{iv}^x = 1$ ($\tilde{z}_{iv}^y = 1$), если проекция прямоугольного элемента i на ось Ox (Oy) находится левее (ниже) проекции запретной области v на ось Ox (Oy), иначе $\tilde{z}_{iv}^x = 0$ ($\tilde{z}_{iv}^y = 0$).

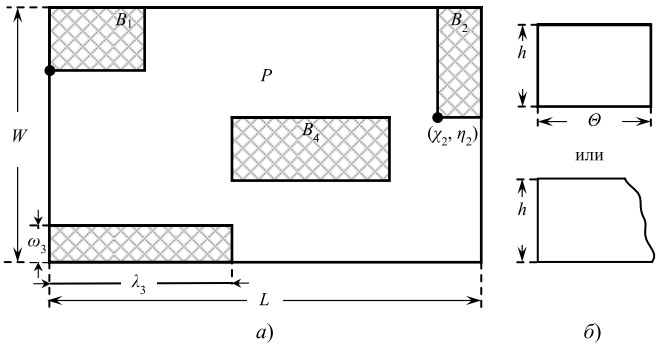


Рис. 2. Иллюстрация постановки комплексной задачи: *а* — МОП P с запретными областями B_v; *б* — размеры ресурса (листы/полубесконечный рулон)

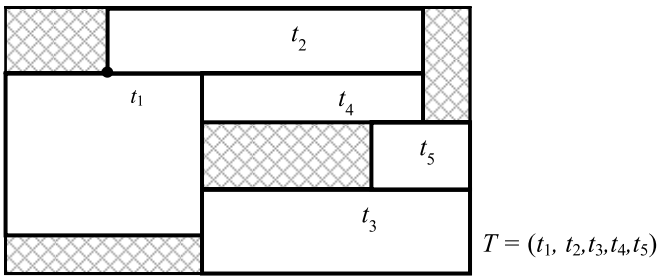


Рис. 3. План покрытия МОП P множеством покрывающих прямоугольников T

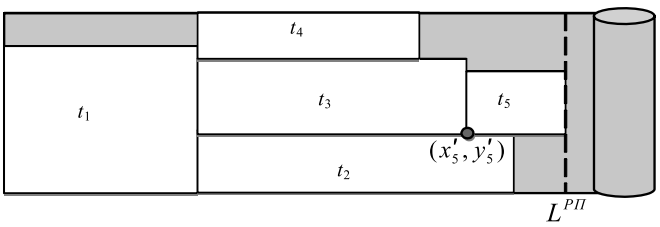


Рис. 4. План раскроя рулона на покрывающие элементы T. Длина занятой части полосы $L^{\text{РП}} = \max\limits_{i=1,...,m} (x'_i + l_i) = x'_5 + l_5$

Необходимо найти:

1. Множество T прямоугольных элементов, покрывающих МОП P, и соответствующий план покрытия, т. е. для каждого $t_i \in T$ требуется отыскать набор значений $\langle x_i, y_i, l_i, w_i \rangle$, где x_i, y_i — координаты левого нижнего угла элемента в системе координат МОП P, l_i и w_i — соответственно длина и ширина прямоугольного элемента, таких что:

$$l_i \leqslant h,\ w_i \leqslant \Theta,\ i = \overline{1, m}; \quad (1)$$

$$x_i + l_i \leqslant L,\ y_i + w_i \leqslant W,\ i = \overline{1, m}; \quad (2)$$

$$x_i \geqslant 0,\ y_i \geqslant 0,\ i = \overline{1, m}; \quad (3)$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geqslant 1,\ i = \overline{1, m},\ j = \overline{1, m},\ i \neq j; \quad (4)$$

$$y_j \geqslant y_i + w_i - W(1 - z_{ij}^y),\ i = \overline{1, m},\ j = \overline{1, m}; \quad (5)$$

$$x_j \geqslant x_i + l_i - L(1 - z_{ij}^x),\ i = \overline{1, m},\ j = \overline{1, m}; \quad (6)$$

$$\tilde{z}_{iv}^x + \tilde{z}_{vi}^x + \tilde{z}_{iv}^y + \tilde{z}_{vi}^y \geqslant 1,\ i = \overline{1, m},\ v = \overline{1, \mu}; \quad (7)$$

$$\eta_v \geqslant y_i + w_i - W(1 - \tilde{z}_{iv}^y),\ i = \overline{1, m},\ v = \overline{1, \mu}; \quad (8)$$

$$\chi_v \geqslant x_i + l_i - L(1 - \tilde{z}_{iv}^x),\ i = \overline{1, m},\ v = \overline{1, \mu}; \quad (9)$$

$$\sum_{i=1}^{m} w_i l_i = S_{\text{МОП}}. \quad (10)$$

Условия (1) накладывают ограничения на размеры покрывающих МОП прямоугольных элементов: они не должны превышать размеры прямоугольного ресурса. Условия (2), (3) задают допустимое положение прямоугольного элемента в области МОП: они не должны выходить за границы МОП. Условия (4)—(6) обеспечивают неперекрытие прямоугольных элементов между собой, (7)—(9) — неперекрытие прямоугольных элементов с запретными областями. Условие (10) описывает необходимость полного геометрического покрытия полезной площади МОП прямоугольными элементами (без "просветов").

2. План раскроя ортогонального ресурса на прямоугольные элементы множества T такой, что выполняются следующие условия:

$$x_i' \geqslant 0,\ y_i' \geqslant 0,\ i = \overline{1, m}; \quad (11)$$

$$x_i' + l_i \leqslant \Theta,\ y_i' + w_i \leqslant h,\ i = \overline{1, m}; \quad (12)$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geqslant 1,\ i = \overline{1, m},\ j = \overline{1, m},\ i \neq j; \quad (13)$$

$$y_j' \geqslant y_i' + w_i - W(1 - z_{ij}^y),\ i = \overline{1, m},\ j = \overline{1, m}; \quad (14)$$

$$x_j' \geqslant x_i' + l_i - L(1 - z_{ij}^x),\ i = \overline{1, m},\ j = \overline{1, m}. \quad (15)$$

Условия (11)—(12) обеспечивают допустимое размещение прямоугольных элементов внутри ортогонального ресурса, условия (13)—(15) — неперекрытие прямоугольных элементов между собой.

Целевые функции $\min F_{cov}(T)$ и $\min F_{cut}(T)$ позволяют сравнивать качество различных решений одной и той же задачи, однако не позволяют объективно сравнивать эффективность решений различных задач, поскольку их значения зависят от исходных данных, например, от размеров ресурса и МОП.

Поэтому качество полученных решений оценивается также с помощью двух относительных показателей:

- для задачи геометрического покрытия рассчитывается коэффициент покрытия

$$k_{cov} = \frac{S_{MOP}}{P_T} \frac{P_{resource}}{S_{resource}},$$

где S_{MOP} и $S_{resource}$ — площади МОП и ресурса соответственно; P_T и $P_{resource}$ — суммарные периметры покрывающих элементов и ресурса соответственно [20];

- для задачи ортогонального раскроя определяется коэффициент раскроя

$$k_{cut} = \frac{\sum_{i=1}^{m} w_i l_i}{S_{resource}}.$$

Коэффициент геометрического покрытия равен 1, если стыки покрывающих элементов находятся лишь на границах МОП и не проходят по внутренней площади МОП. В противном случае $k_{cov} < 1$.

При безотходном решении задачи, т. е. когда отсутствуют потери материала, коэффициент раскроя равен 1. В противном случае $k_{cut} < 1$.

Коэффициенты раскроя и покрытия являются противоречивыми поскольку максимизация размеров покрывающих элементов влечет за собой снижение коэффициента раскроя (увеличение отходов). И, наоборот, при мелких размерах раскраиваемых элементов увеличивается коэффициент раскроя, но уменьшается коэффициент покрытия.

Методы решения комплексной задачи ГПР

Для решения комплексной задачи геометрического покрытия и раскроя [2, 20, 21] предложены гибридные эволюционные алгоритмы. Они основаны на трехэтапном подходе последовательного решения подзадач:

1) декомпозиция МОП на минимальное число прямоугольных областей;

2) определение плана покрытия МОП и размеров прямоугольных покрывающих элементов;

3) определение плана раскроя ресурса на покрывающие элементы.

В работах [20, 21] приведены примеры решения комплексной задачи ГПР в судостроении. В качестве процедуры решения подзадачи декомпозиции МОП на минимальное число прямоугольных областей был применен известный метод матричной декомпозиции. Он разработан ранее Э. И. Хасановой (Дяминовой) [19] для задачи размещения прямоугольных объектов на ортогональном полигоне с запретными участками.

Однако возникает вопрос об эффективности трехэтапного подхода, поскольку параметры ресурса при декомпозиции МОП на прямоугольные области покрытия не учитываются. Для повышения эффективности решения предлагается объединить этапы декомпозиции МОП на прямоугольные области и определения плана покрытия, т. е. проводить декомпозицию МОП на минимальное число прямоугольных областей, размеры которых не превышают размеров ресурса. Для этого предлагается модифи-

цировать метод матричной декомпозиции, добавив в него дополнительное ограничение на размеры прямоугольных областей.

Таким образом, предлагается новый ***метод ограниченной декомпозиции области покрытия*** для поиска рационального решения комплексной задачи ГПР, основанный на подходе в два этапа: первый — решение подзадачи декомпозиции МОП на прямоугольные области, размеры которых не превышают размеры ресурса, и второй — раскрой ресурса на покрывающие элементы, размеры которых равны размерам ограниченных прямоугольных областей. Таким образом, прямоугольные области, полученные в ходе декомпозиции МОП, составляют множество покрывающих элементов мощности m. Рассмотрим каждый этап подробнее.

Подзадача декомпозиции исходного МОП на прямоугольные области ограниченного размера. Задан МОП, подлежащий геометрическому покрытию: $P = \langle W,\ L,\ (\chi_\nu,\ \eta_\nu),\ (\omega_\nu,\ \lambda_\nu),\ \nu = \overline{1, \mu}\,\rangle$.

Требуется найти множество $\Pi = \{\Pi_1, \Pi_2, ..., \Pi_m\}$ прямоугольных областей $\Pi_i = \langle (x_i,\ y_i),\ (w_i,\ l_i)\rangle$ минимальной мощности, где x_i, y_i — координаты нижнего левого угла i-й области; w_i и l_i — ее ширина и длина соответственно; m — искомая мощность множества Π, удовлетворяющее следующим условиям:

- допустимость положения прямоугольной области в МОП;
- неперекрытие прямоугольных областей между собой;
- неперекрытие прямоугольных областей с запретными участками;
- полное геометрическое покрытие полезной площади МОП прямоугольными областями;
- $w_i \leqslant h,\ l_i \leqslant \Theta,\ i = \overline{1, m}$ — дополнительное ограничение, при выполнении которого любой прямоугольный элемент из множества покрывающих элементов может быть целиком вырезан из ресурса.

Процедура матричной декомпозиции МОП на прямоугольные области ограниченных размеров основана на модификации метода матричной декомпозиции [19]. Отличие заключается в дополнительной проверке размеров получаемых прямоугольных областей и, в случае необходимости, — дополнительного разделения их на максимально возможные по размерам (Шаг 1).

Шаг 1. *Представление МОП в матричном виде.*

Шаг 1.1. Через горизонтальные грани запретных участков проведем сквозные линии. В результате МОП оказывается разделен на G прямоугольников с длинами L' и ширинами d_γ, $\gamma = \overline{1, G}$. Если для какого-то из прямоугольников выполняется неравенство $d_\gamma > h$, делим его сквозной горизонтальной линией на два прямоугольника с ширинами h и $(d_\gamma - h)$. Горизонтальное разбиение МОП продолжаем до тех пор, пока не выполнится условие: $d_\gamma \leqslant h$, $\gamma = \overline{1, G}$.

Шаг 1.2. Через вертикальные грани запретных областей также проведем сквозные линии. В результате МОП оказывается разделен на U прямоугольников с длинами q_φ, $\varphi = \overline{1, U}$, и ширинами W'. Если в качестве ортогонального ресурса выступает полубесконечная полоса, то ограничений на длину областей не будет. В противном случае, по аналогии с горизонтальным разбиением, вертикальное деление МОП продолжается до тех пор, пока не выполнится условие: $q_\varphi \leqslant \Theta$, $\varphi = \overline{1, U}$.

В результате МОП оказывается покрытым растровой сетью, каждая ячейка которой либо не содержит запретных областей (пустая), либо является запретной областью или частью запретной области. При этом размеры ячеек не превышают размеров ресурса.

Шаг 1.3. Сопоставим каждой ячейке (γ, σ) число 0, если она свободна, и 1, если она является запретной областью. Тогда исходный МОП может быть описан булевой матрицей (рис. 5).

Шаг 2. *Выбор исходной ячейки и объединение.*

Шаг 2.1. В качестве исходной ячейки выбираем левую нижнюю еще не рассмотренную ячейку МОП. Если она является запретной областью, движемся вверх по столбцу, пока не найдем пустую ячейку, ее принимаем за исходную.

Шаг 2.2. Объединение пустых ячеек, начиная с исходной, вертикальным, горизонтальным или диагональным способом. Способы чередуются случайным образом для каждой исходной ячейки. Объединение ячеек продолжается до тех пор, пока не выполнится хотя бы одно из условий:

- достигнута верхняя граница МОП;
- на пути объединения имеется запретная область;
- ширина бокса достигла h;
- длина бокса достигла Θ (если ресурс поставляется в листах).

Переход на Шаг 2.1.

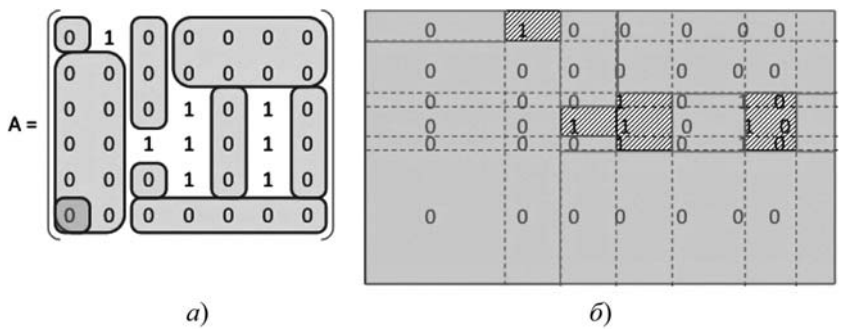


Рис. 5. Иллюстрация процедуры декомпозиции МОП:
а — матричное представление МОП; *б* — графическое

Шаг 3. *"Уровневое" выделение пустых прямоугольных областей.*

Когда достигнута верхняя граница МОП, через правую границу области наибольшей длины проводится условный сквозной рез, выделение следующей прямоугольной области покрытия продолжается справа от реза. К образовавшимся при этом вторичным МОП рекурсивно применяется процедура матричной декомпозиции МОП (возвращаемся к шагу 2). Работа алгоритма продолжается до тех пор, пока не останется пустых ячеек. Если ширина всех выделенных прямоугольных областей равна h, а длина Θ, то подзадачу можно считать решенной: каждая прямоугольная область может быть без остатка покрыта листом прямоугольного ресурса. Если ресурс — полубесконечная полоса и ширина прямоугольной области равна h, то материал для покрытия каждой области получается одним вертикальным разрезом рулонного ресурса.

Благодаря модификации метода матричной декомпозиции МОП на прямоугольные области ограниченных размеров получаем план покрытия исходного МОП покрывающими элементами, размеры которых не превышают размеров ресурса.

Подзадача раскроя ортогонального ресурса. На этапе раскроя ресурса можно применить любой наиболее подходящий для конкретной прикладной ситуации алгоритм. Например, в работе [21] описан алгоритм решения задачи, где на этапе раскроя применяется послойный алгоритм с оценками.

Таким образом, основные шаги ***метода ограниченной декомпозиции области покрытия*** для решения комплексной задачи ГПР следующие:

1. *Решение подзадачи декомпозиции МОП.* Входные данные: размеры МОП P и ресурса. Применение процедуры матричной декомпозиции МОП на прямоугольные области ограниченных размеров. Выходные данные: план покрытия МОП, размеры покрывающих элементов, значение коэффициента покрытия.

2. *Решение подзадачи раскроя.* Входные данные: размер ресурса и размеры покрывающих элементов. Применение алгоритма прямоугольного раскроя. Выходные данные: план раскроя, значение коэффициента раскроя.

В случае, если полученные значения коэффициентов покрытия и раскроя не удовлетворяют, можно реализовать многократное повторение основных этапов метода и выбора лучшего варианта. Это целесообразно в связи со случайным процессом шага 2.2 процедуры матричной декомпозиции.

Вычислительный эксперимент

Проведем сравнение эффективности метода ограниченной декомпозиции области покрытия и гибридного эволюционного алгоритма [20] на основе двухэтапного и трехэтапного подходов к решению соответственно.

Эксперимент 1: решение практической комплексной задачи ГПР. Для сравнения эффективности подходов была рассмотрена реальная задача судостроительной индустрии [21]. На палубах B и C судна снабжения некоторого проекта имеется два помещения, для которых известны размеры [21]. Необходимо покрыть пол помещений морской фанерой одного из четырех размерных типов: тип a (1,22 × 2,44 м); тип b (1,25 × 2,5 м); тип c (0,915 × 1,83 м); тип d (1,22 × 2,135 м). Помещения аппроксимированы МОП с описывающими область прямоугольниками с размерами для палубы B 17,35 × 11,33 м и для палубы C — 15,5 × 14 м, наборами запретных областей с известными размерами и координатами. Далее были применены методы на основе двухэтапного и трехэтапного подходов к решению задачи. Результаты представлены в виде диаграмм на рис. 6, 7.

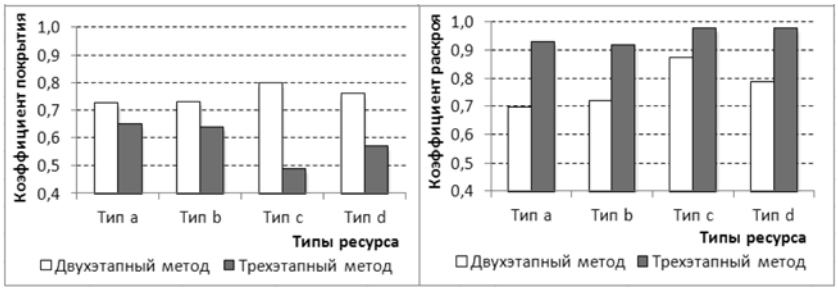


Рис. 6. Результаты решения задачи покрытия области МОП палубы *B*

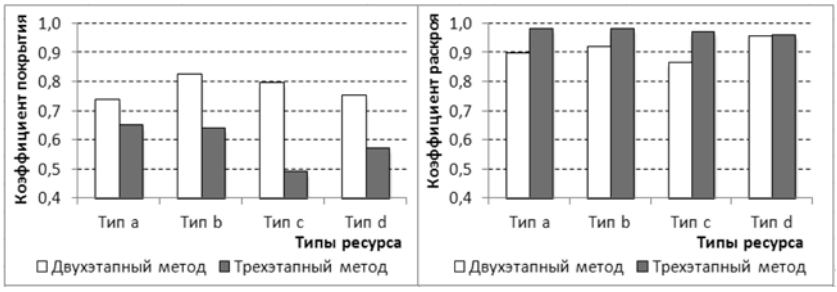


Рис. 7. Результаты решения задачи покрытия области МОП палубы *C*

Для обоих помещений и каждого типа ресурса эксперимент показал, что двухэтапный подход (метод ограниченной декомпозиции области покрытия) позволяет получить решения, лучшие по коэффициенту покрытия (на 8...30 %), т. е. выделяются более крупные покрывающие элементы. Трехэтапный подход позволяет получить решения, лучшие по коэффициенту раскроя (на 0,5...23 %), т. е. более эффективен с точки зрения экономии ресурса. Таким образом, выбор наиболее приемлемого метода решения задачи покрытия в каждом конкретном случае зависит от приоритетов заказчика.

Эксперимент 2: решение безотходной комплексной задачи ГПР. В данном эксперименте сравнивалась эффективность решения безотходных задач ГПР. Под безотходной задачей ГПР будем понимать задачу, коэффициенты раскроя и покрытия оптимального решения которой равны 1. Подобные задачи часто используются для тестирования алгоритмов решения NP-трудных задач [22]. Преимуществом такого рода задач является то, что для них априори известны оптимальные покрытие и раскрой.

Было сгенерировано 10 примеров безотходных задач, оптимальный план покрытия МОП которых состоит из 10 покрывающих элементов. Размеры этих элементов одинаковы и совпадают с размерами ресурса 1220 × 2440. Примеры различаются формой и размерами МОП. Таким образом, оптимальная декомпозиция МОП на прямоугольные области в данных примерах одновременно является и оптимальным планом покрытия.

Каждая из сгенерированных задач была решена с помощью методов на основе двух- и трехэтапного подходов, полученные коэффициенты покрытия и раскроя в каждом случае фиксировались (рис. 8 *а*, *б*).

Как видно из диаграммы на рис. 8, *а*, оба подхода позволяют получать высокие значения коэффициента раскроя: в 8 из 10 примеров было получено оптимальное решение. Средний коэффициент раскроя для двухэтапного метода составил 96,8 %, для трехэтапного — 98,4 %. Причиной этому может быть, во-первых, малая размерность примеров, во-вторых, однотипность формы ресурса.

При решении же задачи геометрического покрытия полученные результаты не оптимальны ни в одном из тестовых примеров. Это означает, что выделяемые в процессе решения прямоугольные области покрытия имеют размеры меньше эталонных. При этом двухэтапный алгоритм в 9 из 10 случаев показал себя существенно лучше трехэтапного (на 4...29 %). Средний коэффициент покрытия для двухэтапного метода составил 81,6 %, для трехэтапного — 68,6 %.

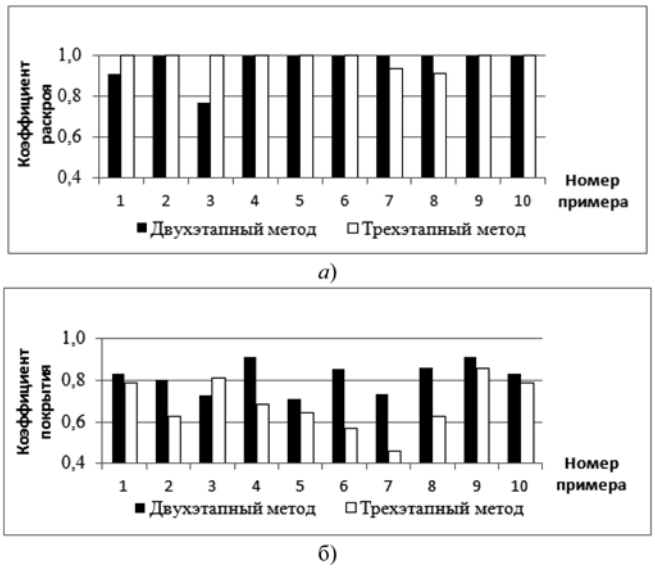


Рис 8. Эксперимент 2: решение безотходных примеров: *а* — коэффициент раскроя; *б* — коэффициент покрытия

Таким образом, проведенный вычислительный эксперимент показал хорошую эффективность исследуемых подходов, а предпочтительный выбор метода решения в каждом конкретном практическом случае зависит от приоритетов заказчика.

Заключение

В целом, можно заключить, что предложенный метод ограниченной декомпозиции области покрытия на основе двухэтапного подхода целесообразно применять, когда важно получить высокий показатель коэффициента покрытия, в то время как метод на базе трехэтапного подхода дает результаты, лучшие по коэффициенту раскроя, т. е. более эффективен с точки зрения экономии ресурса. При этом для предложенного в статье метода ограниченной декомпозиции области покрытия возможны пути повышения эффективности за счет увеличения качества и числа вариантов объединения пустых ячеек в прямоугольные области покрытия. Для этого целесообразно использовать оценки частичных решений, например, на основе и аналогии подетальных норм расхода материала [8], которые были разработаны Л. В. Канторовичем и В. А. Залгаллером для задачи раскроя.

Таким образом, предложенный в статье метод ограниченной декомпозиции является эффективным инструментом решения NP-трудной комплексной задачи ГПР и имеет потенциал для дальнейшего усовершенствования.

Список литературы

1. **Valiakhetova Yu. I., Filippova A. S., Karamova L. M.** Ufa scientific group by E. A. Mukhacheva: applied operational research problems // Вестник УГАТУ. Научный журнал. Уфа: Уфимск. гос. авиац. техн. ун-т, 2013. Т. 17, № 6 (59). С. 83—87.
2. **Телицкий С. В., Филиппова А. С.** Комплексный подход к решению задачи покрытия области заголовками неопределенных размеров // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2012. Т. 2, № 145. С. 61—67.
3. **Телицкий С. В., Валиахметова Ю. И., Хасанова Э. И.** Гибридный алгоритм на основе последовательного уточнения оценок для задач максимального ортогонального покрытия // Вестник Башкирского университета. 2012. Т. 17, № 1 (I). С. 421—425.
4. **Garey M. R., Johnson D. S.** Computers and Intractability: A Guide to the Theory of NP-Completeness. San-Francisco: Freeman, 1979. [Русский перевод: Гэри М., Джонсон Д. Вычислительные машины и трудноразрешимые задачи. М.: Мир, 1982. 416 с.].
5. **Канторович Л. В., Залгаллер В. А.** Расчет рационального раскроя промышленных материалов. Л.: Лениздат, 1951. 198 с.
6. **Мухачева Э. А., Мухачева А. С., Валеева А. Ф., Картак В. М.** Методы локального поиска в задачах ортогонального раскроя и упаковки: аналитический обзор и перспективы развития // Информационные технологии. 2004. № 5. Приложение. С. 2—17.
7. **Filippova A. S., Valiakhmetova J. I.** Optimal use of resources: cutting-packing problems // Advances in Economics and Optimization: Collected Scientific Studies Dedicated to the Memory of L. V. Kantorovich. NY: Nova Science Publishers, 2014. P. 35—48.

8. **Канторович Л. В., Залгаллер В. А.** Рациональный раскрой промышленных материалов. СПб.: Невский Диалект, 2012. 304 с.
9. **URL:** http://paginas.fe.up.pt/~esicup/tiki-index.php (дата обращения: 30.09.15).
10. **Филиппова А. С., Кузнецов В. Ю.** Узадачи о минимальном покрытии ортогональных многоугольников с запретными участками // Информационные технологии. 2008. №9. С. 60—65.
11. **Zak E.** A counterpart of one-dimensional stock cutting: skiving stock problem // Proceedings of the sixteenth triennial conference of the International Federation of Operational Research Societies. 2002. P. 121.
12. **Забелин С. Л., Жеголко К. В., Фроловский В. Д.** Разработка и исследование генетического алгоритма для автоматизации проектных процедур оптимизации геометрического покрытия // Вестник Тамбовского государственного технического университета. Тамбов: изд-во ТГТУ, 2015. Т. 21, № 2. С. 257—265.
13. **Кривуля А. В., Злотник М. В., Романова Т. Е.** Средства математического моделирования в задачах прямоугольного покрытия произвольных многоугольных областей // Радиоэлектроника и информатика, 2007. №4. С. 34—40.
14. **Стоян Ю. Г., Пацук В. Н.** Покрытие многоугольной области минимальным количеством одинаковых кругов заданного радиуса // Доп. НАН України. 2006. № 3. С. 74—77.
15. **Bortfeldt A.** A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces // European Journal of Operational Research. 2006. Vol. 172 (3). P. 814—837.
16. **Мухачева А. С., Чиглинцев А. В., Смагин М. А., Мухачева Э. А.** Задачи двумерной упаковки: развитие генетических алгоритмов на базе смешанных процедур локального поиска оптимального решения // Информационные технологии. 2001. № 9. Приложение. 24 с.
17. **Kenmochi M., Imamichi T, Nonobe K., Yagiura M., Nagamochi H.** Extract algorithms for the two-dimensional strip packing problem with and without rotations // European Journal of Operational Research. 2009. Vol. 198 (1). P. 73—83.
18. **Lesh N., Mitzenmacher M.** Bubble Search: A simple heuristic for improving priority-based greedy algorithms // Information Processing Letters. 2006. Vol. 97 (4). P. 161—169.
19. **Мухачева Э. А., Валиахметова Ю. И., Телицкий С. В., Хасанова Э. И.** Проектирование размещения ортогональных объектов на полигонах с препятствиями // Информационные технологии. 2010. № 10. С. 16—22.
20. **Филиппова А. С., Телицкий С. В., Поречный С. С.** Оптимизация комплексного процесса геометрического покрытия и раскроя // Saarbracken, Germany: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2013. 175 p.
21. **Телицкий С. В., Валиахметова Ю. И., Хасанова Э. И.** Гибридный алгоритм на основе последовательного уточнения оценок для задач максимального ортогонального покрытия // Вестник Башкирского университета. 2012. Т. 17, № 1 (I). С. 421—425.
22. **Мухачева Э. А., Филиппова А. С., Чиглинцев А. В.** Задача прямоугольной упаковки в полубесконечную полосу: численный эксперимент с безотходными задачами E. Hopper на базе алгоритмов блочной структуры // Информационные технологии. 2005. № 7. С. 23—32.

A. S. Filippova, Professor. e-mail: annamuh@mail.ru,
Bashkir State Pedagogical University named after M. Akmulla,
E. I. Dyaminiva, Assistant Professor, e-mail: xasel@mail.ru,
Yu. I. Valiahmetova, Assistant Professor, e-mail: julikcas@inbox.ru,
Ufa State Aviation Technical University

The Limited Decomposition Method for Solving a Complex Problem of Geometric Coverage and Cutting

There is considered some NP-hard complex problem of geometric coverage and cutting. The solutions for it are as follows: a plan of covering the given polygonal domain with rectangular items of maximal size; a plan of cutting maximal number of sheets into covering items. Cutting and covering coefficients are the two criteria to evaluate its efficiency. The method of limited decomposition of the coverage area is offered, it uses matrix representation of the initial data and is based on the two-stage process, namely: solving a subproblem of decomposing of the polygonal domain into rectangular ones, the sizes of which do not exceed the sheet size; solving a corresponding subproblem of rectangular sheet cutting. The computing experiment aimed at researching and comparing of the given method efficiency and that of the hybrid evolutional algorithm, developed earlier, has been carried out. Test problem solving has shown that both methods give high cutting coefficient but the suggested method of limited decomposing brings better cutting coefficient in 90 % of cases.

Keywords: *geometric coverage, cutting, multivariablc orthogonal polygon, matrix decomposition method, limited decomposition method*

References

1. **Valiakhetova Yu. I., Filippova A. S., Karamova L. M.** Ufa scientific group by E. A. Mukhacheva: applied operational research problems, *Vestnik UGATU. Nauchnyj zhurnal,* 2013, vol. 17, no. 6 (59), pp. 83—87.
2. **Telitskij S. V., Filippova A. S.** Kompleksnyj podhod k resheniju zadachi pokrytija oblasti zagototsvkami neopredelennyh razmerov (Comprehensive approach to solving the problem of covering an area with items of indeterminate sizes), *Nauchno-tehnicheskie vedomosti SPbGPU. Informatika. Telekommunikacii. Upravlenie*, 2012, vol. 2, no. 145, pp. 61—67.
3. **Telickij S. V., Valiahmetov'a Yu. I., Hasanova E. I.** Gibridnyj algoritm na osnove posledovatel'nogo utochnenija ocenok dlja zadach maksimal'nogo ortogonal'nogo pokrytija (The hybrid algorithm on the basis of sequential value correction for solving problems of maximal orthogonal covering), *Vestnik Bashkirskogo universiteta*, 2012, vol. 17, no. 1 (I), pp. 421—425.
4. **Garey M. R., Johnson D. S.** Computers and Intractability: A Guide to the Theory of NP-Completeness / San-Francisco: Freeman, 1979. [Russian translation: Gjeri, M. *Vychislitel'nye mashiny i trudnorazreshimye zadachi.* Moscow: Mir, 1982. 416 p.].
5. **Kantorovich L. V., Zalgaller V. A.** *Raschet racional'nogo raskroja promyshlennyh materialov* (Industrial materials rational cutting calculation). L.: Lenizdat, 1951. 198 p.
6. **Muhacheva E. A., Muhacheva A. S., Valeeva A. F., Kartak V. M.** Metody lokal'nogo poiska v zadachah ortogonal'nogo raskroja i upakovki: analiticheskij obzor i perspektivy razvitija (The local search methods in orthogonal cutting-packing problems: analytical survey and development prospects), *Informacionnye tehnologii*, 2004, no. 5, Prilozhenie, pp. 2—17.

7. **Filippova A. S., Valiakhmetova J. I.** *Optimal use of resources: cutting-packing problems*. Advances in Economics and Optimization: Collected Scientific Studies Dedicated to the Memory of L. V. Kantorovich. — NY: Nova Science Publishers, 2014, pp. 35—48.

8. **Kantorovich L. V., Zalgaller V. A.** *Racional'nyj raskroj promyshlennyh materialov* (Industrial materials rational cutting). SPb.: Nevskij Dialekt, 2012. 304 p.

9. **URL:** http://paginas.fe.up.pt/~esicup/tiki-index.php (access date: 30.09.15).

10. **Filippova A. S., Kuznetsov V. Yu.** Zadachi o minimal'nom pokrytii ortogonal'nyh mnogougol'nikov s zapretnymi uchastkami (Problems concerning minimal covering of orthogonal polygons containing prohibited zones), *Informacionnye tehnologii*, 2008, no. 9, pp. 60—65.

11. **Zak E.** A counterpart of one-dimensional stock cutting: skiving stock problem, *Proceedings of the sixteenth triennial conference of the International Federation of Operational Research Societies*, 2002, p. 121.

12. **Zavelin S. L., Zhegolko K. V., Frolovskij V. D.** Razrabotka i issledovanie geneticheskogo algoritma dlja avtomatizacii proektnyh procedur optimizacii geometricheskogo pokrytija (Development and research of a genetic algorithm for automation of geometrical covering optimization procedures), *Vestnik Tambovskogo gosudarstvennogo tehnicheskogo universiteta*, 2015, vol. 21, no, 2, pp. 257—265.

13. **Krivulja A. V., Zlotnik M. V., Romanova T. E.** Sredstva matematicheskogo modelirovanija v zadachah prjamougol'nogo pokrytija proizvol'nyh mnogougol'nyh oblastei (Mathematical modelling means in problems of rectangular covering of random polygon areas), *Radiojelektronika i informatika*. 2007, no. 4, pp. 34—40.

14. **Stojan Yu. G., Patsuk V. N.** Pokrytie mnogougol'noi oblasti minimal'nym kolichestvom odinakovyh krugov zadannogo radiusa (Polygon area covering with a minimal number of similar circles of given radius), *Dop. NAN Ukraini*, 2006, no. 3, pp. 74—77.

15. **Bortfeldt A.** A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *European Journal of Operational Research*, 2006, vol. 172 (3), pp. 814—837.

16. **Muhacheva A. S., Chiglintsev A. V., Smagin M. A., Muhacheva E. A.** Zadachi dvumernoj upakovki: razvitie geneticheskih algoritmov na baze smeshannyh procedur lokal'nogo poiska optimal'nogo reshenija (Two-dimensional packing problems: genetic algorithms development based on mixed procedures of optimal solving local search), *Informacionnye tehnologii*, 2001, no. 9, Prilozhenie, 28 p.

17. **Kenmochi M., Imamichi T., Nonobe K., Yagiura M., Nagamochi H.** Extract algorithms for the two-dimensional strip packing problem with and without rotations, *European Journal of Operational Research*, 2009, vol. 198 (1), pp. 73—83.

18. **Lesh N., Mitzenmacher M.** Bubble Search: A simple heuristic for improving priority-based greedy algorithms, *Information Processing Letters*, 2006, vol. 97 (4), pp. 161—169.

