

Формирование наборов данных для исследования алгоритмов решения задачи покрытия множества

Д. А. Афонский¹, П. И. Васькин¹, А. А. Лисс^{2✉}, С. А. Павлов¹

¹ ЗАО «Морские компьютерные системы», Санкт-Петербург, Россия

² Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

✉ aaliss@etu.ru

Аннотация. Рассматривается автоматизированная генерация тестовых данных для задачи покрытия множества и описания системы проведения экспериментов компании ЗАО МКС для тестирования алгоритмов ее решения. Предложен алгоритм генерации циклических таблиц покрытия с заданными параметрами (число строк, столбцов, отметок) и соблюдением ключевых ограничений (каждая строка – минимум в двух столбцах, множества строк/столбцов не есть подмножества друг друга). Система проведения экспериментов включает веб-приложение, микросервис и СУБД PostgreSQL и позволяет: генерировать тестовые наборы; запускать и тестировать алгоритмы; анализировать результаты и формировать рейтинги алгоритмов по качеству и скорости решения. Подход ускоряет исследования в области дискретной оптимизации и дает основу для развития методов решения задачи покрытия множества.

Ключевые слова: автоматическое формирование наборов данных, задача покрытия множества, циклическая таблица покрытия, правила преобразования таблиц покрытия

Для цитирования: Формирование наборов данных для исследования алгоритмов решения задачи покрытия множества / Д. А. Афонский, П. И. Васькин, А. А. Лисс, С. А. Павлов // Изв. СПбГЭТУ «ЛЭТИ». 2026. Т. 19, № 6. С. 54–62. doi: 10.32603/2071-8985-2026-19-6-54-62.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Original article

Formation of Data Sets for the Study of Algorithms for Solving the Problem of Covering a Set

D. A. Afonskiy¹, P. I. Vaskin¹, A. A. Liss^{2✉}, S. A. Pavlov¹

¹ Marine Computer Systems CJSC, Saint Petersburg, Russia

² Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

✉ aaliss@etu.ru

Abstract. Automated generation is being considered of test data for the task of covering the set and the description of the experimental system of CJSC MKS for testing algorithms for its solution. An algorithm is proposed for generating cyclic coverage tables with specified parameters (number of rows, columns, marks) and compliance with key constraints (each row is in at least 2 columns, sets of rows/columns are not subsets of each other). The experiment system includes a web application, a microservice, and a PostgreSQL database and allows you to: generate test suites; run and test algorithms; analyze the results and generate ratings of algorithms based on the quality and speed of the solution. The approach accelerates research in the field of discrete optimization and provides the basis for the development of methods for solving the set coverage problem.

Keywords: automatic data set generation, the set coverage problem, cyclic coverage table, the rules for converting coverage tables

For citation: Formation of Data Sets for the Study of Algorithms for Solving the Problem of Covering a Set / D. A. Afonskiy, P. I. Vaskin, A. A. Liss, S. A. Pavlov // LETI Transactions on Electrical Engineering & Computer Science. 2026. Vol. 19, no. 6. P. 54–62. doi: 10.32603/2071-8985-2026-19-6-54-62.

Conflict of interests. The authors declare no conflicts of interests.

В [1] представлен систематический обзор и анализ современных алгоритмов решения задачи покрытия множества (ЗПМ). Проведенный в ней анализ показал возросший интерес исследователей к разработке новых и улучшению известных алгоритмов решения ЗПМ. Особенно следует отметить, что существует большое разнообразие алгоритмов на основе метаэвристик, причем каждая метаэвристика порождает целое множество алгоритмов в зависимости от значений параметров алгоритма и характеристик его частей.

Среди ресурсов для тестирования и сравнения алгоритмов решения ЗПМ наиболее известен ресурс [2]. Он связан с работами Джона Э. Беккера (J. E. Becker) [3], [4], который занимался исследованиями в области оптимизации и дискретной математики, включая задачи покрытия множества.

Однако ресурс не обновлялся долгое время, его тестовые данные могут не отражать современные требования или не включать новые типы задач.

Автоматическое формирование наборов данных для исследования алгоритмов решения задачи покрытия множества актуально как с теоретической, так и с практической значимостью данной проблемы.

