

# Международный журнал информационных технологий и энергоэффективности |



Том 5 Номер 3(17)



2020



## СОДЕРЖАНИЕ / CONTENT

### ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- 
1. **Балашов О.В., Букачев Д.С.** Методика оценки труда научных работников и отчётов о научно-исследовательских работах **3**

**Balashov O.V., Bukachev D.S.** Methodology of Estimating the Labor of Scientific Workers and Reports on Scientific Research Works

- 
2. **Зернов М.М., Картавенков А.А.** Способ и алгоритмы реализации генетического оператора рекомбинации на основе турнирной стратегии со случайным исходом **11**

**Zernov M.M., Kartavenkov A. A.** Method and algorithms for implementing the genetic recombination operator based on a tournament strategy with a random outcome

- 
3. **Картавенков А.А.** Оценка турнирных стратегии со случайным исходом в сравнении с другими реализациями генетических операторов **21**

**Kartavenkov A. A.** Evaluation of tournament strategy with a random outcome in comparison with other implementations of genetic operators

- 
4. **Ерш В.С.** Идентификация и анализ устойчивости кластеров социотехнических систем на основе нечеткого когнитивного подхода **32**

**Ersh V.S.** Identification and analysis of sustainability of clusters of sociotechnical systems based on a fuzzy cognitive approach

- 
5. **Раскатова М.В., Новиков К.Д.** Особенности организации и функционирования сетей провайдера **43**

**Raskatova M.V. Novikov K.D.** Peculiarities of organization and functioning of provider networks

---



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 681.3.06

## МЕТОДИКА ОЦЕНКИ ТРУДА НАУЧНЫХ РАБОТНИКОВ И ОТЧЁТОВ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТАХ

<sup>1</sup> Балашов О.В., <sup>2</sup> Букачев Д.С.

<sup>1</sup> Смоленский филиал АО «Радиозавод», Россия, (214027, г. Смоленск, улица Котовского, 2), e-mail: smradio@mail.ru

<sup>2</sup> ФГБОУ ВО Смоленский государственный университет, Смоленск, Россия (21400, г. Смоленск, ул. Пржевальского, 4), e-mail: dsbuka@yandex.ru

Методика предназначена для определения рейтингов научных работников научно-исследовательских подразделений с целью адекватной оценки результативности их труда. На основе обобщения опыта работы в экспертном совете научно-исследовательской организации предложены критерии оценки, а также определены методы получения значений этих критериев.

Ключевые слова: компетенция, решение, лингвистическое описание.

## METHODOLOGY OF ESTIMATING THE LABOR OF SCIENTIFIC WORKERS AND REPORTS ON SCIENTIFIC RESEARCH WORKS

<sup>1</sup> Balashov O.V., <sup>2</sup> Bukachev D.S.

<sup>1</sup> Smolensk branch of joint-stock company "Radio factory", Russia, (214027, Smolensk, street Kotovskogo, 2), e-mail: smradio@mail.ru

<sup>2</sup> Federal State Educational Institution of Higher Education Smolensk State University, Smolensk, Russia (21400, Smolensk, street Przewalski, 4), e-mail: dsbuka@yandex.ru

The methodology is designed to determine the ratings of scientific workers of research units in order to adequately assess the effectiveness of their work. Based on the generalization of experience in the expert council of a research organization, evaluation criteria are proposed, and methods for obtaining the values of these criteria are determined.

Keywords: competence, decision, linguistic description.

В практике работы отечественных научно-исследовательских организаций проблема адекватной оценки труда отдельного научного сотрудника до сих пор остается нерешённой. Это означает, что модели трудовых отношений и управления человеческим фактором должны быть серьезно реформированы в направлении существенно лучшего (на порядок) использования творческих способностей человека. Творческие способности человека проявляются через его *компетенцию*, которая является центральным понятием всей концепции управления персоналом.

*Компетенция* – это рациональное сочетание способностей, личностных качеств и мотивации персонала организации, рассматриваемых во временном интервале [1, 2]. Компетенция персонала – категория динамичная, развивающаяся. Она может быть повышена

за счёт постоянного обучения, самообразования и особенно мотивации. Поэтому задачами управления персоналом является повышение результативности его работы, которая зависит от компетенции и мотивации.

Отсюда следует вывод о том, что для повышения эффективности работы организации необходимо, во-первых, развивать компетенцию каждого работника и, во-вторых, совершенствовать формы мотивации труда. Развитие компетенции может осуществляться как за счёт профессионального образования, организуемого организацией, так и за счёт самообразования работника.

В большинстве научно-исследовательских организаций в оценке результативности участвуют, как правило, руководители всех рангов, а также кадровые работники. Они должны выработать критерии оценки и методы получения значений этих критериев. Выбор критериев оценки результативности труда работника зависит от его принадлежности к той или иной категории работников: руководители, научные работники и другие служащие.

В данной статье речь пойдёт об оценке результатов труда научных работников – непосредственных исполнителей научных исследований. Результаты труда научных работников определяются, исходя из объёма, качества, своевременности, полноты выполнения закреплённых за ними должностных обязанностей.

В практике работы различных организаций наибольшее распространение получили такие методы оценки, как графическая шкала оценки; метод альтернативного ранжирования; метод принудительного распределения; метод критических случаев; рейтинговые шкалы, привязанные к качеству работника; метод управления по целям [3, 4, 5].

В таблице 1 приводятся результаты сравнения методов аттестации [3].

Таблица 1 – Достоинства и недостатки методов оценки результатов деятельности работников

№ п/п	Методы	Достоинства	Недостатки
1	2	3	4
1	Графическая шкала оценивания	Прост в применении; обеспечивает количественный рейтинг для каждого работника	Нормы могут быть неясными, и проблемы могут возникнуть из-за эффекта ореола, центральной тенденции, мягкости, предубежденности
2	Альтернативное ранжирование	Прост в применении; избегает центральной тенденции и др. проблем рейтинговых шкал	Может вызвать несогласие работающих и несправедлив, если все работающие отлично исполняли обязанности
3	Метод принудительного распределения	Неизменное определенное число подчиненных в каждой группе	Результаты оценивания зависят от адекватности первоначального выбора точек отсечения
4	Метод критических случаев	Помогает объяснить работающему, что значит "хорошее" и "плохое" исполнение обязанностей; заставляет тестирующего оценивать подчиненных на основе поведения	Трудно ставить ранги работающих, отличая их друг от друга

Неудовлетворенность организаций традиционными методами аттестации побуждает их искать новые методы оценки персонала. Выбор метода оценки персонала для каждой конкретной организации является уникальной задачей, решить которую может только руководитель самой организации. Одним из подходов к оценке труда научных работников и отчётов о научно-исследовательских работах является отчётно-рейтинговый контроль.

В отчётно-рейтинговой технологии имеется два основных понятия – это отчет и рейтинг [3]. **Отчет** – это логически завершенная часть (раздел, подраздел, пункт и т.д.) отчета о научно-исследовательской работе (НИР). **Рейтинг** представляет собой сумму баллов, набранную научным работником (по определенным правилам) в течение некоторого промежутка времени. Рейтинг может быть текущим (сумма баллов на данный момент времени) и итоговым (полная сумма баллов за всю работу за определенный промежуток времени, как правило, за год). Применение рейтингового контроля дифференцирует научных работников по компетенции и отношению к проводимым научным исследованиям, определяет необходимость их индивидуальной подготовки.

Для активизации научной деятельности работников вводятся коэффициенты. Их роль заключается в дифференциации оценки в зависимости от качества проведения научных исследований, сложности и своевременности сдачи отчётных материалов исследования, оформительской культуры, элементов творчества и т.д.

Баллы, набранные за отчеты, являются составной частью рейтинга. В период проведения исследований может проводиться несколько контрольных мероприятий для определения правильности выбранной (разработанной) методики научных исследований, оценки обоснованности полученных результатов, соответствия направлений проводимых исследований техническому заданию и др. Формы проведения контроля могут быть различными.

Рейтинг формируется на основе баллов, набранных за отчеты. Суммарный рейтинг за работу (А) определяется по формуле

$$A = ((K_1 + K_2) K_3) 100, \dots \dots \dots (1)$$

где 100 – коэффициент для более удобного восприятия значения рейтинга.

Предлагаются следующие показатели.

*Показатель научной новизны  $K_1$*  используется для оценки достигнутых в отчете новых научных методов и результатов исследования.

*Показатель качества изложения материала  $K_2$* . Его значение может изменяться в интервале от 0 до 0,3 в зависимости от количества и качества замечаний, полученных за отчётный материал. Даже если задание выполнено не полностью, у научного работника есть возможность получить баллы за те элементы разработанного материала, с которыми он справился успешно.

*Показатель своевременности выполнения задания  $K_3$*  может значительно влиять на рейтинг, поэтому стимулирует научного сотрудника в срок отрабатывать и сдавать задания в отчет о НИР.

*Показатель научной новизны  $K_1$*  определяется по следующей формуле:

$$K_i = 0,7 (Q_{ni} + Q_{mi} + Q_{pi}), \dots \dots \dots (2)$$

где  $i$  – частная задача исследования, имеющая существенное самостоятельное научное (техническое) значение, решению которой посвящен раздел (подраздел) НИР;

$Q_{ni}$ ,  $Q_{mi}$ ,  $Q_{pi}$  – показатели, характеризующие уровень новизны соответственно постановки, метода решения и результатов решения  $i$ -й частной задачи.

Выделение частных научно-технических задач, имеющих существенное самостоятельное значение, а также определение конкретных значений всех показателей осуществляется коллегиальным решением специалистов (экспертным советом и т. п.), привлекаемых для оценки качества отчета (публикации), при этом за основу берутся и критически анализируются предложения авторов. При обнаружении некорректностей, отрицающих правильность (достоверность) того или иного оцениваемого элемента, его предполагаемая новизна подвергается сомнению, а отчет возвращается на доработку (переработку).

Примеры научно-технических задач, имеющих существенное самостоятельное значение:

- оценка эффективности функционирования системы;
- обоснование требования к устойчивости системы (надежности изделия);
- расчет оптимального значения параметра устройства;
- обоснование решения (рекомендации, предложения) и другие.

Под методом обоснования результатов решения задачи подразумевается выбранный или разработанный с той или иной степенью обоснованности метод решения научной задачи. Оценке подлежат лишь те элементы решения задачи, которые приведены в публикации; ссылки в этом отношении на другие публикации во внимание не принимаются за исключением авторских работ, публикуемых впервые. Отсутствующие элементы обоснования оцениваются самыми нижними рекомендуемыми значениями показателей.

Отсутствие новизны соответствует случаю, когда

- во-первых, рассматриваемый (оцениваемый) на новизну способ, методика, метод уже предложены и опубликованы;
- во-вторых, рассматриваемые материалы уже использовались в других работах научно-исследовательской организации.

Значения показателей уровня новизны (обоснованности) представлены в таблице 2.

Таблица 2 – Значения показателей уровня новизны (обоснованности)

№ п/п	Элементы отчета	Значения $Q_{ji}$
1	2	3
1	Постановка задачи исследования	0; 0,1; 0,2
2	Метод решения задачи исследования	От 0 до 0,6
3	Результат решения задачи	0,1; 0,2

Показатель характеризующий уровень новизны постановки задачи исследования ( $Q_{оп}$ ), может иметь одно из нескольких значений. При отсутствии постановки задачи исследования в явном виде или при использовании известной постановки, показателю уровня новизны присваивается значение  $Q_{оп} = 0$ . В том случае когда постановка задачи осуществлена только

на вербальном уровне (словесное описание), тогда значение  $Q_{оп} = 0,1$ . И наконец,  $Q_{оп} = 0,2$ , когда постановка задачи формализована, то есть осуществлена параметризация, обоснована мера для параметров и доказано математическое соотношение, связывающее переменные.

Для оценки метода решения задачи исследования используются семь значений показателя  $Q_{ом}$ , приведенных в таблице 3.

При использовании известного (аналитического или логического) метода решения задачи коэффициент  $Q_{ом} = 0$ . Если для решения задачи использовались эвристический метод (эрудиция + интуиция) или логический анализ статистических данных, тогда значение коэффициента  $Q_{ом} = 0,1$  или  $0,2$  соответственно. В случае если проведено количественное обоснование метода с помощью известных, но усовершенствованных методик и моделей, т.е. использование отдельных, вновь учитываемых коэффициентов в известной методике или новой расчетной формулы (соотношения), тогда  $Q_{ом} = 0,3$ . Когда количественное обоснование осуществлено с помощью расчётных методик, математических моделей и математических доказательств, обладающих несомненной новизной или подтверждение результата экспериментальной проверкой, или случай результата, не вызывающего сомнения по достоверности (акт реализации, наличие патента на изобретение), то  $Q_{ом}$  присваивается значение от  $0,4$  до  $0,6$  (см. таблицу 3).

Таблица 3 – Оценка метода решения задачи исследования

№ п/п	Метод решения	$Q_{ji}$
1	2	3
1	Известный	0
2	Эвристический (эрудиция + интуиция)	0,1
3	Логический анализ статистических данных	0,2
4	Количественное обоснование с помощью известных методик и моделей (усовершенствованный метод), использование отдельных, вновь учитываемых коэффициентов в известной методике	0,3
5	Количественное обоснование с помощью известных методик и моделей (усовершенствованный метод), использование новой расчетной формулы (соотношения)	0,4
6	Количественное обоснование с помощью расчетных методик, математических моделей и математических доказательств, обладающих новизной	0,5
7	Подтверждение результата экспериментальной проверкой, или случай результата, не вызывающего сомнения по достоверности (акт реализации, наличие патента на изобретение)	0,6

Для результатов решения задачи коэффициенты определяются следующим образом. Если результат получен с использованием известного метода или использовался эвристический метод, тогда значение  $Q_{ор} = 0,1$ . В случае если проведено количественное обоснование результатов с помощью расчетных методик, математических моделей и математических доказательств, усовершенствованных или обладающих новизной, тогда  $Q_{ор} = 0,2$ .

Оценка научной новизны научно-исследовательской работы выставляется по данным таблицы 4.

Таблица 4 – Оценка научной новизны научно-исследовательской работы

№ п/п	Значение $K_2$	Оценка качества изложения материала	Примеч.
1	2	3	4
1	0,700 – 0,600	<b>Очень высокая</b> научная новизна работы	
2	0,599 – 0,400	<b>Высокая</b> научная новизна работы	
3	0,399 – 0,200	<b>Средняя</b> научная новизна работы	
4	0,199 – 0,150	<b>Низкая</b> научная новизна работы	
5	менее 0,150	<b>Очень низкая</b> научная новизна работы	

Показатель качества изложения научных материалов ( $K_2$ ) определяется по следующей формуле:

$$K_2 = 0,3 e^{-\lambda \left( \sum_n r_n O_n \right) \frac{N^*}{N}}, \dots \dots \dots (3)$$

где  $\lambda$  – коэффициент пропорциональности (определен 0,06);  
 $N$  – количество страниц в разделе (подразделе) отчета о НИР;  
 $O_n$  – количество замечаний  $n$ -го типа;  
 $N^*_{ст}$  – количество страниц в разделе (подразделе) отчета о НИР с учетом таблиц и рисунков;  
 $r_n$  – весовой коэффициент замечания  $n$ -го типа.