19. **Muhacheva E. A., Valiahmetova Yu. I., Telitskij S. V., Hasanova E. I.** Proektirovanie razmeshhenija ortogonal'nyh ob''ektov na poligonah s prepjatstvijami (Designing of orthogonal objects location on polygons with some obstacles), *Informacionnye tehnologii*, 2010, no. 10, pp. 16—22.

20. **Filippova A. S., Telitskij S. V., Porechnyj S. S.** *Optimizacija komgleksnogo processa geometricheskogo pokrytija i raskroja* (Optimization of the geometrical cutting-packing complex process) / Saarbrucken, Germany: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2013. 175 p.

21. **Telitskij S. V., Valiahmetova Yu. I., Hasanova E. I.** Gibridnyj algoritm na osnove posledovatel'nogo utochnenija ocenok dlja zadach maksimal'nogo ortogonal'nogo pokrytija, *Vestnik Bashkirskogo universiteta*, 2012, vol. 17, no. 1 (I), pp. 421—425.

22. **Muhacheva E. A., Filippova A. S., Chiglintsev A. V.** Zadacha prjamougol'noj upakovki v polubeskonechnuju polosu: chislennyj jeksperiment s bezothodnymi zadachami E. Hopper na baze algoritmov blochnoj struktury (The problem of rectangular packing into a semi-endless strip: numerical experiment with the wasteless problems), *Informacionnye tehnologii*, 2005, no. 7, pp. 23—32.

УДК 519.857

В. И. Струченков, д-р техн. наук, проф., e-mail: str1942@mail.ru
Московский государственный университет
информационных технологий, радиотехники и электроники (МИРЭА)

Динамическое программирование с использованием множеств Парето в задачах планирования реализации возобновляемых ресурсов

Настоящая статья является продолжением статьи, опубликованной в № 2 за 2016[1]. В ней рассматривается задача составления оптимального поэтапного плана использования возобновляемого однородного ресурса в течение заданного времени. В данной статье предложен эффективный алгоритм динамического программирования с использованием множеств Парето.

Ключевые слова: *ресурс, целевая функция, множество состояний, динамическое программирование, множества Парето*

Введение

В статье [1] рассмотрены две математические модели задачи планирования реализации возобновляемого ресурса. В рамках этих моделей применительно к использованию метода динамического программирования получены рекуррентные соотношения, позволяющие строить оптимальный план без анализа пошаговых вариантов. В вычислительном плане это во много раз эффективнее, чем классический алгоритм динамического программирования, так как практически поиск решения сводится к вычислению трех величин на каждом шаге процесса по простым формулам. Однако эти модели не претендуют на универсальность, так

[1] Струченков В. И. Динамическое программирование в задачах планирования реализации частично возобновляемых ресурсов // Информационные технологии. 2016. № 2.

как возможна различная постановка рассматриваемой задачи. В общем случае модели, рассмотренные в [1], могут потребовать уточнений, которые не позволят получить рекуррентные формулы для построения оптималького плана. Поэтому рассматриваемая задача продолжает оставаться актуальной.

Цель настоящей статьи состоит в анализе задачи построения оптимального плана реализации возобновляемого ресурса в общем случае и разработке алгоритма ее решения по методу динамического программирования с использованием множеств Парето для повышения вычислительной эффективности алгоритмов [2, 3].

Постановка задачи

Задача составления поэтапного плана реализации возобновляемого ресурса состоит в следующем [1, 4]. Заданы:

- период планирования T (например, $T = 10$ лет);
- начальное количество возобновляемого ресурса b_1;
- коэффициент воспроизводства p;
- функция годового дохода $d(x)$ при реализации x единиц ресурса;
- функция сопутствующих затрат в течение года $c(x, b)$, где b — количество ресурса на начало года.

Требуется построить такой план реализации ресурса, чтобы в течение T лет суммарный количественный показатель качества плана (целевая функция) принял максимальное значение.

Будем считать, что годы пронумерованы, начиная с единицы, так что последний год имеет номер T.

Если в начале любого года t ($t = 1, 2, ..., T$) имеется b_t единиц ресурса и в течение этого года реализуется x_t его единиц, то на начало следующего года будет $b_{t+1} = p(b_t - x_t)$ единиц ресурса. Коэффициент воспроизводства p не изменяется в течение рассматриваемого периода планирования.

Если целевая функция — это суммарная прибыль, то задача формализуется в следующем виде: найти вектор $\mathbf{x}(x_1, x_2, ..., x_T)$, при котором достигает максимума

$$\sum_{t=1}^{T} (d(x_t) - c(x_t, b_t)) \text{ и } 0 \leqslant x_t \leqslant b_t;\ b_{t+1} = p(b_t - x_t).$$

Выражение $d(x_t) - c(x_t, b_t)$ только условно можно назвать годовой прибылью, так как реально могут быть еще и сопутствующие затраты, не зависящие от количества реализуемого ресурса. Теоретически наличие в целевой функции постоянного слагаемого несущественно, так как оно не влияет на точку экстремума, т. е. на искомое решение. При анализе математических моделей и построении рекуррентных соотношений для расчета оптимального плана в работе [1] наличие постоянной составляющей затрат на реализацию ресурса не учитывалось. Однако формальный оптимум может соответствовать решению, при котором в каком-либо году $d(x_t) - c(x_t, b_t))$ меньше постоянной составляющей годовых затрат на реализацию ресурса, поэтому годовая прибыль отрицательна, хотя суммарно за все годы прибыль максимальна. Если отрицательная годовая прибыль недопустима, то на каждом этапе появляется дополнительное ограничение на минимальную прибыль и тем самым на минимальное количество реализации ресурса. Это ограничение усложняет алгоритм поиска.

Классическая схема динамического программирования

Для решения задачи применительно к построению оптимального плана вылова форели в работе [4] предлагается классическая схема динамического программирования. Ключевое для метода динамического программирования понятие "состояние системы" формализуется как количество имеющегося ресурса. Соответственно "траектория" (или "путь") — это последовательность состояний, т. е. значений имеющегося ресурса b_t в начале каждого года с номером t ($t = 1, 2, ..., T$). В классическом алгоритме динамического программирования [5, 6] процесс рассматривается от "конца к началу", т. е. начиная с множества состояний $b_T = b_1 p^{T-1} + 1$ в начале последнего года. При реальных b_1, p и T значение b_T может быть велико, поэтому усовершенствование классического алгоритма динамического программирования актуально с точки зрения объема вычислений.

Отметим, что в данной задаче при ее численном решении удобно и более эффективно двигаться от начала к концу. Конкретный вид функций $d(x)$ — доход и $c(x, b)$ — затраты существенного значения не имеет, если есть возможность их вычисления при соответствующих дискретных значениях x_t и b_t.

Далее будем рассматривать задачу в общем виде, двигаясь от начального состояния b_1. Дополнительное ограничение на минимальную годовую прибыль означает, что к последнему году многие из состояний последнего года, о которых речь шла выше (от 0 до $b_1 p^{T-1} + 1$), оказываются недостижимыми, но заранее вычислить, какие именно состояния недостижимы, сложно. Дело в том, что сопутствующие затраты могут зависеть не только от количества ресурса x_t, реализуемого в году t, но и от количества имеющегося в начале года ресурса b_t. Действительно, например, применительно к разведению форели выловить 100 форелей легче, если их много (скажем 10 000), чем если их всего 200 и должно остаться не менее 100.

Однако это не единственное преимущество выбора направления построения оптимального пути от начального состояния в конечные, которых окажется много меньше, чем $b_1 p^{T-1} + 1$.

Алгоритм при движении от начала к концу состоит в следующем.

1. Начиная с первого года, имея b_1 единиц ресурса, рассматриваем все возможные количества его реализации $0 \leqslant x_1 \leqslant b_1$. Вместо нуля может быть заданное минимальное количество реализуемого ресурса $x_{\min}$, что сокращает объем вычислений. Для каждого из возможных значений x_1 вычисляем и запоминаем соответствующие значения целевой функции и состояние с $b_2 = (b_1 - x_1)p$. Переходы с недопустимо малой прибылью отбраковываются.

Тем самым завершается формирование множества состояний после первого года (этапа).

2. Аналогично формируем состояния каждого из последующих этапов, вычисляя суммарные значения целевой функции и оставшиеся количества ресурса. Во избежание полного перебора при достижении одного состояния разными путями в соответствии с принципом оптимальности Р. Беллмана [5, 6] оставляем только путь, по которому это состояние достигается с бóльшим значением целевой функции, и запоминаем соответствующее ему состояние предыдущего этапа (связь).

3. Завершив последний этап, получаем оптимальное значение целевой функции и обратным разворотом по цепочке связей, начиная с оптимального конечного состояния $(b_T + 1)$, восстанавливаем оптимальную последовательность состояний и соответствующие величины реализации ресурса на каждом этапе, т. е. в каждом году.

При наличии дополнительных требований, например, не оставлять ресурса меньше заданного числа, легко скорректировать построение множества состояний на каждом этапе.

Использование множеств Парето

В данной задаче возможно существенное сокращение объема вычислений при использовании более совершенной реализации этого классического алгоритма динамического программирования за счет использования множеств Парето [2, 3].

Предположим, что зависимости $d(x)$ и $c(x, b)$ заданы более сложными формулами, чем линейная и квадратичная модели, рассмотренные ранее [1], таблично или еще каким-то образом. Важно, что мы можем вычислить прибыль за год при известных x и b, т. е. для любого заданного состояния (b) и перехода к новому состоянию. Зададимся вопросом, может ли при различных планах реализации ресурса через несколько лет от начала планирования получиться так, что для одного состояния ресурса осталось больше, чем для второго, и к тому же суммарная прибыль получена больше, чем для второго? Если это так, то очевидно, что второе состояние бесперспективно, и его вообще не надо рассматривать при переходе к следующему этапу, так как наличие лишнего ресурса никак не может помешать принятию тех же решений при дальнейших переходах из первого состояния, что и при дальнейшем продвижении из второго состояния. Получается, что второе состояние "отстало навсегда".

Речь идет о наличии доминируемых (непаретовских) точек в множестве пар чисел (координат точек на плоскости), из которых первое число — это количество оставшегося ресурса, а второе число — это полученная за все предыдущие годы суммарная прибыль [2, 3]. Множество недоминируемых точек называется множеством Парето [2]. Именно эти точки и соответствующие им состояния имеет смысл оставлять на каждом этапе для дальнейшего рассмотрения, исключая все доминируемые точки (состояния) как бесперспективные. Характерно, что для различных функций $d(x)$ и $c(x, b)$ доминируемые точки могут появляться уже после первого этапа, когда из начального состояния (b_1) исходят несколько путей и никакой отбраковки по принципу Р. Беллмана путей, приводящих в одну точку, еще нет.

Будем считать, что при заданном b_1 принимаемое на первом шаге решение x_1' доминирует над другим таким решением x_1'', если соответствующая ему точка с координатами $(b_2', z_1(x_1'))$ доминирует над $(b_2'', z_1(x_1''))$, т. е. $b_2' \geqslant b_2''$ и $z_1(x_1') \geqslant z_1(x_1'')$. Здесь b_2' — количество ресурса на начало второго года, $z_1(x_1')$ — соответственно прибыль, полученная при реализации x_1' единиц ресурса. Аналогичный смысл имеют величины, отмеченные двумя штрихами для другого значения x_1 (рис. 1).

Если прибыль $z(x) = d(x) - c(x, b)$ не является монотонно неубывающей функцией x, то точки, соответствующие участкам убывания ($x_1 > x_1^*$ на рис. 1) являются доминируемыми, и соответствующие значения x можно не рассматривать ни на одном из этапов процесса построения оптимального плана.

Однако и для монотонно возрастающей функции $z(x) = d(x) - c(x, b)$ появление доминируемых

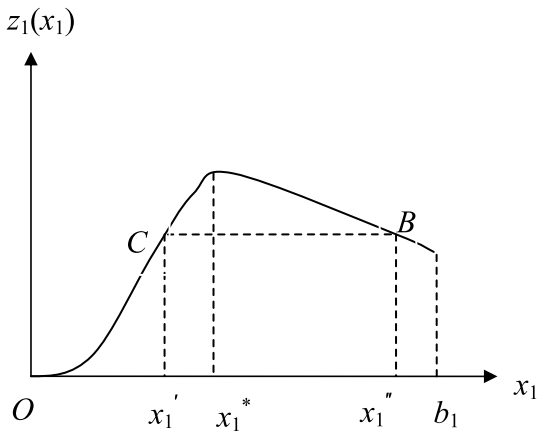


Рис. 1. Доминируемые точки первого шага ($x_1 > x_1^*$)

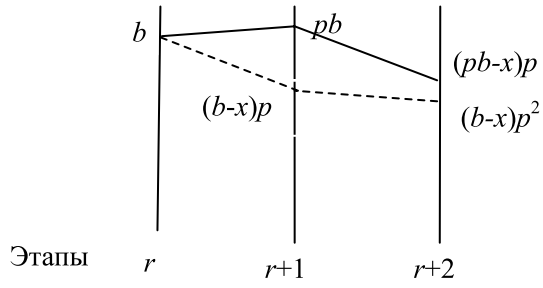


Рис. 2. Пример появления доминируемых состояний

точек (соответственно состояний на различных этапах) также вполне обычно.

В общем случае можно предположить, что $c(x, b)$ является возрастающей функцией x при фиксированном b и убывающей функцией b при фиксированном x. Рассмотрим некоторое состояние b на произвольном этапе r и переходы на два следующих этапа из этого состояния, но по разным вариантам. В каждом варианте на двух переходах реализуется суммарно одно и то же количество ресурса x, но в первом варианте при переходе на $(r + 1)$-й этап нет реализации ресурса (сплошная линия на рис. 2), а во втором варианте нет реализации ресурса при переходе от $(r + 1)$-го к $(r + 2)$-му этапу (штриховая линия на рис. 2).

Очевидно, что $(pb - x)p > (b - x)p^2$ при $p > 1$.

Далее, доход $d(x)$ по этим двум вариантам один и тот же, так как он зависит только от количества реализованного ресурса. Поэтому надо сравнить суммарные за два этапа затраты. Полагая, что при отсутствии реализации ресурса затраты равны нулю, получаем по первому варианту $c_1 = c(x, pb)$, а по второму варианту $c_2 = c(x, b)$, так как по первому варианту реализация осуществляется при наличии ресурса pb, а по второму — при наличии ресурса b. Далее, $pb > b$ и поэтому $c_2 > c_1$, так как функция затрат $c(x, b)$ является убывающей по b при заданном x.

Получается, что состоянию, достигаемому по первому варианту, соответствует бóльший ресурс и бóльшая прибыль, чем состоянию, достигаемому по второму варианту. Другими словами, во втором варианте получаем доминируемое состояние, которое следует исключить из дальнейшего рассмотрения.

Этот вывод остается в силе и при наличии постоянной составляющей затрат, не зависящей от x.

В реальных задачах возможны также дополнительные условия:

- нельзя допускать падение ресурса ниже заданного уровня;
- реализация ресурса планируется количествами, существенно бóльшими, чем единица.

При этих дополнительных условиях появление непаретовских состояний также возможно. Пусть, например, начальное состояние — 1000 единиц ресурса, коэффициент воспроизводства — 1,2, минимальный уровень — 300 единиц ресурса, дискрет использования ресурса — 100 единиц. Первый план состоит в том, что в первый и второй год ресурс не используется, а в третий год из имеющихся к этому году 1440 используются 1000 единиц и 440 остаются. Второй план состоит в том, что во второй год используются 900 единиц (больше нельзя) и далее в конце третьего года остается только 432 единицы ресурса. Это состояние (432) может далее не рассматриваться, так как состояние 440 (по первому плану) ему ни в чем не уступает, если получение прибыли от реализации 1000 единиц ресурса в третьем году предпочтительнее прибыли от реализации 900 единиц ресурса во втором году.

Для реализации динамического программирования с использованием множеств Парето нужно отказаться от ложного стереотипа необходимости разбиения регулярной сетки и решать задачу при движении от начального состояния в конечное. При анализе каждого перехода из каждого состояния нужно вычислить соответствующий ресурс, получаемый в результате такого перехода, и оценку нового состояния (суммарную прибыль от начала до нового состояния). Если в новое состояние приводит несколько путей, то надо оставить лучший из них (как в классической реализации метода Р. Беллмана), иначе нужно проверить, не существует ли таких состояний, которые и по ресурсу и по целевой функции не хуже полученного. Если такие состояния есть, то рассматриваемый переход исключается, если нет, то, наоборот, нужно проверить, не позволяет ли новое состояние исключить из рассмотрения уже имеющиеся. Другими словами, на каждом шаге должны оставаться только такие состояния, которые соответствуют множеству Парето для двухкритериальной задачи: максимум прибыли при максимуме оставшегося ресурса. Такие паретовские множества легко формировать, если на каждом шаге упорядочивать состояния по одному из критериев, например по оставшемуся ресурсу. Фактически, переходя к паретовским множествам, мы используем метод "погружения", т. е. замены исходной однокритериальной задачи более общей двухкритериальной задачей, для решения которой удается найти более эффективный алгоритм, который в итоге приводит и к решению исходной однокритериальной задачи. Действительно, после завершения последнего этапа мы имеем множество Парето, в котором точка с наибольшим значением первого критерия (прибыль) является решением исходной задачи. Однако наличие такого паретовского множества позволяет рассмотреть и субоптимальные решения, если требуется в итоге оставить не менее заданного количества ресурса и субоптимальное решение незначительно уступает оптимальному по целевой функции, но ему соответствует существенно больший остающийся ресурс

Расчеты показали, что такое динамическое программирование с использованием множеств Парето

существенно эффективнее и по памяти и по объему вычислений, чем классическая реализация метода динамического программирования.

Заключение

Для практического использования предложенных алгоритмов необходимо получение реальных данных из практики. Речь идет прежде всего о функциях дохода $d(x)$ и сопутствующих затрат $c(x, b)$, а также коэффициенте воспроизводства p, т. е. о величинах, которые в рассматриваемой постановке задачи являются исходными данными. Расчеты могут быть выполнены при различных значениях количества ресурса на начало периода планирования (b_1) для сопоставления различных планов реализации ресурса. Если функции $d(x)$ и $c(x, b)$ являются аддитивными, то возможность применения динамического программирования для составления оптимальных планов реализации возобновляемых ресурсов сомнений не вызывает. Эффективные в вычислительном плане реализации динамического программирования могут быть получены при использовании множеств Парето на каждом этапе построения оптимального плана.

Заметим, однако, что при выборе в качестве целевой функции экономической эффективности, вычисляемой как отношение суммарной прибыли к суммарным затратам, динамическое программирование применить не удается, так как эта функция не является аддитивной, т. е. не может быть представлена в виде суммы слагаемых, каждое из которых относится к соответствующему этапу. План, оптимальный в смысле экономической эффективности, вообще говоря, не совпадает с планом, дающим максимальную суммарную прибыль. Целесообразность такого плана с меньшей суммарной прибылью не очевидна. В любом случае решение, полученное по методу динамического программирования из условия максимума суммарной прибыли, может служить надежной оценкой в практических расчетах.

Список литературы

1. **Струченков В. И.** Динамическое программирование в задачах планирования реализации частично возобновляемых ресурсов // Информационные технологии. 2016. № 2.
2. **Струченков В. И.** Устаревшие стереотипы и новые алгоритмы решения прикладных задач дискретной оптимизации // Информационные технологии. 2012, № 5. С. 20—29.
3. **Struchenkov V. I.** Combined Algorithms of Optimal Resource Allocation // Applied Mathematics. Scientific Research. 2012. Vol. 3, no. 1.
4. **Косоруков О. А., Мищенко А. В.** Исследование операций: учебник для вузов. М.: Экзамен, 2003, 270 с.
5. **Беллман Р.** Динамическое программирование. М.: ИЛ, 1960.
6. **Беллман Р., Дрейфус С.** Прикладные задачи динамического программирования. М.: Наука, 1965. 458 с.

V. I. Struchenkov, Professor, e-mail: str1942@mail,ru,
Moscow State University of Information Technology, Radio Engineering and Electronics, Moscow, Russia.

Dynamic Programming with Pareto Sets for Planning of the Renewable Resources Implementation

Under study is the problem of optimal planning of the renewable resources implementation, such as the commercial breeding of fish, animals and so on.

The aim of this article is to study the oportunity of the optimal plan calculation using dynamic programming.

The predetermined scheduling period is divided into a number of stages, such as months or years.

At each step of the process of constructing an optimal plan the key concept of the Dynamic Programming "state of the system" is formalized as the number of available resources. Process goes from a given initial state to the final one. In accordance with the principle of optimality R. Bellman in achieving some state by different ways all of them are rejected, except the way that corresponds maximum profit. In addition, the state is rejected if on the same step there is a state with more resources and more profit. As a result, at each step Pareto set is formed. Such algorithm is more efficient than classic algorithm of Dynamic Programming.

Keywords: *resource, the objective function, the set of states, dynamic programming, the optimal path, Pareto sets*

References

1. **Struchenkov V. I.** Dinamicheskoe programmirovanie v zadachah planirovanija realizacii chastichno vozoblovljaenyh resursov, *Informacionnye tehnologii*, 2016, no 2, pp. 94—99 (in Russian).
2. **Struchenkov V. I.** Ustarevshie stereotipy i novye algoritmy reshenija prikladnyh zadach diskretnoi optimizacii, *Informacionnye tehnologii*. 2012, no. 5, pp. 20—29 (in Russian).
3. **Struchenkov V. I.** Combined Algorithms of Optimal Resource Allocation, *Applied Mathematics. Scientific Research*, 2012, vol. 3, no. 1.
4. **Kosorukov O. A., Mishenko A. V.** *Issledovanie operacij. Uchebnik dlja studentov vuzov*, Moscow: Examen, 2003, 270 p. (in Russian).
5. **Bellman R.** *Dinamicheskoe programmirovanie* (Dynamic Programming), Moscow: Foreign Literature, 1960, 234 p. (in Russian).
6. **Bellman R., Drejfus S.** *Prikladnye zadachi dinamicheskogo programmirovanija* (Applied Problems of Dynamic Programming), Moscow: Science, 1965, 458 p. in Russian).

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ
CAD-SYSTEMS

УДК 658.62.018.012

Р. Р. Загидуллин, д-р техн. наук, проф., e-mail: polysoft@list.ru,
Уфимский государственный авиационный технический университет

Автоматизация разработки альтернативных технологических процессов

Рассмотрены вопросы использования и автоматизации разработки альтернативных технологических процессов (АТП) в системах классов САПР ТР/АСТПП с помощью встроенных библиотек программирования скриптов. Предложенный метод формализации АТП как симбиоз традиционной линейной технологической структуры операционной технологии с операторами управления при использовании в системах классов САПР ТП/АСТПП встроенных средств программирования скриптов (сценариев) позволяет создавать в одной записи базового технологического процесса (ТП) несколько различных АТП, что в дальнейшем позволяет автоматизировать процесс активации того или иного АТП в качестве текущего в зависимости от конкретных условий и использовать его в любых системах планирования классов MES (Manufacturing Execution System), APS (Advanced Planning System).

Ключевые слова: *технологический процесс, альтернативный технологический процесс, скрипт, сценарий, САПР ТП, АСТПП, планирование, MES, APS*

Введение

Повышение производительности оборудования является одной из ключевых задач отечественного машиностроения. Производительность станочного парка напрямую связана с плотностью составленных расписаний. Чем меньше простоев, тем выше производительность и фондоотдача. Составление расписаний на сегодняшний день является прерогативой таких информационных систем, как APS, формирующих общий план выпуска для всего предприятия, и MES, отвечающих за построение и пересчет цеховых расписаний при их выполнении. Одним из вариантов решения этой задачи является использование определенным образом структурированных альтернативных технологических процессов (АТП) с последующей активацией необходимого варианта в зависимости от сложившейся производственной ситуации.

Постановка задачи

Наибольший эффект, выраженный пропускной способностью станочной системы (участок, цех, предприятие) как на стадии построения расписания APS, так и на стадии их пересчета MES, достигается в том случае, если каждая деталь, по возможности, имеет более одного технологического процесса, т. е. речь идет о наличии АТП. В этом случае, несмотря на усложнение задачи построения расписания с точки зрения комбинаторики и вычислительной сложности, повышается ее степень свободы и гибкость планирования [1, 2], появляется возможность повысить как производительность отдельных станков, так и увеличить пропускную способность в пределах всей станочной системы. Рассмотрим влияние наличия АТП на следующем примере.

Допустим, что некая деталь представлена двумя АТП, которые различаются числом операций. При этом положим, что первый вариант технологического процесса (ТП), построенный по принципу концентрации операций, имеет на одну операцию меньше, чем второй вариант ТП, т. е. мы имеем два множества ТП — $\text{ТП}_1\{e_1^1, e_2^1, e_3^1, e_4^1\}$ и $\text{ТП}_2\{e_1^2, e_2^2, e_3^2, e_4^2, e_5^2\}$, для которых справедливы выражения $e_i^1 \equiv e_i^2$, $i\{1, 2, 3\}$ и $e_4^1 = e_4^2 \vee e_5^2$, т. е. последняя операция ТП_1 e_4^1 может быть представлена двумя операциями ТП_2 e_4^2 и e_5^2 без потери точности при смене установочных баз при обработке.

В процессе выполнения цеховой программы может возникнуть случай, когда оборудование, через которое проходит ТП_1, находится в состоянии отказа (рис. 1).

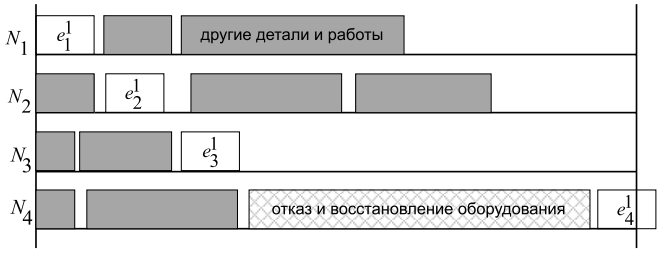


Рис. 1. Фрагмент диаграммы Гантта с вариантом пролеживания операции

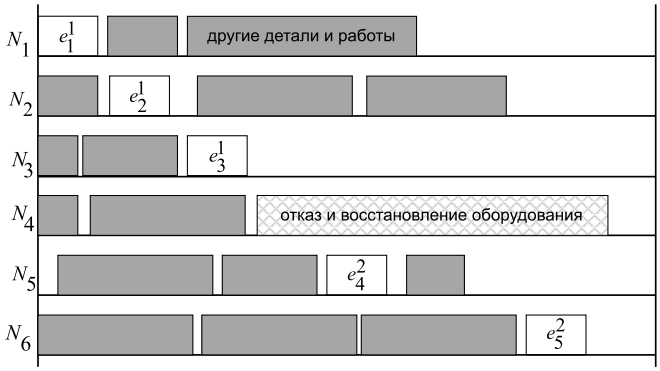


Рис. 2. Фрагмент диаграммы Гантта с вариантом использования АТП

При этом мы видим, что ожидание момента восстановления оборудования N_4 для $ТП_1$ приводит к пролеживанию детали на предстоящей операции e_4^1 и ее запаздыванию (показан случай ее выхода за горизонт планирования).

В то же время возможен вариант, который часто используется на практике (при наличии АТП), когда в подобных случаях при наличии свободного оборудования используется альтернативное продолжение ТП, что представлено на рис. 2.

Решение о переходе на АТП в этом случае должна принимать MES-система, которая на основе анализа текущей ситуации обязана пересчитать расписание. На практике, когда MES-система не используется, решение о переходе на АТП принимает либо мастер, либо специалист планово-диспетчерского бюро. Такой подход, как мы видим из диаграммы (см. рис. 2), часто позволяет выйти из положения, более плотно загрузить имеющееся оборудование и выпустить детали в указанные сроки.

Необходимость создания АТП оговорена в основах технологии машиностроения [3, 4]. Указывается, что при разработке ТП необходимо иметь несколько альтернативных вариантов технологических процессов с последующим расчетом их себестоимости и выбором наиболее экономичного варианта. Основной проблемой при таком классическом подходе является отсутствие связи между задачей разработки АТП и их последующим выполнением, т. е. отсутствует связь с расписанием работы оборудования.

Вопросам разработки АТП с точки зрения декомпозиции и последующего синтеза операций, вариантного проектирование ТП на множестве сетевых и графовых структур, а также с использованием других подходов посвящено большое число работ как отечественных, так и зарубежных авторов [5—8]. Во всех случаях авторы, так или иначе, придерживались точки зрения, аналогичной классике технологии машиностроения [2, 3], т. е. на множестве АТП определялся ТП, оптимальный либо по частному критерию, либо по векторному, что можно отнести к проблеме адекватности оценки всего множества ТП на предприятии при выполнении во времени, с точки зрения эффективности функционирования станочной системы в целом, а не отдельно взятого ТП.

Второй проблемой является процесс формализации АТП и его отражение в системах АСТПП/САПР ТП.

В большинстве случаев разработчики систем проектирования технологической документации классов АСТПП/САПР ТП вопросы формализации АТП избегают ввиду сложности. В таких случаях обычно рекомендуется делать несколько копий ТП, отличающихся друг от друга в той или иной мере. В дальнейшем, в процессе производства, при переходе на тот или иной АТП распечатывается его документация, и процесс производства продолжается уже по документации АТП. Мало того, что это вносит существенную путаницу при расчете окончательной себестоимости, основная проблема состоит в том, что ни одна система планирования классов MES (*Manufacturing Execution System*) или APS (*Advanced Planning System*) не рассчитана на то, чтобы оперировать множеством АТП и выбирать тот, который актуален на текущий момент времени [9], поскольку потребность в АТП возникает до начала момента планирования, в процессе производства.

Вследствие указанных выше причин возникает задача создания методики, позволяющей в существующих на сегодняшний день системах АСТПП, САПР ТП создавать АТП в автоматизированном режиме с возможностью их использования в дальнейшем в MES-системах.

Предложения по автоматизации создания АТП

Любой АТП в любых системах САПР ТП, АСТПП всегда описывается линейно (рис. 3, *а*), как и основной ТП. Такой вариант возможен, но он требует того, чтобы некая система планирования "понимала", что есть вторая запись в базе данных (БД) ТП, представляющая собой АТП. Но здесь воз-

никает вопрос, с какого места, с какой точки продолжения этот АТП должен воспроизводиться, так как АТП, как было отмечено выше, создается, как и основной ТП, с первой операции до последней.

В большинстве случаев АТП представляет собой часть ТП, которая выполняется на другом оборудовании или при неких других условиях. При этом число операций в этой части может отличаться от аналогичной части основного ТП, так как основной ТП, например, может создаваться, с учетом концентрации операций, для оборудования с ЧПУ, а АТП — как для оборудования с ЧПУ, так и для универсального оборудования, где непроизвольно возникает дифференциация операций и их увеличение, а также изменение режимов обработки. При этом получается ветвящаяся структура (рис. 3, *б*).

Поэтому возникает необходимость эту ветвящуюся структуру сделать параллельно штатными средствами той или иной системы САПР ТП/АСТПП в одном процессе разработки ТП.

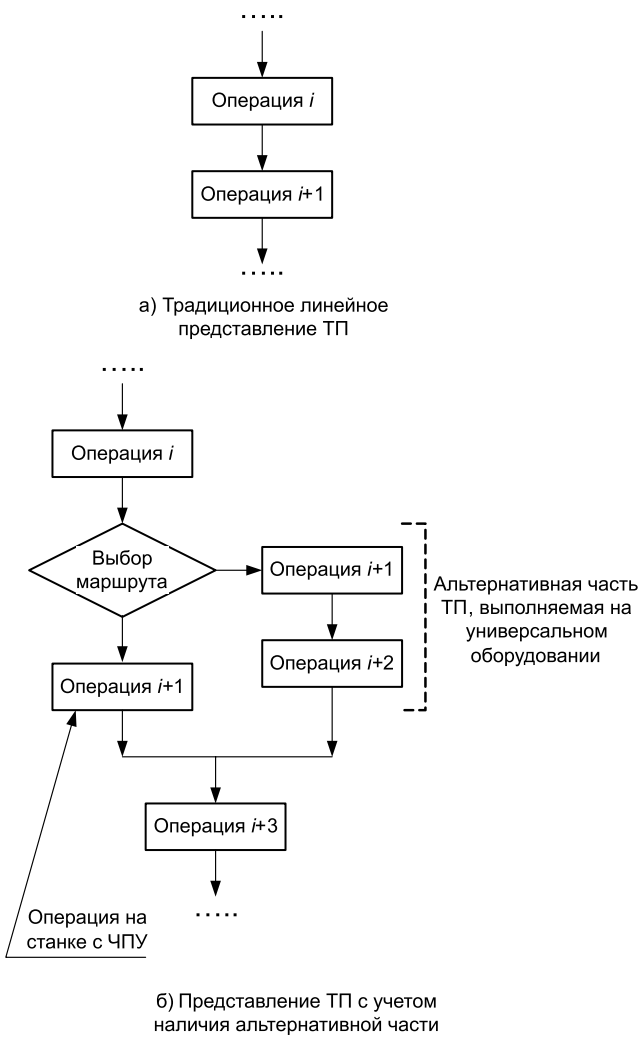


Рис. 3. Варианты учета альтернативных технологических процессов

Такая возможность появляется в том случае, если использовать стандартные методы программирования процессов, поскольку многие системы САПР ТП/АСТПП содержат в себе языковые средства, с помощью которых создаются скрипты (сценарии). В качестве таких встроенных языковых средств часто используется библиотека FastScript [10—12], которая написана полностью на 100 % на Object Pascal. Таким образом, можно программным методом получить параллельный вариант из последовательного (рис. 4) в необходимый момент времени (см. рис. 1, 2).

Это достаточно легко сделать, если ввести маркеры блоков в структуре ТП, как это делается при написании программы с условными переходами, поскольку практически любая система САПР ТП/АСТПП позволяет создавать в структуре ТП различные атрибуты — не только операции, но и другие записи. А поскольку ТП находится в БД, то маркер, по сути, представляет собой запись в БД, которую потом можно прочитать штатными средствами FastScript.

Тогда вид ТП, включающий часть АТП, будет выглядеть так, как это представлено на рис. 5. В качестве маркеров блоков можно использовать тради-

Рис. 4. Принципиальный вариант использования альтернативного ТП в виде отдельного блока при линейном представлении ТП

ционные для программирования операторы "begin" и "end". Таким образом, в одном базовом ТП можно разместить сколько угодно альтернативных частей. Единственное дополнение — это то, что должен присутствовать оператор выбора типа "If", который соотносил бы реальную обстановку (занятость оборудования или его поломку) с соответствующим блоком в зависимости от состояния некоего флага (возможность исполнения ТП на данном этапе или — невозможность). В рассматриваемом примере таких условных переходов, в силу специфики (количество оборудования), много быть не может, не больше одного, как правило.