Задача покрытия множества имеет множество реальных применений:

- размещение пунктов обслуживания;
- назначение экипажей на транспорте;
- проектирование интегральных схем и конвейерных линий;
- построение расписаний;
- логический анализ данных;
- упрощение булевских выражений и т. д.

В каждом из этих случаев необходимо найти оптимальное или приближенное решение, что требует тестирования и сравнения различных алгоритмов на разнообразных наборах данных.

Для разработки, сравнения и улучшения алгоритмов решения задачи покрытия множества требуется обширный набор тестовых данных, которые невозможно получить без автоматизации.

Автоматическое формирование наборов данных позволяет:

- генерировать данные с заданными характеристиками (размерность, распределение элемен-

тов, соотношение количества подмножеств и элементов и т. д.);

- создавать наборы данных, которые охватывают широкий спектр сценариев, включая крайние и пограничные случаи;

- масштабировать генерацию в зависимости от потребностей исследования;

- воспроизводить эксперименты с одинаковыми наборами данных, что важно для сравнения алгоритмов и проверки результатов.

Автоматическое формирование наборов данных экономит время и ресурсы исследователей. Вместо ручного создания каждого набора можно задать параметры генерации и получить множество вариантов за короткое время. Это особенно важно при итеративном процессе разработки и тестирования алгоритмов, когда требуется многократно проверять их работу на разных данных.

С развитием технологий и появлением новых вычислительных возможностей (например, квантовых вычислений) возникает потребность в тестировании алгоритмов на все более сложных и объемных наборах данных. Автоматизация генерации данных позволяет оперативно адаптироваться к этим изменениям и использовать новые возможности для исследования задачи покрытия множества.

Автоматизация позволяет генерировать такие наборы, которые способствуют более объективному анализу и выбору наиболее подходящих методов для конкретных задач.

Определим требования к набору данных, который должен формировать псевдослучайный генератор таблиц покрытия:

- в наборе должны быть только циклические ТП (таблицы покрытия);
- в ТП должно быть m строк;
- в ТП должно быть n столбцов;
- в ТП должно быть n_{otm} отметок.

Цель: сформировать структуру связей «строка ↔ столбец» с соблюдением ограничений:

- каждая строка входит минимум в 2 столбца;
- общее число связей = n_{otm} ;
- множества строк в столбцах не должны быть подмножествами друг друга;
- множества столбцов в строках не должны быть подмножествами друг друга.

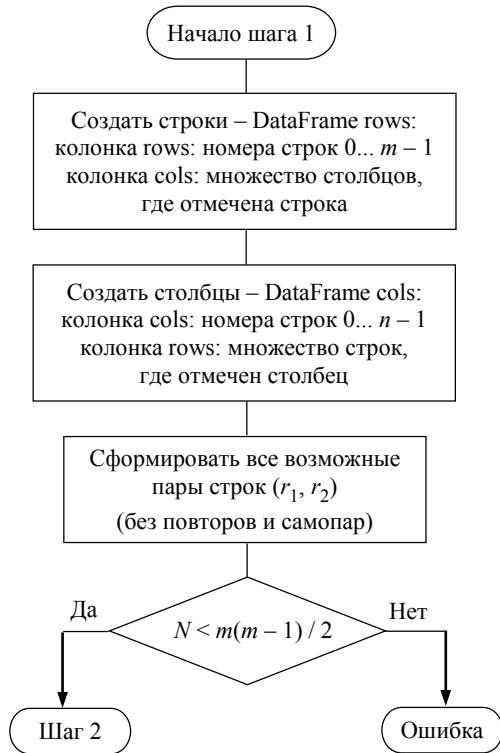


Рис. 1. Блок-схема первого шага алгоритма псевдослучайного генератора циклических таблиц покрытия

Fig. 1. Flowchart of the first step of the pseudo-random cyclic covering tables generator algorithm