Коэффициент  $N^*_{ст}$  определяется по следующей формуле:

$$N^*_{ст} = N_{ст} - N_{рис} - 0,5 N_{табл}, \dots \dots \dots (4)$$

где  $N_{ст}$  – количество страниц, разработанных автором;  
 $N_{рис}$  – количество рисунков, разработанных автором;  
 $N_{табл}$  – количество таблиц, разработанных автором.

Оценка качества проведения научных исследований оценивается по замечаниям. Замечания по качеству выделяются по следующим составляющим (вариант):

**О<sub>1</sub>** (весовой коэффициент  $r_1 = 0,5$ ):

- не доказанные выводы;
- не обоснованные показатели и критерии;
- нарушение логики рассуждений (доказательства);
- использование терминов, не соответствующих требованиям ГОСТ и другим руководящим документам.

**О<sub>2</sub>** (весовой коэффициент  $r_2 = 0,3$ ):

- общеизвестные материалы из учебника без необходимости (лекция, «вода»);
- некорректность формулировок;
- стилистические ошибки;
- отклонение от требований ГОСТ в содержании структурных элементов отчета.

**О<sub>3</sub>** (весовой коэффициент  $r_3 = 0,1$ ):

- подмена тезиса при логическом рассуждении;
- «разрыв» изложения, создание ненужного абзаца;
- повтор материала без необходимости.

**О<sub>4</sub>** (весовой коэффициент  $\gamma_4 = 0,05$ )

- цитирование материала без ссылок на источники;
- орфографическая ошибка;
- грамматическая ошибка.

Перечень и весовые коэффициенты замечаний определены на основании экспертных оценок и могут корректироваться по мере накопления опыта по результатам применения данной методики для оценки отчетов о НИР.

Качественное значение показателя  $K_2$  приведено в таблице 5.

Таблица 5 – Оценка качества изложения материала научно-исследовательской работы

№ п/п	Значение $K_2$	Оценка качества изложения материала
1	2	3
1	0,300 - 0,295	<b>Очень высокое</b> качество изложения
2	0,294 - 0,280	<b>Высокое</b> качество изложения
3	0,279- 0,240	<b>Среднее</b> качество изложения
4	0,239- 0,200	<b>Низкое</b> качество изложения
5	менее 0,200	<b>Очень низкое</b> качество изложения

Общая оценка научно-исследовательской работе выставляется по формуле (1), значения приведены в таблице 6. Однако при определении оценки имеется ряд особенностей. Значение оценки качества изложения научных материалов ( $K_2$ ) определяется по формуле (2), при этом значения количества замечаний ( $N$ ), а также значения коэффициентов  $O_n$  и  $N_{ст}^*$  берется за весь материал отчёта о НИР.

Таблица 6 – Общая оценка научно-исследовательской работы

№ п/п	Значение $K_1 + K_2$	Общая оценка научно-исследовательской работы
1	2	3
1	1,000 - 0,900	<b>Очень высокая</b>
2	0,899 - 0,700	<b>Высокая</b>
3	0,699- 0,400	<b>Средняя</b>
4	0,399- 0,200	<b>Низкая</b>
5	менее 0,200	<b>Очень низкая</b>

Суммарный рейтинг за работу зависит от своевременности сдачи рабочих материалов соисполнителями или от своевременности отправки отчёта о НИР заказчику.

*Показатель своевременности сдачи задания  $K_3$ .* Значения данного коэффициента определены по данным экспертного опроса (таблица 7).

Таблица 7 – Коэффициент своевременности выполнения задания

№ п/п	Задержка в сроках выполнения задания	Кз
1	2	3
1	Задание сдано вовремя	1
2	Задержка в выполнении задания не более чем на одну неделю	0,95
3	Задержка в выполнении задания не более чем на две недели	0,85
4	Задержка в выполнении задания не более чем на три недели	0,7
5	Задержка в выполнении задания не более чем на четыре недели	0,6
6	Задержка в выполнении задания более чем на четыре недели	0,5

Знание рейтинга позволит:

- получать сведения о качестве работы за месяц (квартал, год) каждого научного сотрудника по всем отчетам и определять его место в научном подразделении;
- формировать у руководителей единые требования к знаниям и уровню научной работы научных сотрудников;
- вырабатывать объективные критерии отбора научных сотрудников с целью выдвижения на вышестоящие должности.

#### Список литературы

1. Гусева Е.П. Основные компетенции в оценке персонала // Кадры. 2015. № 1. с. 25-26.
2. Климова Ю.А. Оценка деловых компетенций // Человек и труд 2015. № 7. с. 12-15.
3. Кибанов А.Я., Ивановская Л.В. Управление персоналом. М., ПРИОР, 2011. 387 с.
4. Борисова Е.А. Оценка и аттестация персонала. СПб.: ПИТЕР, 2013. 212 с.
5. Ленкевич М. Оценка творческих сотрудников // Справочник по управлению персоналом. 2011. № 4. с. 36-41.

#### References

1. Guseva E.P. Osnovnyye kompetencii v ocenke personala // Kadry. 2015. № 1.pp 25-26.
  2. Klimova YU.A. Ocenka delovyyh kompetencij // SChelovek i trud. 2015. № 7.pp 12-15.
  3. Kibanov A.YA., Ivanovskaya L.V. Upravlenie personalom. M., PRIOR, 2011. 387 p.
  4. Borisova E.A. Ocenka i attestaciya personala. SPb.: PITER, 2013. 212 p.
  5. Lenkevich M. Ocenka tvorcheskikh sotrudnikov // Spravochnik po upravleniyu personalom. 2011. № 4. pp 36-41.
-



УДК 004.02

## СПОСОБ И АЛГОРИТМЫ РЕАЛИЗАЦИИ ГЕНЕТИЧЕСКОГО ОПЕРАТОРА РЕКОМБИНАЦИИ НА ОСНОВЕ ТУРНИРНОЙ СТРАТЕГИИ СО СЛУЧАЙНЫМ ИСХОДОМ

**Зернов М.М., <sup>1</sup>Картавенков А.А.**

*Филиал ФГБОУ ВО Национальный исследовательский университет «МЭИ» в г. Смоленске, Россия, (214013, г. Смоленск, Энергетический проезд, 1),  
e-mail: <sup>1</sup> kartawenkov.aleksandr@yandex.ru*

**В статье предлагается два варианта способа отбора особей популяции в новое поколение, отличающиеся применением модифицированных вариантов стратегии турнирной селекции. Модификации турнирной стратегии состоят в случайном определении исхода турнира как на основе равной вероятности выбора особей, так и с применением принципа рулетки.**

Ключевые слова: генетический алгоритм, оператор селекции, оператор рекомбинации, отбор методом колеса рулетки, турнирная селекция, равновероятностная турнирная стратегия, турнирная стратегия с использованием колеса рулетки.

## METHOD AND ALGORITHMS FOR IMPLEMENTING THE GENETIC RECOMBINATION OPERATOR BASED ON A TOURNAMENT STRATEGY WITH A RANDOM OUTCOME

**Zernov M.M., Kartavenkov A. A.**

*Smolensk Branch of the National Research University "Moscow Power Engineering Institute", Smolensk, Russia (214013, Smolensk, Energeticheskyy proezd, 1),  
e-mail: kartawenkov.aleksandr@yandex.ru*

**New method of survival selection, using two variants of tournament selection with random outcome, proposed. Variants differ by using equal-probability individuals selection or limited (by tournament group) roulette wheel selection.**

Keywords: genetic algorithm, selection operator, recombination operator, selection by the roulette wheel method, tournament selection, tournament strategy, equal-probability tournament selection, different-probability tournament strategy, tournament selection using the roulette wheel.

## **Введение**

Генетический алгоритм до сих пор за счет своей простоты и универсальности является одним из популярных методов оптимизации [2]. Несмотря на глубокую проработку многих вопросов, постоянно ведутся поиски эффективных способов конструирования решений и управления динамикой эволюционного процесса. Одним из направлений является раскрытие потенциала турнирной стратегии, в том числе и за счет внесения в неё элементов случайности. В работе рассматривается вариант стационарного генетического алгоритма (алгоритма с постоянной численностью популяции) с двумя видами отбора – отбором родительских особей в промежуточную популяцию (классическая селекция) и отбором особей из объединённой популяции «родители+потомки» в новое поколение. Последний вид отбора в некоторых источниках называют оператором рекомбинации [5]. В других же [7-8] называют стратегией выживания (Survivor Selection Policy или Survival Selection). Рассмотренные положения имеют смысл и для случая поколенческих генетических алгоритмов (с ростом численности популяции от поколения к поколению) с частичным отсевом особей при переходе к новому поколению.

Вообще говоря, стратегии отбора, применимые в случае селекции родительских особей, могут быть с успехом применены и в операторе рекомбинации [5,7-8]. Выбор вида стратегии позволяет настроить баланс между направленной и случайной составляющей поиска решения в эволюционном процессе. Применение турнирной стратегии, как правило, делает генетический алгоритм более «эксплуатирующим», т.е. направленным. Стратегия же на основе колеса рулетки позволяет повысить выживаемость менее приспособленных особей и, таким образом, увеличить долю случайности (что важно при оптимизации целевых функций со многими локальными оптимумами).

## **1. Постановка задачи**

В данной работе предлагаются модифицированные варианты реализации оператора рекомбинации (стратегии выживания) на основе турнирной селекции путём введения в результат турнира элемента случайности. Предлагается использовать турниры с равновероятностными исходами (вырожденная стратегия, практически эквивалентная случайному выбору особей в новое поколение) и разноразностными исходами (с вероятностями, определёнными в соответствии с принципом колеса рулетки, но только для конкурирующих в пределах турнира особей).

## **2. Применяемая схема генетического алгоритма**

В дальнейшем, будем рассматривать эволюционный процесс, соответствующий схеме, представленной на рисунке 1. Так, формирование начальной популяции происходит в начале работы алгоритма и представляет собой случайно сгенерированную последовательность неповторяющихся решений. Далее следует процесс выбора родительских пар и их скрещивание. Полученные потомки подвергаются мутации и добавляются в объединённую популяцию родителей и потомков. После этого происходит отбор особей в новое поколение. Так происходит до тех пор, пока не будет достигнут критерий останова генетического алгоритма. Критерием окончания процесса работы алгоритма может служить заданное количество поколений или схождение популяции. Схождением называется такое состояние

популяции, когда все элементы популяции почти одинаковые и находятся в области некоторого экстремума. В такой ситуации кроссинговер почти никак не может изменить популяцию, так как создаваемые при нем потомки представляют собой копии родителей с переменными участками хромосом. Вышедшие из этой области за счет мутации хромосомы склонны к вымиранию, так как чаще всего имеют меньшую приспособленность, особенно если данный экстремум является глобальным. Таким образом, схождение популяции обычно означает, что найдено лучшее или близкое к нему решение [1].

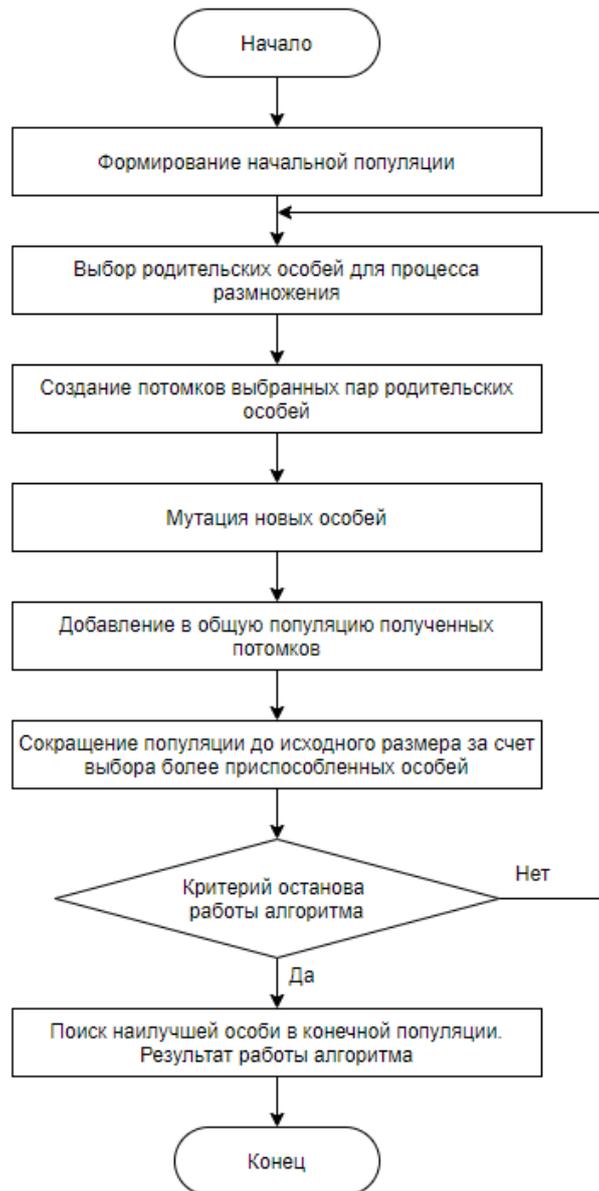


Рисунок 1 – Схема алгоритма генетического алгоритма

### 3. Отбор методом колеса рулетки

В основе данной стратегии лежит сопоставление каждой особи определенного сектора, размер которого пропорционален значению ее приспособленности [6], как показано на рисунке 2. Далее происходит выбор фиксированной точки и «вращение» колеса рулетки.

Особь, напротив которой останавливается точка, попадает в промежуточную популяцию. Так продолжается до тех пор, пока не будет отобрано определенное количество особей.

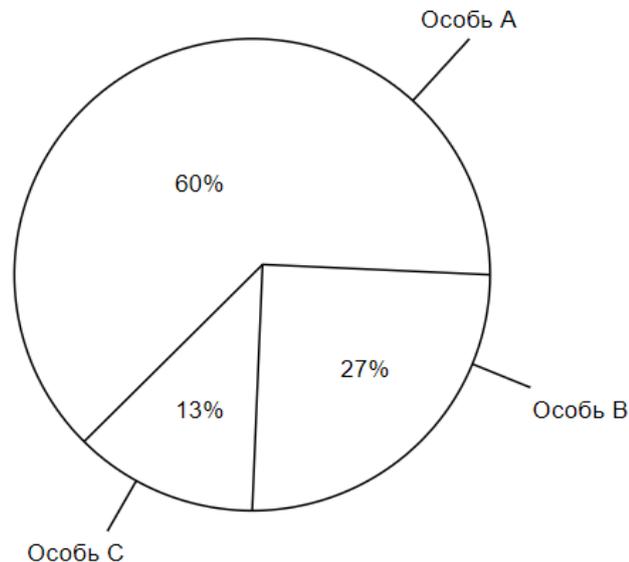


Рисунок 2 – Сопоставление особям секторов

Величина сектора, то есть вероятность выбора особи выполняется по следующей формуле:

$$P(i) = \frac{f(i)}{\sum_{i=1}^N f(i)}, \quad (1)$$

где  $P(i)$  – вероятность выбора  $i$  особи

$f(i)$  – значение функции приспособленности для  $i$  особи,

$N$  – количество анализируемых особей (в популяции в целом или турнирной группе, как будет предложено далее).