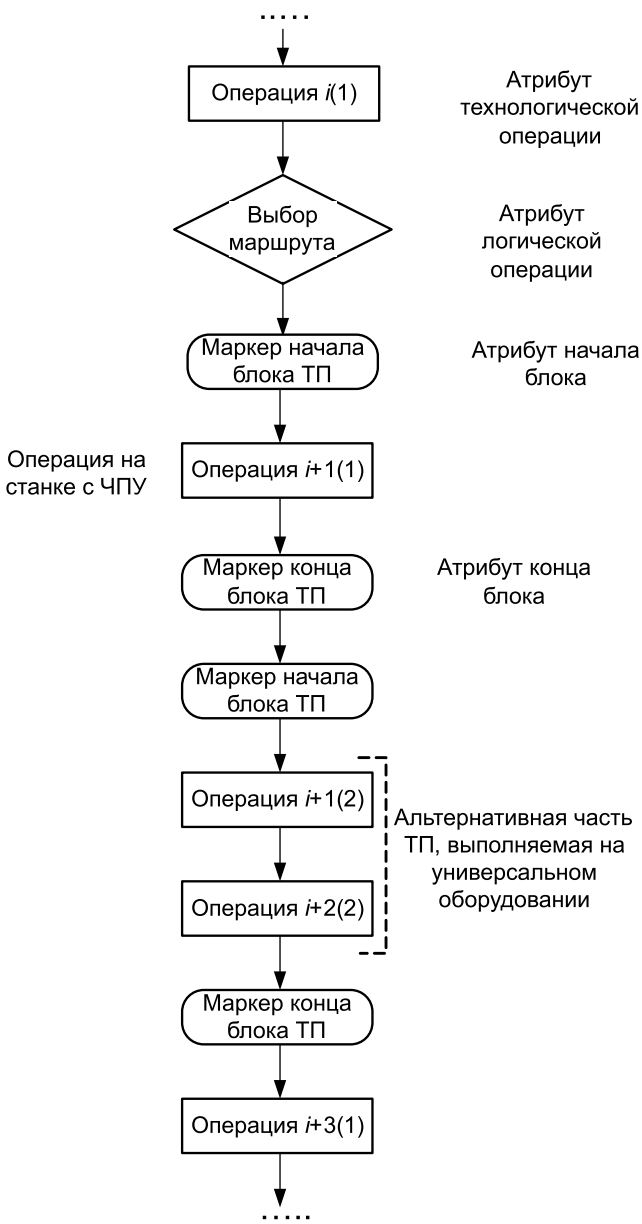


Рис. 5. Предлагаемый программный вариант использования АТП в виде отдельного блока при линейном представлении ТП

На рис. 6 приведен пример тестового ТП, включающего в себя альтернативную часть. Задача тестового примера следующая:

1) необходимо создать в некой САПТ ТП технологический процесс, включающий альтернативную часть и флаги для условных переходов (см. рис. 6);

2) после этого, в зависимости от состояния флагов, распечатать штатными средствами Fast-

```
// Техпроцесс  Example  1

// Флаг  CNC  {true, false }
CNC :=                          Вводить  true или  false

05 Заготовительная

10 Токарная

15 Слесарная

If CNC  = false then
{Возможный пропуск ближайшего
 блока , содержащего  Begin End }

Begin
20 Операция ЧПУ                 Пропускается
                                операция ЧПУ , если  false
End

If CNC  = true then
{Возможный пропуск ближайшего
 блока , содержащего  Begin End }

Begin
25 Операция Универсал  1        Пропускается АТП,
30 Операция Универсал  2        если  true
End

35 Шлифование

40 Промывка
```

Рис. 6. Схема реализации включения/исключения АТП

Таблица 1

Общий вид ТП, включающий часть АТП

№ операции	Основной ТП	Альтернативная часть ТП
05	Заготовительная	
10	Токарная	
15	Слесарная	
20	Операция ЧПУ	
25		Операция Универсал 1
30		Операция Универсал 2
35	Шлифование	
40	Промывка	

Таблица 2

Итоговый ТП без АТП

№ операции	Основной ТП
05	Заготовительная
10	Токарная
15	Слесарная
20	Операция ЧПУ
25	
30	
35	Шлифование
40	Промывка

Таблица 3

Итоговый ТП, использующий АТП

№ операции	Основной ТП
05	Заготовительная
10	Токарная
15	Слесарная
20	
25	Операция Универсал 1
30	Операция Универсал 2
35	Шлифование
40	Промывка

Script (поместить в любой документ или файл) три документа:

2.1) общий вариант ТП, включающий АТП (табл. 1);

2.2) два варианта ТП, в зависимости от состояния флага условного перехода (табл. 2, 3).

В данном случае использовалась конструкция программирования "If (условие) then (блок выполнения)". Если альтернативных блоков может быть несколько и если требуется реализовывать более сложный сценарий итогового ТП на основе базового, то можно ввести дополнительные атрибуты, такие как "Label <number label>" (метка с номером) и атрибут "Go to < number label >" (переадресация управления по номеру указанной метки). Таким образом, за счет использования дополнительных атрибутов можно получить очень гибкий вариант ТП как симбиоз программы и ТП.

При этом в БД хранится общий ТП (см. табл. 1), который является базовым. По сценарию перед активацией той или иной детали, в зависимости от ситуации (см. рис. 1, 2), можно получить, например, два варианта ТП, один из которых уйдет на планирование в MES или APS. Все действия по активации ТП (созданию рабочего ТП по отношению к базовому), инициатором которой является MES-система, должны проводиться непосредственно в системе САПР ТП/АСТПП.

Таким образом, можно достичь следующего:

- повысить гибкость используемой системы САПР ТП/АСТПП;
- добиться универсальности по отношению к любой системе планирования, при этом не потребуется доработка MES- или APS-системы для учета АТП.

Заключение

В работе показано, что наличие АТП повышает эффективность производственных систем только в случае составления или коррекции расписания их работы, что позволяет повысить гибкость планирования и производительность станочных систем в целом.

Предложенный метод формализации АТП, как симбиоз традиционной линейной технологической структуры операционной технологии с операторами управления, при использовании в системах классов САПР ТП/АСТПП встроенных средств программирования скриптов (сценариев) позволяет создавать в одной записи базового ТП несколько различных АТП, что в дальнейшем позволяет автоматизировать процесс активации того или иного АТП в качестве текущего в зависимости от конкретных условий и использовать его в любых системах планирования классов MES, APS.

Список литературы

1. **Загидуллин Р. Р.** Управление машиностроительным производством с помощью систем MES, APS, ERP. 2-е изд., перераб. и доп. Старый Оскол: Изд-во ТНТ. 2015. 372 с.
2. **Загидуллин Р. Р.** Планирование машиностроительного производства. Старый Оскол: Изд-во ТНТ. 2013. 392 с.
3. **Балакшин Б. С.** Основы технологии машиностроения. М.: Машиностроение, 1969. 358 с.
4. **Колесов И. М.** Основы технологии машиностроения. М.: Машиностроение. 1997. 592 с.
5. **Анферов М. А., Селиванов С. Г.** Структурная оптимизация технологических процессов в машиностроении. Уфа: Гилем, 1996. 185 с.
6. **Логашев В. Г.** Технологические основы гибких автоматических производств. Л.: Машиностроение, Ленингр. отд-ние, 1985, 176 с.
7. **Долгов В. А.** Повышение производительности производственных участков в многономенклатурном производстве путем адаптации технологических процессов к их текущему состоянию. Автореферат диссертации на соискание ученой степени доктора технических наук. М.: МГТУ СТАНКИН. 2012. 43 с.
8. **Sivakumar K., Balamurugan C., Ramabalan S.** Concurrent multi-objective tolerance allocation of mechanical assemblies considering alternative manufacturing process selection // The International Journal of Advanced Manufacturing Technology. 2010. Vol. 53, Iss. 5—8. P. 711—732.
9. **Горшков А. Ф., Евтеев Б. Т., Коршунов В. А., Титов В. А., Фролов Е. Б.** Компьютерное моделирование менеджмента. М.: Экзамен. 2007. 622 с.
10. **FastScript 1.9.** Библиотека скриптов. Руководство разработчика. http://www.fast-report.com/public_download/fs_ru.pdf
11. **Schambach M.-P.** Fast script word recognition with very large vocabulary. Logistics & Assembly Syst., Siemens AG, Konstanz, Germany, DOI: 10.1109/ICDAR.2005.111 Conference: Document Analysis and Recognition, 2005 // Proceedings. Eighth International Conference on Source: IEEE Xplore. 2005.
12. **FastScripts** Puts Your AppleScripts in Your Menubar and Lets You Assign Keyboard Shortcuts. Adam Dachis. URL: http://lifehacker.com/5793851/fastscripts-puts-applescripts-in-your-menu-bar-and-lets-you-assign-keyboard-shortcuts

R. R. Zagidullin, Professor, polysoft@list.ru,
Ufa State Aviation Technical University

Automated Development of Alternative Technology

The paper deals with automating the creation of alternative technological processes (ATP) in systems classes CAM using the built-in libraries of programming scripts. The proposed method of formalization of ATP as a symbiosis of the traditional linear technological structure of the operating technology with operators control when used in systems of classes CAM built-in programming scripts allows you to create one record base technological process (TP) several different ATP, which further automates the process of activation of or another ATP as a current depending on the particular circumstances and be used in any system scheduling classes MES (Manufacturing Execution System), APS (Advanced Planning System).

Keywords: *technological process, alternative technological process, the script, CAM, planning, MES, APS*

References

1. **Zagidullin R. R.** *Upravlenie mashinostroitelnym proizvodstvom s pomoshyu system MES, APS, ERP* (Management of machine-building production systems using MES, APS, ERP). 2-e izd., pererab. i dop. Stary Oscol: Izd-vo TNT. 2015. 372 p.

2. **Zagidullin R. R.** *Planirovanie mashinostroinelnogo proizvodstva* (Planning of engineering production). Stary Oscol: Izd-vo TNT. 2013. 392 p.

3. **Balakshin B. S.** *Osnovy tehnologii mashinostroeniya*. Moscow: Mashinostroenie, 1969. 358 p.

4. **Kolesov I. M.** *Osnovy tehnologii mashinostroenija* (Fundamentals of Manufacturing Engineering). M.: Mashinostroenie. 1997. 592 p.

5. **Anferov M. A., Selivanov S. G.** *Strukturnaja optimizacija tehnologicheskih processov v mashinostroenii*. Ufa: Filem, 1996. 185 p.

6. **Logashev V. G.** *Tekhnologicheskie osnovy gibkih avtomaticheskih proizvodstv*. Leningrad: Mashinostroenie, 1985. 176 p.

7. **Dolgov V. A.** *Povyshenie proizvoditel'nosti proizvodstvennyh uchastkov v mnogonomenklaturnom proizvodstve putem adaptacii tekhnologicheskih processov k ih tekushchemu sostojaniju*. Avtoreferat dissertacii na soiskanie uchenoj stepeni doktora tehnicheskih nauk. Moscow: STANKIN, 2012. 43 p.

8. **Sivakumar K., Balamurugan C., Ramabalan S.** Concurrent multi-objective tolerance allocation of mechanical assemblies considering alternative manufacturing process selection. *The International Journal of Advanced Manufacturing Technology*. 2010, vol. 53, Iss. 5—8, pp. 711—732.

9. **Gorshkov A. F., Evteev B. T., Korshunov V. A., Titov V. A., Frolov E. B.** *Kompjuternoe modelirovanie menedgmenta* (Computer simulation management). Moscow: Examen. 2007. 622 p.

10. **FastScript 1.9.** Biblioteka scriptov. Rukovodstvo razrabotchika — http://www.fasl-repoil.com/public_download/fs_ru.pdf

11. **Schambach M.-P.** Fast script word recognition with very large vocabuiary. Logistics & Assembly Syst., Siemens AG, Konstanz, Germany, DOI: 10.1109/ICDAR.2005.111 Conference: Document Analysis and Recognition, 2005. *Proceedings. Eighth International Conference on Source*: IEEE Xplore. 2005.

12. **FastScripts** Puts Your AppleScripts in Your Menubar and Lets You Assign Keyboard Shortcuts. Adam Dachis. URL: http://lilehacker.com/5793851/lastscripls-puts-applescripts-in-your-menubar-and-lets-you-assign-keyboard-shortcuts

УДК 004.67

П. С. Поперечный, аспирант ИППМ РАН, e-mail: pporerechny@elvees.com
Институт проблем проектирования в микроэлектронике РАН

Применение цифровой фильтрации для реализации кодера Рида — Соломона

Предложен способ построения схем кодирования с помощью цифровых фильтров. Данный подход позволяет использовать способы цифровой фильтрации (разбиение на фильтры меньшего порядка, конвейеризация) в целях изменения корректирующей способности кода. Приведено описание данного способа для применения в кодировании РС (Рида — Соломона), аппаратно реализовано устройство кодера, приведены сравнительные характеристики.

Ключевые слова: *РС, корректирующая способность, регистр с линейной обратной связью (РЛОС), КИХ-, БИХ-фильтры (фильтры с конечной/бесконечной импульсной характеристикой), поле Галуа*

Введение

Коды РС (Рида — Соломона) относятся к блочному кодированию и широко используются в системах хранения и передачи информации. Данные коды позволяют исправлять множественные ошибки в блоках данных от нескольких битов до нескольких килобайтов (увеличение блока данных приводит к аппаратным сложностям) [1].

В данной статье предложен метод построения кодера с использованием цифровых фильтров. Хотя арифметика в данном кодировании построена на полях Галуа, основные методы цифровой фильтрации применимы и в конечных полях. Данный подход позволяет разбивать схему на последовательно соединенные фильтры первого порядка с возможностью конвейеризации в целях увеличения быстродействия. Также показано, что при таком подходе кодер легко становится реконфигурируемым, т. е. включением, отключением определенных фильтров в цепи можно изменять корректирующую способность кода.

В настоящее время данные коды массово используются в таких системах хранения информации, как твердотельные накопители, флэш-память и др. Таким образом, ввиду использования разных накопителей для работы с одним устройством необходимо применение кодов с разной корректирующей способностью соответственно. Например, для того чтобы в блоке данных используемый код позволял исправлять до 16 ошибок, необходимо применение определенного порождающего полинома определенной длины. Однако для того чтобы в этом же блоке данных код позволял исправлять например 12 ошибок, необходимо применение другого порождающего полинома с меньшей длиной. То есть для использования одного и того же устройства с разными накопителями необходимо применение разных порождающих полиномов и, как следствие, разных кодеров, что приводит к увеличению аппаратных ресурсов. Однако использование кодера с регулируемой корректирующей способностью (переменный порождающий полином) может также удовлетворить различным требованиям к корректирующей способности [2].

Традиционный способ кодирования РС

Из определения кодов РС следует, что систематическое кодирование осуществляется следующим образом:

$$\frac{m(x)x^{2t}}{g(x)} = q(x) + \frac{r(x)}{g(x)}, \qquad (1)$$

где $m(x)$ — входные незакодированные данные; $g(x)$ — порождающий полином; t — число исправляемых ошибок; $q(x)$ — частное от деления; $r(x)$ — остаток от делении на $g(x)$.

При этом результирующее кодовое слово (закодированные данные) в систематическом виде представляются как

$$c(x) = m(x)x^{2t} + r(x), \qquad (2)$$

где $c(x)$ — кодовое слово.

Таким образом, данные на выходе кодера остаются неизменными, однако к ним добавляются контрольные данные $r(x)$.

Аппаратная реализация выражения (2) выполняется с помощью регистра с линейной обратной связью (РЛОС), представленного на рис. 1. В пер-

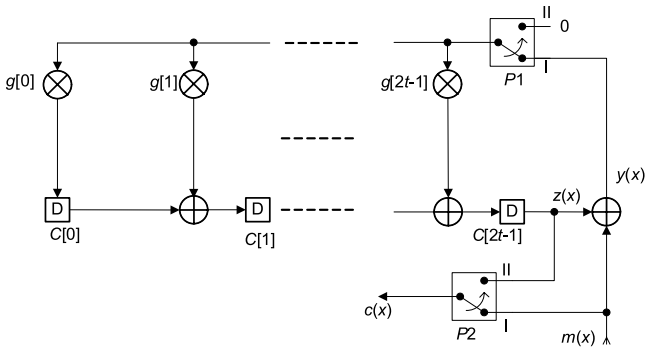


Рис. 1. Традиционная схема кодера РС

вые k тактов (k — число информационных символов) работы схемы данные проходят на выход схемы неизменными, при этом одновременно поступают на вход РЛОС, где с учетом обратной связи происходит вычисление остатка $r(x)$. После k тактов в схеме РЛОС отключается обратная связь, и значение остатка $r(x)$ фиксируется в сдвиговом регистре. В течение последующих $2t$ тактов из схемы выгружается значение остатка $r(x)$ и поступает на выход схемы.

При необходимости изменения требований к корректирующей способности кода РС необходимо изменить порождающий полином $g(x)$, что ведет к изменению схемы РЛОС. В статье предлагается способ построения схемы деления на порождающий полином с возможностью минимальными затратами изменять порождающий полином в процессе работы.

Способ реализации схемы РЛОС с помощью цифровых фильтров

Для реализации выражения (2) применяется схема РЛОС, представленная на рис. 1. Когда переключатели $P1$, $P2$ находятся в положении I, схема РЛОС становится похожей на схему БИХ-фильтра (фильтр с бесконечной импульсной характеристикой), с отличием в том, что выход кодера не совпадает с выходом БИХ-фильтра. А именно, выход РЛОС $z(x)$ выведен с регистра, однако в БИХ-фильтре с подобной реализацией выход данных $y(x)$ выведен с сумматора. Опишем математически работу схемы РЛОС при положении I переключателей $P1$, $P2$. Выход $y(x)$ БИХ-фильтра можно записать следующим образом:

$$y(x) = \frac{m(x)x^{2t}}{g(x)}. \qquad (3)$$

Однако выход $z(x)$ РЛОС в случае кодирования отличается, так как выведен до сумматора, а именно:

$$z(x) = \frac{m(x)x^{2t}}{g(x)} + m(x). \qquad (4)$$

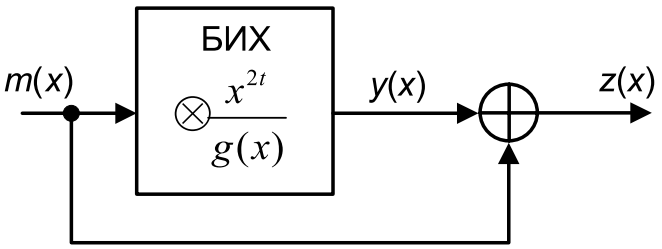


Рис. 2. Эквивалентная схема РЛОС с применением БИХ-фильтра

Эквивалентная схема, описываемая выражением (4), представлена на рис. 2.

Умножив и разделив последнее слагаемое $m(x)$ в выражении (4) на $\frac{x^{2t}}{g(x)}$, получим:

$$z(x) = \frac{m(x)x^{2t}}{g(x)} + \frac{m(x)x^{2t}}{g(x)}\frac{g(x)}{x^{2t}}. \qquad (5)$$

Эквивалентная схема выражения (5) представлена на рис. 3.

Реализация конвейерного кодера РС с помощью цифровых фильтров

После включения переключателей $P1$, $P2$ в положение II в схеме на рис. 1 отключается цепь обратной связи, а на вход схемы подается ноль. В эквивалентной схеме на рис. 3 на вход КИХ-фильтра (фильтра с конечной импульсной характеристикой) вместо выхода БИХ-фильтра подается ноль. Так как в первые k тактов схемы на выход кодера по определению систематического кодирования должны выдаваться данные без изменения, а в последующие $2t$ тактов выдаются данные, вычисленные кодером, можно пренебречь сумматором в схеме 3, добавить переключатель $P3$, в результате получится эквивалентная схема всего кодера, представленная на рис. 4.

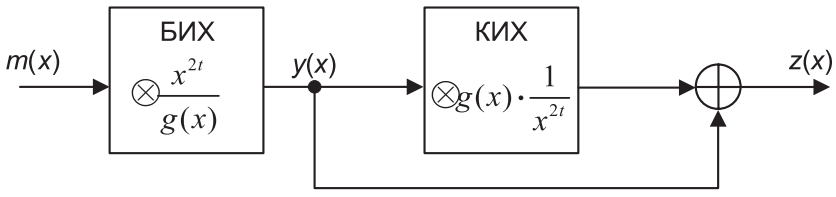


Рис. 3. Эквивалентная схема РЛОС с применением БИХ- и КИХ-фильтров

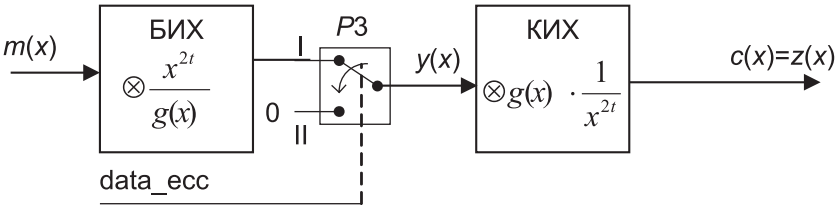


Рис. 4. Эквивалентная схема кодера РС с применением БИХ- и КИХ-фильтров

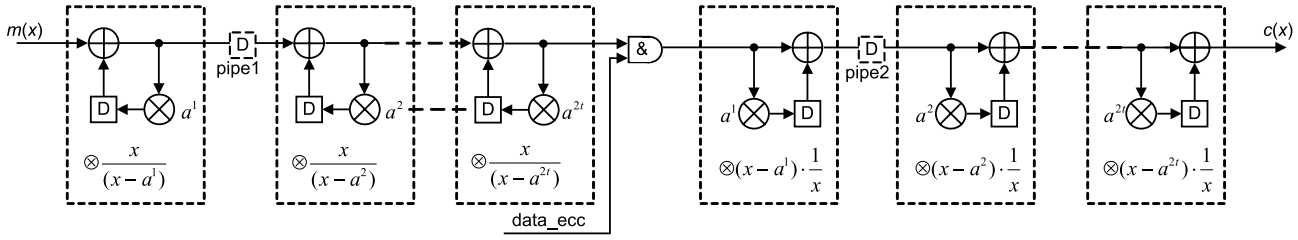


Рис. 5. Функциональная схема кодера РС с применением БИХ- и КИХ-фильтров с конвейеризацией

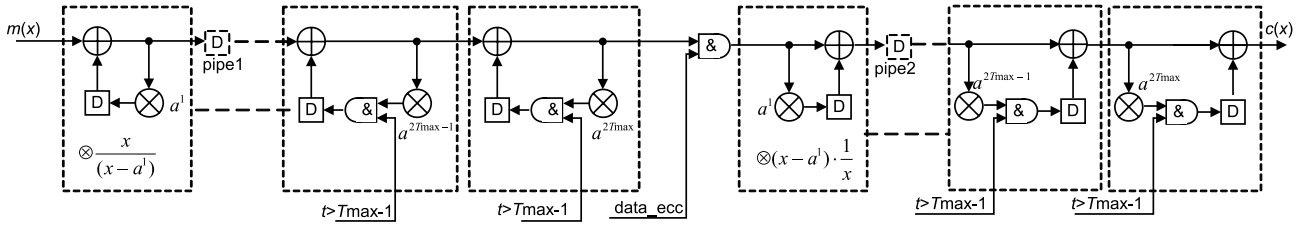


Рис. 6. Функциональная схема реконфигурируемого кодера РС с применением БИХ- и КИХ-фильтров с конвейеризацией

То есть в первые k тактов схемы на выход кодера поступают данные после БИХ- и КИХ-фильтров, соединенных последовательно, а так как оба фильтра имеют один и тот же полином, данные на выходе будут поступать без изменений, что и требуется по определению кодера систематического кода РС. Однако в эту фазу происходит вычисление, и после k-го такта БИХ-фильтр из цепи отключается переключателем $P3$ и продолжает работать лишь КИХ-фильтр, подавая на выход кодера значения контрольных символов.

Такое построение эквивалентной схемы (рис. 4) позволяет избежать общей цепи обратной связи, охватывающей всю схему РЛОС, как в схеме на рис. 1. Порождающий полином $g(x)$ для кодов РС представим в виде

$$g(x) = (x - a)(x - a^1)...(x - a^{2t}), \qquad (6)$$

где $a \in GF(2^m)$ — примитивный элемент поля Галуа размерности m.

Таким образом, схемы БИХ- и КИХ-фильтров можно разбить на последовательно соединенные фильтры первого порядка, как показано на рис. 5. А так как при таком построении схемы отсутствует общая цепь обратной связи с выхода на вход кодера, для уменьшения критического пути через цепь последовательно соединенных сумматоров можно конвейеризовать схему дополнительными регистрами, например pipe1 и pipe2 и т. д. при необходимости. При этом латентность схемы увеличивается на число тактов, равное числу дополнительных конвейерных регистров, и быстродействие становится крайне высоким.

Наилучшим применение схемы на рис. 5 получается при использовании дополнительного сигнала разрешения определенных фильтров. На рис. 6 представлена функциональная схема реконфигурируемого кодера РС с возможной конвейеризацией.

В представлении кодера в виде БИХ-фильтров, рассмотренном в работе [3], отсутствует основное преимущество подхода цифровой фильтрации. А именно, цифровые фильтры, передаточная функция которых описывается в виде сомножителей, можно представить в виде последовательной цепи фильтров меньшего порядка, как показано на рис. 5, 6.

Схема на рис. 6 наделяет код корректирующей способностью от максимально возможного числа исправляемых ошибок $T_{\max}$ до 1. Из выражения (6) следует, что для того чтобы код позволял исправлять t ошибок, необходим порождающий полином степени $2t$. Таким образом, схема реконфигурируемого кодера рассчитана на полином максимальной степени $2T_{\max}$. При необходимости кодировать данные кодом с числом исправляемых ошибок меньше максимального $T_{\max}$ необходимо с помощью сигналов разрешения отключить определенные фильтры (т. е. часть множителей в полиноме (6) заменяются на 1), и результирующий полином получится необходимой степени, а значит, с заданной корректирующей способностью.

Процедура декодирования кодов РС

Декодирование кодов РС структурно показано на рис. 7. Принятые данные (с возможными ошибками) $v(x)$ поступают в схему декодера, одновре-

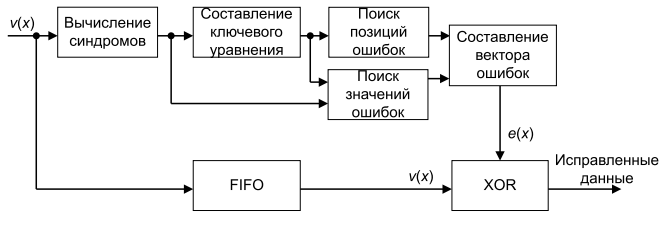


Рис. 7. Структурная схема декодирования кодов РС

менно происходит запись этих данных в буфер FIFO. Декодирование поделено на четыре основных этапа [4]. Сначала данные поступают в схему вычисления синдромов (признаки ошибок) в первые n тактов, потому как кодовое слово стало длины n. Дальнейшие вычисления декодер проводит с вычисленными синдромами. Следующий этап — вычисление полинома локаторов ошибок. Например, в алгоритме Берлекэмпа — Мэсси (BMA) без инверсии для данного этапа требуется $2t$ тактов (где t — число исправляемых ошибок, с которым было закодировано переданное кодовое слово). Далее вычисленные коэффициенты уравнения поступают в схемы поиска позиций ошибок и в схему поиска значений ошибок. После вычисления полинома значений ошибок этот полином вместе с указателями позиций ошибок из схемы поиска позиций ошибок подаются на схему составления вектора ошибок. Данная схема формирует маску $e(x)$ для исправления искаженных данных $v(x)$ при считывании из буфера FIFO. При фиксированном параметре m поля Галуа декодер, реконфигурируемый по числу исправляемых ошибок, реализуется изменением числа тактов, необходимых для работы алгоритма BMA и схемы поиска значений ошибок. Таким образом, добавив к схеме декодера управление в зависимости от параметра t (максимальное число исправляемых ошибок), декодер становится реконфигурируемым в зависимости от корректирующей способности, с которой было закодировано кодовое слово.

Экспериментальные результаты

На основе предложенной методики построения схемы кодирования с помощью цепи БИХ- и КИХ-фильтров разработано аппаратное описание кодера РС. Проведены сравнения по занимаемой площади (число логических вентилей), а также по задержкам на критических путях для ПЛИС семейства Altera (Arria V). Представлена таблица полученных характеристик для кодеров в поле Галуа $GF(2^m)$ при $m = 8$ (длина закодированных данных не будет превышать $n < 2^8$) в зависимости от числа исправляемых ошибок. Анализ проведен в сравнении с каноническим представлением кодера.

Например, для $T = 16$ предложенный кодер будет кодировать данные наиболее типовыми кодами для хранения информации во flash-памяти с числом исправляемых ошибок 16, 15, 14 и т. д. согласно используемой flash-памяти, что делает его гораздо более универсальным в сравнении с каноническим кодером.

Результаты синтеза представлены на графике рис. 8.

Результаты экспериментов по синтезу показали следующее:

Ресурсы ПЛИС (число ячеек), задержка на критическом пути в зависимости от числа исправляемых ошибок канонического кодера (const) и кодера на базе КИХ-БИХ фильтров (var)

Число ошибок t	Число ячеек		Задержка, нс	
	const	var	const	var
2	43	85	2,19	2,75
4	79	204	2,24	3,23
8	145	464	2,49	4,11
16	274	996	2,76	5,23

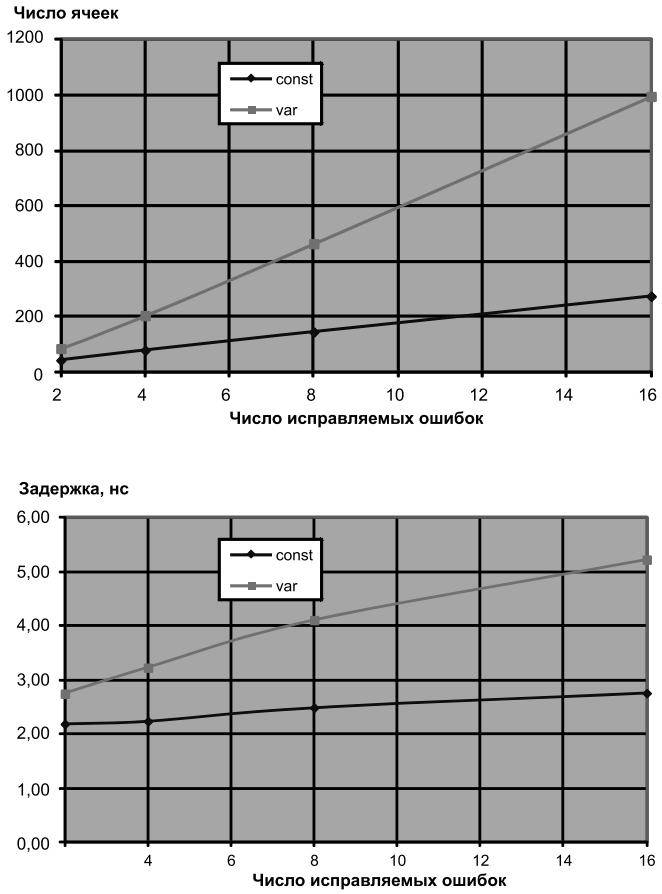


Рис. 8. Ресурсы ПЛИС (число ячеек), задержка на критическом пути в зависимости от числа исправляемых ошибок канонического кодера (const) и кодера на базе КИХ-БИХ фильтров (var)

- задержка на критическом пути предложенного кодера растет быстрее канонического в зависимости от числа исправляемых ошибок. Однако эксперимент проведен без использования конвейерных регистров. Очевидно, что с добавлением хотя бы одного конвейерного регистра (см. рис. 6) задержка уменьшится в 2 раза и быстродействие увеличится в сравнении с каноническим видом кодера;
- для конкретного семейства ПЛИС (Arria V) в среднем занимаемая площадь (в единицах логических ячеек) реконфигурируемого кодера в 2,5...3 раза больше кодера с постоянным многочленом.

Заключение

В данной статье описан способ применения цифровой фильтрации для устройств блочного кодирования. Алгоритмы и способы цифровой фильтрации позволили преобразовать схему кодера РС в удобную последовательность цифровых фильтров. Такое построение схемы делает кодер реконфигурируемым под разную корректирующую способность.

На основе предложенного способа разработан кодер, обладающий преимуществами при определенных условиях, например, при заведомо неизвестном числе исправляемых ошибок. Показано, что разработанное устройство обладает большим быстродействием в сравнении с кодером традиционного вида.

При применении устройства, разработанного предлагаемым способом, для работы с flash-памятью различных типов более 3 делает его выгодным по занимаемой площади. По быстродействию при добавлении конвейерных регистров есть выигрыш в сравнении с устройствами канонического вида, где невозможно сделать конвейеризацию ввиду общей обратной связи [5].

Список литературы

1. **Блейхут Р.** Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986.
2. **Yoo H., Lee Y.,** 7.3 Gb/s Universal BCH Encoder and Decoder for SSD Controllers. IEEE 978-1-4799-2816-3. 2014, pp. 37—38.
3. **Кириллов С. Н., Семин Д. С.** Модифицированный помехозащищенный кодер на основе БИХ-фильтра // Вестник РГРТУ. 2009. № 2.
4. **Вернер М.** Основы кодирования. М.: Техносфера, 2004. 288 с.
5. **Морелос-Сарагоса Р.** Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М.: Техносфера, 2005.

P. S. Poperechny, Postgraduate,
Institute for Design Problems in Microelectronics (IPPM RAS)

Digital Filtering Application for Reed-Solomon Coder Implementation

This article proposes a method for coding circuit development by means of digital filtering. This approach allows to use digital filtering methods (cutting to small-order filters, pipe-lining) to change correcting level capability. The well-known coder scheme for RS includes linear feedback shift register (LFSR), which is very similar to TIR-filter (infinite impulse response). So, it is possible to use digital filtering for coder. Here are algebraic and scheme presentations of this approach. It is shown that we can not just change LFSR by IIR — filter, because both of them have different outputs. As well the coder works at different modes for input data and for output. But if we will use FIR-filter (finite-impulse response) serial-connected to IIR we can overcome these issues. This approach allows using all digital filtering advantages, such as to split one big order filter to many first-order filters.

Exactly this property is very useful for reconfiguration of the RS-coder for differ applications. It means that we can use only one proposed coder for any purposes, for example for different flash-memory types rather than many different coders. Here are descriptions of this method for using in RS, coder hardware description, comparing of synthesis results on FPGA with traditional coder.

Keywords: *RS, correcting level capability, linear feedback shift register (LFSR), FIR-, IIR-filters (finite\infinite impulse response filters), Galois field*

References

1. **Blahut R.** *Teoriva i praktika kodov, kontrolirujushih oshibki.* Moscow: Mir, 1986.
2. **Yoo H., Lee Y.** 7.3 Gb/s Universal BCH Encoder and Decoder for SSD Controllers, *IEEE 978-1-4799-2816-3*, 2014, pp. 37—38.
3. **Kirillov S. N., Semin D. S.** Modificirovannij pomehozashishennij koder na osnove ВГН-filtra, *Vestnik RGRTU*, Rjazan, 2009, no. 2.
4. **Verner M.** *Osnovi kodirovanija.* Moscow: Tehnosfera, 2004. 288 p.
5. **Morelos-Saragosa R.** *Iskusstvo pomehoustojchivogo kodirovanija. Metodi algoritmi primenenije.* Moscow: Tehnosfera, 2005.

УДК: 621.394.147 + 004.056.53

А. В. Еременко[1], канд. техн. наук, доцент кафедры "Инфокоммуникационные системы и информационная безопасность", e-mail: nexus-@mail.ru,
А. Е. Сулавко[2], канд. техн. наук, старший преподаватель кафедры "Комплексная защита информации", e-mail: sulavich@mail.ru,
Д. А. Волков[2], аспирант, e-mail: vlkv.d.a@gmail.com
[1] Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Омский государственный университет путей сообщения" (ОмГУПС (ОмИИТ), г. Омск
[2] Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Омский государственный технический университет" (ОмГТУ), г. Омск

Современное состояние и пути модернизации преобразователей биометрия—код*

Рассмотрена проблема защиты криптографических ключей в процессе их эксплуатации. Объектом исследования в работе выступают преобразователи биометрия—код. Приведен обзор методов, позволяющих использовать биометрические признаки человека в качестве исходного материала для получения криптографических ключей шифрования, а также для идентификации и аутентификации. Определены основные факторы, влияющие на надежность работы рассмотренных методов. Предложены возможные пути модернизации существующих методов генерации криптографических ключей на основе динамических биометрических признаков.