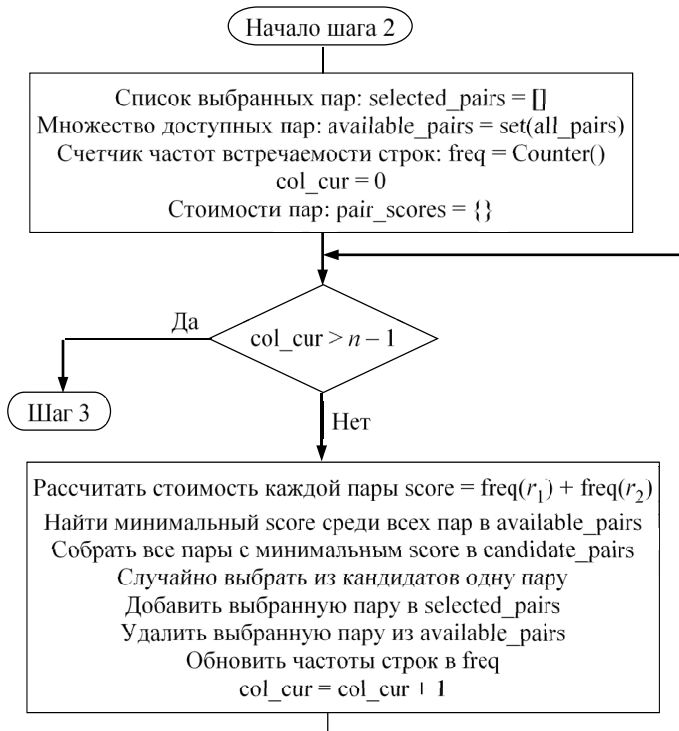


Рис. 2. Блок-схема алгоритма подбора пар строк для предварительного заполнения таблицы покрытия
 Fig. 2. Flowchart of the algorithm for selecting row pairs for pre-populating the coverage table

На рис. 1 приведена блок-схема первого шага алгоритма псевдослучайного генератора циклических таблиц покрытия.

На рис. 2 – блок-схема алгоритма подбора пар строк для предварительного заполнения ТП (с ограничением на частоту строк).

На рис. 3 – блок-схема алгоритма заполнения начальных отметок по выбранным парам.

На рис. 4 – блок-схема алгоритма дополнения отметок до нужного количества.

На рис. 5 – блок-схема алгоритма выбора строки для новой отметки.

На рис. 6 – блок-схема алгоритма попытки поставить отметку.