Также у данного способа возможна модификация, в которой колесо рулетки отбирает особи без повторений. Для этого после каждой итерации отбора следует пересчет процентного соотношения приспособленности для оставшихся особей. Эта модификация более ресурсоемкая, но зато она позволяет решить проблему преждевременной сходимости.

#### 4. Турнирная селекция

Другим популярным методом отбора особей в родительские пары является турнирная селекция, которая представляет собой метод, в котором из популяции, содержащей  $N$  особей, случайно отбираются несколько особей, после чего лучшая из них записывается в промежуточный массив. Эта операция повторяется  $N$  раз (для получения  $N/2$  родительских пар, но можно выбирать и меньшее количество). В дальнейшем особи, записанные в этот массив, будут участвовать в процессе скрещивания [6]. На рисунке 3 изображена схема турнирной селекции.

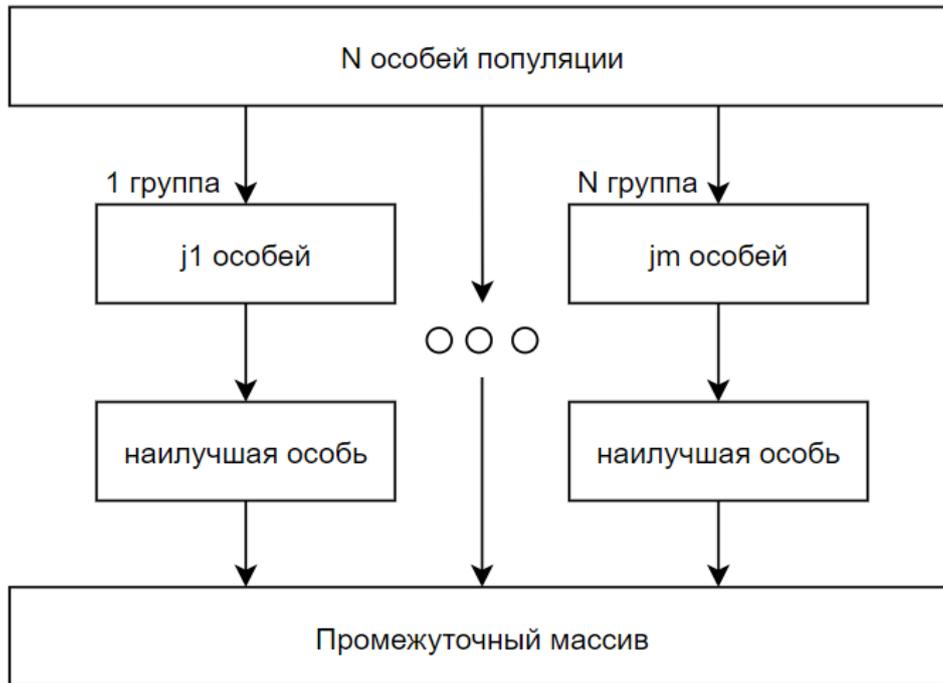


Рисунок 3 – Схема турнирной селекции

Турнирную стратегию помимо отбора особей в родительские пары, также можно использовать в операторе рекомбинации. На рисунке 4 изображена схема турнирной селекции для выбора особей, переходящих в новую популяцию. Возможен и обратный подход – выбор особей, удаляемых из популяции.

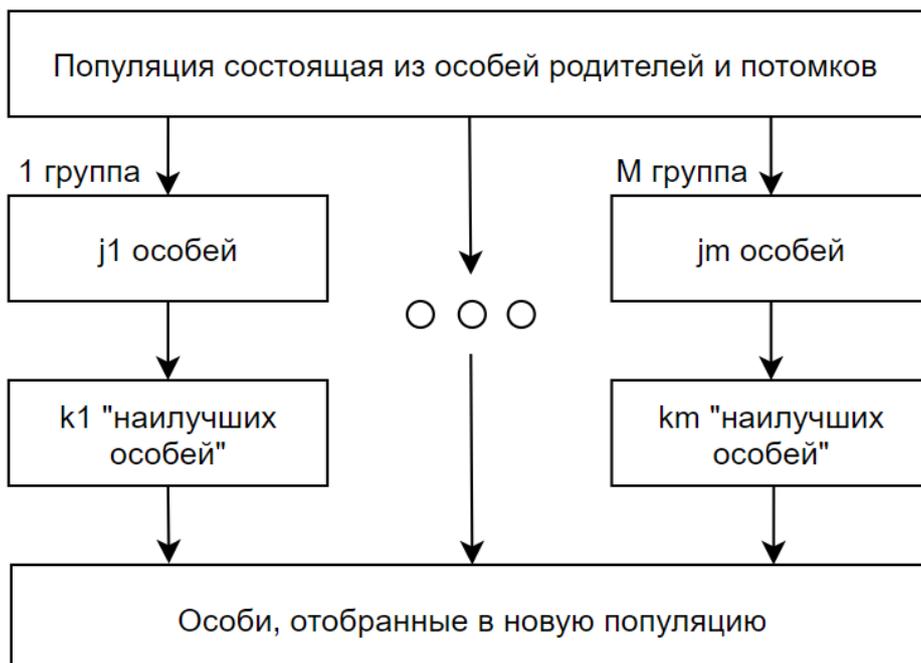


Рисунок 4 – Схема турнирной селекции

Предлагаемые способы реализации оператора рекомбинации, основанные на турнирной стратегии со случайным исходом, имеют ряд общих этапов и отличаются от «классического» способа непосредственно принципом определения победителей.

Общими являются следующие этапы: группировка особей, расчета количества особей, которые должны выйти из группы.

#### 4.1 Разбиение особей на группы

В процессе разбиения особей на группы участвует объединенная популяция родительских и дочерних особей. Все особи этой популяции должны оказаться в турнирных группах, но без дублирования, то есть не должно быть такой ситуации, когда одна и та же особь присутствует в нескольких группах. Вначале требуется рассчитать количество групп. Исходя из одинакового размера, количество групп рассчитывается как:

$$Group\ Amount = \frac{JoinPopulationSize}{Group\ Size}, \quad (2)$$

где *Group Amount* – количество групп,  
*JoinPopulationSize* – размер объединённой популяции родителей и потомков,  
*Group Size* – размер группы.

Если количество групп получается дробным, то производится округление в большую сторону. Последняя группа может оказаться неполной, что не является пробой.

После расчета количества групп следует распределение особей по группам. Данный процесс протекает в нескольких циклах. Внешний имеет размер, равный рассчитанному выше количеству групп, внутренний же равен размеру группы. Внутри последнего и происходит отбор особей. Данный процесс является абсолютно случайным.

Однако при распределении особей по группам возникает проблема поочередного случайного выбора всех элементов в последовательности, так как генератор случайных чисел имеет возможность спустя какое-то количество итераций повторить уже отобранное число. Поэтому простая проверка на наличие выбранной особи в группах не является корректной из-за того, что цикл становится практически бесконечным при случайном выборе последних элементов последовательности. Таким образом, набор особей в группы идёт из дополнительного списка особей, которые еще не были отобраны. Процесс заполнения продолжается до тех пор, пока доступных особей не останется. На рисунке 5 проиллюстрирован процесс группировки особей.

#### 4.2 Расчет количества особей, которые должны выйти из группы

После разбиения популяции на группы появляется новая проблема, а именно, сколько особей должно выйти из каждой группы и попасть в новое поколение с учётом постоянной численности нового поколения.

Прежде всего требуется рассчитать среднее число особей, которые могут выйти из группы. Расчет происходит по формуле 4.2., которая также была найдена эмпирическим путем.

$$AvgPassCandidate = \frac{PopulationSize}{GroupCount}, \quad (4.2)$$

где *AvgPassCandidate* – среднее число особей, которые могут выйти из группы;  
*PopulationSize* – заданный размер популяции;

*GroupCount* – количество турнирных групп, которое рассчитывается на этапе группировки особей;

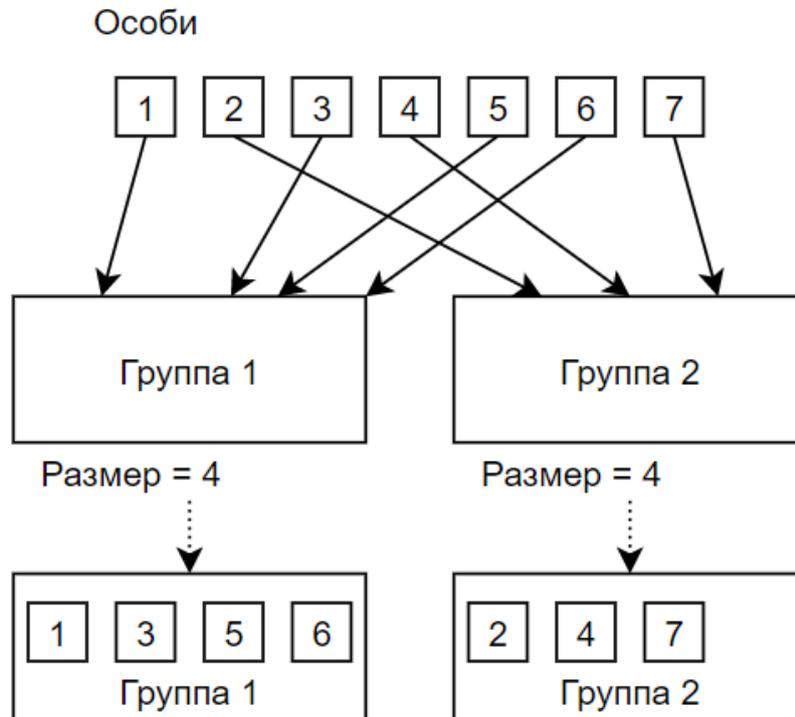


Рисунок 5 – Схема группировки особей

Вообще говоря, рекомендуется задавать размер объединённой популяции (*JoinPopulationSize*) кратным размеру группы, а размер основной популяции кратным количеству групп. Тогда из групп будет отбираться одинаковое количество особей. Однако это условие может и не выполняться. Предположим, что из каждой группы должно выйти *AvgPassCandidate* особей. Далее берется последняя группа, и если она является неполной, а её размер меньше или равен *AvgPassCandidate*, то количество выходящих из нее особей становится равным размеру группы. После этого идет суммирование значений количества выходящих особей из групп. Если это значение оказывается меньше, чем заданный размер популяции, то далее следует процесс дозаполнения, то есть поочередное увеличение для каждой группы значения выходящих в новую популяцию особей. Процесс дозаполнения продолжается до тех пор, пока разница между суммой особей выходящих из каждой группы и заданным размером популяции не станет равна нулю. Таким образом, на выходе будет получен список, который будет содержать количество выходящих особей из каждой группы. На рисунке 6 изображены группы, где красным цветом отмечены особи, которые пройдут отбор.

#### 4.3 Отбор в турнирной селекции

После того, как все особи были разбиты на группы и для каждой этих групп было рассчитано количество выходящих из нее особей, следует непосредственно отбор в группах.

В классической турнирной стратегии отбор представляет собой сортировку особей по их приспособленности. После чего, в зависимости от значения количества выходящих особей, отбираются первые особи текущей группы.

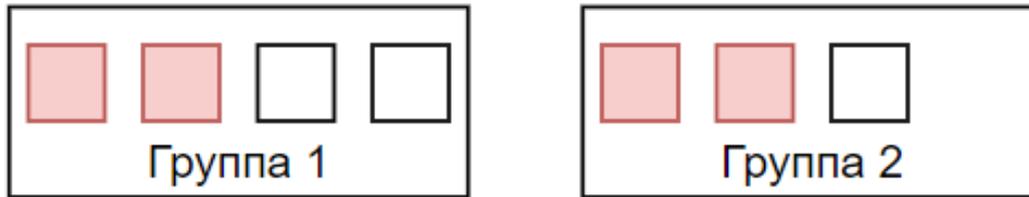


Рисунок 6 – Схема группы с отмеченными особями, которые пройдут отбор

### 5. Равновероятностная турнирная селекция

В качестве крайнего варианта турнирной стратегии при рекомбинации было предложено равновероятностный способ, при котором особи, вне зависимости от их приспособленности, будут с одинаковой вероятностью попадать в следующее поколение. Отличие от простого случайного отбора особей в новую популяцию состоит исключительно в структуре процесса, полностью соответствующей последовательности турнирной стратегии, что облегчает сравнение с другими вариантами (классическим и равновероятностным). Этапы способа представлены ниже:

Этап №1: Объединение родительской и дочерней популяции. Удаление из объединенной популяции особей, выходящих за границу допустимых решений. Таким образом, в процессе отбора будут участвовать заведомо корректные особи.

Этап №2: Разбиение особей популяции на турнирные группы. Таким образом отбор происходит непосредственно в группах.

Этап №3: Расчет числа выходящих особей. Таким образом для каждой группы известно количество особей, которые должны выйти из неё.

Этап №4: Отбор особей в новую популяцию, используя турнирную стратегию, где особи имеют одинаковую вероятность выхода из группы.

«Равновероятностная турнирная селекция» имеет 2 вида реализации оператора рекомбинации – с выходом дублирующих особей в следующее поколение и без. Оба этих варианта подразумевают использование генератора случайных чисел на этапе отбора. Для варианта без дубликатов используется дополнительный список, который будет содержать оставшиеся в группе особи.

В равновероятностном турнире с выходом дублирующих особей при помощи генератора случайных чисел на диапазоне от 0 до индекса последнего элемента группы будет получено число, которое соответствует индексу особи в этой группе. После чего по этому индексу выбирается особь, которая добавляется в следующее поколение. Процесс турнира продолжается пока не будет отобрано требуемое количество особей. После чего турнир проводится для следующей группы.

В равновероятностном турнире без дублирующих особей вначале происходит инициализация списка доступных особей в группе, после чего он принимает в качестве

значения турнирную группу. Далее начинается отбор. Используя генератор случайных чисел на диапазоне от 0 до индекса последнего элемента списка доступных особей группы, будет получено число, которое соответствует индексу особи в данном списке. По этому индексу выбирается особь, которая добавляется в следующее поколение, а также удаляется из списка доступных элементов. Процесс продолжается до тех пор, пока не будет отобрано необходимое число особей. После этого турнир проводится для других групп.

### **6. Разновероятностная турнирная селекция**

Вторым (и основным) вариантом реализации оператора рекомбинации на основе турнирной стратегии со случайным исходом является способ на основе разновероятностной турнирной селекции. В данном способе каждая особь в зависимости от ее приспособленности имеет свою вероятность выхода из группы в соответствии с принципом рулетки (в пределах группы).

От предыдущего, рассматриваемый способ отличается этапом №4.

Этап отбора является комбинацией использования турнирной селекции и колеса рулетки. Таким образом, в каждой турнирной группе происходит вращение колеса рулетки, и особь, значение сектора которой соответствует выбранному при помощи генератора случайных чисел значению, отбирается в следующее поколение. Количество вращений колеса соответствует требуемому количеству особей, которые должны выйти из группы.

«Разновероятностная турнирная селекция» также имеет 2 вида реализации – с выходом из группы дублирующих особей и без.