Ключевые слова: *биометрические технологии, защита персональных данных, помехоустойчивое кодирование, исправляющая способность, идентифицирующие признаки, нечеткие данные, ключевая последовательность*

Введение

Роль биометрических технологий в современном обществе стремительно возрастает. Все большее число получаемых потребителем услуг связано с предоставлением биометрических данных. Биометрические данные требуются при осуществлении государственного контроля — оформления биометрических паспортов с отпечатками пальцев, виз, прохождения миграционного контроля. Растет популярность биометрических технологий при реализации новых проектов в коммерческом секторе: банковском обслуживании, мобильных приложениях, платежных системах, защите информации, учете рабочего времени, контроле доступа и др. Прогнозируется, что к 2019 г. отечественный рынок биометрических технологий преодолеет рубеж 300 млн долл. [1].

При этом крайне актуальными становятся вопросы не только технического, но и правового характера, связанные с обеспечением защиты прав и свобод человека и гражданина при обработке его персональных данных, в том числе защиты прав на неприкосновенность частной жизни, личную и семейную тайну. Согласно Федеральному закону "О персональных данных" № 152-ФЗ, биометрические персональные данные — это "физиологические или биологические характеристики человека, в том числе изображение человека (фотография и видеозапись), которые позволяют установить его личность и используются оператором для установления личности субъекта". По предварительной оценке *Zecurion Analytics* (*Zecurion* — крупнейший разработчик DLP-систем для защиты от внутренних угроз и утечек информации), ущерб мировой экономике от утечек информации за 2013 и 2014 годы составил более 42 млрд долл. При этом доля инцидентов, связанных с утечкой персональных данных, составляет более 60 % [2].

Для того чтобы выполнить требования законодательства по защите биометрических персональных данных операторы применяют методы шифрования и разграничение прав доступа к файлам и базам

* Работа выполнена при финансовой поддержке РФФИ (грант № 15-07-09053).

данных с биометрическими признаками пользователей. Защищенность конфиденциальной информации в большой степени зависит от надежности пароля, выбранного для шифрования или аутентификации субъектов доступа, а также процедур его выработки, изменения и хранения. Использование биометрических данных позволяет решить проблему человеческого фактора при выборе и использовании пароля пользователями информационных сервисов [3].

В случае идентификации пользователей по биометрическим признакам актуальной становится другая проблема — защита биометрического эталона пользователя, хранящегося на удаленном сервере аутентификации, а также биометрических признаков при выполнении процедуры биометрического сканирования. Для аутентификации в разных информационных системах можно использовать отличающиеся пароли. В случае аутентификации, например, по отпечаткам пальцев, число вариантов для изменения скомпрометированного признака пользователя ограничено. Современный уровень технологий позволяет злоумышленникам создавать из отпечатка пальца муляж, который будет принят системой сканирования за "живой" образец. При использовании психофизиологических признаков задача фальсификации усложняется, однако располагая информацией из биометрического эталона, злоумышленник может восстановить сигналы, описывающие биометрический образ автора и использовать их для получения доступа к информационным сервисам или ресурсам.

Но каким образом можно защитить конфиденциальную информацию биометрическими методами и при этом обеспечить защищенность биометрических данных пользователя? Вариант решения обозначенной выше проблемы рассмотрен в работе [4]. Решение заключается в получении из биометрических данных пользователей криптографических ключей, что позволяет объединить преимущества от использования биометрических технологий и криптографических методов, а также защитить биометрические данные пользователей при выполнении сервисных процедур. Предложенный метод позволяет отказаться от хранения биометрических признаков в исходном виде и делает бессмысленной кражу персонального биометрического идентификатора человека, так как он может быть изменен в любой момент при необходимости.

Цель настоящей работы — рассмотреть текущее состояние исследований в области разработки и применения преобразований биометрия—код, определить пути модернизации метода, предложенного авторами ранее в работе [4] для повышения надежности его работы. Интегральную надежность систем генерации криптографических ключей по биометрическим признакам можно определить по оценке риска принятия решения R [5]:

$$R = c_0 \int\limits_{-\infty}^{x_0} \omega_2(x)dx + c_1 \int\limits_{x_0}^{\infty} \omega_1(x)dx,$$

где c_i — вес ошибки; ω_2 — значение ошибки 2-го рода (FAR — *False Accept Rate*); ω_1 — значение ошибки 1-го рода (FRR — *False Rejection Rate*). Равновероятное значение ошибки 1-го и 2-го рода обозначается EER (*Equal Error Rate*).

1. Обзор методов генерации криптографических ключей из биометрической информации пользователей

Идея связывания биометрических образов с криптографическими ключами шифрования возникла как производная от идеи, так называемой, аннулируемой биометрии [6]. Ее суть состоит в устранении существенного недостатка классической биометрии, связанного с невозможностью изменять физиологический биометрический признак в случае его компрометации.

Основной проблемой при генерации ключевых последовательностей по биометрическим данным является техническая невозможность получения одинаковых образов биометрических характеристик при их повторном вводе пользователем. Ситуационные изменения в психофизиологическом состоянии человека, изменение поведенческих характеристик в течение жизни, наложение шумов при получении биометрических образов приводят к проблеме несовпадения биометрических признаков одного и того же человека и генерация ключа при этом становится затруднительной. Для решения этой проблемы в мировой практике изначально сложились два подхода: нечеткое хранилище (*fuzzy vault*) [7] и нечеткий экстрактор (*fuzzy extractor*) [8] — теоретическая работа, ставшая фундаментальной основой для многочисленных современных исследований. Объединяет эти два подхода использование вспомогательной открытой информации, аналогичной той, что применяется при шифровании с открытым ключом. На основе пары открытая строка — предъявленный идентификатор при аутентификации происходит построение криптографического ключа. Различие между нечетким хранилищем и нечетким экстрактором состоит в том, что в первом случае открытая строка представляет собой множество произвольного вида, в то время как во втором случае — конечномерный вектор. Разработанные методы позволяют сгенерировать новый ключ пользователя на основе его биометрического признака. Таким образом, применение биометрии приближается по удобству использования к паролю и другим повторно выдаваемым идентификаторам.

Сравнение достигнутых результатов в области генерации криптографических ключей на основе биометрических признаков

Автор	Тип биометрического признака	FRR/FAR, %	Размер ключа, бит	Условия тестирования
Hao F. и др. [9] Bringer J. и др. [10] Rathgeb C. и Uhl A. [11]	Радужная оболочка глаза	0,47/0 5,62/0 4,64/0	140 42 128	70 испытуемых База данных (БД) ICE 2005 БД ICE 2005
Teoh A. и Kim J. [12] Nandakumar K. [13]	Отпечаток пальца	0,9/0 12,6/0	296 327	БД FVC 2002 БД FVC 2002
Ao A. и Li J. [14]	Изображение лица	7,99/0,11	>4000	294 испытуемых
Maiorana E. и Campisi P. [15]	Подпись	EER >9	>100	БД MCYT
Sutcu Y. и др. [16]	Изображение лица и отпечаток пальца	0,92/0,01	—	БД NIST DB 27 & Face 94
Nandakumar K. и Jain A. [17]	Отпечаток пальца и радужная оболочка глаза	1,8/0,01	224	БД MSU-DBI & CASIAv1
Kelkboom E. и др. [18]	3D-изображение лица	22/0,25	155	БД FRGC

О состоянии исследований в данной области можно судить по таблице, в которой собраны достижения зарубежных ученых по основным направлениям биометрических технологий.

В работе [19], поддержанной австрийским научным фондом (проект № L554-N15), авторы дали оценку робастности результатов, приведенных в таблице с учетом влияния шумовых воздействий на биометрические образы. Авторы пришли к выводу, что результаты, полученные с использованием разработанных методов, являются в определенной степени робастными и на практике должны показать сопоставимые с представленными в таблице оценками ошибки 1-го и 2-го рода.

Как видно из таблицы развитие биометрической криптографии двигается в направлении от использования единичных биометрических признаков к мультимодальным системам генерации криптографических ключей из биометрических признаков. Одна из ключевых проблем в данном подходе связана с формированием общего вектора описания биометрических характеристик с учетом вероятности их единичной ошибки. В работе [20] показана связь эффективности коррекции ошибок с методами группирования битов с разной вероятностью единичной ошибки. Несмотря на предпринятые в данном направлении усилия, единого подхода для решения этого вопроса до сих пор выработано не было.

Для психофизиологических характеристик достичь показателей систем, использующих для генерации ключей физиологические признаки, пока не удается. В работе [21] проведено самостоятельное исследование метода генерации ключей на подписях 126 испытуемых из базы данных рукописных паролей MCYTX. Работа поддержана испанским Министерством науки и технологий (MCYT TIC2003-08382-C05-01) и европейской комиссией по науке и технологиям (IST-2002-507634 Biosecure NoE projects). Результаты в работе получены следующие: ошибка 1-го рода (FRR) составила 57,30 % при ошибке 2-го рода (FAR) 1,18 % для профессиональных подделок и 0,32 % для подделок, выполненных без наблюдения аутентичной подписи и стиля ее написания. Снизить уровень ошибок 1-го рода автор предлагает за счет повторного ввода подписи.

Рассмотрим опыт предшественников. В работе [22] получена средняя ошибка 1-го рода (FRR) 7,05 % и ошибка 2-го рода (FAR) для профессиональных подделок 0 % для 11 испытуемых. Похожий результат был получен в работе [23] для метода замены скомпрометированных ключей с использованием биометрических хешей, получаемых из рукописного почерка. Значение равной ошибки (EER) для этого метода составило < 6,7 % для 40 испытуемых. В работе [24] разработан метод получения ключа из рукописного пароля с ошибкой 1-го рода 28 % при ошибке 2-го рода 1,2 %. Стоит отметить, что данный результат был получен без использования кодов, исправляющих ошибки.

Круг отечественных ученых, занимающихся вопросами совмещения биометрии и криптографии, не столь широк. В работе [25] рассматривается вопрос получения устойчивого криптографического ключа из биометрической характеристики изображения отпечатков пальцев. Предложенные автором методы позволяют строить вектор длиной 600...1728 битов, содержащий около 25 % ошибочных позиций. Для исправления ошибок используют коды Адамара, а также каскадное кодирование, включающее коды Боуза-Чоудхури—Хоквингхема (БЧХ) и репликацию. Такое кодирование обусловлено структурой ошибок: БЧХ-коды корректируют шумовые ошибки, в то время как репликация позволяет справляться с блочными ошибками (свя-

занными с "выпадением" минюций). Таким образом, с вероятностью около 90 % получим исходный ключ с энтропией в 20—36 бит (даже очень защищенные системы редко используют для формирования криптографических ключей материал свыше 200 бит). Для построения зашумленного биометрического вектора из изображения отпечатка пальца на сегодняшний день не существует общепринятого подхода, что связано с особенностями описания шаблона отпечатка пальца. В работе автор опробовал несколько методов для описания ключевых точек отпечатка пальца, лучший результат при энтропии ключа 30 бит для метода с введением внешнего порядка характеризуется уровнем ошибок FRR < 19 % при FAR < 0,01 %.

В работе Е. А. Харина под руководством канд. физ.-мат. наук С. М. Гончарова (см. также [26]) на тему "Генерация ключевых последовательностей на основе биометрических данных пользователей" рассматриваются вопросы генерации ключевых последовательностей на основе клавиатурного почерка и радужной оболочки глаза. Всего для экспериментальной проверки разработанных методов были собраны более 1400 образцов клавиатурного почерка от 26 пользователей, уровень владения клавиатурой которых оценивался как "хороший" или "отличный". В результате экспериментов была также получена зависимость значений ошибок 1-го и 2-го рода от чувствительности системы. Было обнаружено, что уровень ошибки 2-го рода может достигать 0 при вероятности ошибки 1-го рода 50 %. Коэффициент равновероятной ошибки составил более 12 %. Принципиальное отличие разработанной системы генерации ключевых последовательностей на основе радужной оболочки глаза от созданных ранее заключается в последовательном применении двух помехоустойчивых кодов, каждый из которых исправляет ошибки определенного типа. Применение такого подхода, а также маскирования позволило достигнуть уровней ошибок 1-го и 2-го рода 0,47 % и 0 % соответственно и получать ключи длиной 140 бит. Автор также делает вывод, что рисунок радужной оболочки глаза в настоящее время позволяет получать более длинные ключи, чем какая бы то ни было другая биометрическая характеристика, за исключением, быть может, кода ДНК.

Работа [27] посвящена разработке преобразователя биометрия—код доступа на основе электроэнцефалограммы (ЭЭГ) головного мозга человека. В качестве преобразователя используется двухслойная нейронная сеть сигмоидального типа. Нейросетевой преобразователь выдает на выходе 256-битовый ключ. В построенном преобразователе использовались 210 параметров. Прогноз вероятности ошибки 2-го рода составил $\leqslant 10^{-12}$. Участвовало 15 пользователей, проверено порядка 5000 ЭЭГ. В самом худшем случае расстояние Хэмминга до пароля при подделке составило 21 бит в 256-битном ключе. В целом данное направление выглядит перспективным, в качестве недостатка можно отметить необходимость использования специального оборудования и сложную на данный момент процедуру получения биометрического образа человека.

В описании к изобретению [28] указано, что поле кодовых комбинаций для рисунка отпечатка пальца составляет всего 1000 или 10 000 (страница 9 описания, строка 12). В случае доступа злоумышленника к кодам программы автоматически перебрать столь малое поле кодовых комбинаций крайне просто. Реализация атаки займет всего несколько секунд машинного времени, поэтому основным недостатком этого способа является его низкая стойкость к атакам подбора.

Известен способ идентификации человека по его биометрическому образу [29], основанный на использовании нейросетевого преобразователя биометрия—код с повышенной стойкостью к атакам подбора. В случае сохранения рукописного биометрического пароля из пяти букв в тайне данный способ позволяет получить стойкость на уровне порядка 10^{12} попыток атаки подбора (см. таблицу А2 приложения А стандарта [30]). Основным недостатком способа является его недостаточная стойкость к атакам подбора по современным требованиям. Увеличение длины ключа нецелесообразно, так как приведет к необходимости ввода гораздо более сложного рукописного пароля и значительно затруднит работу пользователя с системой. Так, если требуется безопасно обеспечивать доступ к криптографическому ключу длиной 256 бит (стойкость к атакам подбора 10^{77}), потребуется организовывать ввод рукописного пароля, состоящего примерно из шести слов по пять букв. Пользователь обязательно ошибется хотя бы в одном из шести слов и вынужден будет вновь писать шесть рукописных слов. Написать одно рукописное слово обычно удается с одной или двух попыток. Для безошибочного написания двух слов пароля требуется от одной до четырех попыток. Для написания трех слов необходимо уже около 12 попыток. При этом пользователь не знает, в каком слове он ошибся в связи с запретом стандарта (см. п. 7.6 ГОСТ Р52633—2006) снабжать однозначными индикаторами верного результата фрагменты составного биометрического пароля (ключа).

В соответствии с работой [31] код доступа перекодируют в код с обнаружением и исправлением ошибок (стр. 12 описания, п. 7 формулы изобретения, строка 34). Основным недостатком этого изобретения является его низкая способность исправлять нестабильные биометрические данные. Со-

временные коды способны обнаруживать до 80 % ошибок, но исправлять они могут не более 5 % ошибок. По этой причине при разработке нечетких экстракторов подобных тем, что предложены в данной работе, приходится использовать только малую часть наиболее стабильной биометрической информации. Биометрические данные со стабильностью менее 5 % приходится отбрасывать. Как следствие, нейросетевые преобразователи биометрия—код оказываются эффективнее нечетких экстрактов, так как они используют все данные и могут работать с произвольными кодами.

Вторым недостатком изобретения является то, что примеры биометрических образов, введенных с некоторыми ошибками, далее при аутентификации не используются. Это делает все описанные выше способы неэффективными при мультибиометрической аутентификации.

Третьим и главным недостатком является то, что его реализация для нескольких совместно используемых биометрических образов не дает экспоненциального выигрыша по стойкости биометрической защиты к атакам подбора, так как сложность восстановления каждой из частей ключа будет примерно одинаковой и каждую из частей составного ключа можно подбирать независимо.

Целью предлагаемого изобретения в работе [32] является повышение надежности биометрической аутентификации за счет использования нескольких биометрических образов при обеспечении приемлемого уровня дружественности системы, безопасно подсказывающей человеку при вводе какого именно образа он ошибся и тем самым сокращающей трудозатраты людей на высоконадежную мультибиометрическую аутентификацию. Для генерации аутентификатора используют совершенно разные биометрические образы (например, рукописный пароль, голосовой пароль, рисунок отпечатка пальца). При этом для каждого образа применяют свой преобразователь биометрия—код. Интересным решением является обучение нейросетевого преобразователя биометрия—код на самокорректирующийся выходной код, способный обнаруживать и исправлять ошибки. Все коды неудачной биометрической аутентификации в рамках одного сеанса запоминают и затем сравнивают между собой состояния их разрядов. Далее выявляют наиболее нестабильные разряды кодов и осуществляют их направленный перебор до момента появления кода биометрического образа без ошибки или с числом ошибок, поддающимся исправлению выбранным самокорректирующимся кодом.

Положительный технический эффект обусловлен использованием при восстановлении ключа всех введенных пользователем биометрических образов. Решение, предложенное авторами, позволяет сообщать пользователю, находящемуся в измененном состоянии (например, в стрессовой ситуации), о времени, которое потребуется на восстановление ключа. Таким образом, у пользователя есть выбор — продолжить ожидание восстановления (перебора) ключа, либо инициировать новый сеанс восстановления ключа.

Таким образом, в настоящее время ученые используют два подхода для преобразования биометрических характеристик в воспроизводимые битовые последовательности. Первый подход основан на нейронных сетях, второй носит название "нечеткий экстрактор" и работает с использованием помехоустойчивого кодирования. Классические самокорректирующиеся коды малоэффективны. Они требуют очень большой избыточности. Наблюдается экспоненциальный рост избыточности кода, способного править линейно растущий процент ошибок. В работе [33] приводится следующая точка зрения: классические коды на практике не могут исправить более 30 % ошибок, так как их информационная часть оказывается незначительной. Коды Безяева предназначены для биометрии и не поглощают биометрические данные своей избыточностью. Они безопасно хранят рядом синдромы ошибок в виде усеченных хэш-функций [34].

Нейросетевые преобразователи позволяют использовать полный объем выделяемых из биометрического образа человека характеристик, в то время как нечеткие экстракторы работают только со стабильной частью информации. Обучение каждого слоя нейронов сети выполняют своим автоматом, учитывающим особенности той или иной биометрической технологии для входных данных разного качества [35].

Основной проблемой, которую необходимо решить разработчику нейросетевых преобразователей биометрия—код, является оптимальный подбор коэффициентов обучения нейронной сети. Обучение нейросетевых преобразователей биометрия—код должно проводиться на примерах биометрических образов "Свой" и "Чужой" и от этой процедуры зависит защищенность ключа пользователя от попыток его восстановления злоумышленником. Качество защиты определяется законом распределения генерируемых значений биометрических характеристик, используемым в автомате для описания модели формирования признаков при фальсификации злоумышленником аутентичных образов. Несмотря на сложность данного подхода, методы нейросетевого преобразования биометрии в код показывают лучшие результаты.

Преимуществом преобразователей биометрия—код, основанных на помехоустойчивом кодировании, является возможность отказаться от хранения непосредственно самих биометрических данных,

так как вектор биометрических признаков "объединяется" с секретным ключом. Для того чтобы получить ключ, необходимо "знать" биометрические данные, "отсоединив" их от ключа, т. е. хранения самих биометрических данных не требуется. В качестве операций "объединения" и "разъединения" битовых последовательностей, как правило, используется побитовое сложение по модулю 2. При использовании нейросетевого подхода требуется где-то хранить эталон (веса нейронов), следовательно, нужно обеспечить защиту эталона.

Существует точка зрения, что в силу малоинформативности динамических процессов человека (рукописного и клавиатурного почерка, голоса и др.), получаемые при их преобразовании в возобновляемый код ошибки 1-го и 2-го рода будут на порядок больше, чем при преобразовании физиологических признаков (радужная оболочка глаза, отпечатки пальцев, изображение лица, геометрия ладони и др.). Известна причина нестабильности результатов для "психофизиологического" направления: ответ кроется в самой природе этих признаков — психофизиологии человека, которая изменчива.

2. Применение рассмотренных методов на практике

Рассмотренные методы могут быть использованы для повышения защищенности закрытых ключей шифрования конфиденциальной информации (хранение и обмен ключами не требуются, если секретный ключ "привязан" к личности), защиты авторских прав с помощью встраивания водяных знаков в электронные произведения авторов, в области биометрической идентификации пользователей компьютерных систем для повышения уровня защищенности от угрозы неавторизованного доступа.

Заключение

Изучение рассмотренных в статье научных работ, а также имеющийся у авторов задел, позволяют сделать заключение, что достичь прорыва в области генерации криптографических ключей по психофизиологическим признакам человека и снизить ошибки 1-го и 2-го рода до уровня систем, основанных на физиологии, возможно, если идти по следующим путям модернизации.

1. Выделить и определить потенциал признаков рукописного и клавиатурного почерка, голоса для генерации ключей шифрования.

2. Разработать алгоритм маркирования стабильных для каждого пользователя признаков.

3. Разработать алгоритм формирования битового вектора характеристик рукописного, клавиатурного почерка и голоса с учетом вероятности их единичной ошибки (см. работу [20], где была продемонстрирована связь эффективности коррекции ошибок с методами группирования битов с разной вероятностью единичной ошибки).

4. Разработать специальный программный модуль для воспроизведения вычислительного эксперимента по аналогии с тем, как это было реализовано в работе [36]. Предполагается предусмотреть возможности проведения экспериментов с использованием настоящих биометрических данных (реализаций подсознательных движений, введенных субъектами), а также с возможностью генерации реализаций биометрических параметров на основе эталонов (созданных из настоящих биометрических данных). Генерация значений признаков может быть осуществлена с помощью метода Монте-Карло по аналогии с тем, как это осуществлялось в работе [37].

5. Разработать комплексный преобразователь биометрия—код на основе кодов Безяева [34] и/или нейронной сети, настроенной на мультимодальную работу с несколькими типами биометрических признаков и обученной на самокорректирующийся выходной код, способный обнаруживать и исправлять ошибки.

6. Адаптировать генератор ключей к изменению психофизиологического состояния владельца биометрических данных.

Список литературы

1. **Биометрический** рынок России: прогноз на 2015 год и перспективу. М.: Биолинк Солюшенс, 2014. 17 с.

2. **Утечки** конфиденциальной информации. Предварительные итоги 2014 г. М.: Zecurion, 2015. 15 с.

3. **Сулавко А. Е., Еременко А. В., Левитская Е. А.** Разграничение доступа к информации на основе скрытого мониторинга действий пользователей в информационных системах: портрет нелояльного сотрудника // Известия Транссиба/ОмГУПС. 2015. № 1 (21). С. 80—89.

4. **Еременко А. В., Сулавко А. Е.** Исследование алгоритма генерации криптографических ключей из биометрической информации пользователей компьютерных систем // Информационные технологии. 2013. № 11. С. 47—51.

5. **Еременко А. В.** Повышение надежности идентификации пользователей компьютерных систем по динамике написания паролей: Автореф. дис. ... канд. техн. наук. Омск, 2011. 20 с.

6. **Ratha N. K., Connell J. H., Bolle R. M.** Enhancing security and privacy in biometrics—based authentication systems // IBM Systems Journal. 2001. N 40 (3). P. 614—634.

7. **Juels A., Sudan M.** A fuzzy vault scheme // Proc. IEEE Intil. Symp. Inf. Theory. 2002. 408 p.

8. **Dodis Y., Reyzin L., Smith A.** Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data // Proc. from Advances in Cryptology. EuroCrypt. 2004. P. 79—100.

9. **Hao F., Anderson R., Daugman J.** Combining Cryptography with Biometrics Effectively // IEEE Transactions on Computers. 2006. N 55 (9). P. 1081— 1088.

10. **Bringer J., Chabanne H., Cohen G., Kindarji B., Z'emor G.** Theoretical and practical boundaries of binary secure sketches // IEEE Transactions on Information Forensics and Security. 2008. N 3. P. 673—683.

11. **Rathgeb C., Uhl A.** Adaptive fuzzy commitment scheme based on iris-code error analysis // Proc. of the 2nd European Workshop on Visual Information Processing (EUVIP'10). 2010. P. 41—44.

12. **Teoh A., Kim J.** Secure biometric template protection in fuzzy commitment scheme // IEICE Electron. Express. 2007. N 4 (23). P. 724—730.

13. **Nandakumar K.** A fingerprint cryptosystem based on minutiae phase spectrum // Proc. of IEEE Workshop on Information Forensics and Security (WIFS). 2010. P. 1—6.

14. **Ao M., Li S. Z.** Near infrared face based biometric key binding // In Proc. of the 3rd International Conference on Biometrics 2009 (ICB'09) LNCS: 5558. 2009. P. 376—385.

15. **Maiorana E., Campisi P.** Fuzzy commitment for function based signature template protection // IEEE Signal Processing Letters. 2010. N 17. P. 249—252.

16. **Sutcu Y., Li Q., Memon N.** Secure biometric templates from fingerprint-face features // In IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07. 2007. P. 1—6.

17. **Nandakumar K., Jain A. K.** Multibiometric template security using fuzzy vault // In IEEE 2nd International Conference on Biometrics: Theory, Applications, and Systems, BTAS '08. 2008. P. 1—6.

18. **Kelkboom E. J. C., Zhou X., Breebaart J., Veldhuis R. N. S., Busch C.** Multi-algorithm fusion with template protection // In Proc. of the 3rd IEEE Int. Conf. on Biometrics: Theory, applications and systems (BTAS'09). 2009. P. 1—7.

19. **Rathgeb C., Uhl A.** Iris-Biometric Fuzzy Commitment Schemes under Signal Degradation, ICISP'12 // In Proceedings of the 5th international conference on Image and Signal Processing. P. 217—225.

20. **Scotti F., Cimato S., Gamassi M., Piuri V., Sassi R.** Privacy-aware Biometrics: Design and Implementation of a Multimodal Verification System // 2008 Annual Computer Security Applications Conference, IEEE. 2008. P. 130—139.

21. **Santos M. F., Aguilar J. F., Garcia J. O.** Cryptographic key generation using handwritten signature // Proceedings of SPIE, Orlando, Fla, USA, Apr. 2006. 2006. Vol. 6202. P. 225—231.

22. **Vielhauer C., Steinmetz R., Mayerhöfer A.** Evaluating biometric encryption key generation // In Proc. ICPR. 2002. P. 123—126.

23. **Yip K. W., Goh A., Ling D. N. C., Jin A. T. B.** Generation of replaceable cryptographic keys from dynamic handwritten signatures // In Proc. ICB, Lecture Notes in Computer Science 3832, Springer. 2006. P. 509—515.

24. **Hao F., Chan C. W.** Private key generation from on-line handwritten signatures // Information Management & Computer Security, Is. 10. 2002. №. 2. P. 159—164.

25. **Ушмаев О. С., Кузнецов В. В.** Алгоритмы защищенной биометрической верификации на основе бинарного представления топологии отпечатков пальцев // Информатика и ее применения. 2012. Том 6, № 1. С. 132—140.

26. **Харин Е. А., Гончаров С. М., Корнюшин П. Н.** Построение систем биометрической аутентификации с использованием генератора ключевых последовательностей на основе нечетких данных // Матер. 50-й Всерос. межвуз. науч.-техн. конф. Владивосток: ТОВМИ, 2007. С. 112—115.

27. **Гончаров С. М., Боршевников А. Е.** Нейросетевой преобразователь "биометрия — код доступа" на основе мысленного pin-кода // Труды Научно-технической конференции кластера пензенских предприятий, обеспечивающих безопасность информационных технологий. Пенза, 2014. Том 9. С. 46—50.

28. **Медль А., Штефан Э., Мюллер Р.** Способ и система для генерирования набора ключа доступа, а также для аутентификации человека на основе его биометрического параметра. Патент России № 2267159. 2005. Бюл. № 36.

29. **Ефимов О. В., Иванов А. И., Фунтиков В. А.** Способ идентификации человека по его биометрическому образу. Патент России № 2005102541.02.02. 2005.

30. **ГОСТ Р 52633.0—2006.** Защита информации. Техника защиты информации. Требования к средствам высоконадежной биометрической аутентификации. М.: Изд-во стандартов, 2007.

31. **Чмора А. Л., Уривский А. В.** Биометрическая система аутентификации. Патент России № 2316120. 2008. Бюл. № 3.

32. **Иванов А. И.** Способ безопасной биометрической аутентификации // Патент России №2406143. 2010.

33. **Иванов А. И.** Нейросетевая защита конфиденциальных биометрических образов гражданина и его личных криптографических ключей. Пенза, 2014. 57 с.

34. **Безяев А. В., Иванов А. И., Фунтикова Ю. В.** Оптимизация структуры самокорректирующегося биокода, хранящего синдромы ошибок в виде фрагментов хеш-функций // Вестник Уральского федерального округа. Безопасность в информационной сфере. 2014. № 3 (13). С. 4—14.

35. **ГОСТ Р 52633.5—2011.** Защита информации. Техника защиты информации. Автоматическое обучение нейросетевых преобразователей биометрия — код доступа. Введен впервые; Введ. 01.12.2011. М.: Стандартинформ, 2012. 20 с.

36. **Сулавко А. Е.** Программный комплекс для быстрого прототипирования систем распознавания образов // Тезисы II Всероссийской конференции "Теория и практика Успеха": Омск 10—11 апреля 2014 г. С. 21—22.

37. **Епифанцев Б. Н., Ложников П. С., Сулавко А. Е.** Сравнение алгоритмов комплексирования признаков в задачах распознавания образов // Вопросы защиты информации. 2012. № 1. С. 60—66.

A. V. Eremenko, Ph. D., Associate Professor of the Department "Info-communication systems and information security", nexus-@mail.ru, Omsk State Transport University (OSTU),
A. E. Sulavko, Ph. D., Senior Lecturer of the Department "Comprehensive Information Protection", sulavich@mail.ru,
D. A. Volkov, Postgraduate Student, vlkv.d.a@gmail.ru,
Omsk State Technical University (OmSTU)

Current State and Ways to Modernize Converters Biometrics to Code

The article is devoted to the problem of the protection of cryptographic keys during their operation. The object of study in the article are the converters biometrics to code. An overview of the methods for use biometric characteristics of a human as a starting material for producing the cryptographic key encryption, as well as for identification and authentication is performed. The main factors affecting the reliability of the methods are considered. The conclusion about possible ways to modernize the existing methods to generate cryptographic keys based on dynamic biometric features is made.

Keywords: *biometric techniques, personal data protection, error-correcting codes, error correcting capability, biometric features, fuzzy data, key sequence*

References

1. **Biometricheskij** *rynok Rossii: prognoz na 2015 god i perspektivu* [Biometric market in Russia: Forecast for 2015 and for the feature]. Moscow: Biolink Soljushens, 2014, 17 p.

2. **Utechki** *konfidencial'noj informacii*. Predvaritel'nye itogi. Leakage of confidential information. Preliminary results of 2014, Moscow, Zecurion, 2015, 15 p.

3. **Sulavko A. E., Eremenko A. V., Levitskaja E. A.** Razgranichenie dostupa k informacii na osnove skrytogo monitoringa pol'zovatelej komp'juternyh sistem: portret neloial'nogo sotrudnika [Concurrent access to information based on hidden monitoring of computer systems users: Portrait of a disloyal employee]. *Izvestija Transsiba / OSTU*, Omsk, 2015, no. 1 (21), pp. 80—89.

4. **Eremenko A. V., Sulavko A. E.** Issledovanie algoritma generacii kriptograficheskih kljuchej iz biometricheskoj informacii pol'zovatelej komp'juternyh sistem [Investigation of algorithm for generating cryptographic keys from biometric information of users of computer systems]. *Informacionnye tehnologii*, 2013, no. 11, pp. 47—51.

5. **Eremenko A. V.** *Povyshenie nadezhnosti identifikacii pol'zovatelej komp'juternyh sistem po dinamike napisanija parolej* [Improving the reliability of computer systems to authenticate users on the dynamics of handwriting passwords], Abstract. dis. ... cand. tehn. sciences, Omsk, 2011. 20 p.

6. **Ratha N. K., Connell J. H., Bolle R. M.** Enhancing security and privacy in biometrics-based authentication systems, *IBM Systems Journal*, 2001, no. 40 (3), pp. 614—634.

7. **Juels A., Sudan M.** A fuzzy vault scheme, *Proc. IEEE Intil. Symp. Inf. Theory*. 2002. 408 p.

8. **Dodis Y., Reyzin L., Smith A.** Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data, *Proc. from Advances in Cryptology. EuroCrypt*, 2004, pp. 79—100.

9. **Hao F., Anderson R., Daugman J.** Combining Cryptography with Biometrics Effectively, *IEEE Transactions on Computers*, 2006, no. 55 (9), pp. 1081—1088.

10. **Bringer J., Chabanne H., Cohen G., Kindarji B., Z'emor G.** Theoretical and practical boundaries of binary secure sketches, *IEEE Transactions on Information Forensics and Security*, 2008, no. 3, pp. 673—683.

11. **Rathgeb C., Uhl A.** Adaptive fuzzy commitment scheme based on iris-code error analysis, *In Proc. of the 2nd European Workshop on Visual Information Processing (EUVIP'10)*, 2010, pp. 41—44.

12. **Teoh A., Kim J.** Secure biometric template protection in fuzzy commitment scheme, *IEICE Electron. Express*, 2007, no. 4 (23), pp. 724—730.

13. **Nandakumar K.** A fingerprint cryptosystem based on minutiae phase spectrum, *In Proc. of IEEE Workshop on Information Forensics and Security (WIFS)*, — 2010, pp. 1—6.

14. **Ao M., Li S. Z.** Near infrared face based biometric key binding, *In Proc. of the 3rd International Conference on Biometrics 2009 (ICB'09) LNCS: 5558*, 2009, pp. 376—385.

15. **Maiorana E., Campisi P.** Fuzzy commitment for function based signature template protection, *IEEE Signal Processing Letters*, 2010, no. 17, pp. 249—252.

16. **Sutcu Y., Li Q., Memon N.** Secure biometrie templates from fingerprint-face features, *In IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07*, 2007, pp. 1—6.

17. **Nandakumar K., Jain A. K.** Multibiometric template security using fuzzy vault, *In IEEE 2nd International Conference on Biometrics: Theory, Applications, and Systems*, BTAS '08, 2008, pp. 1—6.

18. **Kelkboom E. J. C., Zhou X., Breebaart J., Veldhuis R. N. S., Busch C.** Multialgorithm fusion with template protection, *In Proc. of the 3rd IEEE Int. Conf. on Biometrics: Theory, applications and systems (BTAS'09)*, 2009, pp. 1—7.

19. **Rathgeb C., Uhl A.** Iris-Biometric Fuzzy Commitment Schemes under Signal Degradation, ICISP'12, *In Proceedings of the 5th international conference on Image and Signal Processing*, pp. 217—225.

20. **Scotti F., Cimato S., Gamassi M., Piuri V., Sassi R.** Privacy-aware Biometrics: Design and Implementation of a Multimodal Verification System, *2008 Annual Computer Security Applications Conference, IEEE*, 2008, pp. 130—139.

21. **Santos M. F., Aguilar J. F., Garcia J. O.** Cryptographic key generation using handwritten signature, *Proceedings of SPIE, Orlando, Fla, USA*, Apr. 2006, 2006, vol. 6202, pp. 225—231.

22. **Vielhauer C., Steinmetz R., Mayerhöfer A.** Evaluating biometric encryption key generation, *In Proc. ICPR*, 2002, pp. 123—126.

23. **Yip K. W., Goh A., Ling D. N. C., Jin A. T. B.** Generation of replaceable cryptographic keys from dynamic handwritten signatures, *In Proc. ICB, Lecture Notes in Computer Science 3832, Springer*, 2006, pp. 509—515.