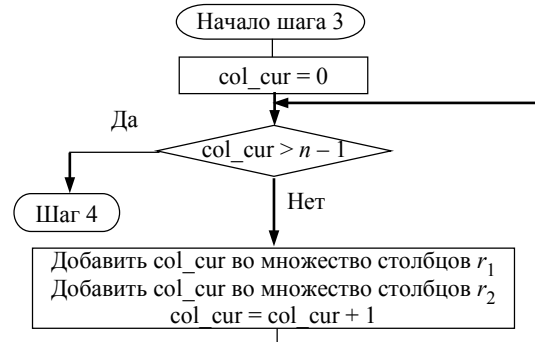


Рис. 3. Блок-схема алгоритма заполнения начальных отметок по выбранным парам

Fig. 3. Flowchart of the algorithm for filling initial marks for selected pairs

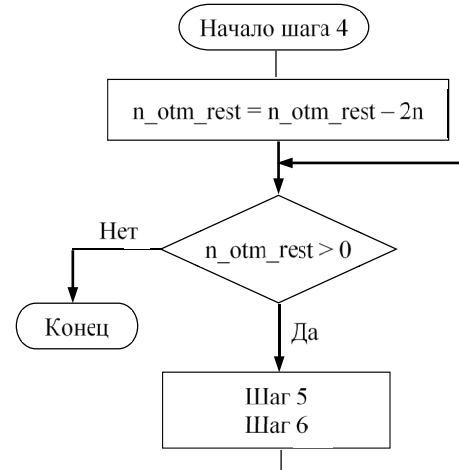


Рис. 4. Блок-схема алгоритма дополнения отметок

Fig. 4. Flowchart of the mark addition algorithm

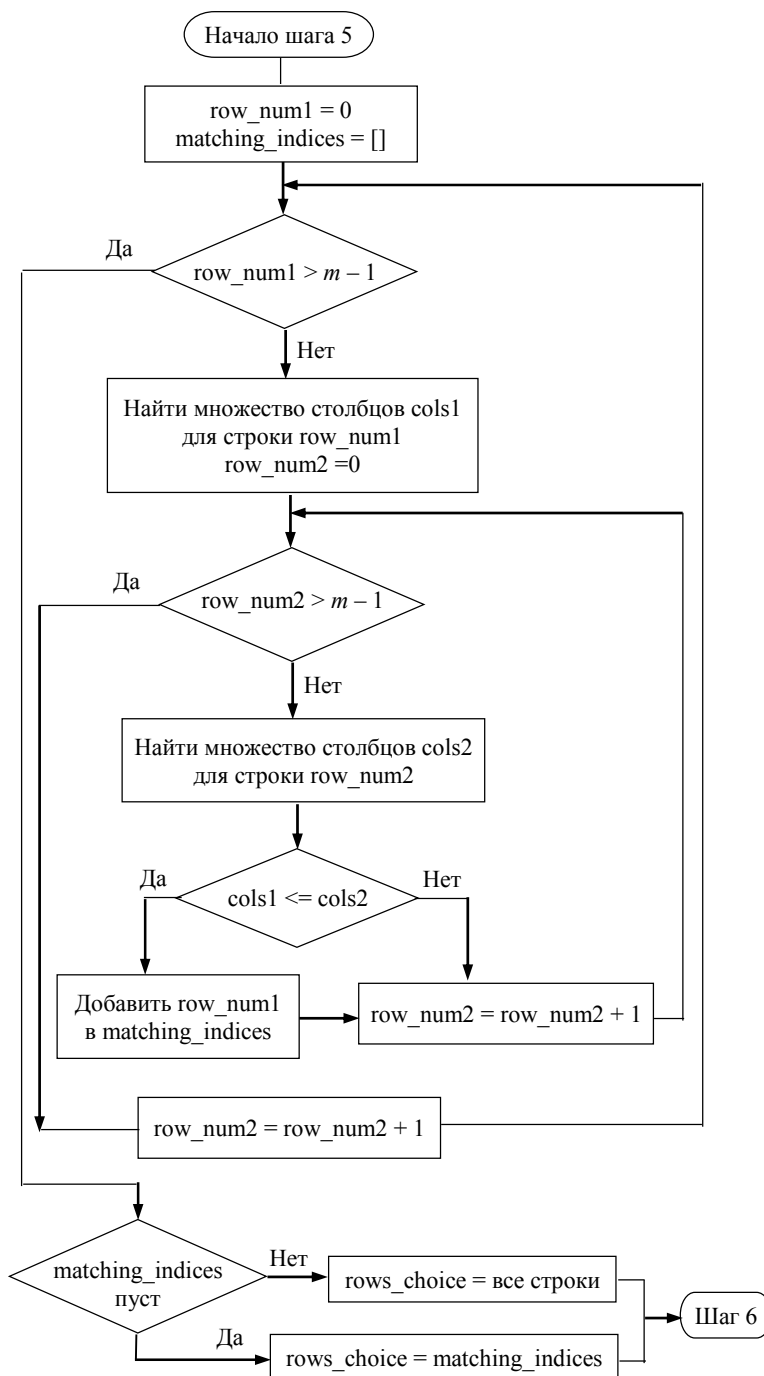


Рис. 5. Блок-схема алгоритма выбора строки для новой отметки
Fig. 5. Flowchart of the algorithm for selecting a row for a new mark

Итоговые свойства результата.

- Полнота покрытия – все столбцы имеют не менее двух отметок и невозможно применить правило удаления столбцов.
- Контроль количества строк – невозможно применить правило удаления строк.
- Случайность – ключевые этапы (выбор пар, столбцов, строк) используют случайные выборки для вариативности генерации.

Практическая значимость. Разработанный алгоритм генерации данных:

- позволяет тестировать и сравнивать алгоритмы ЗПМ в контролируемых условиях;
- дает возможность моделировать сценарии, близкие к реальным задачам (например, с высокой плотностью отметок или асимметричными соотношениями строк/столбцов);
- может быть адаптирован для исследования метаэвристик (генетических алгоритмов, имитации отжига и др.), чувствительных к характеристикам входных данных.

