

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем  
(код и наименование направления подготовки / специальности)

Web-дизайн и мультимедиа  
(направленность (профиль) / специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения для решения задач комбинаторной  
оптимизации»

Обучающийся

Д.С. Кислый  
(Инициалы Фамилия)

(личная подпись)

Руководитель

к.ф.-м.н. О.В. Лелонд

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.ф.н., доцент М.М. Бажутина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

## Аннотация

Тема выпускной квалификационной работы – «Разработка программного обеспечения для решения задач комбинаторной оптимизации».

Как показывает практика, многие компании, особенно занятые в сфере перевозок грузов, нуждаются в программах для формирования оптимального плана для их доставки.

В этой связи разработка программного обеспечения для решения задач комбинаторной оптимизации представляет актуальность и научно-практический интерес.

Объектом исследования бакалаврской работы являются задачи комбинаторной оптимизации.

Предметом исследования бакалаврской работы является программное обеспечение для решения задач комбинаторной оптимизации.

Цель выпускной квалификационной работы – разработка программного обеспечения для решения задач комбинаторной оптимизации.

Методы исследования – методы и алгоритмы решения задач комбинаторной оптимизации, методы и технологии разработки программного обеспечения.

Практическая значимость бакалаврской работы заключается в разработке программы, обеспечивающей решение частной задачи комбинаторной оптимизации.

Результаты бакалаврской работы представляют научно-практический интерес и могут быть рекомендованы для разработки программного обеспечения для решения задач комбинаторной оптимизации.

Бакалаврская работа состоит из 43 страниц текста и содержит 17 рисунков, 4 таблицы и 22 источника.

## **Abstract**

The title of the graduation work is: «Development of software for solving problems of combinatorial optimization».

As practice shows, many companies working in the field of cargo transportation need programs to create an optimal plan for their delivery.

In this regard, the development of software for solving problems of combinatorial optimization has scientific and practical relevance.

The objects of study of the graduation work is the problem of combinatorial optimization.

The subject of study of the graduation work is the software for solving problems of combinatorial optimization.

The aim of the work is to develop of software for solving problems of combinatorial optimization.

The practical significance of the graduation work lies in the development of a program that allows to solve a particular problem of combinatorial optimization.

The results of the graduation work are of scientific and practical interest and can be recommended for the development of software for solving combinatorial optimization problems.

The graduation work consists of 43 pages of text, 17 figures, 4 tables and list of 22 references including foreign sources.

## Оглавление

Введение.....	5
Глава 1 Постановка задачи на разработку программного обеспечения для решения задач комбинаторной оптимизации.....	6
Глава 2 Анализ и выбор технологий разработки программного обеспечения для решения задачи оптимизации маршрутов транспортных средств.....	16
2.1 Анализ программного обеспечения для решения задачи оптимизации маршрута транспортных средств методом Кларка-Райта .....	17
2.2 Анализ и выбор технологий для разработки программы .....	22
Глава 3 Реализация и тестирование онлайн-калькулятора для решения задачи оптимизации маршрутов транспортных средств .....	29
Заключение .....	39
Список используемой литературы .....	41

## Введение

Задачи комбинаторной оптимизации довольно широко применяются на практике.

Достаточно напомнить о таких классических задачах, как задача коммивояжера, задача рюкзака и др.

Следует отметить, что метод комбинаторной оптимизации требует серьёзных вычислительных ресурсов, которые до недавнего времени не были широко доступными [1].

Основная проблема состоит в том, что в комбинаторных задачах нет линейных зависимостей и решить их можно только методом полного перебора или с помощью эвристических алгоритмов.

Это обусловило отсутствие универсального программного обеспечения, предназначенного для решения конкретных задач комбинаторной оптимизации.

Вместе с тем многие компании, особенно занятые в сфере перевозок грузов, нуждаются в таких программах для формирования оптимального плана для их доставки.

В этой связи разработка программного обеспечения для решения задач комбинаторной оптимизации представляет актуальность и научно-практический интерес.

Объектом исследования бакалаврской работы являются задачи комбинаторной оптимизации.

Предметом исследования бакалаврской работы является программное обеспечение для решения задач комбинаторной оптимизации.