24. **Hao F., Chan C. W.** Private key generation from on-line handwritten signatures, *Information Management & Computer Security*, 2002, Is. 10, no. 2, pp. 159—164.

25. **Ushmaev O. S., Kuznecov V. V.** Algoritmy zashhishhennoj biometricheskoj verifikacii na osnove binarnogo predstavlenija topologii otpechatkov pal'cev [Algorithms for secure biometric verification based on the binary representation of the topology of fingerprints]. *Informatika i ee primenenija*, 2012, vol. 6, no. 1, pp. 132—140.

26. **Harin E. A., Goncharov S. M., Kornjushin P. N.** Postroenie sistem biometricheskoj autentifikacii s ispol'zovaniem generatorafkljuchevyh posledovatel'nostej na osnove nechetkih dannyh [Design and construction of biometric authentication using the key sequence generator based on fuzzy data], *Mater. 50-th Vseros. mezhvuz. nauch.-tehn. konf., Vladivostok, TOVMI, 2007*, pp. 112—115.

27. **Goncharov S. M., Borshevnikov A. E.** Nejrosetevoj preobrazovatel' "biometrija — kod dostupa" na osnove myslennogo pin-koda [Neural network converter "biometrics — access code" on the basis of mental pin-code], *Trudy nauchno-tehnicheskoj konferencii klastera penzenskih predprijatij, obespechivajushhih bezopasnost' informacionnyh tehnologij*, Penza, 2014, vol. 9, pp. 46—50.

28. **Medl' A., Shtefan Je., Mjuller R.** *Sposob i sistema dlja generirovanija nabora kljucha dostupa, a takzhe dlja autentifikacii cheloveka na osnove ego biometricheskogo parametra* [A method and system for generating a set of access key, and for authenticating a person based on his biometric], Patent RU № 2267159. 2005. Bulletin № 36.

29. **Efimov O. V., Ivanov A. I., Funtikov V. A.** Sposob identifikacii cheloveka po ego biometricheskomu obrazu [A method of identifying a person by his biometric images], Patent Ru № 2292079, 2005.

30. **GOST P 52633.0—2006.** Zashhita informacii. Tehnika zashhity informacii. Trebovanija k sredstvam vysokonadezhnoj biometricheskoj autentifikacii. [Protection of information. Security technique. Requirements for means of highly reliable biometric authentication]. Moscow, 2007.

31. **Chmora A. L., Urivskij A. V.** Biometricheskaja sistema autentifikacii, [Biometric authentication], *Patent Ru № 2316120*, 2008, Bulletin N 3.

32. **Ivanov A. I.** *Sposob bezopasnoj biometricheskoj autentifikacii* [Way to secure biometric authentication], Patent Ru № 2406143, 2010.

33. **Ivanov A. I.** *Nejrosetevaja zashhita konfidencial'nyh biometricheskih obrazov grazhdanina i ego lichnyh kriptograficheskih kljuchej* [Neural protection of sensitive biometric image of the citizen and his personal cryptographic keys], Penza, 2014, 57 p.

34. **Bezjaev A. V., Ivanov A. I., Funtikova Ju. V.** Optimizacija struktury samokorrektirujushhegosja bio-koda, hranjashhego sindromy oshibok v vide fragmentov hesh-funkcij [Optimization of structure of bio self-correcting code stored syndromes errors in the form of fragments of hash functions], *The Bulletin of the Ural Federal District. Security in the field of information*, 2014, no. 3 (13), pp. 4—14.

35. **GOST P 52633.5—2011.** Protection of information. Security technique. Automatic learning of neural network converters biometrics — access code: 01.12.2011, Moscow, Standartinform, 2012, 20 p.

36. **Sulavko A. E.** Programmnyj kompleks dlja bystrogo prototipirovanija sistem raspoznavanija obrazov [Software package for rapid prototyping systems, pattern recognition] *Tezisy II Vserossijskoj konferencii "Teorija i praktika Uspeha",* Omsk, 10—11 April 2014, pp. 21—22.

37. **Epifancev B. N., Lozhnikov P. S., Sulavko A. E.** Sravnenie algoritmov kompleksirovanija priznakov v zadachah raspoznavanija obrazov [Comparison of algorithms for aggregation features in pattern recognition problems] *Voprosy zashhity informacii*, 2012, no. 1, pp. 60—66.

УДК 004.75

Н. И. Червяков, д-р техн. наук, проф., зав. каф., e-mail: k-fmf-primath@stavsu.ru,
М. А. Дерябин, аспирант, мл. науч. сотр., e-mail: maxim.deryabin@gmail.com
ФГАОУ ВПО Северо-Кавказский федеральный университет, г. Ставрополь

Новый метод порогового разделения секрета, основанный на системе остаточных классов

Рассмотрены основные принципы построения схем разделения секрета на основе системы остаточных классов. На основе анализа абсолютной и вычислительной стойкости известных модулярных схем предложен новый подход к построению вычислительно-стойкой схемы разделения секрета, основанный на переводе секрета в систему остаточных классов. Его основной особенностью является малый размер долей секрета относительно размера разделяемой информации, что позволяет эффективно хранить и передавать информацию. Проведенный в работе анализ показал, что при выборе в качестве системы компактных последовательностей взаимно простых чисел вычислительная стойкость предлагаемой схемы приближается к вычислительной стойкости схемы Асмута—Блума.

Ключевые слова: *пороговые схемы разделения секрета, Китайская теорема об остатках, система остаточных классов, совершенные схемы*

Введение

Схемы разделения секрета (СРС) представляют собой протоколы распределения некоторой секретной информации (секрета) среди группы участников, каждый из которых получает свою долю секрета. При этом восстановление секрета возможно лишь в случае объединения долей секрета некоторой коалиции участников. Такую коалицию называют разрешенной коалицией. Использование СРС позволяет надежно скрыть информацию, распределив ее, например, в различные не связанные друг с другом хранилища. Одним из самых распространенных видов СРС являются пороговые схемы. В данных схемах секрет, разделенный среди n участников, восстановить могут только k и более участников, где порог k является важным параметром схемы и $2 < k \leqslant n$. Среди таких схем одной из самых известных является схема Шамира [1], получившая широкое распространение благодаря доказанной высокой стойкости.

Широко известны также СРС, основанные на Китайской теореме об остатках. Данные схемы базируются на модулярном представлении числа — в виде набора остатков от деления на заранее подобранные взаимно простые числа. Китайская теорема об остатках [2] позволяет восстанавливать позиционное значение на основе модулярного представления, обеспечивая алгоритмическую основу для восстановления секрета. К схемам такого типа относятся схемы Асмута—Блума [3] и Миньотта [4].

Однако важным вопросом при исследовании СРС является стойкость предложенного метода, которая понимается как вероятность восстановления секрета для неразрешенной коалиции участников. Понятие стойкости здесь используется с точки зрения устойчивости схемы к атакам различного рода. Основным видом атак в области пороговых СРС является попытка восстановления секрета группой участников, число которых не превышает заданный "порог" схемы. Для схем, основанных на Китайской теореме об остатках, важным результатом является введенная в работе [5] асимптотическая совершенность СРС. В асимптотически совершенной СРС количество получаемой информации при объединении долей секрета неразрешенной коалицией стремится к нулю при правильном выборе параметров схемы, основным из которых является набор модулей системы остаточных классов. В работах [5 и 6] показана асимптотическая совершенность схемы Асмута—Блума при достаточно близком расположении оснований системы. Однако описание схемы не будет полным без анализа ее вычислительной стойкости.

Важным свойством любой информационной системы является ее эффективность, которая полностью зависит от поставленных перед ней задач. Параметрами эффективности в данном контексте выступают скорость и точность системы, а также затрачиваемые ресурсы. Задачей проектировщиков является разработка системы таким образом, чтобы требуемая точность работы была достигаема за наименьшее время и с наименьшими затратами. Для СРС можно выделить два основных направления их применения. Первым является хранение или передача ключей, информации небольшого размера. В данном случае создаваемая система нечувствительна к ресурсам, а обеспечение стойкости является приоритетной задачей. Вторым направлением является потоковая передача информации и ее хранение, которые ограничены в плане ресурсов. Так как в данном случае количество обрабатываемой информации намного выше, то важными параметрами информационной системы для такого

случая являются скорость обработки и объем выходных данных. В первом случае для конкретного объема информации необходимо использовать совершенную СРС. При этом вероятность потери ключа будет равна вероятности его подбора. Во втором случае абсолютной безопасностью можно пренебречь в пользу скорости и повышения объема обрабатываемой информации. Основную роль здесь играет доказанная вычислительная стойкость используемой схемы. Вычислительная стойкость характеризуется сложностью раскрытия секрета, неразрешенной коалицией участников и оценивается вычислительной сложностью полного перебора элементов данного множества. Стойкость такого рода отражает стойкость схемы при практической реализации.

СРС находят дальнейшее развитие при реализации множественных схем разделения секрета [7], взвешенных схем [8] и верифицируемых схем [9]. Однако подробный анализ вычислительной стойкости данных схем ранее не проводили. В связи с этим чаще всего на практике используют вычислительно неэффективную схему Асмута—Блума, которая доказано является асимптотически совершенной [5, 6].

В данной работе предлагается новый подход к разделению секрета, основанный на переводе позиционного секрета в систему остаточных классов (СОК). Отличительной особенностью данного подхода является ориентация на эффективность схемы в условиях разделения большого объема информации. В разделе 1 приведены общие понятия, касающиеся СОК и СРС на ее основе. Кроме того, в нем более детально обсуждается вопрос об абсолютной и относительной стойкости схемы. В разделе 2 предложена новая схема разделения секрета на основе использования СОК и проведено исследование стойкости данной схемы при практической реализации в сравнении с классическими пороговыми схемами на СОК — схемами Асмута—Блума и Миньотта.

1. Схемы разделения секрета с использованием СОК и их свойства

Пусть каждый участник схемы имеет уникальный номер или идентификатор, все множество которых назовем универсальным множеством номеров и обозначим U (в простейшем случае $U = \{1, 2, ..., n\}$, где n — число участников схемы). Множество номеров разрешенной коалиции назовем разрешенным подмножеством множества U и обозначим I. Вместе с тем, неразрешенным подмножеством будем называть подмножество $\tilde{I}$ номеров участников любой коалиции, не имеющей права восстанавливать секрет. Областью определения секрета S при этом будем называть множество, объединяющее все возможные значения секрета s.

Система остаточных классов представляет собой одну из самых распространенных непозиционных систем счисления. Каждая конкретная СОК определяется системой взаимно простых оснований $\{p_1, p_2, ..., p_n\}$ и диапазоном представления $P = \prod_{i=1}^{n} p_i$. Позиционное число X такое, что $0 \leqslant X < P$, в этой системе представляется в виде кортежа из n чисел $(x_1, x_2, ..., x_n)$, полученных по формуле

$$x_i = X \bmod p_i,\ i = 1, 2, ..., n.$$

СОК имеет множество приложений, среди которых можно отметить ускорение операций за счет параллельной реализации базовых арифметических операций, контроль целостности информации, цифровая обработка сигналов. Хорошо разработанные методы обратного перевода чисел из СОК в позиционную систему вместе с эффективной реализацией вычисления остатка от деления делают данную систему подходящей для использования в качестве основы СРС [10—12].

В основе модулярного исчисления лежит Китайская теорема об остатках, согласно которой число X, $0 \leqslant X < P$, представленное остатками $(x_1, x_2, ..., x_n)$ в системе модулей $\{p_1, p_2, ..., p_n\}$, можно единственным образом вычислить по формуле

$$X = \left| \sum_{i=1}^{n} \left| P_i^{-1} \right|_{p_i} P_i x_i \right|_P,$$

где $P_i = \frac{P}{p_i}$, $\left| P_i^{-1} \right|_{p_i}$ — мультипликативная инверсия P_i по модулю p_i для $i = 1, 2, ..., n$.

Рассмотрим далее две СРС, основанные на СОК. На этапе генерирования параметров в схеме Асмута—Блума вначале выбирается число p_0, которое определяет множество всех возможных секретов. Произвольный секрет s следует выбирать так, что $s \in Z_{p_0}$. Далее система оснований $p_1 < p_2 < ... < p_k < p_{k+1} < ... < p_n$ подбирается так, что

$$\prod_{i=1}^{k} p_i > p_0 \prod_{i=0}^{k-2} p_{n-i},$$

последнее неравенство принято называть условием Асмута—Блума. На этапе разделения секрета генерируется случайное число r такое, что $s' = s + rp_0 < \prod_{i=1}^{k} p_i$. Секрет s такой, что делится таким образом, что $s_i = s' \bmod p_i$ есть доля секрета для каждого участника с номером i, где $i = 1, 2, ..., n$. В данном методе любое разрешенное множество участников с номерами из I может единственным образом восстановить секрет; при этом $|I| = m \geqslant k$. Вначале с помощью Ки-

тайской теоремы об остатках вычисляется позиционное число x на основе модулярного представления $(s_{i_1}, s_{i_2}, ..., s_{i_m})$ в СОК с основаниями $p_{i_1}, p_{i_2}, ..., p_{i_m}$, где $i_j \in I$ для всех $j = 1, 2, ..., m$. Исходный секрет восстанавливается как остаток от деления числа x на p_0: $s = x \bmod p_0$.

Для обсуждения вопроса стойкости схемы Асмута—Блума рассмотрим случай, когда объединены доли некоторой неразрешенной коалиции участников с номерами из $\tilde{I}$. Тогда $|\tilde{I}| \leqslant k - 1$, пусть $P = \prod_{i=1}^{k} p_i$ и $\tilde{P} = \prod_{i \in \tilde{I}} p_i$. Все, что будет известно в таком случае — это число $\tilde{s} = s' \bmod \tilde{P}$. Так как по условию Асмута—Блума $P/\tilde{P} > p_0$ и $(\tilde{P}, p_0) = 1$, то набор чисел $\tilde{s}$, таких что $\tilde{s} \equiv s' \bmod \tilde{P}$ и $\tilde{s} < P$, покрывает все классы вычетов по модулю m_0. Таким образом, как показано в работе [3], коалиция, объединяющая менее k долей секрета, не получает никакой полезной информации о секрете, что говорит о стойкости схемы. Однако серьезным ее недостатком является увеличение размера долей секрета относительно размера самого секрета, что приводит к существенному избытку выходной информации. Данного недостатка лишена следующая пороговая схема разделения секрета — (k, n)-пороговая схема Миньотта.

В схеме Миньотта система оснований $p_1 < p_2 < ... ... < p_k < p_{k+1} < ... < p_n$ подбирается как последовательность Миньотта, числа в которой удовлетворяют неравенству

$$\alpha = \prod_{i=0}^{k-2} p_{n-i} < \prod_{i=1}^{k} p_i = \beta.$$

Для достижения стойкости секрет s выбирается. При этом для произвольного секрета s из промежутка (α, β) числа $s_i = s \bmod p_i$ есть доли секрета для каждого участника с номером i, где $i = 1, 2, ..., n$. Восстановить секрет может любое разрешенное множество участников с номерами I, при этом $|I| = m \geqslant k$, секрет s вычисляется с использованием Китайской теоремы об остатках на основе числа $(s_{i_1}, s_{i_2}, ..., s_{i_m})$, представленного в СОК с основаниями $p_{i_1}, p_{i_2}, ..., p_{i_m}$, где $i_j \in I$ для всех $j = 1, 2, ..., m$.

Для того чтобы обеспечить достаточный уровень стойкости схемы необходимо использовать последовательности Миньотта с большим значением $\frac{\beta - \alpha}{\beta}$ [4]. Как показано в работе [6], схема Миньотта не является стойкой в абсолютном смысле, однако она нашла большое применение на практике за счет уменьшения объема выходных данных.

Ниже рассмотрим понятия, играющие важную роль в теории абсолютной стойкости схемы. Основываясь на данном аппарате, можно оценить энтропию для схем разделения секрета. Информационную энтропию в условии отсутствия информации о долях секрета обозначим как $H(s \in S)$, где s есть секрет, распределенный в области определения секрета S. В данном случае энтропия максимальна, так как рассматривается только общедоступная информация. Энтропию секрета при условии, что известны доли секрета от участников, чьи номера принадлежат некоторому множеству $I \subseteq U$, обозначим как $H(s \in S \mid s_i : i \in I)$. В случае, если I представляет собой множество номеров разрешенного подмножества участников, то $H(s \in S \mid s_i : i \in I) = 0$, так как в данном случае секрет должен быть правильно восстановлен. Для обратного случая, в котором объединяется неразрешенное множество участников, важно добиться максимума условной энтропии. Данное условие приводит к формулировке условия совершенности схемы на основе вероятностного подхода. При этом важнейшим понятием теории СРС является убыль неопределенности схемы, под которой понимается величина

$$\Delta(s_i : i \in I) = H(s \in S) - H(s \in S | s_i : i \in I).$$

Убыль неопределенности в случае, если I является множеством номеров разрешенного подмножества участников, равна безусловной энтропии области определения секрета: $\Delta(s_i : i \in I) = H(s \in S)$. Пусть теперь множество участников является неразрешенным. СРС является *совершенной*, если для всех неразрешенных подмножеств участников с номерами $\tilde{I}$ следует

$$\Delta(s_i : i \in \tilde{I}) = 0.$$

Однако как было отмечено ранее, для анализа стойкости СРС, основанных на СОК, было введено дополнительное понятие, позволяющее исследовать свойства СРС данного типа общепринятыми методами [5]. Поэтому в данном случае было введено понятие *асимптотически совершенной СРС*. Схема является асимптотически совершенной, если для всех неразрешенных подмножеств участников с номерами $\tilde{I}$ и для любого $\varepsilon > 0$ существует такое p, что при условии выбора на этапе генерации параметров СРС системы оснований $p_0 < p_1 < ... < p_n$ так, что $p_i > p$ $(i = 0, 1, ..., n)$ для нее следует

$$\Delta(s_i : i \in \tilde{I}) < \varepsilon.$$

Открытым остается вопрос о том, как именно необходимо подбирать параметры СРС на СОК, чтобы она обладала свойством асимптотической совершенности. В работе [5] показана асимптотическая совершенность схемы Асмута—Блума при

использовании "достаточно близко расположенных" взаимно простых чисел в качестве системы оснований СОК. В работе [6] в качестве системы модулей предлагается использовать так называемые компактные последовательности взаимно простых чисел.

Согласно [6] последовательность $p_0 < p_1 < ... < p_n$ называется компактной последовательностью взаимно простых чисел с начальным значением p_0, если $p_n < p_0 + p_0^{\theta}$ для некоторого действительного числа $\theta \in (0, 1)$.

Рассмотрим структуру компактной последовательности на конкретном примере. Пусть $p_0 = 20$. Возьмем несколько взаимно простых с ним чисел: $p_1 = 21$, $p_2 = 29$ и $p_3 = 43$. Вычислим полученное значение θ исходя из условия $p_n = p_0 + p_0^{\theta}$. Тогда $\theta = \log p_0(p_n - p_0)$ и в нашем случае $\theta \approx 1{,}015$. Очевидно, что выбранная последовательность 20, 21, 29, 43 компактной не является. Заменим в ней число $p_3 = 43$ на $p_3 = 31$. Для полученной последовательности 20, 21, 29, 31 значение $\theta = 0{,}756$, откуда следует, что она является компактной.

Компактные последовательности взаимно простых чисел играют важную роль в исследовании стойкости СРС. Использование данных последовательностей в качестве системы оснований СОК позволяет строить эффективные и стойкие схемы, что обусловлено близостью чисел на числовой прямой. В работе [6] показано, что схема Асмута—Блума является асимптотически совершенной при использовании компактной последовательности в качестве системы оснований СОК. Стоит отметить, что схема Миньотта асимптотически совершенной не является. Однако исследование стойкости не будет полным без анализа вычислительной стойкости методов. В дальнейшем будем полагать, что в качестве системы оснований использована именно компактная последовательность взаимно простых чисел.

Обратимся далее к понятию вычислительной стойкости схемы. Пусть в некоторый момент времени некоторому аналитику удалось собрать доли неразрешенного подмножества участников с номерами $\tilde{I}$ для некоторой конкретной СРС. Задача аналитика в таком случае состоит в том, чтобы восстановить секрет на основе имеющихся данных. В реальной ситуации множество S можно разделить на два подмножества. Первое подмножество S_1 будет состоять из всех вариантов секрета, которые не подходят на роль секрета при известных данных. Второе подмножество S_2 будет содержать все оставшиеся возможные варианты секрета. Например, если в схеме Миньотта известна доля секрета s_j для модуля p_j, $0 \leqslant j \leqslant n$, то секрет обязательно должен удовлетворять условию: $s \equiv s_j \bmod p_j$. Следовательно, в данном случае $S_1 = \{s : s \in S \wedge s \not\equiv s_j \bmod p_j\}$ и $S_2 = \{s : s \in S \wedge s \equiv s_j \bmod p_j\}$. Отметим, что в случае, если СРС совершенна, то для нее $S_1 = \varnothing$ и $S_2 = S$.

Таким образом, для того чтобы найти исходный секрет, необходимо перебрать все варианты, входящие в S_2, и стойкость схемы зависит от мощности этого множества и от вычислительной сложности полного перебора. Необходимо генерировать параметры схемы так, чтобы аналитик не смог с использованием современных вычислительных ресурсов подобрать секрет за приемлемое время. Схему, отвечающую данным условиям, будем называть *вычислительно стойкой*.

В качестве меры вычислительной стойкости примем мощность множества S_2: $f(\tilde{I}) = |S_2|$. Для схемы Асмута—Блума, принимая в расчет ее асимптотическую совершенность и условие Асмута—Блума, $f(\tilde{I}) = |S| = |Z_{p_0}| = p_0$ для любого $\tilde{I}$.

Легко видеть, что не все абсолютно стойкие схемы являются вычислительно стойкими, но вместе с тем вычислительно стойкие схемы не всегда совершенны. Наиболее важной в практическом смысле является именно вычислительная стойкость.

В данном пункте нами введены основные понятия, используемые при построении СРС, основанных на СОК. Рассмотрены наиболее известные СРС на СОК — схема Асмута—Блума и схема Миньотта. Приведены основные подходы, применяемые для анализа стойкости СРС и их расширение для СРС на СОК. Далее рассматривается предлагаемая схема, основанная на использовании СОК, обладающая похожей стойкостью, по способная разделять секрет гораздо большего размера.

2. Предлагаемая схема разделения секрета, основанная на использовании СОК

В схемах Асмута—Блума и Миньотта на область секрета S накладываются существенные ограничения. В первом случае она равна Z_{p_0}, во втором случае существенно усекается за счет исключения из диапазона достаточно большого промежутка чисел. Рассмотрим теперь схему разделения секрета, основанную на простом переводе секрета из области $S = [0, P)$ в избыточную систему остаточных классов с основаниями $p_1, p_2, ..., p_n$. Базовым преимуществом данной схемы является существенная экономия объема информации при ее передаче и хранении. Однако предлагаемая схема, очевидно, не является совершенной, так как при условии раскрытия части долей секрета энтропия существенно уменьшается.

Предлагаемая (k, n)-пороговая схема разделения секрета, основанная на использовани СОК.

Генерация параметров. Подбирается компактная последовательность взаимно простых чисел $p_1 < p_2 < ... < p_k < p_{k+1} < ... < p_n$, при этом $p_n < p_1 + p_1^{\theta}$, где $\theta \in (0, 1)$; секрет s выбирается из промежутка $(0, P)$,

где P представляет собой рабочий диапазон СОК и определяется согласно выражению

$$P = \prod_{i=1}^{k} p_i,$$

доли секрета s_i имеют множества представления Z_{p_i} для участника с номером i, где $i = 1, 2, ..., n$.

Разделение секрета. Доли секрета s вычисляются так, что $s_i = s \bmod p_i$ есть доля секрета для каждого участника с номером i, где $i = 1, 2, ..., n$.

Восстановление секрета. Любое разрешенное множество участников с номерами I может единственным образом восстановить секрет, при этом $|I| = m \geqslant k$; секрет s вычисляется с использованием Китайской теоремы об остатках на основе числа $(s_{i_1}, s_{i_2}, ..., s_{i_m})$, представленного в СОК с основаниями $p_{i_1}, p_{i_2}, ..., p_{i_m}$, где $i_j \in I$ для всех $j = 1, 2, ..., m$.

Рассмотрим далее основные свойства предлагаемой схемы. Ключевой характеристикой схемы является ее вычислительная стойкость, которая напрямую зависит от значений используемых оснований. Следующее утверждение позволяет установить нижнюю границу для выбора оснований.

Утверждение 1. В предлагаемой схеме для любого неразрешенного подмножества участников с номерами $\tilde{I}$ следует, что $P > \tilde{P} = \prod_{i \in \tilde{I}} p_i$ при $p_1 > 2^{k-1}$.

Доказательство. Используя тот факт, что в пороговых СРС максимальным неразрешенным подмножеством является подмножество с номерами n, $n - 1$, ..., $n - k + 2$ и учитывая определение компактных последовательностей, получим

$$\tilde{P} < \prod_{i=0}^{k-2} p_{n-i} < (p_1 + p_1^{\theta})^{k-1} =$$

$$= p_1^{k-1}(1 + p_1^{\theta-1})^{k-1} < p_1^{k-1} 2^{k-1}.$$

Вместе с тем $P > p_1^k$. Отсюда

$$\frac{P}{\tilde{P}} > \frac{p_1^k}{p_1^{k-1} 2^{k-1}} = \frac{p_1}{2^{k-1}}.$$

Для соблюдения условия $P > \tilde{P}$ необходимо выполнение неравенства $P/\tilde{P} > 1$. Данное неравенство обязательно будет выполнено в случае, если выполнится неравенство $\frac{p_1}{2^{k-1}} > 1$, или, что равносильно, $p_1 > 2^{k-1}$. *Утверждение доказано.*

Иными словами, предлагаемая схема применима при выборе модуля p_1 на этапе генерирования параметров таким, что $p_1 > 2^{k-1}$.

Утверждение 2. Для предлагаемой схемы при объединении долей неразрешенного подмножества участников с номерами $\tilde{I}$ мощность множества перебора определяется выражением

$$f(\tilde{I}) = \left\lfloor \frac{P}{\tilde{\tilde{P}}} \right\rfloor,$$

где $\tilde{\tilde{P}} = \prod_{i \in \tilde{I}} p_i$.

Доказательство. Так как доли участников с номерами, принадлежащими $\tilde{I}$, известны, то на основе СОК, основаниями которой являются все числа p_{i_j} такие, что $i_j \in \tilde{I}$, можно восстановить число $\tilde{s} \equiv s \bmod \tilde{P}$. На основе последнего сравнения $s = a\tilde{P} + \tilde{s}$. Единственным неизвестным параметром секрета остается a. Именно множество всех возможных чисел a необходимо перебирать. Однако s определено в СОК с диапазоном P, следовательно, $0 \leqslant a\tilde{P} + \tilde{s} < P$, откуда

$$-\frac{\tilde{s}}{\tilde{\tilde{P}}} \leqslant a < \frac{P - \tilde{s}}{\tilde{\tilde{P}}}.$$

Принимая во внимание, что число s, а вместе с ним и число a, является целым неотрицательным числом, и так как $\tilde{s} < \tilde{\tilde{P}}$, получим

$$0 = \left\lceil -\frac{\tilde{s}}{\tilde{\tilde{P}}} \right\rceil \leqslant a \leqslant \left\lfloor \frac{P - \tilde{s}}{\tilde{\tilde{P}}} \right\rfloor = \left\lfloor \frac{P}{\tilde{\tilde{P}}} \right\rfloor.$$

То есть число a лежит в промежутке $\left[0, \left\lfloor \frac{P}{\tilde{\tilde{P}}} \right\rfloor\right]$, мощность которого равна

$$f(\tilde{I}) = \left\lfloor \frac{P}{\tilde{\tilde{P}}} \right\rfloor + 1.$$

Утверждение доказано.

Исследуем далее вопрос о том, как связаны мощности множеств перебора схемы Асмута—Блума и предлагаемой схемы при одинаковом выборе параметров СОК. Пусть схема Асмута—Блума определяется набором оснований $p_0 < p_1 < p_2 < ... < p_k < p_{k+1} < ... < p_n$. Для обеспечения асимптотической идеальности СРС Асмута—Блума потребуем, чтобы данная последовательность была компактной с начальным значением p_0, или $p_n < p_0 + p_0^{\theta}$ для $\theta \in (0, 1)$. При этом последовательность $p_1 < ... < p_n$ будет компактной с начальным значением p_1. Данную СОК будем использовать в качестве основы для предлагаемой СРС.

Мощность множества перебора Асмута—Блума константна и равна p_0. Обозначим $f(\tilde{I})$ мощность

множества перебора предлагаемой схемы. Значение $f(\tilde{I})$ можно получить из утверждения 2:

$$f(\tilde{I}) = \left\lfloor \frac{P}{\tilde{P}} \right\rfloor + 1.$$

Пусть далее $\tilde{I}_{\max}$ есть множество номеров неразрешенного подмножества с наибольшим диапазоном. Тогда, как было сказано ранее,

$$\tilde{P}_{\max} = \prod_{i=0}^{k-2} p_{n-i}.$$

Установим связь между значением $f(\tilde{I}_{\max})$ и k. Построим две СРС, в одной из которых возьмем в качестве порога k, в другой — $n - k + 1$. При этом положим, что $2 \leqslant k \leqslant \left\lfloor \frac{n}{2} \right\rfloor$. Рассчитаем $f(\tilde{I}_{\max})$ для второй СРС:

$$\frac{\prod_{i=1}^{n-k+1} p_i}{\prod_{i=0}^{n-k-1} p_{n-i}} = \frac{p_1 p_2 \cdots p_k p_{k+1} \cdots p_{n-k+1}}{p_{k+1} p_{k+2} \cdots p_{n-k+1} p_{n-k+2} \cdots p_n} =$$

$$= \frac{p_1 p_2 \cdots p_k}{p_{n-k+2} \cdots p_n} = \frac{\prod_{i=1}^{k} p_i}{\prod_{i=0}^{k-2} p_{n-i}}.$$

Выражение справа представляет собой значение $f(\tilde{I}_{\max})$ для первой СРС. Следовательно, значения $f(\tilde{I}_{\max})$ симметричны по k относительно значения $\left\lfloor \frac{n}{2} \right\rfloor$. Пусть теперь $2 \leqslant k < \left\lfloor \frac{n}{2} \right\rfloor$ и $k_1 = k + 1$. Пусть значение f_{k_1} равно значению $f(\tilde{I}_{\max})$ для СРС с порогом k_1. Оценим величину f_{k_1}:

$$f_{k_1} = \frac{\prod_{i=1}^{k_1} p_i}{\prod_{i=0}^{k_1-2} p_{n-i}} = \frac{\prod_{i=1}^{k+1} p_i}{\prod_{i=0}^{k-1} p_{n-i}} =$$

$$= \frac{p_{n-k+2}}{p_{k+1}} \frac{\prod_{i=1}^{k} p_i}{\prod_{i=0}^{k-2} p_{n-i}} = \frac{p_{n-k+2}}{p_{k+1}} f_k,$$

где f_k представляет собой значение $f(\tilde{I}_{\max})$ для СРС с порогом k. Так как $2 \leqslant k < \left\lfloor \frac{n}{2} \right\rfloor$, то $n - k + 2 > \left\lfloor \frac{n}{2} \right\rfloor$, следовательно, $n - k + 2 > k + 1$, учитывая ограничения, накладываемые на модули СОК, получим, что $p_{n-k+2} > p_{k+1}$, следовательно, $\frac{p_{n-k+2}}{p_{k+1}} > 1$. Учитывая сделанные ранее рассуждения, получим, что $f_{k_1} < f_k$. Иными словами, наихудшим случаем, в котором $f(\tilde{I}_{\max})$ принимает наименьшее значение, является случай $k = \left\lfloor \frac{n}{2} \right\rfloor$. Ввиду симметричности $f(\tilde{I}_{\max})$ относительно данного значения целесообразно рассматривать k в границах $\left[2, \frac{n}{2}\right]$, так как для промежутка $\left[\frac{n}{2}, n-1\right]$ рассуждения ведутся похожим образом. Особым случаем является СРС, в которой $k = n$. Для такой СРС $f(\tilde{I}_{\max}) = p_1$.

Докажем далее ряд важных утверждений, позволяющих достаточно точно оценить величину $f(\tilde{I}_{\max})$.

Утверждение 3. Для любой последовательности $p_1 < p_2 < \ldots < p_n$ следует, что

$$\frac{P}{\tilde{P}_{\max}} \leqslant p_1. \qquad (1)$$

Доказательство. Рассматривая $\tilde{P}_{\max}$, заметим, что так как последовательность p_i возрастает, то $p_k \leqslant p_n$, $p_{k-1} \leqslant p_{n-1}$, ..., $p_2 \leqslant p_{n-k+2}$. При этом равенства $p_i = p_{n-i+2}$, $i = 2, 3, \ldots, k$, достигаются только при $k = n$. Это означает, что

$$P' = \prod_{i=2}^{k} p_i \leqslant \tilde{P}_{\max}.$$

Так как $P = p_1 P'$, то $\frac{P}{\tilde{P}_{\max}} \leqslant p_1$. *Утверждение доказано.*

Выражение (1) показывает верхнюю границу для $f(\tilde{I}_{\max})$. Для оценки нижней границы $f(\tilde{I}_{\max})$ докажем следующее утверждение.

Утверждение 4. Для любой последовательности $p_1 < p_2 < \ldots < p_n$, такой, что $p_n < p_1 + p_1^{\theta}$, $0 < \theta < 1$, и любого k, такого, что $2 \leqslant k < \left\lfloor \frac{n}{2} \right\rfloor$, следует, что

$$\frac{P}{p_i \tilde{P}_{\max}} > \left(\frac{1}{1 + p_1^{\theta - 1}} \right)^{k-1}. \qquad (2)$$

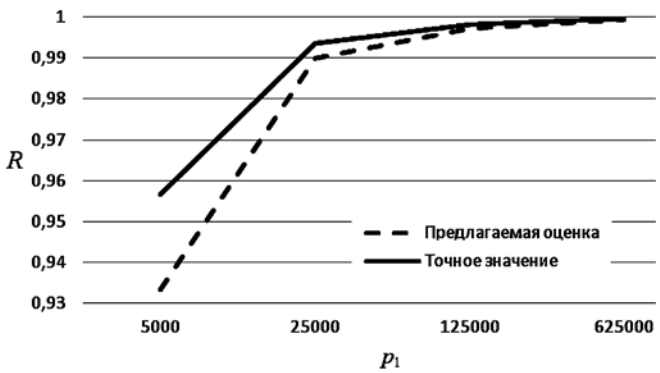
Доказательство. Так как последовательность компактна с начальным значением p_1, то $\tilde{P}_{\max} < (p_1 + p_1^{\theta})^{k-1}$. Вместе с тем, так как последовательность возрастает, то $P > p_1^k$. Следовательно

$$\frac{P}{\tilde{P}_{\max}} > \frac{p_1^k}{(p_1 + p_1^{\theta})^{k-1}} = p_1\left(\frac{p_1}{p_1 + p_1^{\theta}}\right)^{k-1} = p_1\left(\frac{1}{1 + p_1^{\theta-1}}\right)^{k-1},$$

откуда следует неравенство (2).

Утверждение доказано.