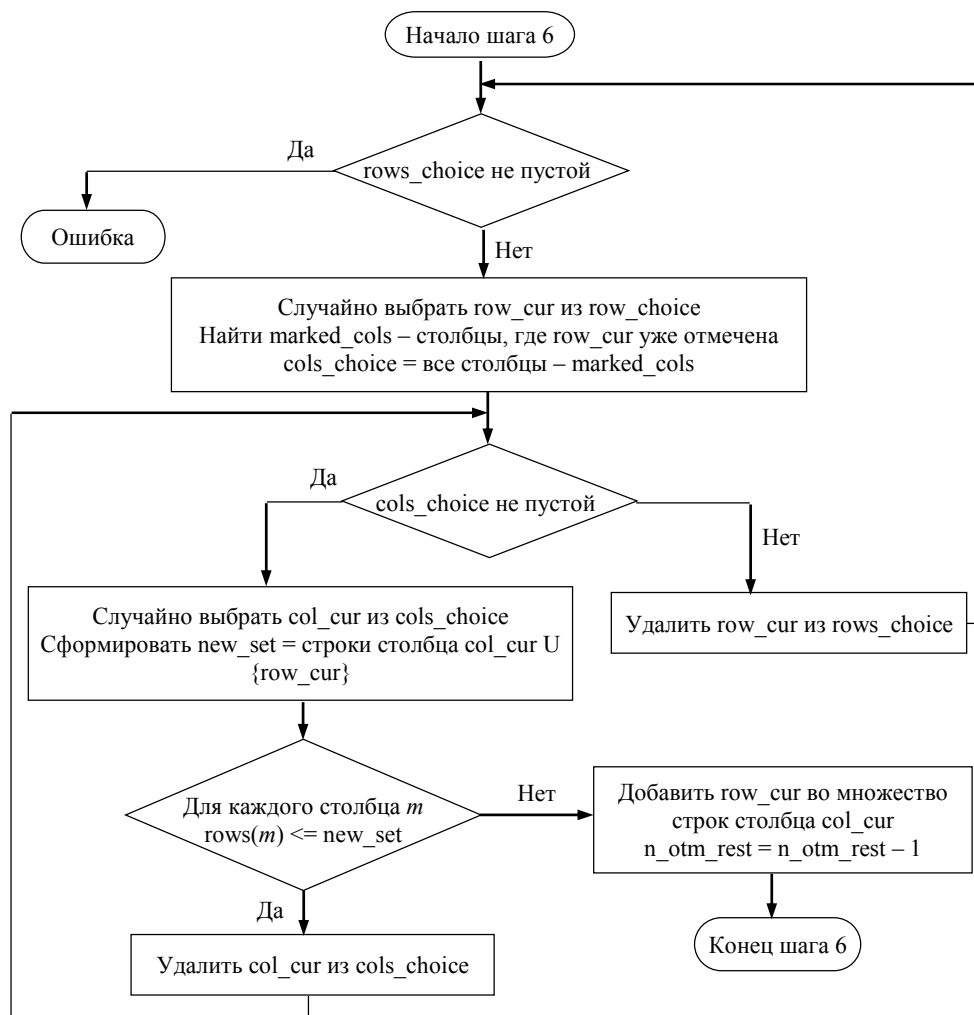


Рис. 6. Блок-схема алгоритма попытки поставить отметку
Fig. 6. Flowchart of the algorithm for attempting to place a mark

Перспективы развития. Дальнейшие исследования могут быть направлены:

- на расширение генератора для поддержки взвешенных и дробных вариантов ЗПМ;
- интеграцию с платформами для автоматизированного тестирования алгоритмов (например, через API (application programming interface));
- оптимизацию производительности для генерации сверхбольших наборов данных (актуальных для квантовых вычислений);
- разработку критериев качества сгенерированных данных (например, оценка «трудности» экземпляров ЗПМ).

Рассмотрим демонстрационный пример работы предлагаемого алгоритма. Будем генерировать ТП с параметрами: $n_rows = 10$; $n_cols = 15$; $n_otm = 60$. После второго шага все столбцы имеют по две отметки (табл. 1). В результате получаем циклическую ТП, имеющую 60 отметок (табл. 2).

Заключение. Автоматизированная генерация тестовых данных для ЗПМ – необходимый инструмент для:

- разработки новых алгоритмов;
- объективной оценки существующих методов;
- масштабирования исследований в области дискретной оптимизации.

Предложенный алгоритм закладывает основу для систематического тестирования и совершенствования подходов к решению ЗПМ. Компанией ЗАО «Морские компьютерные системы» создан стенд для тестирования и сравнения алгоритмов решения ЗПМ.

Цель и область применения. ЗАО «Морские компьютерные системы» ведет исследования в области разработки и практического применения алгоритмов решения задачи покрытия множества:

- оптимизация внутренних производственных процессов предприятия;
- автоматизация бизнес-процессов сторонних заказчиков в рамках разработки заказного ПО.