### **Заключение**

В статье предложено 2 варианта способа реализации оператора отбора особей в новое поколение генетического алгоритма (оператора рекомбинации) на основе турнирной стратегии со случайным исходом. Варианты способа имеют общую структуру и отличаются применяемыми алгоритмами отбора особей из турнирных групп. Первый вариант, на основе равновероятностного отбора, формирует вырожденную стратегию эквивалентную случайному отбору. Второй вариант, на основе разновероятностного отбора позволяет сочетать свойства турнирной стратегии и стратегии на основе колеса рулетки. В дальнейшем предстоит провести практический анализ результатов применения рассмотренных вариантов, а также рассмотреть способы на основе обратного турнира – в котором выбираются выбывающие особи.

### **Список литературы**

1. Батищев Д.И., Неймарк Е.А., Старостин Н.В. Применение генетических алгоритмов к решению задач дискретной оптимизации. Учебно-методический материал по программе повышения квалификации «Информационные технологии и компьютерное моделирование в прикладной математике». – Нижний Новгород: Нижегородский государственный университет им. Н.И. Лобачевского, 2007. – 85 с.
2. Букатова И. Л., Михасев Ю. И., Шаров А. М. Эвоинформатика: Теория и практика эволюционного моделирования. – М.: Наука, 1991. – 206 с.

3. Бураков М. В. Генетический алгоритм: теория и практика: учеб. пособие – СПб.: ГУАП, 2008. – 164 с.
4. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / под ред. В.М. Курейчика. – 2-е изд., испр. и доп. – М.: Физматлит, 2006. – 320 с.
5. Емельянов В. В., Курейчик В. В., Курейчик В. М. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 432 с.
6. Панченко Т.В. Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с
7. Genetic Algorithms - Survivor Selection// Материалы сайта [www.tutorialspoint.com](http://www.tutorialspoint.com) [Электронный ресурс]: - Режим доступа: [https://www.tutorialspoint.com/genetic\\_algorithms/genetic\\_algorithms\\_survivor\\_selection.htm](https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_survivor_selection.htm), свободный. – Загл. с экрана. – Яз. англ. Дата доступа: 20 сент. 2020
8. Luke S. Essentials of Metaheuristics. A Set of Undergraduate Lecture Notes. Zeroth Edition. Online Version 2.3. February, 2016 [Электронный документ]: - Режим доступа: <https://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf>, свободный. – Загл. с экрана. – Яз. англ. Дата доступа: 20 сент. 2020

## References

1. Batishchev D.I., Neymark E.A., Starostin N.V. Application of genetic algorithms to solving discrete optimization problems. Educational and methodical material according to the development program "Information technologies and computer modeling in applied mathematics". – Nizhny Novgorod: N.I. Lobachevsky State University of Nizhni Novgorod, 2007. – 85 p.
  2. Bukatova I. L., Mikhasev Yu. I., Sharov A. M. Evoinformatics: Theory and practice of evolutionary modeling. – М.: Science, 1991. - 206 p.
  3. Burakov M.V. Genetic algorithm: theory and practice: textbook – St. Petersburg: GUAP, 2008. – 164 p.
  4. Gladkov L.A., Kureichik V.V., Kureichik V.M. Genetic algorithms/ed. V.M. Kureichik. – 2nd ed., Correct and additional – М.: Fizmatlit, 2006. – 320 p.
  5. Emelyanov V.V., Kureichik V.V., Kureichik V.M. Theory and practice of evolutionary modeling. – М.: Fizmatlit, 2003. – 432 p.
  6. Panchenko T.V. Genetic algorithms: educational manual/ed. Yu. Yu. Tarasevich. – Astrakhan: Publishing House "Astrakhan University," 2007. – 87 p.
  7. Genetic Algorithms - Survivor Selection. Available at: [https://www.tutorialspoint.com/genetic\\_algorithms/genetic\\_algorithms\\_survivor\\_selection.htm](https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_survivor_selection.htm) (accessed 20 September 2020)
  8. Luke S. Essentials of Metaheuristics. A Set of Undergraduate Lecture Notes. Zeroth Edition. Online Version 2.3. February, 2016. Available at: <https://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf> (accessed 20 September 2020)
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.02

## ОЦЕНКА ТУРНИРНЫХ СТРАТЕГИИ СО СЛУЧАЙНЫМ ИСХОДОМ В СРАВНЕНИИ С ДРУГИМИ РЕАЛИЗАЦИЯМИ ГЕНЕТИЧЕСКИХ ОПЕРАТОРОВ

**Картавенков А.А.**

*Филиал ФГБОУ ВО Национальный исследовательский университет «МЭИ» в г. Смоленске, Россия, (214013, г. Смоленск, Энергетический проезд, 1),  
e-mail: kartawenkov.aleksandr@yandex.ru*

**В данной статье ведется анализ и оценка способов «Равновероятностная турнирная селекция» и «Разновероятностная турнирная селекция» («Турнирная селекция с использованием колеса рулетки») в сравнении с «Классической турнирной селекцией».**

Ключевые слова: анализ, оценка, генетический алгоритм, оператор селекции, равновероятностная турнирная селекция, классическая турнирная селекция, разновероятностная турнирная селекция.

## EVALUATION OF TOURNAMENT STRATEGY WITH A RANDOM OUTCOME IN COMPARISON WITH OTHER IMPLEMENTATIONS OF GENETIC OPERATORS

**Kartavenkov A. A.**

*Smolensk Branch of the National Research University "Moscow Power Engineering Institute", Smolensk, Russia (214013, Smolensk, Energeticheskyy proezd, 1),  
e-mail: kartawenkov.aleksandr@yandex.ru*

**In this article, we analyze and evaluate the methods "Equal Probability Tournament Selection" and "Differential Tournament Selection" ("Tournament Selection Using the Roulette Wheel") in comparison with the "Classic Tournament Selection".**

Keywords: analysis, evaluation, genetic algorithm, selection operator, equal probability tournament selection, classical tournament selection, tournament selection with different probabilities.

### Введение

В настоящее время исследования в области ГА снова начали набирать популярность. Это связано с большей доступностью компьютеров с высокой вычислительной мощностью, чего не было в 70-е годы прошлого века, когда данный алгоритм только зарождался и набирал свою популярность [4]. Таким образом, если раньше приоритетной задачей являлось нахождение приемлемого решения за короткое время, то сейчас проблема быстрого действия алгоритма перестала быть главнейшей, а её место заняла проблема качества найденного решения.

Именно поэтому сейчас продолжают исследования различных модификаций и реализаций генетических операторов. Одними из таких модификаций являются способы реализации генетического оператора рекомбинации на основе турнирной стратегии со случайным исходом.

### **1. Постановка задачи**

В качестве конкретной решаемой задачи в данной статье будет рассмотрена оценка качества работы способов «Равновероятностной турнирной селекции» и «Разновероятностной турнирной селекции» («Турнирная селекция с использованием колеса рулетки») генетического оператора рекомбинации в сравнении с «Классической турнирной селекцией», которая представляет собой использование турнирной стратегии в роли оператора отбора наиболее приспособленных особей в следующее поколение.

### **2. Выбор ключевых показателей и методики оценки**

Прежде всего требуется выделить ключевые показатели, по которым будет происходить оценка. Как отмечалось выше, в настоящее время ключевым показателем генетического алгоритма является качество решения, под которым понимается близость найденного решения к точному [3]. Также в качестве другого ключевого показателя можно выделить скорость сходимости алгоритма при поиске решений с заданной точностью, то есть за какое количество итераций (поколений) будет получено удовлетворительное решение [1]. Таким образом, качество работы способов «Равновероятностной турнирной селекции» и «Разновероятностной турнирной селекции» определяется данными показателями.

Методика оценки содержит следующие инструкции, соблюдение которых позволяет дать корректную оценку работы способов «Равновероятностной турнирной селекции» и «Разновероятностной турнирной селекции»:

1) Оценка качества решения заключается во множественном прогоне генетического алгоритма с различными стратегиями оператора рекомбинации и с одинаковыми тестовыми данными. После этого происходит расчет лучшего решения на каждом прогоне и среднего решения для всех прогонов. На основании среднего решения строится оценка качества решения, а именно разница между ним и точным решением.

2) Оценка сходимости алгоритма также заключается во множественном прогоне генетического алгоритма с различными стратегиями, однако акцент здесь делается на расчет поколения, на котором алгоритм получил решение удовлетворяющее заданную точность. Это называется сходимостью. Для каждого прогона записывается значение поколения, на котором закончился расчет. После этого на основании этих значений рассчитывается среднее значение сходимости для выбранной стратегии.

### **3. Тестовые данные**

Поскольку при каждом запуске генетического алгоритма происходит генерация начальной популяции, возникает проблема разных тестовых данных. Для того, чтобы решить ее, следует производить генерацию начальной популяции только в начале анализа. Однако процесс генерации начальной популяции находится внутри расчета генетического алгоритма. Поэтому в метод расчета следует добавить условие, которое сообщает, что данный процесс

работает в режиме оценки и что нет необходимости генерировать начальное поколение для генетического алгоритма, так как уже имеются тестовые данные необходимые для расчета. Данная логика не противоречит работе генетического алгоритма, потому как при невыполнении условия расчет происходит как обычно, то есть используя генерацию начального поколения в начале генетического алгоритма.

Таким образом, при оценке каждый запуск генетического алгоритма будет сопровождаться использованием только одних тестовых данных, то есть одной и той же начальной популяции. Данное нововведение позволяет объективно оценить каждую реализацию генетического алгоритма вне зависимости от удачно или недостаточно удачно сгенерированной начальной популяции.

#### 4. Оценка качества решения

Как было уже сказано ранее данная оценка рассчитывается путем множественного запуска генетического алгоритма при различных значениях оператора отбора особей в следующее поколение, и расчета близости точного решения к среднему, которое получено на основании лучших решений каждого прогона.

Так, в начале произведем расчет оценки для малых значений настроек генетического алгоритма:

- Функция принадлежности  $y = x$ , которая находится в диапазоне от 0 до 50;
- Погрешность решения = 0;
- Требуется найти глобальный максимум;
- Количество поколений = 5;
- Размер популяции = 4;
- Количество прогонов = 5;
- Анализируемый метод отбора особей в следующее поколение.
- Анализируемые стратегии: равновероятностная турнирная селекция, классическая турнирная селекция, равновероятностная турнирная селекция;
- Начальная популяция: 5, 22, 36, 48.

Точным глобальным максимумом для функции  $y = x$  на отрезке от 0 до 50 является значение 50. На основании этих данных получаем значения лучших решений каждого прогона для выбранных стратегий. Помимо этого, производим расчет среднего лучшего решения для каждой стратегии. Данные результаты представлены таблицей 1.

Как отмечалось выше, ключевым показателем данной оценки является качество решения, то есть близость найденного решения к точному. Таким образом, лучшей стратегией будет та, чья оценка будет иметь наименьшее значение. Расчет оценки качества производится по формуле 1, которая указана ниже [2]:

$$Evaluation = Exact\ Solution - Avg.\ Solution, \quad (1)$$

где *Evaluation* – оценка качества решения,  
Exact Solution – точное решение,  
Avg. Solution – среднее решение выбранной стратегии.  
На таблице 2 изображена оценка каждой стратегии.

Таблица 1 – Лучшие решения для функции  $y=x$  в диапазоне от 0 до 50, найденные на каждом прогоне для каждой турнирной стратегии

№ п/п	Способы	Прогоны					
		1	2	3	4	5	Ср. Знач
1	2	3	4	5	6	7	8
1	Равновероятностная турнирная селекция	32	40	46	46	48	42
2	Классическая турнирная селекция	50	50	40	50	42	47
3	Турнирная селекция с использованием колеса рулетки	48	50	48	50	50	49
4	Точное решение	50					50

Таблица 2 – Оценка решений для каждой турнирной стратегии

№ п/п	Способы	Оценка качества решений
1	2	3
1	Равновероятностная турнирная селекция	8
2	Классическая турнирная селекция	3
3	Турнирная селекция с использованием колеса рулетки	1

Также по результатам оценки можно построить график, который изображен на рисунке 1. Так, по оси У берутся значения наилучшего решения, а по оси Х – значение прогона.

Стоит отметить, что чем больше количество прогонов, количество популяций, размер популяции и диапазон допустимых решений, тем лучше оценка качества решений стратегий. Таким образом, далее рассмотрена оценка качества решения тех же самых турнирных стратегий, но уже с другими значениями настроек генетического алгоритма:

- функция принадлежности  $y = x^2 + 2 * x + 10$ , которая находится в диапазоне от 0 до 250;
- погрешность решения = 5;
- требуется найти глобальный максимум;
- количество поколений = 15;
- размер популяции = 8;
- количество прогонов = 15;
- анализируемый метод отбора особей в следующее поколение.
- анализируемые стратегии: равновероятностная турнирная селекция, классическая турнирная селекция, равновероятностная турнирная селекция;
- начальная популяция: 81, 128, 114, 189, 105, 193, 164, 234.

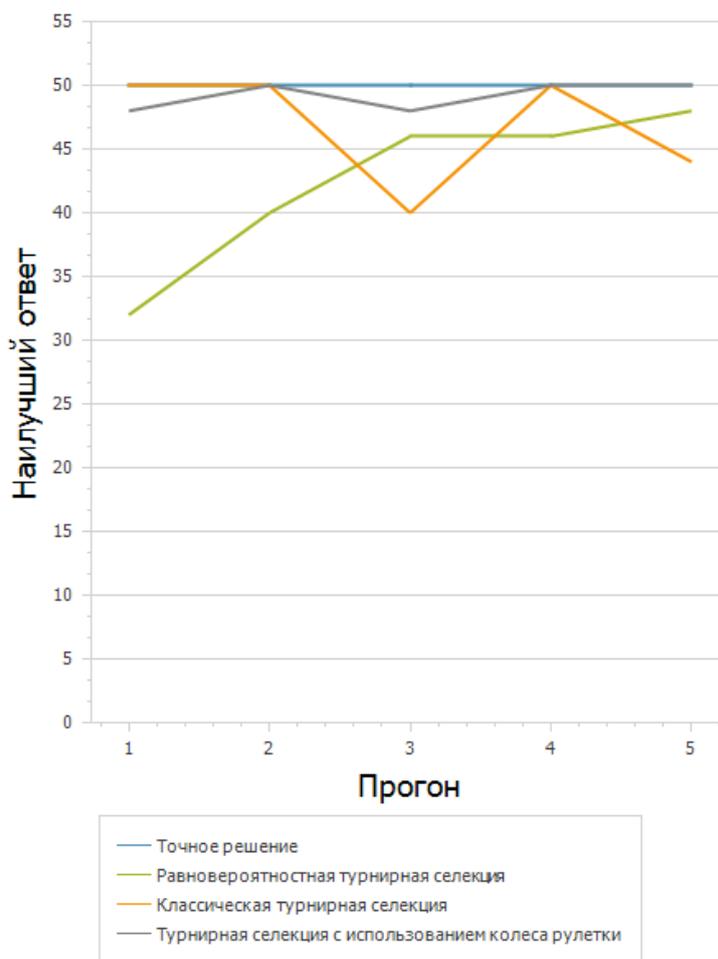


Рисунок 1 – График с результатами прогонов для каждой стратегии

Точным значением глобального максимума для функции  $y = x^2 + 2 * x + 10$  на отрезке от 0 до 250 является 250. На основании этих данных получаем значения лучших решений каждого прогона для выбранных стратегий. Помимо этого, производим расчет среднего лучшего решения для каждой стратегии. Данные результаты представлены таблицей 3.