Основываясь на утверждениях 3 и 4, можно достаточно точно определить границы для величины $\frac{P}{\tilde{P}_{\max}}$, которые напрямую зависят от значения p_1. Рассмотрим далее конкретный пример, позволяющий увидеть, как быстро величина $\frac{P}{\tilde{P}_{\max}}$ сходится к p_1. На рисунке изображен график изменения величины $R = \frac{P}{p_1 \tilde{P}_{\max}}$ при выборе в качестве системы оснований СОК минимальных компактных последовательностей для различных значений p_1. Из графика видно, что при увеличении p_1 значение R приближается к 1 и, следовательно, $\frac{P}{\tilde{P}_{\max}}$ приближается к p_1. При этом предлагаемая оценка (2) позволяет оцепить величину R снизу. Утверждения 3 и 4 позволяют оценить степень близости $\frac{P}{\tilde{P}_{\max}}$ к p_1 в зависимости от заданных p_1, θ, k и n до этапа генерирования самой последовательности.



Оценка величины $R = \frac{P}{p_1 \tilde{P}_{\max}}$ в зависимости от значения p_1 при $n = 15$, $k = 7$

Доказанное утверждение 4 имеет важное значение для оценки вычислительной стойкости рассматриваемых пороговых схем. При фиксированных θ и k значение $\frac{P}{\tilde{P}_{\max}}$ находится в следующих границах:

$$p_1\left(\frac{1}{1 + p_1^{\theta-1}}\right)^{k-1} 1 < \frac{P}{\tilde{P}_{\max}} \leqslant p_1. \quad (3)$$

При этом легко показать, что для фиксированных $0 < \theta < 1$ и $2 \leqslant k \leqslant n$

$$\lim_{p_1 \to \infty}\left(\frac{1}{1 + p_1^{\theta-1}}\right)^{k-1} = 1 \text{ и } \frac{1}{1 + p_1^{\theta-1}} < 1.$$

Следовательно

$$\left(\frac{1}{1 + p_1^{\theta-1}}\right)^{k-1} = (1 - \varepsilon),$$

где $0 < \varepsilon < 1$. Причем, чем больше p_1, тем ближе ε к 0. Тогда из неравенства (3) следует

$$p_1(1 - \varepsilon) < \frac{P}{\tilde{P}_{\max}} \leqslant p_1.$$

На основании данного выражения можно получить следующую оценку для $f(\tilde{I}_{\max})$:

$$p_1 - [\varepsilon p_1] < f(\tilde{I}_{\max}) - 1 \leqslant p_1. \quad (4)$$

Последнее неравенство позволяет определить степень близости величины $f(\tilde{I}_{\max})$ к p_1 без генерации самой последовательности. Так как $p_1 > p_0$, то за счет всех ограничений, накладываемых на $\frac{P}{\tilde{P}_{\max}}$, получим, что с возрастанием числа p_0 мощность множества перебора предлагаемой схемы $f(\tilde{I}_{\max})$ будет приближаться к числу p_1. Можно сделать вывод, что мощность множества перебора для предлагаемой схемы разделения секрета при выборе достаточно больших модулей эквивалентна мощности множества перебора схемы Асмута— Блума, которая равна p_0.

Сопоставим теперь предлагаемую схему со схемой Миньотта. Базовым конструктивным требованием схемы Миньотта является включение секрета s в промежуток $\left(\alpha = \prod_{i=0}^{k-2} p_{n-i}, \beta = \prod_{i=1}^{k} p_i\right)$. Доказанные ранее утверждения относительно значения мно-

жества перебора предлагаемой схемы позволяют отойти от данного правила в пользу увеличения общего диапазона представления секрета. Основываясь на предположении равномерного распределения секрета в промежутке [0, P), компактности множества $p_0 < p_1 < ... < p_n$ и достаточно большом значении p_0, легко показать, что вероятность попадания секрета в промежуток [0, α) приближается к вероятности "угадать" произвольный секрет в схеме Асмута—Блума. Отметим, что в используемых обозначениях $\alpha = \widetilde{P}_{\max}$ и $\beta = P$. Действительно, область определения секрета в схеме Асмута—Блума равна Z_{p_0} и определяется значением p_0. При равномерном распределении секрета на данном множестве вероятность выбора произвольного секрета равна $1/p_0$. Вместе с тем, вероятность попадания числа в промежуток [0, α) = [0, $\widetilde{P}_{\max}$) равна $\frac{|[0, P_{\max})|}{|[0, P)|} = \frac{\widetilde{P}_{\max}}{P}$. Но согласно утверждению 3, при достаточно большом p_0 величина $\frac{P}{\widetilde{P}_{\max}}$ эквивалентна величине p_0.

Отсюда следует, что выбор параметров, определяющих предлагаемую СРС, позволяет уйти от ограничений, накладываемых на параметры схемы Миньотта. Равномерность распределения секрета может быть достигнута за счет предварительного шифрования информации. Рассмотрим далее примеры генерации параметров предлагаемой СРС на основе использования СОК.

Пример. Пусть $p_1 = 1024$, $n = 10$ и $k = 5$ и требуется, чтобы отклонение от мощности перебора Асмута—Блума не превышало 10 %. Определим, каким должно быть θ в данном случае. Согласно оценкам (3) и (4) получим

$$\theta < \log_{p_1}\left(\frac{1}{\sqrt[k-1]{1-\varepsilon}} - 1\right) + 1,$$

где ε есть требуемое отклонение. В нашем случае $\varepsilon = 0{,}05$. Подставив в формулу имеющиеся данные, получим $\theta < 0{,}477$. Следовательно, числа последовательности, обеспечивающей требуемую мощность множества перебора, должны попадать в промежуток [1024, 1051). Для данной СРС $f(\widetilde{I}_{\max}) > 921$. *Конец примера.*

Стоит отметить, что доказательство возможности генерирования компактных последовательностей на основе заданного начального значения является довольно сложной теоретико-числовой задачей. При этом, чем меньше у числа p_1 уникальных делителей, тем выгоднее его использовать в качестве основы для такой последовательности, так как при этом снижается количество чисел в промежутке от p_1 до $2p_1$, невзаимно простых с p_1. Генерирование компактных последовательностей есть предмет дальнейших исследований. На данный момент можно ограничиться лишь практическими рекомендациями, заключающимися в выборе в качестве p_1 числа достаточно большой величины и с наименьшим количеством делителей.

В данном пункте рассмотрена предлагаемая схема разделения секрета, основанная на использовании СОК. Утверждения 1—5 позволяют оценить качества каждой конкретной СРС. Во-первых, согласно утверждению 1, $p_1 > 2^{k-1}$. Таким образом, зафиксирована граница, начиная с которой, необходимо генерировать последовательность оснований. Отметим, что p_1 должно быть существенно больше числа 2^{k-1} для обеспечения максимальной вычислительной стойкости. Во-вторых, важным параметром схемы является величина θ, определяющая компактную последовательность. Чем ближе θ к нулю, тем лучше свойства СРС в плане вычислительной стойкости, что следует из утверждения 4 и неравенства (3).

Заключение

Важным свойством схемы является минимальная избыточность информации в процессе разделения секрета. Для СРС Асмута—Блума количество информации, получаемой в результате вычисления долей секрета, существенно превосходит количество информации, которое несет в себе сам секрет. Вместе с тем, предлагаемая схема способна разделить секрет величины P, при этом области определения долей секрета в свою очередь равны p_i для участника с номером i. Очевидно, что каждый участник получает доли гораздо меньшего размера, чем исходный секрет, в отличие от СРС Асмута—Блума. Оценка (4) позволяет сделать вывод, что при одинаковом выборе параметров СОК для предложенной схемы и схемы Асмута—Блума, их вычислительная стойкость совпадает. Важными условиями для генерирования системы оснований СОК являются выбор p_1 так, что $p_1 > 2^{k-1}$, и выбор θ как можно ближе к 0. Наиболее подходящим применением предложенной СРС является использование ее для хранения или передачи больших объемов информации, так как она ориентирована на многократное использование выбранных параметров. Одним из перспективных направлений будущей работы является разработка вычислительно эффективного метода генерирования набора оснований СОК по заранее заданным p_1, θ и k.

Работа выполнена при поддержке базовой части государственного задания СКФУ №2563.

Список литературы

1. **Shamir A.** How to share a secret // Communications of the ACM. 1979. N. 22 (11). P. 612—613.

2. **Червяков Н. И., Галушкин А. И., Евдокимов А. А., Лавриненко А. В., Лавриненко И. Н.** Применение искусственных нейронных сетей и системы остаточных классов в криптографии. М.: Физматлит, 2012. 280 с.

3. **Asmuth C. A., Bloom J.** A Modular Approach to Key Safeguarding // IEEE Transactions on Information Theory. 1983. 29 (2). P. 208—210.

4. **Mignotte M.** How to Share a Secret // Lecture Notes in Computer Science. 1983. Vol. 149. P. 371—375.

5. **Quisquater M., Preneel B., Vandewalle J.** On the Security of the Threshold Scheme Based on the Chinese Remainder Theorem. // Lecture Notes in Computer Science. 2002. Vol. 2274. P. 199—210.

6. **Barzu M., Tiplea F. L., Dragan C. C.** Compact sequences of co-primes and their applications to the security of CRT-based threshold schemes // Information Sciences. 2013. N. 240. P. 161—172.

7. **Hsu C.-F., Harn L.** Multipartite Secret Sharing Based on CRT // Wireless Personal Communications. 2014. Vol. 78 (1). P. 271—282.

8. **Harn L., Fuyou M.** Weighted secret sharing based on the Chinese remainder theorem // International Journal of Network Security. 2007. Vol. 16 (6). P. 420—426.

9. **Liu Y., Harn L., Chang C.-C.** A novel verifiable secret sharing mechanism using theory of numbers and a method for sharing secrets // International Journal of Communication Systems. 2014. N. 28 (7). P. 1282—1292.

10. **Червяков Н. И.** Ускоренный алгоритм определения позиционных характеристик и его нейросетевая реализация // Нейрокомпьютеры: разработка, применение. 2001. № 10. С. 19—21.

11. **Червяков Н. И., Бабенко М. Г., Ляхов П. А., Лавриненко И. Н.** Эффективный алгоритм точного определения универсальной позиционной характеристики модулярных чисел и его применение для вычисления основных проблемных операций в системе остаточных классов // Инфокоммуникационные технологии. 2014. Т. 12. № 1. С. 4—18.

12. **Chervyakov N. I., Averbukh V. M., Babenko M. G., Lyakhov P. A., Gladkov A. V., Gapochkin A. V.** Approximate method of implementation non-modular operations in the residue number system // Fundamental research. 2012. N. 6. P. 189—193.

N. I. Chervyakov, Professor, Chief of Department
of Applied Mathematics and Mathematical Modeling, k-fmf-primath@stavsu.ru,
M. A. Deryabin, Postgraduate Student, Junior Researcher
of Department of Organization of Scientific Research, maxim.deryabin@gmail.com
North Caucasus Federal University, Stavropol

New Method of Threshold Secret Sharing Based on Residue Number System

This paper researches basic principle of constructing secret sharing schemes based on residue number system (RNS). Perfect and computational secrecy analysis of existing secret sharing schemes based on RNS is followed by a new approach to constructing a secret sharing scheme based on conversion of the secret to RNS. The main feature of the proposed method is the fact that the size of each share is small relatively to the size of the secret. Mentioned above feature results in effective data storage and transfer. Performed analysis has shown that using compact sequences of co-prime numbers as moduli set leads to computational secrecy of proposed scheme being close to the computational secrecy of Asmuth—Bloom scheme.

Keywords: *threshold secret sharing schemes, Chinese reminder theorem, residue number system, perfect schemes*

References

1. **Shamir A.** How to share a secrety, *Communications of the ACM*, 1979, no. 22 (11), pp. 612—613.

2. **Chervyakov N. I., Galushkin A. I., Evdokimov A. A., Lavrinenko A. V., Lavrinenko I. N.** *Primenenie iskusstvennyh nejronnyh setej i sistemy ostatochnyh klassov v kriptografii*, Moscow, Fizimatlit, 2012. 280 p.

3. **Asmuth C. A., Bloom J.** A Modular Approach to Key Safeguarding, *IEEE Transactions on Information Theory*, 1983, 29 (2), pp. 208—210.

4. **Mignotte M.** How to Share a Secret, *Lecture Notes in Computer Science*, 1983, vol. 149, pp. 371—375.

5. **Quisquater M., Preneel B., Vandewalle J.** On the Security of the Threshold Scheme Based on the Chinese Remainder Theorem, *Lecture Notes in Computer Science*, 2002, vol. 2274, pp. 199—210.

6. **Barzu M., Tiplea F. L., Dragan C. C.** Compact sequences of co-primes and their applications to the security of CRT-based threshold schemes, *Information Sciences*, 2013, no. 240, pp. 161—172.

7. **Hsu C.-F., Harn L.** Multipartite Secret Sharing Based on CRT, *Wireless Personal Communications*, 2014, vol. 78 (1), pp. 271—282.

8. **Harn L., Fuyou M.** Weighted secret sharing based on the Chinese remainder theorem, *International Journal of Network Security*, 2007, vol. 16 (6), pp. 420— 426.

9. **Liu Y., Harn L., Chang C.-C.** A novel verifiable secret sharing mechanism using theory of numbers and a method for sharing secrets, *International Journal of Communication Systems*, 2014, 28 (7), pp. 1282—1292.

10. **Chervyakov N. I.** Uskorennyj algoritm opredelenija pozicionnyh harakteristik i ego nejrosetevaja realizacija, *Nejrokomp'jutery: razrabotka, primenenie*, 2001, no. 10, pp. 19—21.

11. **Chervjakov N. I., Babenko M. G., Ljahov P. A., Lavrinenko I. N.** Jeffektivnyj algoritm tochnogo opredelenija universal'noj pozicionnoj harakteristiki moduljarnyh chisel i ego primenenie dlja vychislenija osnovnyh problemnyh operacij v sisteme ostatochnyh klassov, *Infokommunikacionnye tehnologii*, 2014, vol. 12, no. 1, pp. 4—18.

12. **Chervyakov N. I., Averbukh V. M., Babenko M. G., Lyakhov P. A., Gladkov A. V., Gapochkin A. V.** Approximate method of implementation non-modular operations in the residue number system, *Fundamental research*, 2012, no. 6, pp. 189—193.

БАЗЫ ДАННЫХ
DATABASE

УДК 004.02: 004.312

В. Н. Жураковский, канд. техн. наук, доц.,
С. И. Силин, канд. техн. наук, ассистент, e-mail: sm2-2@inbox.ru,
Московский государственный технический университет имени Н. Э. Баумана

Выбор алгоритмов обработки данных в современной вычислительной технике на основе системного анализа

Рассмотрены проблемы выбора алгоритмической реализации параллельной обработки данных на базе ПЛИС. Рассмотрены различные параметры, которые влияют на возможность реализации алгоритма в ПЛИС. Показано, что исследование и оптимизацию процесса разработки целесообразно проводить с помощью методики и средств системного анализа. Проведена формализация критериев выбора формы реализации на основе приведенных показателей.

Ключевые слова: *ПЛИС, системный анализ, алгоритм, параллельная обработка сигналов, радиолокация, встроенные системы, системы на кристаллах, микроэлектроника, схемотехника, системотехника, проектирование систем*

Введение

К бортовым встраиваемым системам предъявляется ряд требований по массогабаритным характеристикам, энергопотреблению, стоимости, удобству настройки, снижению зависимости параметров алгоритмов от характеристик используемой элементной базы. В связи с этим в последнее время стало широко распространено применение цифровых программируемых микросхем, таких как цифровые сигнальные процессоры (ЦСП) и программируемые логические интегральные схемы (ПЛИС).

В данной статье речь пойдет о проблемах, возникающих при использовании ПЛИС [1]. Эти микросхемы применяются все более и более часто в большом числе приложений. Подобный рост популярности связан с их высоким быстродействием, массовым параллелизмом, простотой отработки проектов, возможностью изменения и настройки параметров алгоритмов в зависимости от конкретных условий их применения. Также в последнее время появилась технология, позволяющая переносить проекты, разработанные для ПЛИС, в заказные интегральные схемы с жесткой логикой без изменения самого проекта, что позволяет упростить и удешевить процесс запуска серийного производства изделий.

Особенности реализации параллельных алгоритмов

Несмотря на все достоинства разработка программно-математического обеспечения для ПЛИС также имеет ряд трудностей, с которыми приходится столкнуться разработчику. Каждая из микросхем обладает определенными характеристиками по производительности (максимальной тактовой частоте) и числу доступных для использования ячеек (объему).

Сами ячейки являются разнородными по типу и назначению [2, 3]:

- специализированные входные элементы (рис. 1, позиция *а*);
- комбинационные логические блоки и регистры, в некоторых ПЛИС объединенные в логические модули (рис. 1, позиция *б*);
- оперативная память (рис. 1, позиция *в*);
- блоки цифровой обработки сигнала (ЦОС, рис. 1, позиция *г*);
- модули фазовой автоподстройки частоты (ФАПЧ); и др.

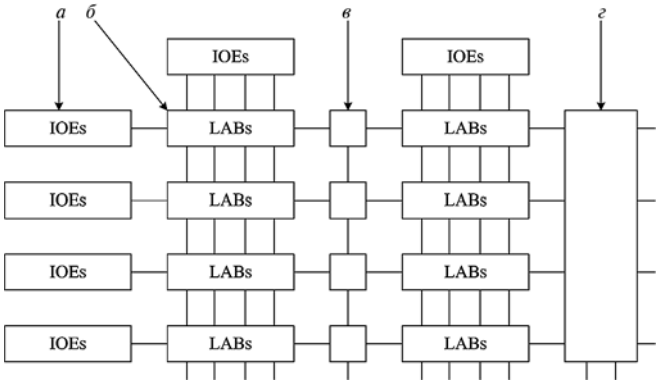


Рис. 1. Типовая архитектура ПЛИС на примере ПЛИС Stratix II компании Altera

Эти ячейки не являются взаимозаменяемыми, и каждый из них располагается внутри кристалла в определенном месте. Специфика и расположение используемого блока относительно других используемых блоков оказывает влияние на производительность.

Существуют методы проектирования, позволяющие разработать модули таким образом, чтобы получить требуемую производительность [4, 5]. Но многие алгоритмы, используемые в современных системах высокоскоростной обработки (системах цифровой связи, системах первичной обработки радиосигналов или системах траекторного сопровождения [6]), требуют существенных вычислительных затрат и большого объема оперативной памяти. Например, при использовании гистограммных алгоритмов определения координат излучающего объекта [7] необходимо за несколько миллисекунд построить проекцию следа пеленгационного конуса на земной поверхности, для чего приходится вычислять значения трансцендентых математических функций в арифметике с плавающей точкой, что достаточно проблематично реализовать в ПЛИС. При этом все построенные и накопленные проекции следов должны сохраняться в памяти вычислителя, что требует десятки мегабайтов, а при использовании параметрической селекции число карт увеличивается кратно числу разделений, что приводит к необходимости использования модулей оперативной памяти на несколько гигабайтов и установки вычислительных модулей, поддерживающих 64-разрядную адресацию. В качестве другого примера можно привести современные алгоритмы фильтрации, предполагающие использование фильтров Калмана. Такие алгоритмы используются при оценке точного положения объекта на траектории [8] и связаны с большим объемом матричных вычислений, точность которых также зависит от использования целочисленных вычислений или арифметики плавающей точки. Все это приводит к росту сложности системы, а это критически сказывается на возможности использования аппаратуры на борту, а не в стационарном режиме. Поэтому зачастую выбор того или иного алгоритма обусловливается поиском компромисса между желаемой точностью результата, желаемой скоростью обработки и затратой аппаратных ресурсов на его получение. При этом различные реализации одного и того же алгоритма могут требовать различные ресурсы, например, вычислительного времени, числа регистров, ячеек памяти, умножителей и пр. В статье [9] рассмотрены особенности применения ПЛИС к различным задачам, возникающим в процессе разработки различного оборудования.

Выбор алгоритма и правильного способа его реализации эвристическими методами приводит к трудно предсказуемым результатам, поэтому целью данной работы является разработка методики, позволяющей провести анализ используемых реализаций тех или иных функциональных модулей и выполнить многокритериальную оценку проекта. Подобная методика должна позволять сравнить несколько систем между собой и сделать выбор лучшей из них по некоторым критериям.

Для решения подобной задачи целесообразно использовать методы системного анализа, предполагающего возможность декомпозиции системы на ряд подсистем (модулей) и проведения оценки отдельных подсистем и синтеза системы, оптимальной по заданным условиям [10, 11].

Применение системного анализа для получения критериев качества алгоритмов

Согласно идеологии системного анализа декомпозиция системы на модули имеет смысл до тех пор, пока членение подсистем низшего уровня не приводит к потере системообразующих свойств [10, 11]. Подобный подход соответствует минимизации числа связей между отдельными подсистемами, так как при потере системообразующих свойств связи между подсистемами начинают соответствовать так называемым промежуточным данным, которые используются для получения того или иного свойства.

Другая сторона системного анализа позволяет свести оценку качества системы к задачам численной оптимизации, т. е. поставить в соответствие некоторым параметрам системы некоторую численную величину, характеризующую то, насколько система соответствует заявленным критериям качества.

Таким образом, чтобы получить возможность применить системный анализ к разработке программного обеспечения для ПЛИС, необходимо выработать критерии и выделить целевую функцию.

Очевидно, что критерии должны удовлетворять определенным физическим процессам, происходящим в кристалле. Основные параметры, которые стремятся улучшить разработчики систем реального времени, — это точность результата, время реакции, разрешение по времени, количество затраченных системных ресурсов (ячеек ПЛИС) и быстродействие.

Под *быстродействием* (производительностью) будем понимать максимальную частоту тактирования ПЛИС f_{max}, при которой разрабатываемый проект способен функционировать в заданном кристалле.

Под *временем реакции* будем понимать максимальное время $t_{обр}$, которое должно пройти с момента появления данных на входе до момента появления результата на выходе.

Под *разрешающей способностью* по времени будем понимать минимальный интервал времени между появлением двух данных на входе, чтобы система могла провести их корректную обработку. Обозначим эту величину как $t_{вх.\ min}$.

Под *количеством затраченных ресурсов* будем понимать число занятых ячеек общего назначения, число ячеек памяти и число специализированных ячеек (модулей ЦОС). Введем две величины M — число использованных ячеек данного типа и $M_{\max}$ — максимальное число ячеек данного типа в выбранной схеме. Введем $\varepsilon = M/M_{\max}$ — относительное число использованных ячеек. Полному использованию одного типа блоков соответствует значение 1. Тогда полному заполнению ПЛИС соответствует значение 3.

Чем меньше использовано ресурсов, тем быстрее можно провести сборку проекта (трассировку между блоками внутри кристалла), тем проще получить желаемую производительность, поэтому при заданном размере кристалла оптимизацию можно проводить на основе минимизации величины

$$\varepsilon_M(\text{alg}) = \frac{M_{mem}(\text{alg})}{M_{mem\max}} + \frac{M_{\log ic}(\text{alg})}{M_{\log ic\max}} + \frac{M_{dsp}(\text{alg})}{M_{dsp\max}}, \quad (1)$$

где ε_M — относительное заполнение ПЛИС при реализации данного алгоритма; M_{mem} — число используемых алгоритмом ячеек памяти; $M_{\log ic}$ — число логических ячеек общего назначения, используемых алгоритмом; M_{dsp} — число ячеек цифровой обработки сигналов (умножителей и фильтров), используемых алгоритмом; alg — обозначение выбранного алгоритма.

При этом критерием для выбора из двух алгоритмов alg_j и alg_i будет являться выражение

$$\varepsilon_M(\text{alg}_i) > \varepsilon_M(\text{alg}_j) \Rightarrow \text{alg}_j \succ \text{alg}_i. \quad (2)$$

Учтем, что невозможно использовать ячеек больше, чем число доступных блоков, т. е. для каждого типа ячеек $\varepsilon_{type} \leqslant 1$ и

$$M_{type}(\text{alg}) \leqslant M_{type\ \max}, \quad (3)$$

где type — тип ячеек: logic, mem, dsp.

Условные ограничения на использование алгоритмов

На проект накладываются ограничения по требуемому быстродействию, разрешающей способности и времени реакции, которые позволяют исключить возможность применения той или иной реализации проекта, если не выполняются условия по достигнутым параметрам:

$$\begin{aligned} f^{\text{д}}_{\max} &\geqslant f_{\max}; \\ t^{\text{д}}_{\text{вх}\ \min} &\leqslant t_{\text{вх}}, \end{aligned} \quad (4)$$

где верхний индекс обозначает достигнутое значение соответствующей величины

$$t^{\text{д}}_{\text{обр}} \leqslant t_{\text{обр}}.$$

Проводить анализ проекта целиком не очень удобно, поэтому целесообразно применить методологию анализа и синтеза из области системного анализа. Тогда мы можем рассмотреть наш проект как систему, которую можно расчленить на ряд подсистем, каждую их которых можно разбить еще дальше, до тех пор, пока дальнейшее дробление потеряет под собой физическое обоснование. Применительно к программно-аппаратному обеспечению подобный подход будет означать разбиение на отдельные функциональные блоки — алгоритмы. Тогда величину относительного заполнения для отдельного алгоритма можно представить, как

$$\varepsilon_{Mi}(\text{alg}_i) = \frac{M_{mem\,i}(\text{alg}_i)}{M_{mem\ \max}} + \frac{M_{\log ici}(\text{alg}_i)}{M_{\log ic\max}} + \frac{M_{dsp}(\text{alg}_i)}{M_{dsp\max}}, \quad (5)$$

а суммарную величину (1) при условии использования выбранного перечня алгоритмов alg как

$$\varepsilon_M(\text{alg}) = \sum_i \varepsilon_{Mi}(\text{alg}_i). \quad (6)$$

Тогда из (1) и (6) следует, что

$$\varepsilon_M(\text{alg}) = \frac{M_{mem}(\text{alg})}{M_{mem\ \max}} + \frac{M_{\log ci}(\text{alg})}{M_{\log ic\max}} + \frac{M_{dsp}(\text{alg})}{M_{dsp\max}}. \quad (7)$$

Полученная формула (7) может применяться только в том случае, если все функциональные блоки синхронизированы между собой, т. е. если выходы каждого из блоков настроены таким образом, чтобы соответствующие друг другу выходы блоков соответствовали друг другу по времени.

Рассмотрим конструкцию из N параллельно функционирующих модулей с общим источником данных D_k в квант времени k. Выходные данные каждого из модулей i Pi_k являются реакцией на входные данные, поступившие в k-й квант времени, считываются потребителем, включенным последовательно с параллельной цепью (рис. 2). Каждый из

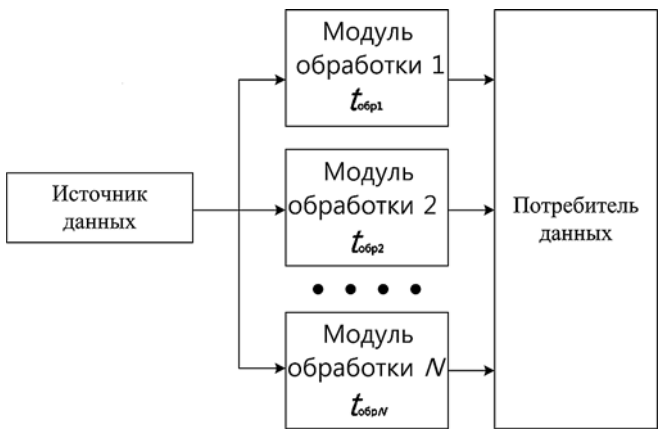


Рис. 2. Схема параллельной обработки данных

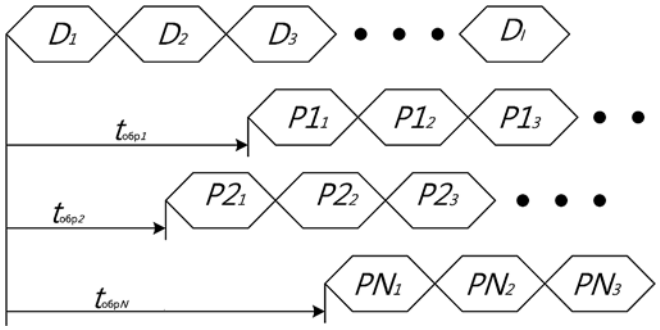


Рис. 3. Временные диаграммы при параллельной обработке данных

модулей обладает своим временем обработки $t_{\text{обр}i}$, поэтому данные поступают на входы потребителя асинхронно (рис. 3). При этом во многих случаях они должны анализироваться в один момент времени (для диаграммы на рис. 3 в момент времени $t_{\text{обр}N}$), т. е. необходимо ввести согласующие линии задержки в остальные каналы поступления данных.

Эквивалентное число тактов для времени обработки данных модулем i при заданном алгоритме alg и требуемых времени реакции $t_{\text{обр}i}$ и предельного быстродействия $f_{i\max}$ для этого модуля составит

$$N_{\text{обр}i}(\text{alg}) = \frac{t_{\text{обр}i(\text{alg})}}{f_{i\max}}.$$

Для корректного анализа данные должны поступить на вход в одно и то же время.

Значит, число тактов, на которое надо задержать сигнал, составит для каждого модуля

$$\Delta N_{\text{обр}i}(\text{alg}) = \max(N_{\text{обр}i}(\text{alg})) - N_{\text{обр}i}(\text{alg}).$$

Число элементов, которые будут затрачены на согласование времени прихода информации от разных модулей, определяется выражением

$$M_{delayi}(\text{alg}) = \Delta N_{\text{обр}i}(\text{alg}) N_{out}(\text{alg}),$$

где N_{out} — разрядность выходной шины модуля при заданном алгоритме.

Суммарное число элементов определяется выражением

$$M_{delay}(\text{alg}) = \sum_i M_{delayi}(\text{alg}).$$

Задержка может быть построена на блоках общего назначения, тогда

$$M_{delay}(\text{alg}) = M_{delay}^{\log ic}(\text{alg}) + M_{delay}^{mem}(\text{alg}),$$

где $M_{delay}^{\log ic}$ — число ячеек общего назначения, используемых для согласования по времени; M_{delay}^{mem} — число ячеек памяти, используемых для согласования.

То есть наличие большого рассогласования во временах обработки параллельных модулей приводит к существенным дополнительным затратам на согласующие линии задержки.

Тогда из (3) мы получаем новое ограничивающее условие:

$$\begin{aligned}
&\sum_i (M_{memi}(\text{alg}_i) + M_{delayi}^{mem}(\text{alg}_i)) < M_{mem\ \max},\\
&\sum_i (M_{\log ici}(\text{alg}_i) + M_{delayi}^{\log ic}(\text{alg}_i)) < M_{\log ic\ \max},\\
&\sum_i M_{dspi}(\text{alg}_i) < M_{dsp\ \max}.
\end{aligned} \qquad (8)$$

А выражение (6) для алгоритма в составе системы записывается как

$$\varepsilon_{Mi}(\text{alg}_i) = \frac{\sum_i M_{memi}(\text{alg}_i) + M_{delayi}^{mem}(\text{alg}_i)}{M_{mem\max}} + \frac{\sum_i M_{\log ici}(\text{alg}_i) + M_{delayi}^{\log ci}(\text{alg}_i)}{M_{\log ic\max}} + \frac{\sum_i M_{dspi}(\text{alg}_i)}{M_{dsp\max}}. \qquad (9)$$

Таким образом, относительное заполнение, которое является основным критерием выбора предпочтительности системы, оказывается зависимым от времени обработки каждого из алгоритмов. Попробуем получить выражение для расчета параметров $t_{\text{обр}}^{\text{д}}$, $t_{\text{вх min}}^{\text{д}}$ и $f_{\max}^{\text{д}}$ в зависимости от того, какой алгоритм используется в данном конкретном случае.

В первом приближении можно считать, что время реакции и разрешающая способность при прочих равных параметрах обратно пропорциональны быстродействию:

$$\frac{t_{\text{вх min2}}}{t_{\text{вх min1}}} = \frac{t_{\text{обр2}}}{t_{\text{обр1}}} = \frac{f_{\max 1}}{f_{\max 2}}$$

или

$$t_{\text{вх min}} = \frac{K_{\text{вх min}}}{f_{\max}},\ t_{\text{обр}} = \frac{K_{\text{обр}}}{f_{\max}}. \qquad (10)$$

Коэффициенты $K_{\text{вх min}}$ и $K_{\text{обр}}$ имеют размерность числа тактов системной частоты и показывают, сколько тактов требуется на реакцию алгоритма и обработку соответственно. В общем случае они не являются константами, а зависят от параметров и структуры системы.

Во-первых, они зависят от выбранного алгоритма и его технической реализации. Для примера сравним два алгоритма преобразования Фурье — прямого преобразования (DFT) и быстрого (FFT). Для определенности допустим, что в один момент времени можно проводить только попарные арифметические операции, тогда сумма S четырех чисел a, b, c и d

$$S = a + b + c + d$$

вычисляется за два такта как попарные суммы S_1 и S_2 за первый такт и последующее вычисление суммы S во второй:

$$S_1 = a + b;$$

$$S_2 = c + d;$$

$$S = S_1 + S_2.$$

Тогда при условии полной параллельности всех вычислений выполнение алгоритма DFT займет N тактов системной частоты. Алгоритм FFT потребует $[\log_2 N] + 1$ тактов, где N — число точек, по которым считается преобразование. Подобная реализация затруднительна для большого числа точек, так как хранить большие объемы информации в неспециализированных ячейках с памятью (регистрах) нерационально, а специализированные ячейки (оперативная память RAM) не позволяет получить доступ к нескольким позициями одновременно. Поэтому в основном используется последовательная реализация алгоритма, при которой затраты по времени возрастают в N раз, т. е. $([\log_2 N] + 1)N$ для FFT и N^2 для DFT.

Это значит, что минимально возможное число тактов, за которые произойдет реакция системы,

$$K_{\text{обр min}}(\text{alg}) = o(\text{alg}),$$

т. е. невозможно получить время реакции, меньшее чем число операций, соответствующих математической сложности алгоритма в его текущей реализации. Подобное справедливо только для условий идеальных математических операций (т. е. любая операция занимает ровно один такт). Для реальных математических операций в ПЛИС существенную роль играет разрядность M. В этом случае одна операция в зависимости от $f_{\max}$ может занимать не один квант времени, а несколько. Обозначим эту величину как $N_{\text{доп}}(f_{\max}, M)$. Дополнительно на времени вычисления могут сказываться дополнительные расходы $N_{\text{расх}}$, обусловленные принципом функционирования ячеек. Например, если алгоритм не предусматривает последовательное чтение памяти, а выбирает следующую ячейку для чтения исходя из результатов последней операции, то сказывается дополнительная задержка на оперативной памяти в два такта (один такт на запись адреса, один такт на выдачу результата). Вместе с тем, если проводится последовательная обработка в некотором алгоритме, например, сложение всех чисел в блоке оперативной памяти, то выбор следующей позиции для чтения не зависит от значения предыдущей позиции. Поэтому данные будут прочитаны за число тактов, равное размеру памяти в словах, плюс дополнительные два такта на работу с памятью — задержку получения данных на выходе памяти относительно входа. То есть подобный вклад может быть аддитивным, мультипликативным или комбинированным (когда есть и аддитивная, и мультипликативная добавки). Исходя из этого можно записать выражение для числа тактов обработки в общем виде:

$$K_{\text{обр}}(\text{alg}) =$$

$$= (N_{\text{доп}}(f_{\max}, M) + N^{\text{мул}}_{\text{расх}}(\text{alg}))o(\text{alg}) + N^{\text{адд}}_{\text{расх}}(\text{alg}), \quad (11)$$

где $N^{\text{адд}}_{\text{расх}}$ — аддитивные дополнительные расходы, а $N^{\text{мул}}_{\text{расх}}$ — мультипликативные.

Таким образом, из выражений (10) и (11) следует:

$$t_{\text{обр}}(\text{alg}) =$$

$$= \frac{(N_{\text{доп}}(f_{\max}, M) + N^{\text{мул}}_{\text{расх}}(\text{alg}))o(\text{alg}) + N^{\text{адд}}_{\text{расх}}(\text{alg})}{f_{\max}},$$

где M определяется исходя из ряда соображений, например, таких как требуемая точность $\varepsilon_{\text{треб}}$ и диапазон значений оцифрованного параметра $\Delta_{\text{диап}}$.