Стенд для тестирования и сравнения алгоритмов позволяет:

- запускать алгоритмы решения ЗПМ;
- генерировать тестовые наборы данных (ТП);

Табл. 1. ТП после второго шага
 Tab. 1. Coverage table after the second step

Строки	Столбцы														
	c ₀	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	c ₇	c ₈	c ₉	c ₁₀	c ₁₁	c ₁₂	c ₁₃	c ₁₄
r ₀					+					+		+			
r ₁	+					+								+	
r ₂					+				+						
r ₃				+						+	+				
r ₄		+						+				+			
r ₅			+					+						+	
r ₆		+				+							+		
r ₇			+						+						+
r ₈				+			+						+		
r ₉	+						+				+				+

Табл. 2. Окончательный вид ТП
 Tab. 2. The final view of the coverage table

Строки	Столбцы														
	c ₀	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	c ₇	c ₈	c ₉	c ₁₀	c ₁₁	c ₁₂	c ₁₃	c ₁₄
r ₀		+		+	+					+		+			+
r ₁	+	+		+		+		+		+		+		+	
r ₂					+				+						
r ₃				+						+	+				
r ₄		+	+		+	+	+	+	+			+			+
r ₅			+					+					+	+	
r ₆		+				+	+	+			+		+		+
r ₇			+			+			+	+		+	+	+	+
r ₈	+		+	+			+		+				+	+	
r ₉	+				+		+	+			+				+

– тестировать алгоритмы на сгенерированных ТП;

– сохранять результаты в базе данных (БД) на основе СУБД (система управления БД) PostgreSQL.

Архитектурное решение. Стенд построен на основе программного обеспечения собственной разработки ЗАО «Морские компьютерные системы», включающего три ключевых компонента:

DWA (Database Web Application) – веб-приложение, обеспечивающее:

- пользовательский веб-интерфейс к БД;
- визуализацию данных (таблицы, представления, графики);
- управление доступом и модификацию данных.

DBMS (DataBase Micro Service) – микросервис, реализующий:

- бизнес-логику взаимодействия с БД;
- обработку запросов к алгоритмам решения ЗПМ.

СУБД PostgreSQL – реляционная СУБД, обеспечивающая:

- хранение тестовых наборов ТП;

– сохранение результатов тестирования алгоритмов;

– транзакционность и целостность данных.

Функциональные возможности веб-приложения DWA. DWA предоставляет расширенный интерфейс для работы с БД стенда. Его основные возможности перечислены далее.

• Визуализация и фильтрация данных:

- отображение таблиц и представлений БД;
- фильтрация данных по полям и условиям (в том числе по диапазонам значений, текстовым шаблонам, логическим выражениям);
- сортировка и пагинация результатов.

• Управление доступом пользователей. Для каждого пользователя настраиваются права на:

- видимость конкретных таблиц/представлений;
- редактирование данных в таблицах;
- добавление новых строк;
- удаление строк (логическое и физическое);
- выполнение специализированных операций (например, запуск алгоритмов).

• Визуализация данных. DWA позволяет строить графики на основе отфильтрованных данных

таблиц или представлений. Поддерживаются графики следующих типов:

Линейный график (3 варианта):

– числовые показатели – отображает среднее, минимум и максимум по выбранному числовому значению; каждое значение рисуется отдельным цветом;

– категориальные данные – строит отдельную линию для каждого уникального значения категориального поля;

– общее количество – единая линия, показывающая суммарное количество записей в каждый из интервалов времени.

Тренды на базе линейного графика – обработка данных с помощью алгоритма экспоненциального скользящего среднего (exponential moving average – ЕМА). Параметры:

– окно сглаживания (количество точек для усреднения);

– тип линейного графика (из перечисленных выше);

– визуализация исходного ряда и тренда на одном поле.

Столбчатая диаграмма – позволяет:

– группировать данные по нескольким полям (в том числе по временным интервалам);

– агрегировать значения (сумма, среднее, количество, максимум/минимум);

– упорядочивать столбцы по временной шкале;

– настраивать цвета и подписи для каждой группы.

Сценарии использования стенда:

• **Генерация тестовых наборов ТП**

– настройка параметров генерации (размерность ТП, объем тестового набора);

– сохранение сгенерированных ТП в БД;

– просмотр и анализ ТП через веб-интерфейс.