Ниже, на таблице 4 изображена оценка каждой стратегии.

Также на рисунке 2 представлен график с результатами каждого прогона для каждой стратегии.

По результатам оценки качества решений выявлено, что способ «Турнирной селекции с использованием колеса рулетки» находит более качественное решение по сравнению с остальными турнирными стратегиями.

### 5. Оценка сходимости алгоритма

Как было уже сказано ранее оценка сходимости алгоритма также заключается во множественном прогоне генетического алгоритма с различными стратегиями, однако акцент здесь делается на расчет номера поколения, на котором алгоритм получил решение удовлетворяющее заданную точность. Это называется сходимостью [5]. Для каждого прогона записывается значение поколения, на котором закончился расчет. После этого на основании

этих значений рассчитывается среднее значение сходимости для выбранной стратегии. Таким образом, можно утверждать, что сходимость прямо зависит от точности решения.

Таблица 3 – Лучшие решения для функции  $y=x^2 + 2 * x + 10$  в диапазоне от 0 до 250, найденные на каждом прогоне для каждой турнирной стратегии

Прогоны	Способы		
	Равновероятностная турнирная селекция	Классическая турнирная селекция	Турнирная селекция с использованием колеса рулетки
1	2	3	4
1	225	250	250
2	179	239	250
3	204	250	250
4	246	214	250
5	250	250	250
6	240	250	250
7	223	236	249
8	223	250	249
9	221	250	250
10	250	243	250
11	215	250	250
12	241	250	250
13	250	237	250
14	224	250	250
15	192	250	249
Среднее	226	245	250

Таблица 4 – Оценка решений для каждой турнирной стратегии

№ п/п	Способы	Оценка качества решений
1	2	3
1	Равновероятностная турнирная селекция	24
2	Классическая турнирная селекция	5
3	Турнирная селекция с использованием колеса рулетки	0

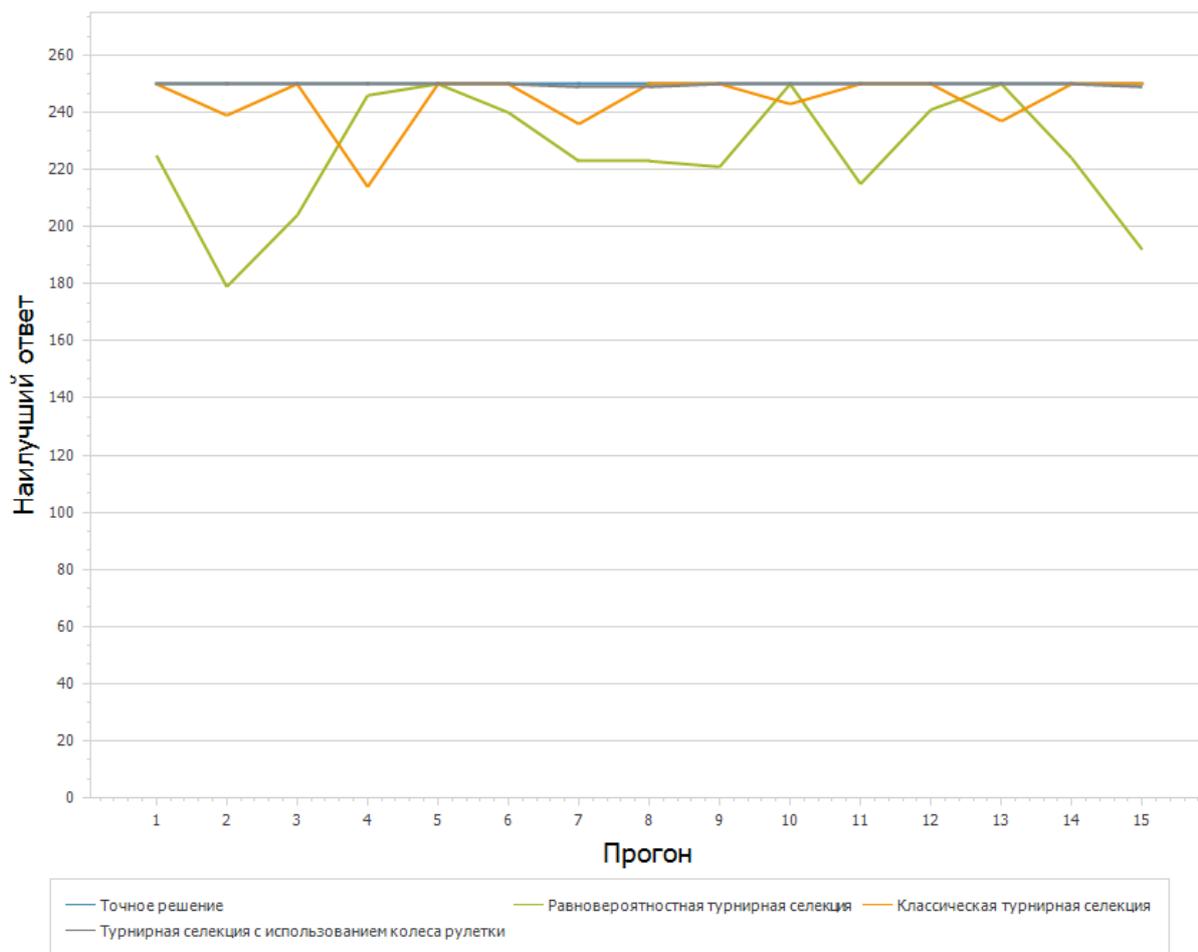


Рисунок 2 – График с результатами прогонов для каждой стратегии

Для расчета оценки сходимости выставим следующие настройки генетического алгоритма:

- функция принадлежности  $y = x^2 + 2 * x + 10$ , которая находится в диапазоне от 0 до 250;
- погрешность решения = 5;
- требуется найти глобальный максимум;
- количество поколений = 15;
- размер популяции = 8;
- количество прогонов = 15;
- анализируемый метод отбора особей в следующее поколение.
- анализируемые стратегии: равновероятностная турнирная селекция, классическая турнирная селекция, разноразностная турнирная селекция;
- начальная популяция: 81, 128, 114, 189, 105, 193, 164, 234.

Точным значением глобального максимума для функции  $y = x^2 + 2 * x + 10$  на отрезке от 0 до 250 является 250. На основании этих данных получаем значения лучших решений каждого прогона для выбранных стратегий. Помимо этого, производим расчет среднего лучшего решения для каждой стратегии. Данные результаты представлены таблицей 5.

Таблица 5 – Количество итераций, за которое генетическим алгоритмом было найдено удовлетворительное решения для функции  $y=x^2 + 2 * x + 10$  в диапазоне от 0 до 250

Прогоны	Способы		
	Равновероятностная турнирная селекция	Классическая турнирная селекция	Турнирная селекция с использованием колеса рулетки
1	2	3	4
1	15	4	10
2	15	3	15
3	15	5	4
4	15	4	15
5	15	7	2
6	15	12	2
7	15	15	15
8	15	15	10
9	15	5	6
10	1	12	15
11	15	9	6
12	15	5	9
13	6	4	15
14	15	6	14
15	15	15	4
Среднее	13	11	10

После расчета средних значений количества итераций, за которое сойдётся генетический алгоритм, следует расчет оценки сходимости для каждой стратегии. Расчет выполняется по формуле 2 [2]. Таким образом, лучшей стратегией будет та, чья оценка будет иметь наибольшее значение.

$$Evaluation = Generation Amount - Avg.Number of Last Generation, \dots (2)$$

где *Evaluation* – оценка сходимости алгоритма,  
*Generation Amount* – количество поколений,  
*Avg.Number of Last Generation* – среднее значение количества итераций, за которое сойдется алгоритм.

Таблица 6 – Оценка сходимости для каждой турнирной стратегии

<b>№ п/п</b>	<b>Способы</b>	<b>Оценка сходимости алгоритма</b>
<b>1</b>	<b>2</b>	<b>3</b>
1	Равновероятностная турнирная селекция	2
2	Классическая турнирная селекция	4
3	Турнирная селекция с использованием колеса рулетки	5

Поскольку оценка сходимости прямо зависит от выбранной точности решения. То ниже она будет рассмотрена при тех же значениях настроек генетического алгоритма, но уже с большим значением погрешности решения:

- функция принадлежности  $y = x^2 + 2 * x + 10$ , которая находится в диапазоне от 0 до 250;
- погрешность решения = 10;
- требуется найти глобальный максимум;
- количество поколений = 15;
- размер популяции = 8;
- количество прогонов = 15;
- анализируемый метод отбора особей в следующее поколение.
- анализируемые стратегии: равновероятностная турнирная селекция, классическая турнирная селекция, разноразмерная турнирная селекция;
- начальная популяция: 81, 128, 114, 189, 105, 193, 164, 234.

Результаты расчетов представлены на таблице 7.

На основании полученных значений можно построить оценку сходимости алгоритма для выбранных стратегий. На таблице 8 представлены результаты оценки.

По результатам оценки сходимости алгоритма выявлено, что способ «Турнирной селекции с использованием колеса рулетки» имеют примерно такую же сходимость как и способ «Классической турнирной селекции».

### **Заключение**

В данной статье приведены ключевые показатели генетических алгоритмов, на основе которых построены оценки реализаций турнирной селекции оператора рекомбинации. Оценка показала, что разработанный способ «Разноразмерной турнирной селекции» («Турнирная селекция с использованием колеса рулетки») имеет отличные показатели. Так, этот способ находит более качественное решение за меньшее количество итераций чем другие турнирные стратегии.

Таблица 7 – Количество итераций, за которое генетическим алгоритмом было найдено удовлетворительное решения для функции  $y=x^2 + 2 * x + 10$  в диапазоне от 0 до 250.

Прогоны	Способы		
	Равновероятностная турнирная селекция	Классическая турнирная селекция	Турнирная селекция с использованием колеса рулетки
1	2	3	4
1	15	1	9
2	15	3	9
3	13	15	15
4	15	5	15
5	15	6	2
6	15	7	15
7	15	15	3
8	5	4	4
9	1	3	1
10	15	4	15
11	15	7	3
12	15	1	15
13	8	15	9
14	15	8	15
15	15	15	4
Среднее	13	10	10

Таблица 8 – Оценка сходимости для каждой турнирной стратегии

№ п/п	Способы	Оценка сходимости алгоритма
1	2	3
1	Равновероятностная турнирная селекция	2
2	Классическая турнирная селекция	5
3	Турнирная селекция с использованием колеса рулетки	5

### Список литературы

1. Батищев Д.И., Неймарк Е.А., Старостин Н.В. Применение генетических алгоритмов к решению задач дискретной оптимизации. Учебно-методический материал по программе повышения квалификации «Информационные технологии и компьютерное моделирование в прикладной математике». – Нижний Новгород: Нижегородский государственный университет им. Н.И. Лобачевского, 2007. – 85 с.
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / под ред. В.М. Курейчика. – 2-е изд., испр. и доп. – М.: Физматлит, 2006. – 320 с.

3. Емельянов В. В., Курейчик В. В., Курейчик В. М. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003. – 432 с.
4. Жиглявский А. А., Жилинскас А. Г. Методы поиска глобального экстремума. – М.: Наука, 1991. – 248 с.
5. Рутковская, Д., Пилиньский, М., Рутковский, Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. – 2-е изд., стереотип. – М.: Горячая линия – Телеком, 2013. – 384 с.

### References

1. Batishchev D.I., Neymark E.A., Starostin N.V. Application of genetic algorithms to solving discrete optimization problems. Educational and methodical material according to the development program "Information technologies and computer modeling in applied mathematics". – Nizhny Novgorod: N.I. Lobachevsky State University of Nizhni Novgorod, 2007. - 85 p.
  2. Gladkov L.A., Kureichik V.V., Kureichik V.M. Genetic algorithms/ed. V.M. Kureichik. – 2nd ed., Correct and additional – М.: Fizmatlit, 2006. - 320 p.
  3. Emelyanov V.V., Kureichik V.V., Kureichik V.M. Theory and practice of evolutionary modeling. – М.: Fizmatlit, 2003. – 432 p.
  4. Zhiglyavsky A. A., Zhilinskias A. G. Methods of searching for a global extremum. – М.: Science, 1991. – 248 p.
  5. Rutkovskaya, D., Pilinsky, M., Rutkovsky, L. Neural networks, genetic algorithms and fuzzy systems: Per. with Polish I. D. Rudinsky. – 2nd ed., stereotype. – М.: Hotline – Telecom, 2013. – 384 p.
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.82

## ИДЕНТИФИКАЦИЯ И АНАЛИЗ УСТОЙЧИВОСТИ КЛАСТЕРОВ СОЦИОТЕХНИЧЕСКИХ СИСТЕМ НА ОСНОВЕ НЕЧЕТКОГО КОГНИТИВНОГО ПОДХОДА

**Ерш В.С.**

Филиал ФГБОУ ВО «Национальный исследовательский университет «МЭИ» в г. Смоленске, Россия, (214013, г. Смоленск, Энергетический проезд, 1), e-mail: [victory.s.e@mail.ru](mailto:victory.s.e@mail.ru)

Дано описание обмена сообщениями, видов сообщений и моделей, которые реализуются системой обмена сообщениями. Проведен сравнительный анализ брокеров сообщений, среди которых Apache ActiveMQ, Apache Kafka, RabbitMQ, IBM MQ, NATS, ZeroMQ. Результаты анализа сведены в сравнительную таблицу. Уделено внимание расширенным возможностям брокеров сообщений по потоковой обработке данных, осуществляемой с помощью библиотеки Apache Kafka Streams и расширенного протокола сообщений NATS Streaming.

Ключевые слова: Обмен сообщениями, брокер сообщений, потоковая обработка данных.

## IDENTIFICATION AND ANALYSIS OF SUSTAINABILITY OF CLUSTERS OF SOCIOTECHNICAL SYSTEMS BASED ON A FUZZY COGNITIVE APPROACH

**Ersh V.S.**

Smolensk Branch of the National Research University "Moscow Power Engineering Institute", Smolensk, Russia (214013, Smolensk, Energeticheskoy proezd, 1), e-mail: [victory.s.e@mail.ru](mailto:victory.s.e@mail.ru)

The description of messaging, types of messages and models that are implemented by the messaging system is given. A comparative analysis of message brokers, including Apache ActiveMQ, Apache Kafka, RabbitMQ, IBM MQ, NATS, ZeroMQ, was carried out. The analysis results are summarized in a comparative table. Attention is paid to the advanced capabilities of message brokers for data streaming. It is done by the Apache Kafka Streams library and the extended message protocol NATS Streaming.

Keywords: Message passing, message broker, data streaming.

Обмен сообщениями – это технология высокоскоростного асинхронного взаимодействия между программами с гарантией доставки информации [1]. Программы взаимодействуют между собой, обмениваясь пакетами данных, называемыми сообщениями. Канал, или очередь, – это логический маршрут, объединяющий программы и использующийся для транспортировки сообщений. Канал напоминает массив сообщений, доступный для одновременного использования многими приложениями.

Отправитель – это приложение, которое отправляет сообщение путем его размещения в канале. Получатель – это приложение, которое получает сообщение путем его считывания из канала, а затем удаляет его.