То есть

$$M(\text{alg}) = \frac{\Delta_{\text{диап}}}{\varepsilon_{\text{треб}}} + M_{\text{доп}}(\text{alg}). \quad (12)$$

Ввод добавки $M_{\text{доп}}(\text{alg})$ обусловлен тем, что для работы некоторых алгоритмов (например, алгоритмов последовательного приближения) требуется, чтобы разряд, обеспечивающий заданную точность, был не самым младшим в вычислителе. Например, для вычисления арктангенса по алгоритму CORDIC с точностью до заданного бита требуется увеличить разрядность операций на два-три бита, т. е. $M_{\text{доп}}(\text{atan2}_{\text{CORDIC}}) = 2$.

Максимальное число разрядов, которое можно получить при заданной тактовой частоте, можно рассчитать, приняв во внимание соображение, что в современных кристаллах задержка на связях между элементами много больше, чем задержка переключения ячеек. Тогда можно воспользоваться следующей формулой:

$$N_{\text{доп}}(f_{\max}, M) = \left[\frac{M}{M_0(f_{\max})}\right] + 1,$$

где M_0 — это число ячеек, через которые успеет пройти сигнал между активными фронтами тактовых сигналов, которое, в свою очередь, зависит от времени распространения сигнала между ячейками. Примем упрощение, при котором будем считать, что сигналы распространяются между соседними ячейками. Тогда

$$N_{\text{доп}}(f_{\max}, M) = \left\lfloor \frac{M}{\frac{1}{f_{\max} t_0}} \right\rfloor + 1 = [M f_{\max} t_0] + 1, \quad (13)$$

где t_0 — время распространения сигнала между соседними ячейками. Подставив формулу (12) в выражение (13), получим

$$N_{\text{доп}}(f_{\max}, \Delta_{\text{диап}}, \varepsilon_{\text{треб}}) = \left\lfloor \frac{M}{\frac{1}{f_{\max} t_0}} \right\rfloor + 1 =$$

$$= \left\lfloor \left(\frac{\Delta_{\text{диап}}}{\varepsilon_{\text{треб}}} + M_{\text{доп}}(\text{alg}) \right) f_{\max} t_0 \right\rfloor + 1. \quad (14)$$

Таким образом, из выражений (10), (11) и (14) мы получаем формулу, которой можно описать зависимость времени обработки одного пакета данных:

$$t_{\text{обр}}(f_{\max}, \Delta_{\text{диап}}, \varepsilon_{\text{треб}}, \text{alg}) =$$

$$= \frac{\left(\left(\frac{\Delta_{\text{диап}}}{\varepsilon_{\text{треб}}} + M_{\text{доп}}(\text{alg}) \right) f_{\max} t_0 + 1 + N^{\text{мул}}_{\text{расч}}(\text{alg}) \right) o(\text{alg}) + N^{\text{адд}}_{\text{расч}}(\text{alg})}{f_{\max}}. \quad (15)$$

Второй важной характеристикой, как уже указывалось, является величина $t_{\text{вх min}}$. Данный параметр определить проще всего. Минимальный интервал времени между двумя пакетами, который модуль обработки сможет корректно принять, зависит от характера алгоритма и характера его реализации. Характер алгоритма зависит от того, является ли алгоритм итерационным или непрерывным, т. е. выполняет действия с пришедшим пакетом сразу при его поступлении или ему требуется время на обработку. Например, интегратор является непрерывным по данной терминологии, так как для его работы ему достаточно прибавить поступающие данные к уже имеющимся. Вместе с тем, цифровой фильтр с конечной импульсной характеристикой (КИХ-фильтр) является итерационным, так как в общем случае на каждой итерации необходим пересчет свертки всех записанных данных со всей импульсной характеристикой (ИХ), т. е. потребуется $W_{\text{итер}}$ операций умножения и сложения.

Характер реализации представляет собой способ реализации алгоритма в цифровой технике — итерационный или конвейерный. В первом случае мы экономим ресурсы ценой увеличения $t_{\text{вх min}}$, во втором ценой ресурсов снижаем $t_{\text{вх min}}$. Для описания воспользуемся длиной конвейера $L_{\text{конв}}$ (параллельно вычисляемых операций). Если $L_{\text{конв}} = W_{\text{итер}}$, то новые данные можно подавать каждый такт, так как все операции будут обрабатываться одновременно. Если же $L_{\text{конв}} = 1$, то число одновременно выполняемых операций равно единице, поэтому потребуется $W_{\text{итер}}$ тактов на вычисление результата, в течение которых новые данные невозможно будет обработать.

То есть

$$K_{\text{вх min}}(\text{alg}) = \frac{W_{\text{итер}}(\text{alg})}{L_{\text{конв}}(\text{alg})}. \quad (16)$$

Тогда из выражений (1) и (8) получаем разрешающую способность

$$t_{\text{вх min}}(\text{alg}) = \frac{W_{\text{итер}}(\text{alg})}{L_{\text{конв}}(\text{alg}) f_{\max}}. \quad (17)$$

Третьей важной величиной является достигнутое быстродействие $f_{\max}$. Эта величина определяет устойчивость работы микросхемы к изменяющимся условиям окружающей среды, допустим, к флуктуациям температуры или джиттеру тактового сигнала. Достигнутое быстродействие можно описать также как и желаемое быстродействие, то есть через максимальную длину, на которую распространится сигнал за период тактового сигнала

$$f_{\max} = \frac{1}{l_{\max}(\text{alg}) t_0}, \quad (18)$$

где $l_{\max}$ — максимальная длина пути (в ячейках) между двумя регистрами в модуле.

Полученные формулы (15), (17) и (18) позволяют рассчитать требуемые параметры алгоритмов при выбранных ограничивающих $f_{\max}$, $t_{\text{обр}}$ и $t_{\text{вх min}}$ либо рассчитать достигнутые параметры $f^{\text{д}}_{\max}$, $t^{\text{д}}_{\text{обр}}$ и $t^{\text{д}}_{\text{вх min}}$ при фиксированных алгоритмах.

Из выражения (18) видно, что тактовая частота получается в прямой зависимости от времени на обработку и минимального интервала между поступлениями данных, т. е. она является следствием необходимости обеспечения своевременной обработки данных. Таким образом, в выражения (15) и (17) можно подставить значение частоты из выра-

жения (18). Тогда выражения для расчета временных параметров алгоритма примут вид.

$$t_{\text{вх min}}(\text{alg}) = \frac{W_{\text{итер}}(\text{alg}) l_{\max}(\text{alg}) t_0}{L_{\text{конв}}(\text{alg})};$$

$$t_{\text{обр}}(l_{\max}, \Delta_{\text{диап}}, \varepsilon_{\text{треб}}, \text{alg}) =$$

$$= \left(\frac{\frac{\Delta_{\text{диап}}}{\varepsilon_{\text{треб}}} + M_{\text{доп}}(\text{alg})}{l_{\max}} + 1 + N^{\text{мул}}_{\text{расх}}(\text{alg}) \right) \times \tag{19}$$

$$\times\, o(\text{alg}) l_{\max}(\text{alg}) t_0 + N^{\text{адд}}_{\text{расх}}(\text{alg}) l_{\max}(\text{alg}) t_0.$$

Анализ данных величин позволяет найти в проекте участки, которые снижают возможность достижения требуемой производительности и, проанализировав их, изменить таким образом, чтобы они удовлетворяли заданным условиям.

Существуют два основных способа учета параметров (19) для ограничивающего условия (4): анализ "сверху вниз" и анализа "снизу вверх". В первом случае мы после разбиения системы на части задаем для каждой подсистемы свои временные ограничения, чтобы рассчитать временные параметры и применить ограничивающее условие (4) к каждой из подсистем. Второй вариант — рассчитать временные параметры для каждой из подсистем, из них получить достигнутые временные параметры системы, например, для времени обработки это будет время наиболее длинного пути распространения данных от входа к выходу. И применить ограничивающее условие (4) к временным параметрам самой системы.

Для первого случая и с введенным упрощением из (4) и (19) получаем ограничения, накладываемые на каждую подсистему в проекте:

$$\left(\left(\frac{\frac{\Delta_{\text{диап}}}{\varepsilon_{\text{треб}}} + M_{\text{доп}}(\text{alg})}{l_{\max}} + 1 + N^{\text{мул}}_{\text{расх}}(\text{alg}) \right) o(\text{alg}) + \right.$$

$$\left. + N^{\text{адд}}_{\text{расх}}(\text{alg}) \right) l_{\max}(\text{alg}) t_0 \leqslant t_{\text{обр}}; \tag{20}$$

$$\frac{W_{\text{итер}}(\text{alg}) l_{\max}(\text{alg}) t_0}{L_{\text{конв}}(\text{alg})} \leqslant t_{\text{вх}}.$$

Рассмотрим два простых примера, иллюстрирующих выбор ограничивающих параметров для условия (4) с упрощением по частоте и определение наилучшей системы согласно критерию (2) при расчете по формуле (9).

Выбор ограничивающих величин $t_{\text{вх}}$ и $t_{\text{обр}}$ диктуется требованиями к обработке и принципам работы модуля. Рассмотрим две ситуации, в которой на ПЛИС происходит обработка некоторого сигнала с частотой дискретизации $f_{\text{д}}$.

В зависимости от назначения устройства, мы можем получить разные $t_{\text{вх}}$. Представим два случая, в одном из них ПЛИС предназначена для цифровой обработки звука в музыкальной аппаратуре, во втором ПЛИС используется в ЦОС эхолота.

4. Примеры расчета критериев

В музыкальной аппаратуре поступающий сигнал необходимо обрабатывать постоянно. То есть аппаратура не может сделать паузу, подумать и после этого выдать результат. Таким образом, мы получаем величину $t_{\text{вх}}$, равную $1/f_{\text{д}}$. При этом данные будут обрабатываться в реальном времени. Данные, поступившие через $t_{\text{вх}}$, будут обработаны, а не буферизованы. Если бы $t^{\text{д}}_{\text{вх min}} \leqslant t_{\text{вх}}$ не выполнялось, то данные записывались бы в буфер в надежде, что будет время обработать их позже. Но так как в данном приложении все отсчеты и моменты времени равноправны, то увеличивалась бы задержка передачи данных от входа к выходу с течением времени, либо происходило бы неучтенное алгоритмами прореживание, что привело бы к неправильным или нестабильным результатам работы аппаратуры.

Теперь рассмотрим приложение ЦОС для эхолота. Предположим, что максимальная глубина, которую можно анализировать эхолотом, равна $h_{\max}$. Тогда время, которое будет потрачено на получение отсчетов сигнала, будет равно $t_{\text{ан}} = \frac{2h_{\max}}{v_{\text{с}}}$, где $v_{\text{с}}$ — скорость распространения сигнала в среде. Допустим, что эхолот посылает зондирующий сигнал каждые $t_{\text{з}}$. В этом случае мы должны принять $N = t_{\text{ан}} f_{\text{д}}$ отсчетов, но эти отсчеты мы можем обрабатывать в течение большего времени. Тогда $t_{\text{вх}} = \frac{t_{\text{з}}}{N} = \frac{t_{\text{з}}}{t_{\text{ан}} f_{\text{д}}}$. В зависимости от периода следования зондирующих сигналов мы получаем разные результаты. Например, если сигнал излучается через интервалы $t_{\text{ан}}$, то задача сводится к предыдущей, т. е. $t_{\text{вх}} = \frac{1}{f_{\text{д}}}$. Если же период излучения больше, то мы получаем два контура. Один из них должен успевать обрабатывать сигналы на входной частоте (частоте АЦП) и обеспечивать буферизацию входных сигналов, а второй — обеспечивать возможность чтения из буфера и обработку записанных отсчетов за время между посылками. Если среднее время обработки отсчета получится больше $\frac{t_{\text{з}}}{t_{\text{ан}} f_{\text{д}}}$, то возникнут те же самые проблемы, когда мы бу-

дем вынуждены либо терять информацию, либо "плыть" по времени, прошедшем от появления отсчета на входе до появления результата на выходе.

Аналогичная ситуация со временем обработки. Оно диктуется несколькими факторами. В глобальном смысле (в масштабе всего проекта) время определяется как максимальное время реакции на изменение. На примере той же обработки звука, если задержка будет слишком большой, то это будет сказываться на качестве игры, потому что фактически будет получаться, что исполнители не слышат ни того, как играют они, ни того, как играют соседи. То есть время реакции в данном случае определяется разрешающей способностью человека по времени.

Для эхолота время обработки зависит от сферы применения эхолота. Например, рассмотрим применение эхолота для поиска рифов при движении на плавательном транспортном средстве. Путь максимальное расстояние, на котором действует эхолот — $h_{\max}$, скорость транспортного средства v, время реакции транспортного средства на информационное сообщение об обнаруженных рифах $t_{\text{р}}$, скорость распространения звука в среде $v_{\text{зв}}$, а максимальное ускорение судна a. Тогда максимальное время остановки составит $t_{\text{ост}} = \frac{2h_{\max}}{v_{\text{зв}}} + t_{\text{р}} + \frac{v}{a} + t_{\text{обр}}$.

Пройденный путь составит

$$S_{\text{ост}} = v\frac{v}{a} - \frac{a\left(\frac{v}{a}\right)^2}{2} + v\left(\frac{2h_{\max}}{v_{\text{зв}}} + t_{\text{р}} + t_{\text{обр}}\right) = \frac{v^2}{2a} + v\left(\frac{2h_{\max}}{v_{\text{зв}}} + t_{\text{р}} + t_{\text{обр}}\right).$$

Пройденный путь должен быть меньше, чем $h_{\max}$, т. е.

$$\frac{v^2}{2a} + v\left(\frac{2h_{\max}}{v_{\text{зв}}} + t_{\text{р}} + t_{\text{обр}}\right) < h_{\max}.$$

Из этого выражения вычисляется требуемое время $t_{\text{обр}}$.

Накладываемые таким образом ограничения позволяют исключить из рассмотрения ряд алгоритмов, которые заведомо не удовлетворяют этим условиям.

Разработанные методы были применены к проектированию модулей цифровой обработки, что отражено в [8].

Заключение

Таким образом, необходимость поиска лучших алгоритмов для реализации заданной системы на ПЛИС ставит перед разработчиком ряд задач. Необходимо провести разбиение системы на составляющие блоки, рассчитать параметры возможных алгоритмов и граничные величины параметров (1) для каждого алгоритма. После этого необходимо провести сравнение алгоритмов по критерию (16) при учете системы условий (13) для каждого из алгоритмов, системы условий (15) для текущей реализации системы, а также при выполнении условий (13) для всей системы целиком.

Разработанная система критериев позволяет сократить время разработки программы, найти в проекте "узкие" места, не позволяющие достигнуть заложенных требований, выбрать из всех алгоритмов те, которые будут соответствовать заданным требованиям по скорости, точности и производительности. При этом появляется возможность оптимального использования алгоритмов на заданной аппаратной базе (выбранной микросхеме), либо, наоборот, становится возможным выбрать микросхему, на которой данный проект стоит реализовывать.

Список литературы

1. **Тарасов И. Е.** Разработка цифровых устройств на основе ПЛИС Xilinx с применением языка VHDL. М.: Горячая линия — Телеком, 2005. 253 с.

2. **Altera** corporation, Stratix II device handbook [Электронный ресурс]. URL: https://www.altera.com/en_US/pdfs/literature/hb/stx2/stratix2_handbook.pdf (дата обращения: 15.04.2015).

3. **Virtex-6** Family Overview [Электронный ресурс]. URL: http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf (дата обращения: 15.04.2015).

4. **Johnson H. W., Graham M.** High-Speed Digital Design. USA: Prentice Hall, 1993. 384 с.

5. **Кондратов К. С., Жураковский В. Н., Силин С. И.** Принципы проектирования встраиваемых систем на основе программных средств // Инженерный вестник. 2014. №11. С. 536—544.

6. **Логвиненко А. С., Жураковский В. Н.** Повышение точности измерения параметров сигналов в цифровом тракте // Инженерный вестник. 2014. №10. С. 614—651.

7. **Жураковский В. Н., Кондратов К. С.** Алгоритм определения местоположения наземных объектов в условиях низкой точности входных данных // Наука и образование: электронное научно-техническое издание. 2013. № 12. С. 307—314.

8. **Шахтарин Б. И., Микаэльян С. В.** Траекторный фильтр в системе координат измерителя для системы слежения за целями по угломерным данным // Научный вестник Московского государственного технического университета гражданской авиации. 2013. № 193. С. 21—25.

9. **Smetana D.** Designing a general-purpose FPGA DSP card for EW, radar applications using the latest generation of FPGAs [Электронный ресурс] // Military Embedded Systems. 29.01.2014. URL: http://mil-embedded.com/articles/designing-general-purpose-applications-using-latest-generation-fpgas (дата обращения: 03.09.2015).

10. **Николаев В. И., Брук В. М.** Системотехника: методы и приложения. Л.: Машиностроение, Ленингр. отд-ние, 1985. 199 с.

11. **Антонов А. В.** Системный анализ. М.: Высшая школа, 2004. 454 с.

12. **Еременков А. И., Жураковский В. Н., Силин С. И.** Аналого-цифровой модуль приема и передачи данных. Разработка алгоритмов и программного обеспечения // Инженерный вестник. 2014. №10. С. 652—665.

V. N. Zhurakovskiy, Associate Professor, **S. I. Silin,** Assistant, e-mail: sm2-2@inbox.ru,
Bauman MSTU

System Analysis Applied to the Data Processing Algorithms Selection for Modern Computing Systems

Main purpose of this work is development of criteria system that could be applied to develop sophisticated devices based on field programmable gate array (FPGA), systems on the chip (SOC) and complex programmable logic devices (CPLD). In the first chapter of this article authors reviewed typical modern approaches to fulfilling this task. Authors provided explanation of the basis which the further apply to developed theory. In the second chapter authors developed criteria and objective function that could be used during system development to select the most suitable algorithm and realization. In the third chapter authors developed limitations to be used during selection process. Limitations are based on the typical structure of programmable integrated circuits and properties of typical problems arise during development process. In the fourth chapter authors placed example of different signal processing algorithms that could be analyzed with developed criteria and limitation systems. In the fifth chapter conclusion to work is placed. This theory suits to give mathematical and theoretical explanation of the selection process during DSP algorithms development.

Keywords: *FPGA, system analysis, algorithm, parallel signal-processing, radiolocation, embedded systems, systems on the chip, electronics, circuit technique, systems engineering, system design*

References

1. **Tarasov I. E.** *Razrabotka cifrovyh ustrojstv na osnove PLIS Xilinx s primeneniem jazyka VHDL*. Moscow: Gorjachaja linija — Telekom, 2005, 253 p.

2. **Altera** *corporation*, Stratix II device handbook [Electronic resource]. URL: https://www.altera.com/en_US/pdfs/literature/hb/stx2/stratix2_handbook. pdf (date of circulation: 15.04.2015).

3. **Virtex-6** *Family Overview* [Electronic resource]. URL: http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf (date of circulation: 15.04.2015).

4. **Johnson H. W., Graham M.** *High-Speed Digital Design*. Prentice Hall, 1993, 384 p.

5. **Kondrashov K. S., Zhurakovskij V. N., Silin S. I.** Principy proektirovanija vstraivaemyh sistem na osnove programmnyh sredstv, *Inzhenernyj vestnik*, 2014, no. 11, pp. 536—544.

6. **Logvinenko A. S., Zhurakovskij V. N.** Povyshenie tochnosti izmerenija parametrov signalov v cifrovom trakte, *Inzhenernyj vestnik*, 2014, no. 10, pp. 641—651.

7. **Zhurakovskij V. N., Kondrashov K. S.** Algoritm opredelenija mestopolozhenija nazemnyh obektov v uslovijah nizkoj tochnosti vhodnyh dannyh, *Nauka i obrazovanie: jelektronnoe nauchno-tehnicheskoe izdanie*, 2013, no. 12, pp. 307—314.

8. **Shahtarin B. I., Mikajel'jan S. V.** Traektornyj fil'tr v sisteme koordinat izmeritelja dlja sistemy slezhenija za celjami po uglomernym dannym, *Nauchnyj vestnik Moskovskogo gosudarstvennogo tehnicheskogo universiteta grazhdanskoj aviacii*, 2013, no. 191, pp. 21—25.

9. **Smetana D.** Designing a general-purpose FPGA DSP card for EW, radar applications using the latest generation of FPGAs [Electronic resource], *Military Embedded Systems*. 29.01.2014. URL: http://mil-embedded.com/articles/designing-general-purpose-applications-using-latest-generation-fpgas (date of circulation: 03.09.2015).

10. **Nikolaev V. I., Bruk V. M.** *Sistemotehnika: metody i prilozhenija*. Leningrad: Mashinostroenie, Leningr. otd-nie, 1985, 199 p.

11. **Antonov A. V.** Sistemnyj analiz. Moscow: Vysshaja shkola, 2004. 454 p.

12. **Eremenkov A. I., Zhurakovskij V. N., Silin S. I.** Analogo-cifrovoj modul' priema i peredachi dannyh. Razrabotka algoritmov i programmnogo obespechenija, *Inzhenernyj vestnik*, 2014, no. 10, pp. 652—665.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОРГАНИЗАЦИОННЫХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ
INFORMATION TECHNOLOGIES IN THE ORGANIZATIONAL AND SOCIO-ECONOMIC SYSTEMS

УДК 004.514

А. С. Зуев, канд. техн. наук, зав. каф., Zuev_Andrey@mail.ru,
И. С. Фадеев, диспетчер кафедры, fadeev@protonmail.ch
Московский государственный университет информационных технологий, радиотехники и электроники

Виртуальные ситуационные центры — новый инструмент управления социально-экономическими системами

Показаны возможности применения современных технологий и устройств виртуальной реальности для реализации виртуальных ситуационных центров как инструментов управления социально-экономическими системами на уровне государства и отдельных экономических субъектов. Обосновывается целесообразность дополнения системы управления, основанной на ситуационных центрах, их виртуальными моделями, отображаемыми с помощью современных устройств компьютерной техники.

Ключевые слова: *виртуальная реальность, ситуационный центр, управление социально-экономическими системами, мобильный телефон*

Введение

Одним из перспективных направлений развития концепций человеко-компьютерного взаимодействия [1] является технология виртуальной реальности [2], позволяющая замещать реальное окружающее человека пространство виртуальным окружением с обеспечением опций интерактивного взаимодействия с находящимися в нем объектами. В общем случае данная технология может быть описана следующим образом — видимый человеком отраженный от реальных объектов свет перекрывается и вместо него демонстрируется замещающее изображение, воспринимаемое как реальное окружающее пространство. Первый шлем виртуальной реальности, реализующий основные идеи данной технологии, был создан в США в 1968 году и получил название "Дамоклов меч" [3]. Однако в результате развития средств электронно-вычислительной техники широкое распространение технология виртуальной реальности начала получать в последние годы и нашла применение в индустрии развлечений и производстве, медицине и образовании, военном деле и маркетинге. С учетом динамики развития средств электронно-вычислительной техники данная технология будет занимать все большее место во всех сферах жизни информационного общества, в том числе в задачах управления социально-экономическими системами. В статье рассматривается пример такой задачи, имеющей важное значение не только для системы государственного управления, но и для бизнеса — реализация виртуальных ситуационных центров посредством использования современных персональных устройств компьютерной техники (мобильные телефоны, планшеты, смартфоны и т. п.).

Ситуационный центр как инструмент управления социально-экономическими системами

Ситуационные центры (центры стратегического управления, визионариумы, ситуационные комнаты, командные пункты управления, мультимедийные центры и т. д.) представляют собой инновационные комплексы методических, информационных и аппаратно-программных средств, предназначенные для работы руководителей или групп экспертов [4]. Актуальность создания таких специализированных объектов на различных уровнях государственного и корпоративного управления обусловлена многи-

ми факторами, в том числе необходимостью оперативного принятия решений на основании большого потока информации в условиях дефицита времени.

В настоящее время в Российской Федерации число ситуационных центров (СЦ) различного назначения велико, в том числе их используют:

- федеральные органы государственной власти (президент, правительство, федеральные министерства, агентства и др.);
- региональные органы субъектов РФ и местного самоуправления (краевые и областные администрации, мэрии и др.);
- крупные промышленные предприятия;
- образовательные учреждения и др.

Ситуационный центр — это сложный объект, обладающий организационной и пространственной (топологической) структурой, включающей в общем случае [4]: зал совещаний руководства; операторскую; комнату аналитиков; зал дежурной смены (для кризисных центров и центров мониторинга). При его проектировании учитывается множество аспектов: охрана и безопасность, размещение и углы обзора рабочих мест, электропитание, средства связи, коммунальное обеспечение. Решение такого комплекса задач требует привлечения большого числа различных специалистов и связано со значительными финансовыми затратами.

В то же время ситуационный центр имеет ряд объективных недостатков.

1. Необходимость присутствия экспертов и лица, принимающего решения (ЛПР), в данной локации. Доставка ЛПР и экспертов в центр может быть затруднена, особенно при крупномасштабных чрезвычайных ситуациях, требующих повышенной оперативности принятия решений.

2. Мобильность ЛПР и экспертов ограничена территорией СЦ. Вне территории СЦ эксперты и ЛПР не могут выполнять свою роль, а их локализация делает систему управления уязвимой к воздействиям, направленным на затруднение (подавление) функционирования СЦ.

3. Необходимость коммунального обеспечения объекта, содержания штата обслуживающего персонала и приобретения дорогостоящего оборудования обусловливает значительный объем затрат на его содержание.

4. Потенциальная доступность представляемой ЛПР и экспертам информации широкой нецелевой аудитории — обслуживающий персонал, охрана, секретари.

5. Необходимость контроля помещений СЦ на наличие "закладок" — непредусмотренной аппаратуры передачи информации. Проверка СЦ на наличие "закладок" может быть достаточно сложной задачей, обусловленной его топологией и требующей регулярного решения.

6. Низкая загруженность по времени использования, обусловленная относительной редкостью возникновения ситуаций, требующих эксплуатации СЦ.

Представляет интерес поиск возможностей применения современных технологий и устройств виртуальной реальности для воссоздания окружения ситуационного центра с предоставлением оператору соответствующих ему возможностей и исключением перечисленных выше недостатков.

Виртуальный ситуационный центр и его разработка

В настоящее время, помимо полноценных шлемов виртуальной реальности, оснащенных специализированным программно-аппаратным обеспечением [5], виртуальное окружение может быть реализовано с применением шлемов виртуальной реальности для смартфонов [6], в которых в качестве экрана могут быть использованы современные мобильные телефоны. С помощью подобных устройств может быть воссоздана не только обстановка конкретного ситуационного центра, но и симулировано нахождение на конкретном рабочем месте в нем.

Виртуальный ситуационный центр не имеет перечисленных выше недостатков, а именно:

1) ЛПР и эксперты могут участвовать в работе центра из текущего местоположения и перемещаться в пределах областей, обеспечивающих доступ к телефонной связи и/или Интернету;

2) затраты минимизированы, так как потребность в дорогостоящем оборудовании, коммуникациях и обслуживающем персонале отсутствует;

3) передаваемая и выводимая с помощью устройства виртуальной реальности информация недоступна нецелевой аудитории, возможности ее считывания и перехвата существенно ограничены;

4) приложение виртуального СЦ используется "по необходимости" и требует наличия либо специального устройства, либо шлема виртуальной реальности для смартфонов и мобильного телефона.

Виртуальный СЦ, реализованный с использованием персонального мобильного телефона, обеспечивает потенциально бóльшую оперативность включения ЛПР или эксперта в работу СЦ. Сеансы связи можно устанавливать на основании имеющегося в телефоне списка контактов, а для защиты передаваемой информации применимы методы, используемые в работе реальных ситуационных центров. Наличие возможности работать в обстановке, предоставляемой виртуальным СЦ, увеличивает устойчивость системы управления, основан-

ной на использовании реальных СЦ, в том числе к воздействиям, направленным на затруднение (подавление) их работы. В результате виртуальный СЦ дополняет функциональность реального центра, а в перспективе развития электронно-вычислительной техники может его заменить.

Разработку виртуального СЦ можно разделить на два основных этапа: трехмерное моделирование интерьера и окружающих объектов и написание программного кода, отвечающего за реализацию необходимого функционала.

Для первого этапа можно использовать любой редактор трехмерной графики (например, "Blender" [7]), в котором создаются (моделируются) все виртуальные аналоги физических объектов, используемых в СЦ, начиная от помещения и мебели и заканчивая оборудованием, с которым будет взаимодействовать ЛПР.

На втором этапе, вместо написания большого количества программного кода, целесообразно использовать так называемый "движок" (англ. *engine*) — программный компонент, обеспечивающий взаимодействие объектов в виртуальном трехмерном пространстве. При этом весь необходимый функционал можно реализовывать с помощью программных скриптов. С практической точки зрения наиболее мощным и удобным "движком" на сегодняшний день представляется программный инструмент "Unity" [8].

Особенность приложений виртуальной реальности заключается в том, что изображение на экране шлема или мобильного телефона необходимо выводить отдельно для правого и левого глаза. Рациональный способ реализации этой особенности — использование специальных комплектов средств разработки (англ. *software development kit*, SDK). Конкретный SDK выбирается в зависимости от целевой платформы, на которой планируется использовать приложение (например, "Oculus SDK" [9], "Fibrum SDK" [10]).

В результате применения перечисленных выше средств разработки получается готовый проект, который может быть многократно скомпилирован под различные устройства виртуальной реальности и мобильные телефоны.

Заключение

Внедрение технологии виртуальной реальности в различные сферы жизни информационного общества будет являться долгосрочной тенденцией в развитии концепций человеко-компьютерного взаимодействия, а ее широкая практическая реализация будет связана с существенными социальными, культурными, психологическими и экономическими аспектами. Существующие примеры, перспективы и задачи прикладного применения данной технологии впечатляющи, а ожидаемые результаты могут стать будущим компьютерной техники на ближайшее десятилетие. Целесообразно ожидать, что развитие *"hardware"*-составляющей данной технологии будет идти по пути миниатюризации.

В настоящее время разработка приложения виртуального ситуационного центра выполняется в учебной лаборатории "Технологии дополненной и виртуальной реальности" на кафедре "Прикладная и бизнес-информатика" Федерального государственного бюджетного образовательного учреждения высшего образования "Московский государственный университет информационных технологий, радиотехники и электроники". Коллектив исследователей приглашает к сотрудничеству заинтересованных специалистов и заказчиков.

Список литературы

1. **Магазанчик В. Д.** Человеко-компьютерное взаимодействие: учебн. пособие. М.: Университетская книга; Логос, 2007. 256 с.
2. **Виртуальная** реальность. [Электронный ресурс] // Википедия — свободная энциклопедия [Офиц. сайт]. URL: https://ru.wikipedia.org/wiki/Виртуальная_реальность (дата обращения: 11.12.2015).
3. **Осипов М. П.** Системы виртуальной реальности: учебно-метод. пособие. Н. Новгород: Типография ННГУ, 2012. 48 с.
4. **Ильин Н. И., Демидов Н. Н., Новикова Е. В.** Ситуационные центры. Опыт, состояние, тенденции развития. М.: МедиаПресс, 2011. 336 с.
5. **Шлемы** виртуальной реальности — обзор самых популярных устройств. [Электронный ресурс] // Virtual Environment Group [Офиц. сайт]. URL: http://ve-group.ru/shlemyi-virtualnoy-realnosti-obzor-samyih-populyarnyih-ustroystv/ (дата обращения: 11.12.2015).
6. **ООР "Фибрум"** [Офиц. сайт]. URL: http://fibrum.com/ (дата обращения: 11.12.2015).
7. **Blender** [Офиц. сайт]. URL: http://www.blender.org/ (дата обращения: 11.12.2015).
8. **Unity** [Офиц. сайт]. URL: http://unity3d.com/ru/unity (дата обращения: 11.12.2015).
9. **Oculus SDK** [Офиц. сайт]. URL: https://developer.oculus.com/downloads/ (дата обращения: 11.12.2015).
10. **Fibrum SDK** [Офиц. сайт]. URL: http://fibrum.com/sdk/#sdk_download (дата обращения: 11.12.2015).

A. S. Zuev, Ph. D., Head of the "Applied and Business Informatics" Department, Zuev_Andrey@mail.ru,
I. S. Fadeev, fadeev@protonmail.ch
Moscow State University of Information Technologies, Radioengineering and Electronics

Virtual Command Centers — New Management Tool for Socio-Economic Systems

The article describes the possibilities of application of modern technologies and virtual reality devices to implement virtual command centers as an instrument of socio-economic systems at the level of the state and the individual economic actors. Expediency of addition of management system, based on the command centers, with their virtual models, which are processed with modern computer hardware is proved.

Virtual command center has no objective disadvantages as compared with real physical command centers. Namely, a person who makes decision and experts can participate in the center's work from the places of their current location and move within areas that provide access to the telephone and/or internet; costs are minimized, since there is no need for expensive equipment, communications and service personnel; information that transmitted and displayed by the virtual reality device is not available for non-targeted audience, the possibility of its reading and interception is severely limited; application of the virtual command center is used "as needed" and requires either special devices or head-mounted display for smartphones and mobile phones.

Keywords: *virtual reality, command center, the management of socio-economic systems, mobile phone*

References

1. **Magazanchik V. D.** *Cheloveko-komp'jutemoe vzaimodejstvie: Uchebn. posobie* (Human-computer Interaction: Study guide). Moscow: Universitetskaja kniga; Logos, 2007, 256 p.

2. **Virtual'naja** *real'nost'* (Virtual reality). [Electronic resource], Wikipedia, the free encyclopedia. [Official website]. URL: https://ru.wikipedia.org/wiki/Virtual_reality (accessed: 11.12.2015).

3. **Osipov M. P.** *Sistemy virtual'noj real'nosti:* Uchebno-metodicheskoe posobie (The systems of virtual reality: Educational methodology handbook). — N. Novgorod, Tipografij a NNGU, 2012, 48 p.

4. **Il'in N. I., Demidov N. N., Noyikova E. V.** *Situacionnye centry. Opyt, sostojanie, tendencii razvitija* (Command centers. Experience, condition, development trends). Moscow: MediaPress, 2011, 336 p.

5. **Shlemy** *virtual'noj real'nosti — obzor samyh populjarnyh ustrojstv* (Head-mounted displays — the most popular devices overview), [Electronic resource], Virtual Environment Group [Official website], URL: http://ve-group.ru/shlemyi-virtualnoy-realnosti-obzor-samyih-populyarnyih-ustroystv/ (accessed: 11.12.2015).

6. **OOP** "Fibrum" [Official website], URL: http://fibrum.com/ (accessed: 11.12.2015).

7. **Blender** [Official website]. URL: http://www.blender.org/ (accessed: 11.12.2015).

8. **Unity** [Official website]. URL: http://unity3d.com/ru/unity (accessed: 11.12.2015).

9. **Qculus** *SDK* [Official website], URL: https://developer.oculus.com/downloads/ (accessed: 11.12.2015).

10. **Fibrum** *SDK* [Official website], URL: http://fibrum.com/sdk/#sdk_download (accessed: 11.12.2015).

ИНФОРМАЦИЯ

Двенадцатая Конференция CEE-SECR /Разработка ПО Digital October, Москва, 27—28 (29) октября 2016 г.

2016 CEE-SECR Software Engineering Conference in Russia

Разработка ПО/CEE-SECR (www.secr.ru) — ежегодная независимая конференция, собирающая около 1000 участников, среди которых разработчики ПО, исследователи, инженеры-практики, лидеры общественного мнения, предприниматели и инвесторы. За годы работы конференция CEE-SECR завоевала заслуженный авторитет главного профессионального события в индустрии разработки программного обеспечения в России и странах Восточной Европы.