• **Запуск и тестирование алгоритмов**

– выбор алгоритма из реестра доступных реализаций;

– задание параметров алгоритма;

– автоматический запуск на выбранном наборе ТП;

– фиксация результатов (качество, время).

• **Анализ результатов**

– сравнение алгоритмов по качеству и скорости;

– построение графиков показателей;

– формирование отчетов.

• **Управление данными и доступом**

– администрирование пользователей и ролей;

– аудит изменений в БД;

– экспорт/импорт данных (CSV, JSON, Excel).

Преимущества решения:

– унификация интерфейса – единый веб-доступ к данным и функционалу стенда;

– гибкость настройки – возможность адаптировать права, фильтры и визуализацию под задачи;

– масштабируемость – архитектура на микросервисах позволяет наращивать нагрузку и функционал;

– прозрачность – все результаты сохраняются в БД и доступны для повторного анализа;

– интеграция – открытые API DBMS позволяют подключать сторонние инструменты и сервисы.

Перспективы развития:

– внедрение механизмов машинного обучения для предсказания эффективности алгоритмов на новых ТП;

– расширение библиотеки алгоритмов (жадные, генетические, точные методы);

– поддержка распределенных вычислений для больших ТП;

– добавление интерактивных дашбордов для мониторинга экспериментов.

Система проведения экспериментов (СПЭ).

Назовем рассмотренный ранее стенд системой проведения экспериментов (СПЭ). Чтобы алгоритм решения ЗПМ попал в СПЭ, он должен быть оформлен по шаблону, предоставляемому разработчику программного модуля. Это требование необходимо для исследования алгоритма и настройки его параметров, а также для сравнения алгоритма с другими, уже включенными в библиотеку. Алгоритмы сравниваются по времени получения решения и качеству решения (числу строк в приближенно-минимальном решении).

Алгоритм включается в библиотеку модератором ресурса (scr@mcs.ru), после того как разработчик продемонстрировал его работоспособность на тестовых наборах. Тестовые наборы данных (НД) имеют небольшие значения свойств. НД для исследования алгоритма предоставляются после того, как разработчик согласует с модератором ресурса план проведения экспериментов. После проведения экспериментов разработчик получит результаты исследования.

После окончательной настройки параметров алгоритма разработчик может сделать запрос на получение рейтинга алгоритма. Рейтинговые НД созданы модератором ресурса и включены в СПЭ (в настоящее время список рейтинговых НД состоит из 50 наборов данных).

Ранжирование алгоритма привязано к регламентному набору данных. Для каждого регламентного набора подсчитывается три рейтинга:

- 1) с акцентом на качество решения;
- 2) с акцентом на время получения решения;
- 3) при равноценном влиянии на рейтинг качества и времени получения.

Пул кандидатов на получение рейтинга ограничен (100 алгоритмов). Если алгоритм имеет очень низкий рейтинг, то он исключается из пула, даже если пул еще не заполнен. Любое изменение в пуле влечет пересчет рейтинга. В СПЭ реализованы комплексные рейтинги: для выбранного подмножества регламентных наборов данных. Комплексный рейтинг равен среднему рейтингу на соответствующих наборах данных.

Уточним, что в СПЭ под алгоритмом понимается конкретная реализация алгоритма. Свойства алгоритма в системе следующие:

- 1) разработчик программного модуля;
- 2) метод (эвристика);
- 3) средства реализации;
- 4) параметры алгоритма.

По запросу разработчика программного модуля модератор СПЭ может сгенерировать НД по заданным параметрам ТП (число строк, столбцов, отметок), но нужно иметь в виду, что параметры должны обеспечивать возможность генерации циклических ТП.

Список литературы

1. Васькин П. И., Лисс А. А. Алгоритмы решения задачи покрытия множества // Изв. СПбГЭТУ «ЛЭТИ». 2026. Т. 19, № 3. С. 21–42. doi: 10.32603/2071-8985-2026-19-3-24-42.
2. Beasley J. E. OR-Library collection of test data sets for a variety of OR problems. 2005. URL: <https://people.brunel.ac.uk/~mastjib/jeb/info.html> (дата обращения: 28.01.2026).