Сообщение представляет собой некоторую структуру данных – строку, байтовый массив, запись или объект [2]. Оно может быть интерпретировано непосредственно как содержащиеся в нем данные, как команда, которую необходимо выполнить получателю, или как описание события, произошедшего на стороне отправителя. Сообщение состоит из двух частей – заголовка и тела. Заголовок сообщения содержит метаданные, которые используются системой обмена сообщениями и игнорируются получателем сообщения. Это могут быть данные об отправителе, или информация, куда следует передать сообщение. Тело сообщения содержит полезную информацию, которая, как правило, игнорируется системой обмена сообщениями.

В распределённой компьютерной среде существует большое количество моделей для обмена сообщениями [3], которые различаются по способу передачи данных. Одной из них является асинхронная модель, где система посылает сообщение и продолжает обработку. Пользователям асинхронной системы сообщений не нужно ждать ответа от получателя, так как они могут положиться на инфраструктуру работы с сообщениями, которая обеспечивает доставку.

Этой связующей инфраструктурой является *система обмена сообщениями*. Она имеет много общего с системой баз данных. Основная задача системы баз данных – обеспечить надежное хранение информации, а основная задача системы обмена сообщениями – гарантировать доставку сообщений от отправителя получателю.

Необходимость наличия системы обмена сообщениями обуславливается ненадежностью сетей передачи данных. Сбой в компьютерной сети – одна из самых распространенных причин неудавшейся доставки сообщения от отправителя к получателю. Система обмена сообщениями позволяет гарантировать доставку информации за счет повторной отправки сообщения (до тех пор, пока оно не будет принято получателем). В идеальных условиях сообщение доставляется с первого раза, однако, к сожалению, так бывает далеко не всегда.

Рассмотрим несколько моделей [3], которые реализуются системой обмена сообщениями.

*Модель «Точка-точка»* гарантирует, что каждое отдельное сообщение будет потреблено только одним получателем. У канала может быть много получателей, одновременно обрабатывающих сообщения, но только один из них сможет успешно принять конкретное сообщение.

В классических системах обмена сообщениями данная модель обычно реализуется через очереди [4]. Очередь действует, как буфер FIFO, на который может подписаться один или несколько получателей. Каждое сообщение доставляется только одному из подписанных получателей. Если несколько получателей попытаются одновременно потребить некоторое сообщение, очередь позаботится о том, чтобы эта операция удалась только одному из них.

Обмен сообщениями типа *«Точка-точка»* используется, когда вариант использования требует однократного действия с сообщением (один получатель).

*Модель «Публикация-подписка»* используется для реализации широковещательной рассылки. Имеется один входной канал, который разбивается на несколько выходных каналов, по одному на каждого подписчика. Когда оповещение о событии публикуется в канале, оно распределяется по всем подписанным пользователям.

Обмен сообщениями типа «Публикация-подписка» обычно используется, когда сообщения носят информационный характер, и потеря одного сообщения не критична. Например, канал может передавать показания температуры от группы датчиков один раз в секунду [5]. Ничего страшного не произойдет, если система, которая получает эти данные, пропустит сообщение, потому что новое сообщение поступит в ближайшее время.

Существует несколько видов сообщений [6]:

- Сообщение с командой, отправляя которое приложение вызывает функциональность другого удаленного приложения. Сообщения с командой обычно отправляются по каналу «точка-точка», чтобы каждая команда была потреблена и выполнена только один раз.
- Сообщение с данными документа просто передает данные от одного приложения другому. Получатель сам решает, что с ними делать. В качестве данных может выступать элементарная единица данных, объект или структура данных, которую можно разложить на несколько небольших частей [7].
- Сообщения о событиях, которое применяется, когда несколько приложений хотели бы использовать оповещения о событиях, чтобы согласовывать свои действия. Когда у субъекта появляется событие, о котором следует оповестить остальных, он создает объект события, упаковывает его в сообщение и размещает в канале как сообщение о событии. Наблюдатель получает сообщение о событии, извлекает событие и обрабатывает его.

Рассмотрим современные брокеры сообщений и их возможности.

### **1. Apache ActiveMQ**

Apache ActiveMQ – система обмена сообщениями, которая была выпущена в 2004 году. Она имеет открытый исходный код, который реализует Java Message Service (JMS). ActiveMQ, кроме Java, можно также использовать из таких языков программирования, как .NET, C, Delphi Perl, Python, PHP и Ruby [7]. ActiveMQ поддерживает большое количество транспортных протоколов, среди которых STOMP, REST, AMQP, MQTT, OpenWire, WebSockets [7].

Данный брокер обеспечивает функции кластеризации, хранения сообщений, кэширования, поддержку возможности использования баз данных. Он предоставляет поддержку взаимодействия с более чем одним клиентом или сервером. Связь управляется с помощью таких функций, как кластеризация компьютеров и возможность использовать любую базу данных в качестве поставщика сохраняемости, помимо виртуальной памяти, кеша и сохраняемости журнала. ActiveMQ использует несколько режимов для обеспечения высокой доступности, среди них механизмы блокировки на уровне строк файловой системы и базы данных, совместное использование хранилища сохраняемости через общую файловую систему.

В ActiveMQ ведется запись в журналы на диске. Это такая структура данных на диске, в которую можно добавлять данные, но нельзя изменять или удалять, и состоящая из нескольких файлов. Такой механизм является эффективным для хранения и извлечения сообщений, так как доступ к диску для обеих операций является последовательным. И на жестких дисках, и на SSD накопителях последовательный доступ оказывается быстрее произвольного.

ActiveMQ поставляется с рядом подключаемых стратегий сохранения сообщений. Они идут в форме адаптеров персистентности, которые можно рассматривать, как движки сохранения сообщений. Когда брокер получает персистентные сообщения, они сначала записываются на диск в журнал. Журнал содержит лог всех входящих сообщений, а также сведения о тех сообщениях, которые были подтверждены, как прочитанные получателем. Входящие сообщения сериализуются брокером в независимое от протокола представление объекта, а затем маршализируются в двоичную форму, которая записывается в конец журнала. Работа брокера – это автоматический процесс, не требующий расширенного администрирования и мониторинга.

## **2. Apache Kafka**

Apache Kafka – брокер сообщений, созданный корпорацией LinkedIn как распределенная программа [8]. Он отличается высокой скоростью работы, масштабируемостью, отказоустойчивостью, возможностью передавать данные пакетами и способностью секционировать.

Работа брокера осуществляется следующим образом: приложения-отправители посылают сообщения на узел Kafka, затем указанные сообщения обрабатываются приложениями-потребителями. Указанные сообщения сохраняются, а потребители подписываются для получения новых сообщений. Kafka гарантирует, что сообщения будут упорядочены в той последовательности, в которой они поступали на узел. Kafka не отслеживает, прочитал ли получатель сообщения и поэтому не удаляет их после прочтения. Вместо этого сообщения хранятся на узле их в течение некоторого времени и по его истечении удаляются. Получатели сами опрашивают Kafka, не появилось ли у него новых сообщений, и указывают, какие записи им нужно прочесть.

Kafka является распределенной системой [9]. Хранение и пересылка сообщений происходит параллельно на разных серверах системы, что гарантирует высокую надежность. Если несколько серверов перестанут работать, сообщения все равно будут пересылаться и обрабатываться дальше.

Система легко масштабируется горизонтально, что дает возможность увеличить производительность путем добавления новых серверов. Максимальное количество машин, которых можно добавить в систему, не ограничено.

Записи в Apache Kafka хранятся в виде журнала транзакций [8]. Он представляет собой очередь сообщений, в которую можно только добавлять записи, удалять или модифицировать их нельзя. Такой подход обеспечивает упорядоченность записей.

## **3. RabbitMQ**

RabbitMQ также, как и Kafka, является распределенным и горизонтально масштабируемым брокером сообщений. Он создан на основе системы Open Telecom Platform, а в качестве движка базы данных для хранения сообщений используется Mnesia [10]. Брокер написан на языке программирования Erlang, не широко используемом языке программирования, но хорошо приспособленном для таких задач. RabbitMQ является

отличным решением для построения сервис-ориентированной архитектуры и распределения отложенных ресурсоемких задач.

Данная система обмена сообщениями выступает в качестве сервиса-посредника и имеет множество функций: поддержку шифрования, разграничение права доступа, сохранение сообщений на диск, работу в кластерах и дублирование сервисов. Для построения кластерных решений поддерживается горизонтальное масштабирование.

RabbitMQ позволяет взаимодействовать различным программам при помощи протокола AMQP. Брокер включает в себя сервер, библиотеки поддержки протоколов HTTP, XMPP и STOMP, клиентские библиотеки для Java и .NET Framework. Имеется реализация клиентов для большинства популярных языков программирования, среди которых Perl, Python, Ruby, PHP [10].

Так как в RabbitMQ каждый получатель запрашивает/выгружает разное количество сообщений, то распределение работы может стать неравномерным, что повлечет задержки и потерю порядка сообщений во время обработки. Для предупреждения этого и балансировки нагрузки каждый получатель настраивает предел предварительной выборки – ограничение на количество скопившихся неподтвержденных сообщений.

RabbitMQ включает четыре способа маршрутизации в отличие от Apache Kafka, реализующего запись на диск без маршрутизации. [11] Это позволяет использовать мощный и гибкий набор шаблонов обменов сообщениями. RabbitMQ позволяет пересылать сообщения при помощи маршрутизации с применением прямого и разветвляемого обменов. Также брокер дает возможность выборочной маршрутизации с тематическим обменом, основываясь на соответствии символьной подстановке в предлагаемом ключе маршрутизации, а обмен заголовками представляет некий альтернативный подход к маршрутизации сообщений с применением самого сообщения.

#### **4. IBM MQ**

IBM MQ – это сервер управления очередями сообщений, который был разработан компанией IBM в 1993 году под названием MQSeries. Позже он был переименован в IBM MQ [12]. Данный брокер доступен на большом количестве платформ, таких как UNIX, Linux и Microsoft Windows.

Архитектура IBM MQ не зависит от времени. Сообщения отправляются, независимо от того, работает ли программа-получатель в данный момент времени. Если она не работает, сообщение попадает в очередь и будет доставлено, когда получатель станет доступным. Порядок сообщений при этом сохранится, также сообщениям можно задать уровень приоритета.

С помощью IBM MQ можно отправлять различные сообщения. Например, это могут быть файлы большого размера либо сообщения с командой, которая запускает приложения. Последние используются для построения архитектуры системы управляемой сообщениями. Также возможно преобразование данных для различных протоколов и архитектур. Можно изменить кодировку или порядок разрядов в байте.

В IBM MQ хорошо продумана архитектура маршрутизации. Она позволяет доставлять сообщения по альтернативному пути через сеть диспетчеров. IBM MQ может быть реализован

в виде кластера, в котором обработка сообщений выполняется несколькими экземплярами для увеличения производительности.

Основной компонент IBM MQ – диспетчер очереди, который отвечает за действия, непосредственно не связанные с перемещением данных: хранение файлов, временную привязку, запуск приложений и другие [13]. Диспетчеры открыты для взаимодействия через прямое программное соединение для программ, выполняющихся с диспетчером очереди на одном компьютере, и через сетевое соединение. Сетевое соединение позволяет повысить гибкость системы, а прямое, в свою очередь, улучшает быстродействие.

Для взаимодействия диспетчеров между собой организуются специальные каналы. Один канал используется для данных в одном направлении, второй канал – для обратных данных. В сети TCP/IP каждому каналу назначается отдельный порт.

## 5. NATS

NATS – это относительно новый брокер сообщений с открытым исходным кодом. Изначально он был написан на языке программирования Ruby, а позже перенесен на Go. Клиенты NATS доступны на многих наиболее широко используемых языках, среди которых Go, C и C #, Java, PHP, Python и Rust [14].

Основными принципами разработки NATS являются производительность, масштабируемость и простота использования. Чтобы подключить новый узел, процессу NATS достаточно указать адрес любого узла кластера, после чего он скачивает всю топологию и определяет другие узлы.

Производительность обеспечивается за счет того, что все сообщения в кластере доставляются напрямую от отправителя получателю. Промежуточные шаги отсутствуют, что гарантирует минимальную задержку. Но из-за того, что сообщения не записываются на диск, сервисы-получатели должны аккуратно завершать свою работу. Сначала они отписываются от новых сообщений, после чего обрабатывают те, которые получили раньше и только после этого останавливают процесс.

Сообщения в NATS группируются по темам. Каждый узел кластера знает подписчиков других узлов и на какие темы они подписаны. Кроме того, клиенты NATS также знают топологию кластера и способны самостоятельно переподключаться в случае потери связи со своим узлом [15].

## 6. ZeroMQ

ZeroMQ – это библиотека обмена сообщениями, которая позволяет организовать быстрый асинхронный обмен сообщениями между приложениями. Она работает на различных архитектурах и поддерживается большинством языков программирования. Изначально она позиционировалась как интерфейс для обмена сообщениями, а затем – как коммуникационный протокол, основанный на TCP/IP [16].

ZeroMQ отличается от рассмотренных выше систем тем, что не является полноценной системой. Это простой программный интерфейс, который дает возможность создать собственное связующее программное обеспечение любой сложности.

Преимуществом ZeroMQ является высокая производительность. Она достигается за счёт того, что в данной системе отсутствуют издержки протокола AMQP. Зато присутствует

широковещательный протокол с гарантированной доставкой, а также набор вызовов для многопоточной рассылки сообщений нескольким адресатам. ZeroMQ использует собственную разработку, заключающуюся в агрегированной отправке нескольких сообщений в одном TCP-пакете. Это позволяет уменьшить количество системных вызовов.

По сравнению с сокетами ZeroMQ обладает большей простотой использования. Для того, чтобы передать сообщение в ZeroMQ достаточно инициировать отправку сообщения, а дробление и отправка делается в отдельном потоке, асинхронно с выполнением пользовательского кода. Такая особенность удобна для реализации механизмов событийной обработки. В ZeroMQ отсутствует типизация сообщений, они не интерпретируются интерфейсом и являются массивом двоичных данных. Таким образом, через ZeroMQ можно передавать что угодно, например, сообщения JSON или двоичные форматированные данные типа BSON, Protocol Buffers или Thrift, не испытывая при этом неудобств.

Несмотря на то, что ZeroMQ является низкоуровневым интерфейсом, он обеспечивает хорошую масштабируемость. В ZeroMQ реализована децентрализованная схема обмена сообщениями. Сокет ZeroMQ равномерно распределяет нагрузку по сети за счет возможности быть подключенным к нескольким получателям. Одновременно с этим существует возможность получения сообщения от множества отправителей.

Таким образом, в ZeroMQ успешно реализован компромисс между эффективностью и функциональностью, позволяющий организовать быстрый асинхронный обмен сообщениями между высоконагруженными приложениями.

С учетом вышеизложенной информации о брокерах, было проведено сравнение, результаты которого приведены в таблице 1.

В плане масштабируемости рассмотренные брокеры имеют равнозначные возможности. Все они также поддерживают популярные языки программирования и протоколы. Все брокеры, кроме IBM MQ, являются программным обеспечением с открытым исходным кодом. Они хранят сообщения на дисках. Apache Active MQ, RabbitMQ и NATS дополнительно к этому поддерживают хранение в базах данных.

Что касается *потоковой обработки данных*, то по результатам сравнения выделяются два брокера: Apache Kafka и NATS.