В состав программного комитета конференции CEE-SECR входят эксперты мирового уровня, представляющие как ИТ-индустрию, так и научно-исследовательские организации.

Подробности: http://2016.secr.ru/

ДИСКУССИОННЫЙ КЛУБ
DISCUSSION CLUB

УДК 004.8

В. И. Голубев, канд. техн. наук,
РНТО РЭС им. А. С. Попова (секция им. акад. А. И. Берга)

Информация как отображение объектов окружающего мира в коре головного мозга

Механизм сознания формируется таким образом. Сигналы от объекта через визуальный или акустический анализатор попадают в проекционные поля коры головного мозга и регистрируются в колонках этой области. Далее они передаются в ассоциативные поля коры. Здесь потоки импульсов образуют кольца связанных нейронов, в которых пачки импульсов циркулируют с затуханием. Это есть кольца связанных нейронов, кольца реверберации, которые сформированы за счет горизонтальных связей в нейронных сетях. Эта циркуляция происходит с затуханием амплитуды сигналов, но ее достаточно, чтобы записать этот процесс в биохимической памяти пептидов и создать долговременную память о происходящем процессе. Отдельные импульсы от нейронов этих колец возвращаются в таламус, другие идут на гиппокамп и далее на таламус через хвостатые ядра мозга. Задержка импульсов в длинных ядрах мозга необходима для синхронизации прямых импульсов от проективных областей и задержанных в длинных ядрах мозга, в гиппокампе. Эта задержка создает долговременную память.

Ключевые слова: *материя, энергия, сознание, информация, мозг, мысль, идеи, cortex, нейрон, таламус, гиппокамп, реверберация, колонки, бочонки*

Большое число желающих определить, что такое **информация**, показывает, что все определения далеки от истины. Отсутствие четкого определения информации привело к большому количеству ошибок в рассуждениях разных авторов о тех или иных свойствах не только параметров предмета исследований, но и трактовке этих параметров. Информация неразрывно связана с сознанием, а это мало кто замечает. Поэтому нужно поставить вопрос, кто и когда изобрел информацию, или, когда она возникла. Если этот вопрос поставлен правильно, то в нем и надо искать ответ. Очевидно, что сведения о сознательной деятельности людей были записаны на бумаге 1000 лет назад, на папирусах 2000 лет назад, а последние археологические раскопки находят записи людей на камнях 10 и 20 млн лет назад. Сведения, которые человек получает от зрительного или слухового анализатора, необходимо зафиксировать в памяти и запомнить, для того чтобы в нужный момент можно было их воспроизвести. Запись сведений в коре головного мозга это и есть информация в отличие от предмета, который есть или материя m или энергия $mv^2/2$. А до этой записи — это сигналы, которые ее передают и записывают на разных носителях — все это лишь сведения, параметры, свойства, которые характеризуют объект. Наш коллега по общ. им. А. С. Попова Н. А. Коротков вплотную подошел к понятию информации [1]. Но, не имея достаточного уровня знаний по физиологии нейронных структур мозга, остановился на полпути, и далее посредством обобщений по психологии и философии совсем запутал этот вопрос, представив зрительный образ, как объективную реальность (ОР) и субъективную реальность (СР). Вот тут и встает этот деликатный философский вопрос, который рассматривал Коротков: "Где грань между данными о предмете, его параметрами и информацией о нем?" Анализируя работу Ленина о махизме, Коротков только запутывает этот вопрос, так как Ленин рассматривал его с позиций материализма с критикой ошибочных позиций идеализма. "Для истинного естествоиспытателя, не сбитого с толку профессорской философией, как и для всякого реалиста, ощущение есть действительная непосредственная связь сознания с внешним миром, есть превращение энергии внешнего раздражения в факт сознания..." Гениально! Вот по всем таким суждениям Ленина его мозг и был признан гигантским. О рассуждениях более объективных не могло быть речи, так как о работе мозга тогда было ничего неизвестно, а ответ надо искать в глубинах мозговой деятельности. И мы теперь имеем возможность судить об информации с позиций материальных исследования нейронных структур мозга и сравнить восприятие и представление информации человеком и компьютером,

одновременно разобраться с вопросом: чем отличается мышление человека от мышления компьютера. И когда заявляют, что человек может мыслить, а компьютер не может, это простая глупость, которую говорят узкие специалисты, не способные одним взглядом охватить весь спектр проблем, связанных с этим вопросом. Это опровергается игрой компьютера в шахматы, а также исправлением ошибок в тексте, когда компьютер некоторое время думает, а затем эти ошибки исправляет или подсказывает, как надо их исправить. И еще в ряде случае компьютер думает долго, и лишь после этого выдает решение. Что нужно компьютеру, чтобы он мог думать самостоятельно — это отдельная большая тема. Умственная работа компьютера во многих сферах деятельности превосходит способности человека, лишь в области идей человек пока значительно превосходит компьютер, но выдавать идеи могут лишь отдельные люди с резонансом в области идей [10] по шкале стадий переработки информации.

Мне как специалисту по мозгу, мышлению и схемам компьютеров и мониторов предстоит распутать этот клубок и дать четкое определение работы мозга в сфере сознания, а также дать строгое определение информации.

Разобраться с механизмом мышления пыталось очень большое число специалистов по мозгу: Прибрам, Хьюбел, Маунткасл, Бехтерева, Иваницкий, Сергин и др. Более ста авторов написали книги по мозгу и мышлению, но не достигли глубин в анализе связи работы нейронных ансамблей в коре головного мозга и связи этих структур с информацией как третьей субстанцией природы. Первая — материя, третья — сознание, а между ними находится информация. Из анализа всей литературы по этим вопросам следует вывод, что информация есть отображение внешнего мира в коре головного мозга. Слово отражение, которым пользовались раньше, устарело после развития теории функций комплексного переменного (ТФКП). Является ли такое отображение конформным или отображением какого-то другого вида — этот вопрос остается открытым. А главное, надо разобраться, как думает человек, и почему люди думают по-разному. Вот в чем заключалась проблема, которую ставил перед учеными Клинтон в 1990 г.: понять механизм мышления, чтобы разобраться с вопросом, почему люди плохо понимают друг друга. Надо было разобраться с этим вопросом за 10 лет. Это время уже прошло, а сдвигов не видно, воз и ныне там. Почему? Потому что для решения таких сложных вопросов необходимо привлекать мыслящих людей, а не тех, которые умеют списывать диссертации, использовать устаревшие мысли, играть словами и формулами. Надо выбирать людей со склонностью к поиску истины. Таким был В. М. Бехтерев, на творческом пути стояли Прибрам, Маунткастл, Бехтерева, Чернавский и ряд других, которые упорно исследовали работу мозга с целью разобраться с механизмом мышления, но так и не нашли его, хотя продвинулись заметно вперед; Я еще в школе заметил, что многие думают по-своему, не так, как другие. В институте эта вера усилилась, и после 5-го курса я сформулировал принцип избирательности мышления на основе постулатов статистической радиофизики. Этот принцип был сформулирован как теория избирательности [10]. Эта теория оказалась значительно больше и важнее теории относительности, но обосновать различия в видах мышления никак не удавалось, все логические соображения, все, что бралось в обоснование этих отличий, быстро разваливалось после соответствующих расчетов или чтения научной литературы по анатомии и физиологии мозга. Я даже перешел на работу в Институт им. Анохина, (1986), так понравились мне его многие высказывания по работе мозга.

И вот только в 2010 году, после того как я уехал из Москвы, где дышать было нечем вследствие большой загазованности воздуха автомобильными газами, прожил 8 лет в г. Домодедово, на свежем воздухе, мозги просветлели, стали сиять, идеи пошли сплошным потоком, и возникла идея, что все решает баланс вертикальных и горизонтальных связей в коре. Теперь это надо подробно обосновать и описать. Проекционные поля для всех сенсорных систем колончатые, а аналитические поля — у всех людей разные. У одних они — колончатые, у других — бочоночные и еще возможны разные промежуточные варианты.

Горизонтальные и вертикальные связи изучали Поляков, Глумов, Эдельман, Коган и другие, но никто серьезных выводов из этого не сделал. И вот теперь приходится снова возвращаться к этому вопросу при исследовании информации. Задача информации передать данную конкретную мысль так, чтобы она оказалась в коре мозга слушателей или читателей, или наблюдателей экрана телевизора, картины, компьютера и т. п. Она должна содержать мысль, т. е. то, что будет храниться в коре головного мозга.

Существует материя m, у нее есть энергия, это или mv^2 (кинетическая энергия), или АТФ (биологическая энергия), или gh (потенциальная энергия), и т. д. Между массой и энергией есть сведения о них, которые давно определяют как информацию, однако в понимании этого слова большая путаница. Большинство приняло определение Хартли, но многие с этим не согласны, и выдают свое определение и характеристики этого понятия. Чернавский дает все основные определения этого слова, но все они отображают устаревшие определения этого понятия. Давнее философское представление, что существует материя и сознание, в данном случае дает возможность отнести информацию к категории сознания. Но для этого необходимо определить

сознание достаточно четко и провести его физическое описание, определение и объяснение этого слова, исключить всякие спекуляции вокруг него, которых накопилось чрезвычайно много. Прежде всего отделим сознание, которое присуще только мыслящей материи, от осознания сигналов окружающей среды, присущее всем субъектам живой материи, всем биологическим телам природы [2, 7]. Следует напомнить, что все функции мозга замыкательные. Осознание происходит в коре головного мозга ***при замыкании данного поля коры через гиппокамп на таламус***. Пачки импульсов, паттерны, проходят по этому пути широким потоком и замыкание этого потока приводит к осознанию того предмета, который выделяет гиппокамп из всех других импульсных потоков. Такое замыкание приводит работу мозга к тому, что осознается только этот предмет, а все остальное оказывается за пределами сознания. Мозг думает только об этом предмете (рис. 1). Сигналы со зрительного [2—4, 8] сенсорного поля поступают через блок GL таламуса на проекционное поле коры 17 (обозначения полей по Бродману), с него на ассоциативное поле 19, где образуются кольца rev-ring, от этого кольца идут импульсы в поле 20, в котором сходятся сигналы со всех ассоциативных полей, и далее с него через hcg (структура hippocampalgyrus) на hippocampus и через Corpus Mammilarus на таламус. Осознание произошло, если не было никаких сбоев и помех. То же самое имеет место в звуковых полях коры (41, 22). Переключение внимания на другой предмет осуществляет таламус, импульсы с которого проходят через гиппокамп и отыскивают в нем, как в картотеке, данные об этом новом предмете. Затем включается в замыкательную функцию мозга соответствующее поле коры, и после этого в сознании оказывается этот новый предмет, мозг думает о нем. Гиппокамп по моим исследованиям [3, 4] (рис. 2) является устройством с запаздывающей обратной связью. Вот чего не знают физиологи, поэтому не могут разобраться с этим блоком. Эта связь необходима для выравнивания времени реверберационного кольца (rev-ring) в коре со временем задержки в таламусе, которое намного короче. Вот на этом принципе и осуществляется длительная память. Доказать это экспериментально довольно трудно, поэтому все выводы по изучению механизма мышления должны осуществляться на основе "Логических экспериментов".

Литературных данных о работе нейронных структур и коры накопилось достаточно много, пора проводить их логическую обработку в глобальном масштабе, а не по отдельным функциональным системам. Чтобы память была действительно длительной, необходимо эту замыкательную функцию повторять несколько раз, чтобы изучаемый материал был закреплен на уровне пептидов, которые дают биохимическое закрепление связей между аксонами и синапсами соответствующих колец. На рис. 2 рассматривается схема температурной регуляции в организме, где каждой температуре соответствует своя нейронная задержка со своим rev-ring, и импульсы через hpg и Corpus Mam. поступают не на таламус, а на главный блок физиологической регуляции — гипоталамус, который повышает или понижает температуру тела по сигналам рецепторов T_p и X_p. В этой регуляции задействованы поля коры 7, 13, 14, 20, в которых образуются кольца температуры тела.

Реверберацию мало кто понимает, поэтому Эдельман, Бехтерева, Сергин, Иваницкий пользуются словесным описанием этого явления, называя его "циклическим процессом прогонки информации". Я больше 10 лет работал с реверберaторами, вначале с ленточными, затем с цифровыми. Ввиду такой длительной работы с этими устройствами я всем своим мозгом и нутром прочувствовал, что мысль — это реверберационный процесс. Это явление замечали Беритов, Лоренто де Но, Верцеано,

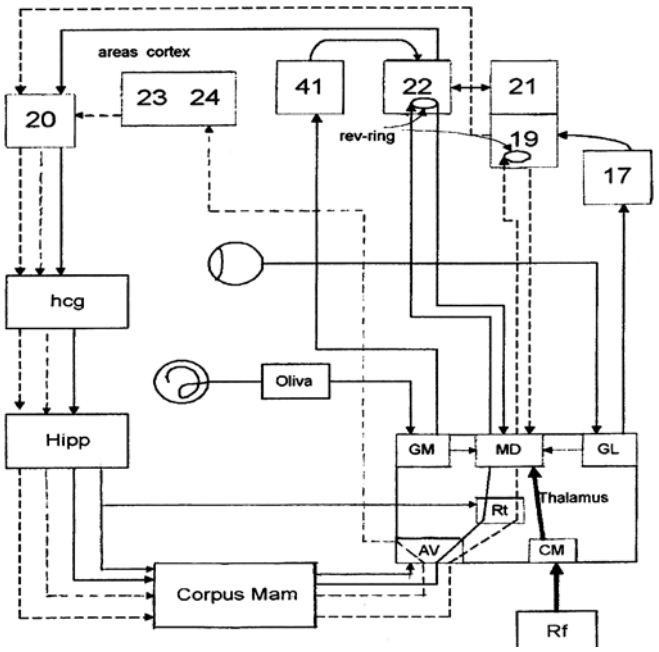


Рис. 1

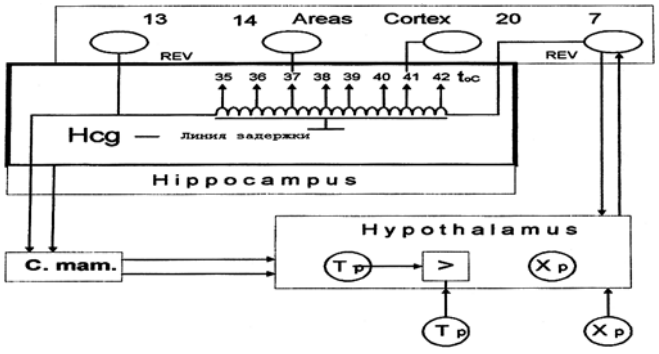


Рис. 2

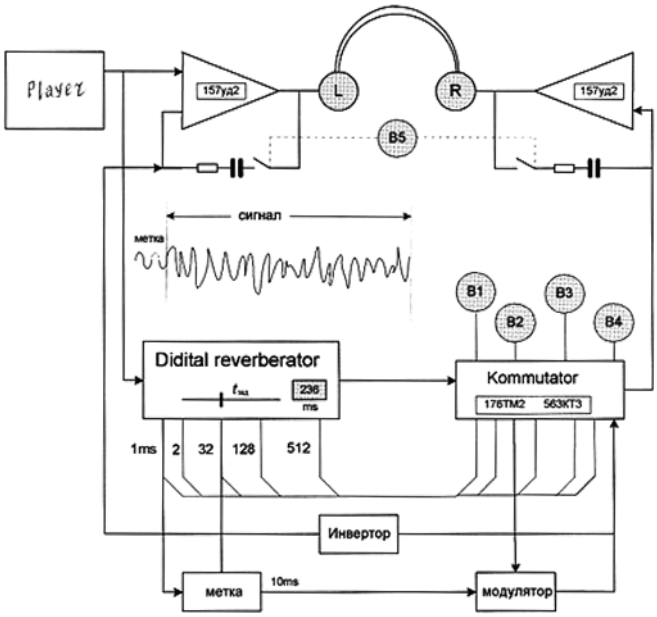


Рис. 3

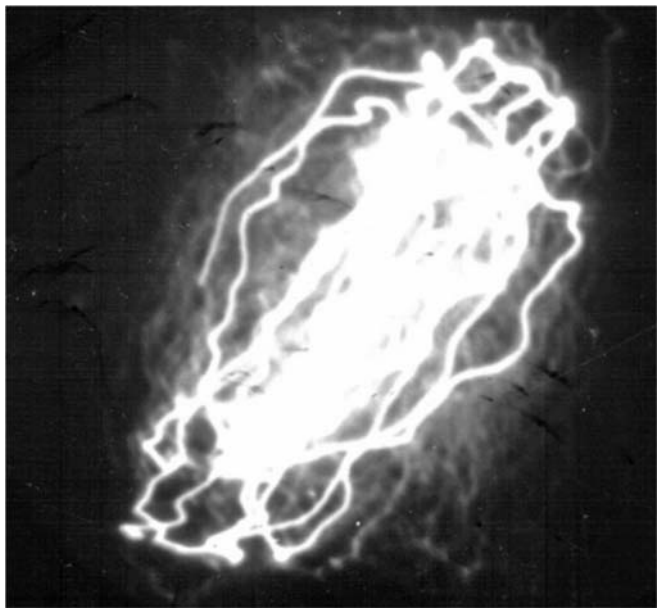


Рис. 4

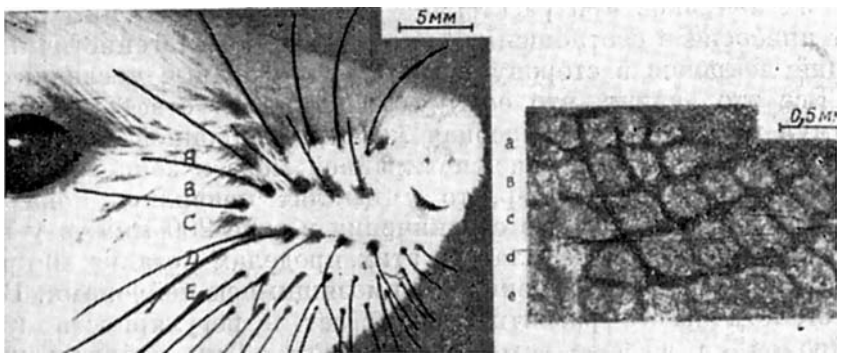


Рис. 5

Маунткасл, но прочувствовать мало, надо осмыслить, чтобы объяснить это явление другим исследователям этого вопроса. Сразу следует заметить, что мысли бывают быстрыми и медленными — от 30 мс в случае быстрой речи до 350 мс в случае медленной мелодии, и для каждой из них необходимо свое время реверберации, 3—10—15 с [6, 9]. Это хорошо проверять с помощью мелодичной музыки. Так, с помощью схемы (рис. 3) можно описать механизм ***осознания*** современными средствами радиоэлектроники, регулируя время задержки и время реверберации. В этой схеме можно сфотографировать "мысль" (рис. 4), фазовая картина тока в правом и левом динамике. Обычно длина rev-ring меняется от 50 до 400 мс в зависимости от ритма, с которым идет музыка или другие звуковые сигналы. Время реверберации можно регулировать от 1 до 10 с. Задержка сигнала в одном наушнике приводит к смещению точки восприятия звука вдоль меридиана, проходящего через уши, и при этом не наблюдается никаких различий между правым и левым полушарием мозга, о чем твердят поклонники принципа Сперри.

В мозге время реверберации определяется скоростью выработки нейропептидов, которая с возрастом падает, и человек не может вспомнить, о чем он думал 5 с назад. Пока реверберационный процесс не затух полностью, можно направить внимание на то, какую погоду сказали, и восстановить цифру. С возрастом медиаторы вырабатываются настолько плохо, что становится трудным вспомнить, о чем подумал или слышал 3—4 с назад. Но выработку медиаторов (и других пептидов) можно существенно увеличить, если применить инсулин, соматотропный гормон и большую двигательную активность. А также слушать реверберацонные сигналы.

Механизм формирования мысли и ее запись в нейронных структурах относится ко всем видам биологических тел, у которых мозг имеет кору и таламус, этот вывод следует из большого объема исследований за 100 лет. В коре этих животных прописываются замкнутые реверберационные кольца, которые дают подробное описание циклических движений. Например, движение усов грызунов имеет многочисленные подтверждения разными экспериментами (рис. 5). В коре образуются заметные следы связей между нейронами, соответствующие движению усов грызуна. Схема, приведенная на рис. 3, была разработана для измерения величины rev-ring. В таламусе имеются соответствующие им малые rev-rings (Verceano, Беритов и др.), которые являются картотекой колец в коре. Движение этих колец в коре и в таламусе осуществляется синхронно и характеризует движение мысли. Этими движениями исходно управляют ядра

ретикулярной формации. Аналогично у человека можно проследить образование циклических колец в коре, если он будет повторять периодически какую-то фразу или мелодию и постепенно превратит ее в безусловный рефлекс. Подробнее этот механизм здесь не рассматривается, так как мы исследуем информацию, а не принципиальную схему работы мозга.

У человека существуют большие пласты таких реверберационных колец из разных областей знаний, за счет чего у него образуется СОЗНАНИЕ, способное думать обобщенно о разных проблемах жизни и сопоставлять между собой эти проблемы. Причем перескоки с одного кольца на другое происходят за несколько миллисекунд, без вовлечения в этот цикл гиппокампа и всей замыкательной функции мозга. Отсюда многие замечают, что самое быстрое — это мысль, так как перескок происходит без включения в этот процесс rev-ring. С помощью такого СОЗНАНИЯ можно составить представление об ***информации*** с учетом того большого пласта отклонений от объективного представления, которое выработалось за долгое время ввиду отсутствия строгого определения этого понятия. Здесь следует отметить, что причиной заблуждений в исследовании понятия информации является узкая специализация большинства ученых. Специалисты с узким спектром знаний не могут разобраться с предметом исследований, который требует знания многих смежных наук. Анатомия, физиология, гистология и биохимия являются основными предметами для того чтобы понять информацию как элемент сознания, который формируется в коре и связанных с ней структурах мозга. И надо хорошо знать еще радиотехнику, теорию информации, импульсную технику, работу компьютера и его программы, а также машинные языки мозга и компьютера, и многое другое. И вот вследствие того, что исследователи информации всего этого не знают, они путаются в решении вопроса определения информации, выдают свои субъективные суждения, основанные на их представлениях.

Какие представления об информации мы имеем на сегодняшний день? Рассмотрим главные. Первое дают Хартли, Винер, Шеннон и др. Все они рассматривают частные случаи информации, в основном выделения предметных значимых сигналов в средствах связи при наличии шумов или помех. В свое время это была важная проблема, но в настоящее время, когда мы перешли на спутниковое и цифровое телевидение, когда мобильная связь обеспечивает чистый звук связи, о шумах можно забыть, и к информации подходить с других позиций. Другие неортодоксальные представления об информации Саночкина, инж. Макеева, Туманова ... и автора являются ошибочными или нелогичными, или сугубо субъективными. Повторяю, информация в каналах ее передачи есть сведения про объект материи ***m***, эта передача тоже материальна, энергия ***gh***, или ***pV*** тоже есть материя. Большинство признает, что информация занимает промежуточное положение между материей и энергией, отсюда появляется гибрид **миэ**, который является основным атрибутом окружающего нас мира. Однако философский анализ показывает, что важнее признать разделение мира на материю и сознание. Логично информацию отнести к области сознания. Но это надо обосновать. Материалистический взгляд на этот вопрос сводится к тому, что материальные предметы отражаются в сознании человека. Вот это отображение и было переведено здесь на физический язык, что давно пытаются сделать в журнале "Сознание и физическая реальность". Но спектр их знаний не позволяет этого сделать. Узкие специалисты не могут решать многие из современных проблем, так как процесс познания достиг такого уровня, когда простого и даже вдумчивого созерцания недостаточно для открытия новых истин. Вот тут и назревает необходимость логического экспериментирования. И этот же метод необходимо применить к вопросу происхождения человека и возникновения жизни на Земле.

Информация хранится в коре головного мозга, все остальные носители информации на бумаге, на пластинке, на магнитной ленте, на кинопленке, на дискете, на DVD и, наконец, на флешке — все это материя, ***m***, это не информация, а ее материальные носители. Информация — это отображение предмета в коре головного мозга. Но по установившейся традиции все материальные носители сведений называются информацией, и отойти от этой тенденции уже невозможно. Наскальные записи, сделанные еще в меловой период миллионы лет назад, также являются материальными носителями информации, которые превращаются в истинную информацию только после того, как ее расшифруют специалисты археологии в коре своего мозга. Затем они разъясняют нам это на современном понятном нам языке, и мы узнаем, как жили предыдущие цивилизации, что они нам хотели сообщить и далее понять, почему они погибли.

Современное представление об информации подразумевает передачу и хранение сведений о материальных объектах на бумаге, на магнитной ленте — и в последнее время на микросхемах памяти в виде двоичных кодов. Все это может быть выражено в математических формулах, в результате к игре слов разной информации прибавилась игра с формулами. Однако к решению таких вопросов необходимо подключать образное мышление! Независимо от того, на какой носитель будет записана передаваемая информация, она в итоге попадет в кору головного мозга, где должна быть осознана. Без осознания, если она не содержит смысла, информация будет вскоре забыта, окажется просто поте-

рянной или записанной на механический носитель, где она будет храниться для будущих поколений.

Отсутствие четкого определения информации привело к большому числу ошибок в рассуждениях разных авторов о тех или иных свойствах не только сведений и параметров предметов исследований, но и в трактовке этих параметров. И главный вопрос: когда появилась информация? Нельзя началом возникновения информации считать работы Винера и Шеннона, ибо информация возникла намного раньше появления радио и кибернетики. Вначале информация передавалась с помощью записей на бумаге. Отсюда следует вывод, что информация формируется в коре головного мозга человека и предназначена для передачи ее другим людям. А также потомкам. Надо признать, что информация возникла с появлением сознания у человека. Именно он каждому предмету придумал графическое изображение, буквенное описание, звуковые оформления, были сигналы-слова, отличные от звуков животных. Можно ли считать звуки животных информацией, например, пение птиц? Нет, в них нет мелодии, значит, в них нет информации. Так же как в музыке без мелодии. Это бессмысленные звуки, они ничего не передают, кроме того, что есть он, кто издает эти звуки. Так же как вой волка или лай собаки. Но он сообщает, что кто-то появился. Это не информация, это сообщение, но это сообщение многие могут считать информацией, пусть примитивной, но все же информацией. Для кого? Любая информация для кого-то предназначена. Эта информация предназначена для хозяина собаки. Или сообщение другим волкам, что он есть, и по интонации другие должны понять, что он сообщает. Также и люди в древнем мире передавали свои соображения другим людям с помощью звуков или с помощью записей на песке, на деревьях, на камнях.

Требование сокращения единиц информации для отображения конкретной мысли стало насущной необходимостью нашего времени. Ценность информации $Z = C_{\text{мысл}}/\Sigma_{\text{слов}}$, только так можно уплотнить большое количество сведений для выделения главной существенной мысли. Не игра слов, а поток мыслей позволит нам уплотнить мышление человека, поднять его на такую высоту, с которой будут четко видны решения всех вопросов и территориальных, и национальных, и политических, и экономических и, главное, экологических. Ибо экология через два года станет главным вопросом современности. Об этом много говорили на всех конференциях и на всех уровнях. Но на высоких уровнях на это не реагируют, так как не понимают существа проблемы и не находят подходящего выхода. Все хотят энергии все больше и больше, а это чревато экологической катастрофой. Перерасход энергии человечеством, превышение нормы, стало недопустимо большим. Норма равна 3—4 млрд т. н. э., а люди сейчас вырабатывают больше 14 млрд т. н. э. Также и математики играют с формулами, как болтливые люди со словами. Игра с формулами, как и игры со словами, являются достоянием людей созерцающих, мало думающих. Но не тех, кто ищет смысл или истину. Смысл, как правило, есть только там, где поиск истины завершается успехом, т. е. новой идеей или открытием. В большинстве случаев большое число слов доказывает, что смысла в них нет. Новый подход к определению информации показывает, что через какие бы каналы мы не передавали информацию, она всегда приходит в кору головного мозга человека. Значит, информация — это средство передачи сведений о предметах или явлениях материального мира в кору головного мозга. В соответствии с предыдущей статьей по этой теме [8], чем меньше слов уходит на описание данного вопроса, тем более ценной является информация. Поэтому я стараюсь быть кратким. В соответствии с теорией функций комплексного переменного один объект с пространственной характеристикой может быть трансплантирован в другое пространство и такое преобразование является конформным, если углы пересечения ортогональных линий сохраняют угол и направление. Простой звук, записанный на магнитофоне, будет представлен соответствующей комбинацией магнитных доменов на магнитной ленте. Это есть конформное отображение. Этот же звук может быть записан на HDD компьютера, где сигнал будет предварительно закодирован в двухтактном виде по системе работы жесткого диска в компьютере. Это тоже конформное отображение, но по совсем другой формуле. После прохождения этих сигналов через соответствующие декодеры мы получим тот же звук, который в итоге попадает в слуховые поля головного мозга человека.

Список литературы

1. **Коротков Н. А.** К вопросу о роли информации в механизме формирования зрительного образа в сознании человека // Труды РНТО РЭС им. Попова. 2007. Вып. LXII. С. 105—107.
2. **Голубев В. И.** Механизм осознания и мышления // Труды РНТО РЭС им. А. С. Попова. 2005. Вып. LX. С. 87—91.
3. **Голубев В. И.** Функциональная схема двигательной регуляции и стриарного демпфирования колебаний. Майская сессия НТО РЭС. Москва, 2000. С. 222—224.
4. **Голубев В. И.** Замыкательная функция мозга в системе физиологической регуляции. Труды LVII Научной сессии НТО РЭС. 2002. Т. 2. С. 45—46.
5. **Батуев А. С.** Нейрофизиология коры головного мозга. Л.: Университет, 1984. 219 с.
6. **Сергин В. Я.** Сознание как система внутреннего видения // ЖВНД. 1994. Т. 44, вып. 4—5. С. 627—639.
7. **Голубев В. И.** Автомобильные проблемы. М.: Сайнс-Пресс, 2006. С. 90—97.
8. **Голубев В. И.** Новый подход к определению ценности информации // Доклады РНТО РЭС. 2014. Вып. LXIX. С. 132—136.
9. **Голубев В. И.** Акустический способ выключения сознания при измерении длительности энграммы // Труды РНТО РЭС им. Попова. 2007. Вып. LXII. С. 166—169.
10. **Гвидон В.** Паралич власти. М.: Прогресс, 1993. С. 26—28.
11. **Прибрам К.** Языки мозга. М.: Прогресс, 1975. 464 с.

V. I. Golubev, Ph. D.,
RNTO im. A. S. Popov

Information is Display of Subjects of World Around in a Cerebral Cortex

The mechanism of consciousness is formed in this way. The signals from the object through a visual or acoustic analyzer fall within the field of projection of the cerebral cortex and are recorded in the columns in the area. Then they are transferred to the field of associative cortex. There momentum flux forming rings connected neurons, wherein a burst circulate with damping. It has rings connected neurons, rings reverb, which are formed by horizontal connections in neural networks. This circulation occurs with attenuation of the signal amplitude, but it is enough to record the process in biochemical memory peptides and create lasting memories of an event process. Separate impulses from neurons in the rings back to the thalamus, hippocampus, others go on and on to the thalamus through the caudate nucleus of the brain. The delay in the long pulse brain nuclei required for direct synchronization pulses from the projective areas and detained in long nuclei of the brain, the hippocampus. This delay creates a long-term memory.

Keywords: *matter, energy, consciousness, information, brain, thoughts, ideas, cortex, neuron, thalamus, hippocampus, reverb, kolumns, drums*

References

1. **Korotkov N. A.** To a question of information role in the mechanism of formation of a vision in consciousness of the person, *Works RNTO RES of Popov,* 2007. Is. LXII, pp. 105—107.

2. **Golubev V. I.** Mekhanizm of understanding and thinking, *Works RNTO RES of A. S. Popov*, issue LX, Moscow, 2005, pp. 87—91.

3. **Golubev V. I.** Function chart of motive regulation and striarny damping of fluctuations, *May session of NTO RES,* Moscow, 2000, pp. 222—224.

4. **Golubev V. I.** Switching function of a brain in system of physiological regulation, *The LVII Scientific session of NTO RES, Works*, 2002, vol. 2, pp. 45—46.

5. **Batuyev A. S.** *Neyrofiziologiya of a cerebral cortex*, L, University, 1984, 219 p.

6. **Sergin V. Ya.** Soznaniye as system of internal vision, *ZhVND*, 1994, vol. 44, no. 4—5, pp. 627—639.

7. **Golubev V.** *The Motor-car problems*, Moscow, Sayns-Press, 2006, pp. 90—97.

8. **Golubev V. I.** New approach to determination of value of information. *Reports of RNTO RES*, issue LXIX, 2014, pp. 132—136.

9. **Golubev V. I.** Akustichesky a way of switching off of consciousness at measurement of duration of an engramma, *Works RNTO RES of Popov*, issue, 2007, pp. 166—169.

10. **Gvidon V.** *Paralich of the power*, Moscow, Progress, 1993, pp. 26—28.

11. **Pribram K.** *Languages of a brain*, Moscow, Progress, 1975, 464 p.

VII Всероссийская конференция

"Проблемы разработки перспективных микро- и наноэлектронных систем"

(серия "МЭС")

МЭС-2016

пройдет 03—07 октября 2016 г. в Зеленограде

Среди учредителей конференции — Российская академия наук, Российский фонд фундаментальных исследований, ФАНО России, Национальный исследовательский университет "МИЭТ", Южный федеральный университет, Префектура Зеленоградского АО г. Москвы, ОАО "НИИМЭ и Микрон". В число официальных партнеров и спонсоров конференции входят Фонд образовательных и инфраструктурных программ Роснано, Фонд "Сколково", фирма Intel, ПАО "ИНЭУМ им. И. С. Брука", ЗАО "ПКК Миландр", ОАО НПЦ "ЭЛВИС", ЗАО "Мегратек" и многие другие предприятия. Организатор конференции — Федеральное государственное бюджетное учреждение науки Институт проблем проектирования в микроэлектронике Российской академии наук (ИППМ РАН), соорганизаторы конференции — Казенное предприятие г. Москвы "Корпорация развития Зеленограда" и Московское научно-техническое общество радиотехники, электроники и связи имени А. С. Попова.

Рейтинг конференций серии "МЭС" неуклонно растет. Увеличивается число участников (в 2014 г. — более 230), к организации конференции привлекаются новые предприятия. Труды конференции, в том числе англоязычный реферативный сборник статей, включены в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук (Перечень ВАК). В текущем году планируется включение Трудов конференции в раздел Russian Science Citation Index (RSCI) базы научных публикаций Web of Science.

Как и прежде, конференция "МЭС" этого года посвящена актуальным вопросам автоматизации проектирования микроэлектронных схем, систем на кристалле, IP-блоков и новой элементной базы по следующим научным направлениям:

- Теоретические аспекты проектирования микро- и наноэлектронных систем (МЭС);
- Методы и средства автоматизации проектирования микро- и наноэлектронных схем и систем (САПР СБИС);
- Опыт разработки цифровых, аналоговых, цифроаналоговых, радиотехнических функциональных блоков СБИС;
- Особенности проектирования СБИС для нанометровых технологий;
- Системы на кристалле перспективной РЭА.

Также особое внимание будет уделено вопросам разработки интегральных схем для космического применения.

В рамках конференции пройдет сессия-презентация научно-технических достижений российских и зарубежных компаний, а также организаций, способствующих развитию микроэлектроники и информационных технологий в России. Запланировано проведение выставки программных продуктов и оборудования, используемого в микроэлектронной отрасли.

Более подробную информацию о конференции можно получить на web-сайте http://www.mes-conference.ru.

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Т. В. Пчелкина*.

Сдано в набор 25.12.2015. Подписано в печать 24.02.2016. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT316. 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 "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".

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