3. Beasley J. E. An algorithm for set covering problem // Europ. J. of Operational Research. 1987. Vol. 31, no. 1. P. 85–93.
4. Beasley J. E. A lagrangian heuristic for set-covering problems // Naval Research Logistics (NRL). 1990. Vol. 37, no. 1. P. 151–164.

Информация об авторах

Афонский Дмитрий Алексеевич – ведущий инженер-программист отдела разработки информационных систем ЗАО «Морские компьютерные системы», ул. Бабушкина, 80, Санкт-Петербург, 192174, Россия.
E-mail: ada@mcs.ru

Васькин Павел Ильич – канд. техн. наук, доцент, директор по направлению ИС и БД, ЗАО «Морские компьютерные системы», ул. Бабушкина, 80, Санкт-Петербург, 192174, Россия.
E-mail: vpi@mcs.ru
<https://orcid.org/0000-0003-1819-0167>

Лисс Анна Александровна – канд. техн. наук, доцент, зав. кафедрой математического обеспечения и применения ЭВМ (МО ЭВМ) СПбГЭТУ «ЛЭТИ».
E-mail: aaliss@etu.ru

Павлов Сергей Анатольевич – ведущий инженер-программист отдела разработки прикладного ПО, ЗАО «Морские компьютерные системы», ул. Бабушкина, 80, Санкт-Петербург, 192174, Россия.
E-mail: pavlov@mcs.ru

Вклад авторов:

Афонский Д. А. – разработка веб-приложения и микросервиса, на основе которых была создана система проведения экспериментов.

Васькин П. И. – разработка и реализация алгоритма псевдослучайной генерации циклических таблиц покрытия; формализация рейтинговой системы для сравнения алгоритмов.

Лисс А. А. – концептуальная постановка задачи исследования; подготовка вводной части.

Павлов С. А. – разработка и реализация системы проведения экспериментов; реализация рейтинговой системы.

References

1. Vas'kin P. I., Liss A. A. Algoritmy reshenija zadachi pokrytija mnozhestva // Izv. SPbGJeTU «LJeTI». 2026. T. 19, № 3. S. 21–42. doi: 10.32603/2071-8985-2026-19-3-24-42. (In Russ.)

2. Beasley J. E. OR-Library collection of test data sets for a variety of OR problems. 2005. URL: <https://people.brunel.ac.uk/~mastjib/jeb/info.html> (data obrashhenija: 28.01.2026).

3. Beasley J. E. An algorithm for set covering problem // Europ. J. of Operational Research. 1987. Vol. 31, no. 1. P. 85–93.

4. Beasley J. E. A lagrangian heuristic for set-covering problems // Naval Research Logistics (NRL). 1990. Vol. 37, no. 1. P. 151–164.

Information about the authors

Dmitry A. Afonskiy – Dr Sci. (Eng.), leading software engineer at the Information Systems Development Department, Marine Computer Systems CJSC, 80 Babushkina St., Saint Petersburg, 192174, Russia.
E-mail: ada@mcs.ru

Pavel I. Vaskin – Cand. Sci. (Eng.), Associate Professor, Director of Information Technology and Database, Marine Computer Systems CJSC, 80 Babushkina St., Saint Petersburg, 192174, Russia.
E-mail: vpi@mcs.ru
<https://orcid.org/0000-0003-1819-0167>

Anna A. Liss – Cand. Sci. (Eng.), Associate Professor, Head of the Computer Software and Application Department, Saint Petersburg Electrotechnical University.
E-mail: aaliss@etu.ru

Sergey A. Pavlov – leading software engineer of the Department of Application Software Development, Marine Computer Systems CJSC, 80 Babushkina St., Saint Petersburg, 192174, Russia.
E-mail: pavlov@mcs.ru

Author contribution statement:

Afonskiy D. A. – development of a web application and microservice, on the basis of which a system for conducting experiments was created.

Vaskin P. I. – development and implementation of an algorithm for pseudorandom generation of cyclic coverage tables; formalization of a rating system for comparing algorithms.

Liss A. A. – conceptual formulation of the research task; preparation of the introductory part.

Pavlov S. A. – development and implementation of a system for conducting experiments; implementation of a rating system.

Статья поступила в редакцию 09.02.2026; принята к публикации после рецензирования 16.04.2026; опубликована онлайн 29.06.2026.

Submitted 09.02.2026; accepted 16.04.2026; published online 29.06.2026.