Обработка большого массива данных в реальном времени предоставляет широкие возможности для использования в современных системах. Apache Kafka Streams и NATS Streaming позволяют обрабатывать информационные потоки в режиме реального времени.

Библиотека Apache Kafka Streams позволяет распределенным приложениям в режиме реального времени обрабатывать данные, хранящиеся в Kafka, реализуя стандартные классы этой платформы потоковой обработки без развертывания отдельных кластеров [17]. Благодаря этому инструменту можно организовать потоковую обработку данных прямо внутри кластера, без привлечения дополнительных технологий.

Kafka Streams API полностью скрывает сложность обслуживания отправителей и потребителей сообщений Kafka, позволяя сосредоточиться на логике обработки информационных потоков. Она способна выполнять обработку данных с сохранением и/или без сохранения состояния в режиме реального времени [18].

Таблица 1 – Сравнительная таблица систем обмена сообщениями

№ п/п	Название	Apache ActiveMQ	Apache Kafka	RabbitMQ	IBM MQ	NATS	ZeroMQ
1	2	3	4	5	6	7	8
1	Язык программирования, на котором разработана система	Java	Java	Erlang	–	Ruby, Go	C++
2	Поддерживаемые языки	C, C#, Haxe, Node.js, Perl, Racket, Python, Ruby on Rails	C, C++, Go, Java, .NET, Python, Scala	Java, Ruby, Python, PHP, Perl, Rust, Go, JavaScript, C, C++	C, C#, C++, Java, Visual Basic, Assembler, Node.js, Ruby, Python	Go, C, C#, Java, PHP, Python, Rust	C, C++, C#, Java, Python, Ruby, Go, NodeJS, Perl, Rust
3	Открытый исходный код	Да	Да	Да	Нет	Да	Да
4	Хранение сообщений	БД / Диск	Диск	БД / Диск	Диск	БД / Диск	Диск
5	Масштабируемость	Да	Да	Да	Да	Да	Да
6	Поддерживаемые протоколы	OpenWire, Stomp, WSIF, WS, AUTO, AMQP, MQTT, REST	Binary over TCP	HTTP, MQTP, STOMP, AMQP	AMQP, MQI, REST, JMS	Google Protocol Buffer	IPC, TCP, UDP, TIPC, WebSoc ket
7	Возможность потоковой обработки	Нет	Да	Нет	Нет	Да	Нет

Расширенный протокол сообщений NATS Streaming также предоставляет пользователям возможность потоковой обработки больших объемов данных. Он реализует собственный расширенный формат сообщений с использованием Google Buffer Protocol [19]. Эти сообщения передаются как полезные данные двоичного сообщения через базовую платформу NATS.

NATS Streaming предлагает настраиваемое сохранение сообщений: в памяти или в базе данных. Подсистема хранения использует общедоступный интерфейс, который позволяет пользователям разрабатывать свои собственные реализации. Протокол гарантирует доставку сообщений за счет того, что сообщения сохраняются в памяти и повторно доставляются по мере необходимости.

Apache Kafka Streams и NATS Streaming основаны на следующих важных концепциях потоковой обработки [20]:

- управление состоянием приложения;
- быстрые и эффективные агрегации и объединения;
- различие между временем события и временем его обработки;
- непрерывная обработка данных, поступающих с опозданием и помехами.

Описанные выше сервисы применяются для управления запросами о состоянии приложений в режиме реального времени и безкластерной разработки распределенных сервисов с последующим масштабированием. Они широко используются на практике в различных IT-проектах государственных и частных компаний.

### Список литературы

1. Хоп Г., Вульф Г. Шаблоны интеграции корпоративных приложений. М.: ООО «И.Д. Вильямс», 2007. 672 с.
2. Беседина К.В. Современные подходы к интеграции корпоративных приложений / К. В. Беседина // Academy. 2017. № 5 (20). С. 41-43.
3. Линева Ф. А. Обзор систем обмена сообщениями / Ф. А. Линева // Молодой ученый. 2017. № 19 (153). С. 29-32.
4. Фаулер М. Архитектура корпоративных программных приложений. М.: ООО «И.Д. Вильямс», 2016. 544 с.
5. Оринштейн Д. Интеграция корпоративных приложений. М.: Microsoft Press, 2002. 454 с.
6. Пролетарский А.В., Березкин Д.В., Гапанюк Ю.Е., Козлов И.А., Попов А.Ю., Самарев Р.С., Терехов В.И. Методы ситуационного анализа и графической визуализации потоков больших данных Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2018. № 2. С. 98-123.
7. Jakub Korab. Understanding Message Brokers. Learn the Mechanics of Messaging though ActiveMQ and Kafka. O'Reilly, 2017. 132 p.
8. Apache Kafka Documentation [Электронный ресурс] URL: <https://kafka.apache.org/documentation/gettingStarted> (дата обращения 15.09.20).
9. Estrada R. Apache Kafka 1.0 Cookbook: Over 100 practical recipes on using distributed enterprise messaging to handle real-time data Packt Publishing, 2017. 250 с.
10. Alvaro Videla, Jason J. W. Williams RabbitMQ in Action: Distributed Messaging for Everyone 2012. 312 с.
11. Чадов А. Ю.; Михальченко К. Э. Сравнительный анализ AMQP брокеров сообщений для использования в качестве элемента децентрализованной системы разграничения доступа Комплексная защита информации: материалы XXIV научно-практической конференции. Витебск. 21–23 мая 2019 г.: УО ВГТУ. Витебск, 2019. С. 261-267.

12. Introduction to IBM MQ [Электронный ресурс] URL: [https://www.ibm.com/support/knowledgecenter/en/SSFKSJ\\_8.0.0/com.ibm.mq.pro.doc/q001020\\_.htm](https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.pro.doc/q001020_.htm) (дата обращения 15.09.20).
13. Марц Н., Уоррен Дж. Большие данные. Принципы и практика построения масштабируемых систем обработки данных в реальном времени. М.: Вильямс, 2016. 368 с.
14. NATS Documentation [Электронный ресурс] URL: <https://docs.nats.io/> (дата обращения 15.09.20).
15. Andrade H.C.M., Gedik B., Turaga D.S. Fundamentals of stream processing: application design, systems, and analytics. Cambridge University Press, 2014. 558 p.
16. ZeroMQ. An introduction [Электронный ресурс] URL: <http://wiki.zeromq.org/blog:introduction> (дата обращения 15.09.20).
17. Andrade, Henrique C.M., Bugra G., Deepak S. Turaga. Fundamentals of Stream Processing: Application Design, Systems, and Analytics. NY: Cambridge University Press, 2014. 529 p.
18. Нархид Н., Шапира Г., Палино Т. Apache Kafka. Поточковая обработка и анализ данных. СПб.: Питер, 2019. 320 с.
19. William P. Bejeck Jr. Kafka Streams in Action: Real-time apps and microservices with the Kafka Streams API Manning Publications, 2018. 280 p.
20. Самарев Р.С. Обзор состояния области потоковой обработки данных / Р. С. Самарев // Труды Института системного программирования РАН. 2017. № 1 (29). С. 231-260.

## References

1. Hop G., Wolf G. Patterns of Enterprise Application Integration. М.: J.D. Williams LLC, 2007. 672 p.
2. Besedina K.V. Modern Approaches to Integrating Corporate Applications / K.V. Besedina // Academy. 2017. No 5 (20). pp. 41-43.
3. Linev F.A. Review of Messaging Systems / F.A. Lynev // Young Scientist. 2017. No 19 (153). pp. 29-32.
4. Fowler M. Architecture of Corporate Software Applications. - М.: J.D. Williams LLC, 2016. 544 p.
5. Orinstein D. Corporate Application Integration. М.: Microsoft Press, 2002. 454 p.
6. Proletarian A.V., Berezkin D.V., Gapanyuk Y.E., Kozlov I.A., Popov A.Y., Sama-rev R.S., Terekhov V.I. Methods of situational analysis and graphic visualization of big data flows - Herald of the Moscow State University by N.E. Bauman. Ser. Instrumentation. 2018. No 2. pp. 98-123.
7. Jakub Korab. Understanding Message Brokers. Learn the Mechanics of Messaging though ActiveMQ and Kafka. O'Reilly, 2017. 132 p.
8. Apache Kafka Documentation [E-resource] URL: <https://kafka.apache.org/documentation/gettingStarted> (Address date 15.09.20).
9. Estrada R. Apache Kafka 1.0 Cookbook: Over 100 practical recipes on using distributed enterprise messaging to handle real-time data Packt Publishing, 2017. 250 p.
10. Alvaro Videla, Jason J. W. Williams RabbitMQ in Action: Distributed Messaging for Everyone 2012. 312 p.
11. Chadov A. Yu.; Michalchenko K.E. Comparative analysis of AM'P brokers messages for use as an element of a decentralized access delineation system / Comprehensive information

- protection: materials of the XXIV Scientific and Practical Conference. Vitebsk. May 21-23, 2019: VGTU. pp. 261-267.
12. Introduction to IBM MQ [E-resource] URL: [https://www.ibm.com/support/knowledgecenter/en/SSFKSJ\\_8.0.0/com.ibm.mq.pro.doc/q001020\\_.htm](https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.pro.doc/q001020_.htm) (Address date 15.09.20).
  13. Martz N., Warren J. Big data. Principles and practices in building scalable real-time data processing systems. M.: Williams, 2016. 368 p.
  14. NATS Documentation [Электронный ресурс] URL: <https://docs.nats.io/> (Address date 15.09.20).
  15. Andrade H.C.M., Gedik B., Turaga D.S. Fundamentals of stream processing: application design, systems, and analytics. Cambridge University Press, 2014. 558 p.
  16. ZeroMQ. An introduction [E-resource] URL: <http://wiki.zeromq.org/blog:introduction> (Address date 15.09.20).
  17. Andrade, Henrique C.M., Bugra G., Deepak S. Turaga. Fundamentals of Stream Processing: Application Design, Systems, and Analytics. – NY: Cambridge University Press, 2014. 529 p.
  18. Narhid N, Shapira G, Palino T. Apache Kafka. Streaming and data analysis. St. Petersburg: Peter, 2019. 320 p.
  19. William P. Bejeck Jr. Kafka Streams in Action: Real-time apps and microservices with the Kafka Streams API Manning Publications, 2018. 280 p.
  20. Samarev R.S. Review of the State of Streaming Data / R.S. Samarev // Proceedings of the Institute of System Programming ras. 2017. No 1 (29). pp 231-260.
-



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 519

## ОСОБЕННОСТИ ОРГАНИЗАЦИИ И ФУНКЦИОНИРОВАНИЯ СЕТЕЙ ПРОВАЙДЕРА

<sup>1</sup>Раскатова М.В., <sup>2</sup>Новиков К.Д.

ФГБОУ ВО «Национальный исследовательский университет «МЭИ», Москва, Россия, (111250, г. Москва, улица Красноказарменная, д.14),  
e-mail: 1) [ultimatedoker@gmail.com](mailto:ultimatedoker@gmail.com), 2) [marina@raskatova.ru](mailto:marina@raskatova.ru)

В статье описываются варианты построения сетей провайдера, а также способы их расширения. Рассматриваются типичные отклонения от общепринятых стандартов построения сетей передачи данных, их причины и возможные последствия. Описывается существующее программное обеспечение для эксплуатации сетей, а также предлагается вариант улучшения и облегчения трудового процесса инженеров.

Ключевые слова: создание инфраструктуры, сеть передачи данных, централизованное управление.

## PECULIARITIES OF ORGANIZATION AND FUNCTIONING OF PROVIDER NETWORKS

<sup>1</sup>Raskatova M.V. <sup>2</sup>Novikov K.D.

<sup>1</sup>National Research University "Moscow Power Engineering Institute, Moscow, Russia (111250, Moscow, street Krasnokazarmennaya, 14),  
e-mail: 1) [ultimatedoker@gmail.com](mailto:ultimatedoker@gmail.com), 2) [marina@raskatova.ru](mailto:marina@raskatova.ru)

The article describes the options for building provider networks, as well as ways to expand them. Typical deviations from generally accepted standards for building data transmission networks, their causes and possible consequences are considered. Describes the existing software for the operation of networks, as well as proposed options to improve and facilitate the work process of engineers.

Keywords: creation of infrastructure, data transmission network, centralized management.

Первой в России сетью передачи данных принято считать «Релком». Эта сеть была организована в 1990 году, специалистами кооператива «Демос». На тот момент, никто даже не задумывался, в какие колоссальные объемы выльется их эксперимент по соединению ЭВМ в Ленинграде и Новосибирске, с использованием обыкновенного аналогового телефонного модема, но именно с этих событий началась эпоха развития интернет-технологий в Российской Федерации.

По мере развития интернета, как такового, российские провайдеры перепробовали практически все существующие технологии для предоставления портов передачи данных широкому кругу населения. Первой, действительно распространенной технологией обмена информацией, можно по праву считать xDSL, во всех её вариациях. Одновременно со

стартом подключения первых абонентов, возник вопрос о том, какую топологию сети использовать и каким образом должно происходить расширение провайдерских сетей.

В настоящее время, все без исключения провайдеры используют смешанную топологию. Это обусловлено тем, что сети подвержены постоянному расширению и нуждаются в своевременном улучшении показателей отказоустойчивости. Эталонная сеть, к которой стремятся все провайдеры, должна представлять собой резервированную древовидную структуру, в которой соблюдается баланс отказоустойчивости и стоимости. Пример такой топологии, можно увидеть на рисунке 1.

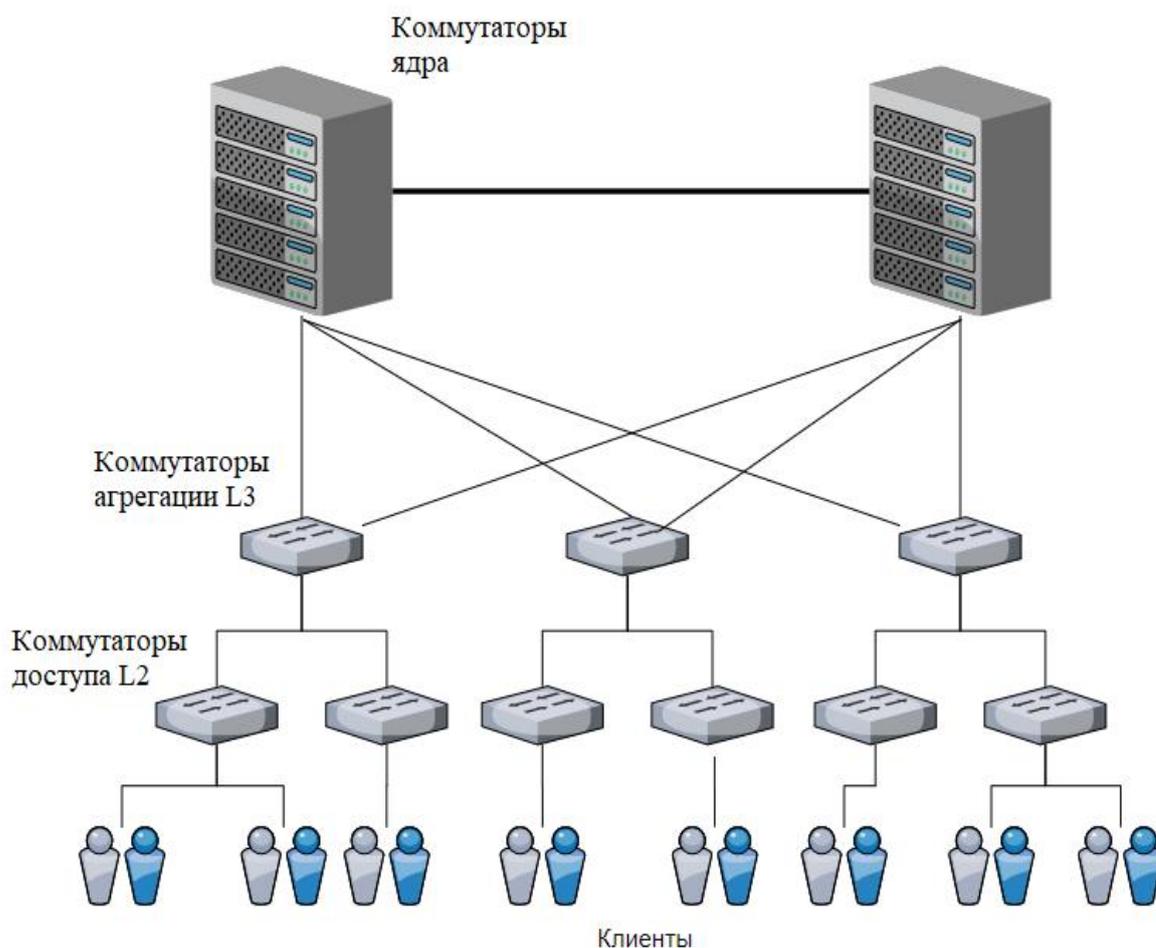


Рисунок 1 – Эталонная модель построения сети

Узловые коммутаторы, обладающие очень высоким запасом производительности и надежности, в эталонной модели резервируются с нескольких ядер. Из этого вытекает очевидное правило, что чем выше в древовидной модели сети находится оборудование, тем выше должен быть его уровень отказоустойчивости.

Существует два основных типа компаний, специализирующихся на предоставлении широкополосного доступа в интернет: ориентированные на физических лиц и на юридических. Помимо этого, особо крупные компании, такие как «ВымпелКом», с юридической точки зрения могут работать как с первым типом клиентов, так и со вторым,

однако на практике, можно сказать, что разными клиентами занимаются разные подразделения внутри компании, практически не связанные между собой.

Эталонная модель, в которой резервированием не обладает только окончное коммутационное оборудование, едва ли является достижимой на практике. Причин всего две: во-первых, стремление руководства к экономии средств, а во-вторых, желание клиентов получить порт СПД в кратчайшие сроки.

Помимо всего вышеперечисленного, в России существует частая практика покупки небольших компаний, более крупными конкурентами. Иногда, при покупке компании провайдера, выкупленная сеть продолжает функционировать с относительной автономностью, без вмешательства компании-владельца. В противовес этому, часто встречается ситуация, когда небольшую городскую сеть выкупают для возможности использования её узлов. Это позволяет подключать новые объекты, не строя промежуточную трассу, а использовать уже существующие линии коммуникаций. Конечно, можно просто арендовать канал до объекта, но далеко не всегда, это обладает достаточной степенью окупаемости.

При варианте, в котором крупная компания выкупает относительно небольшую сеть, для её дальнейшей эксплуатации, зачастую приходится организовывать коммутационные стыки и переносить сервера авторизаций, базы данных клиентов и т.д. с одного ядра сети, на другое, находящееся с “другой” стороны стыка сетей. Безусловно, можно осуществить подобный перенос, обеспечив при этом достаточный уровень надежности, но на практике, всё это упирается в коммерческую часть вопроса.

Как правило, реальные крупные сети являют собой некую “кашу” из различных топологий, моделей оборудования и протоколов передачи информации. Можно сказать, что главной причиной вышеописанных проблем является временной фактор. Если основу коммутационной среды строят согласно определенному плану соблюдая установленные проектом нормы, то дальнейшее расширение этой сети будет происходить через какой-то неопределенный промежуток времени. Необходимо будет учитывать следующие факторы, которые невозможно предсказать при создании изначального проекта сети:

- некоторые модели коммутационного оборудования будут сняты с производства, появятся другие производители, предлагающие более высокий уровень цены\качества;
- возможность присоединения к существующей сети другой, имеющей отличную конфигурацию и топологию;
- появление новых технологий и протоколов, обладающих существенными преимуществами над существующими;
- экономическая ситуация в компании, которая может не позволить сделать полноценное резервирование, при подключении новых узлов сети;
- возможность потери прав и возможностей, на эксплуатацию узлового оборудования на каком-либо сегменте сети.

Рассмотрим последний пункт подробнее. Довольно типичной проблемой у провайдеров, является вопрос о том, где физически установить коммутационное оборудование. В качестве очевидного примера, напрашиваются компании, работающие с юридическими лицами. Зачастую, узел связи устанавливается непосредственно на

клиентской территории, с соответствующим договором, после чего, от установленного узла происходит подключение дочерних объектов. Спустя какое-то время, клиент, на территории которого установлено узловое оборудование, закрывает фирму или просто переезжает на другой адрес, а новый арендатор, по каким-то причинам не хочет продолжать сотрудничать с компанией-провайдером. В результате приходится переносить узел связи на какой-то другой физический адрес, что влечет за собой потерю надлежащего уровня резервирования и временную неработоспособность сегмента сети. Пример типичной реально существующей топологии сети, показан на рисунке 2.

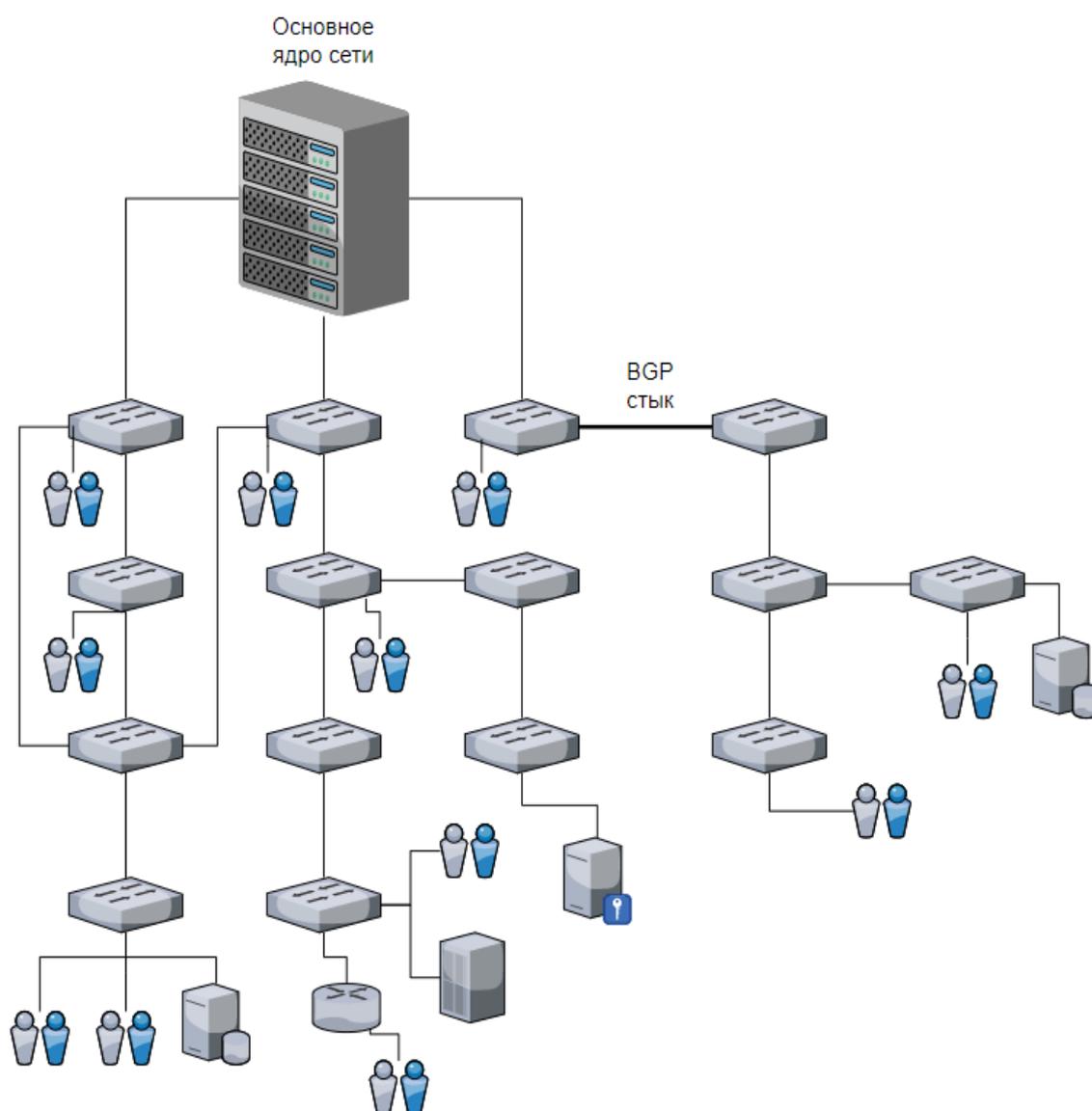


Рисунок 2 – Пример фактически существующих сетей

Не лишним будет упомянуть, ещё одно отличие реально функционирующих сетей, от эталона: клиенты, у обычных провайдеров, могут включаться не только с окончного оборудования, но и с узловых высокопроизводительных коммутаторов. Подобная ситуация возникает в случаях, когда подключить клиента непосредственно с узлового коммутатора менее трудно и/или ресурсозатратно.

Не редки ситуации, когда Radius-сервер находится по другую сторону BGP стыка (например, в ранее рассмотренной ситуации, когда маленькая сеть объединяется с более крупным провайдером), что может повлечь собой отказ в работе интернета у всех клиентов находящихся за стыком, при обрыве магистрального кабеля, даже если в рассматриваемом сегменте сети присутствует резервный Uplink стык со сторонним провайдером.

Все подобные ситуации, с недостатком резервирования, возможно решить, но обычно, в виду высокой ресурсозатратности, владелец компании-провайдера делает упор на покупку хорошо зарекомендовавшего себя оборудования и оперативное устранение проблем, возникающих на сети, а редкие перебои в работе иногда дешевле просто компенсировать клиенту, чем строить дорогостоящие резервные магистрали.

Если рассматривать эталонную модель сети, в которой всё установленное коммутационное оборудование создано одной и той же фирмой производителем, то управление сетью будет осуществляться унифицированными командами, для которых возможно создание не сложных скриптов. К сожалению, в настоящей среде передачи данных, оборудование одного производителя – непозволительная роскошь. Эти обстоятельства ведут к тому, что инженерам отдела эксплуатации среды передачи данных требуются обширные навыки работы со множеством различных коммутаторов, каждый из которых обладает уникальным синтаксисом и алгоритмом работы.

При эксплуатации крупных сетей, важными аспектами работы являются: мониторинг, конфигурация, хранение истории в удобном виде и прикладные задачи. Разберем каждый аспект в отдельности:

### 1. Мониторинг

Под мониторингом сети подразумевается контроль работоспособности отдельных узлов и сегментов сети. Обычно, для этого достаточно использования обыкновенного ICMP протокола: если он сообщит о недоступности конечного адреса, то необходимо уведомить инженера. Для разрешения вопроса контроля работы сети, существует множество приложений, как платных, так и бесплатных, позволяющих наблюдать за состоянием сети, вне зависимости от производителя оконечного оборудования. Пожалуй, одной из самых популярных утилит, является Zabbix. Он является open source решением, способным удовлетворить преобладающее большинство требований провайдеров.

### 2. Конфигурация

Гораздо сложнее обстоят дела с конфигурацией сложных сетей. Дело в том, что широкое разнообразие оборудования влечет за собой необходимость создавать отдельные скрипты для каждого производителя, либо делать всё в ручном режиме. Даже такие элементарные действия, как блокировка порта СПД, на коммутаторах разных производителей, происходит с использованием различных команд, несмотря на одинаковую логику (открыть конфигурацию интерфейса и внести в неё строку shutdown).

### 3. Хранение истории

Довольно часто у инженеров, которые занимаются эксплуатацией сетей, возникает необходимость узнать информацию о бывшем пользователе порта СПД. К примеру, при подключении нового абонента, необходимо проверить, была ли ранее проложена в помещение клиента витая пара. Имея эту информацию в удобном виде, можно существенно облегчить включение клиента, просто сконфигурировав уже сконмутированный порт передачи данных.

Обычно подобные данные ведутся в различных таблицах, которые доступны только инженерам, либо в программах собственной разработки.

#### 4. Прикладные задачи

В этот пункт, пожалуй, можно внести такие несущественные и разнообразные задачи, как расчет значений политик, которые используются при создании ограничителя пропускной способности порта, генерацию надежных паролей, которые будет удобно вводить руками (под удобством подразумевается отсутствие в нём неоднозначностей, таких как латинская буква O и цифра 0), а также создание\закрытие\изменение статусов клиентских заявок и т.д.

Несмотря на относительно высокую трудоемкость, в отношении эксплуатации сетей, процесс обслуживания СПД можно довольно сильно упростить, разработав универсальное приложение, способное выполнять основную конфигурацию оборудования, без каких-либо зависимостей от производителя устройств. Для этого потребуется однократное создание некой таблицы соотношения синтаксиса устройств различных производителей. В дальнейшем, потребуется объединить вышеописанную утилиту, в единый модуль управления сетями, который будет позволять работать с основными аспектами эксплуатации унифицированными командами в одном приложении.

Подводя итог, можно с уверенностью утверждать, что эксплуатация и расширение крупных современных сетей, является довольно сложной задачей, требующей широкого спектра знаний, а также умения идти на компромисс в вопросах резервирования узлов. Также хочется сказать, что несмотря на огромное разнообразие общепринятых норм и стандартов, они далеко не всегда соблюдаются на практике, что может привести к множеству проблем, даже для самых ответственных и опытных инженеров.

### Список литературы

1. Абросимов Л.И. Базисные методы проектирования и анализа сетей ЭВМ. М.: Университетская книга, 2015.
2. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. 5-е изд. СПб.: Питер, 2016.
3. Куроуз Д., Росс К. Компьютерные сети. Настольная книга системного администратора. М.: Издательство «Э», 2016
4. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. СПб.: Питер, 2012.
5. История русского интернета // Habr URL: <https://habr.com/ru/post/5395/> (дата обращения: 20.09.2020).

### References

1. Abrosimov L.I. Basis methods of computer network design and analysis. Moscow: University Book, 2015.
  2. Oliver V.G., Oliver N.A. Computer networks. Principles, technologies, protocols. 5th ed. St. Petersburg: Peter, 2016.
  3. Kurose D., Ross K. Computer Networks. Desktop book of the system administrator. Moscow: "E" Publishing House, 2016
  4. Tanenbaum E., Weatherall D. Computer Networks. - 5th ed. - St. Petersburg: Peter, 2012.
  5. History of the Russian Internet // Habr URL: <https://habr.com/ru/post/5395/> (Request date: 20.09.2020).
-