Цель выпускной квалификационной работы – разработка программного обеспечения для решения задач комбинаторной оптимизации.

Для достижения данной цели необходимо выполнить следующие задачи:

- выполнить постановку задачи на разработку программного обеспечения для решения задачи комбинаторной оптимизации;

- проанализировать и выбрать технологии для разработки программного обеспечения для решения задач комбинаторной оптимизации;
- выполнить реализацию программы и протестировать ее работоспособность.

Методы исследования – методы и алгоритмы решения задач комбинаторной оптимизации, методы и технологии разработки программного обеспечения.

Практическая значимость бакалаврской работы заключается в разработке программы, обеспечивающей решение частной задачи комбинаторной оптимизации.

Данная работа состоит из введения, трех глав, заключения, списка используемой литературы и приложений.

Первая глава посвящена постановке задачи на разработку программного обеспечения для решения задач комбинаторной оптимизации.

Вторая глава посвящена анализу и выбору технологии для разработки программного обеспечения для решения задачи комбинаторной оптимизации.

В третьей главе представлены реализация программы и результаты ее тестирования.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Бакалаврская работа состоит из 43 страниц текста и содержит 17 рисунков, 4 таблицы и 22 источника.

## **Глава 1 Постановка задачи на разработку программного обеспечения для решения задач комбинаторной оптимизации**

Комбинаторная оптимизация – это подмножество математической оптимизации, связанное с исследованием операций, теорией алгоритмов и теорией сложности вычислений. Она включает в себя алгоритмические методы для решения задач дискретной оптимизации в пределах конечного набора возможностей [22].

Одним из основных классов проблем комбинаторной оптимизации является проблема маршрутизации транспорта (Vehicle Routing Problems, VRP).

Растущие требования клиентов к качеству доставки грузов и сильная конкуренция на рынке заставили перевозчиков срочно оптимизировать свои расходы.

Проблема маршрутизации транспорта (ПМТ) является важной проблемой в распределительной сети и играет ключевую роль в снижении затрат и улучшении обслуживания транспортных компаний.

Проблема заключается в том, чтобы посетить группу клиентов, используя парк транспортных средств, соблюдая ограничения на транспортные средства, клиентов, водителей и т. д. Основная цель состоит в том, чтобы минимизировать затраты на маршрутизацию каждого транспортного средства.

Проблема планирования транспортных средств может быть сформулирована нижеследующим образом.

Имеется конечное множество клиентов, для каждого из которых известны местоположение и потребность в некотором товаре, который должен быть доставлен с одного склада транспортными средствами доставки при соблюдении следующих условий и ограничений [12]:

- требования всех клиентов должны быть удовлетворены;
- каждый клиент обслуживается только одним транспортным

средством;

- вместимость транспортных средств не может быть нарушена (для каждого маршрута общая потребность не должна превышать вместимость).

Последнее условие обусловило появление такого класса задач комбинаторной оптимизации, как Capacitated Vehicle Routing Problem (CVRP) или задача маршрутизации транспортных средств с вместимостью (МТСВ).

Задача МТСВ – это ПМТ, в которой транспортные средства с ограниченной грузоподъемностью должны забирать грузы из разных мест или доставлять их в разные места. Грузы имеют вес или объем, а транспортные средства имеют максимальную грузоподъемность.

Разработка программного обеспечения для решения данной задачи комбинаторной оптимизации представляет научно-практический интерес.

Рассмотрим математическую постановку задачи и выберем алгоритм для решения.

Решение задачи МТСВ заключается в удовлетворении спрос множества клиентов, используя парк транспортных средств с минимальными затратами.

Существуют различные варианты формулировки проблемы МТСВ.

Целевая функция и большинство ограничений одинаковы для всех вариантов [13].

Проблема МТСВ может быть формализована как модель линейного целочисленного программирования. Общая протяженность маршрута, на котором выполняются все требования клиентов, должна быть сведена к минимуму.

Двоичная переменная  $x_{ijk}$  имеет значение 1, если дуга между  $i$ -м и  $j$ -м узлами находится на оптимальном маршруте и обслуживается  $k$ -ым транспортным средством (1):

$$x_{ijk} \in \{0,1\} \quad \forall k \in \{1, \dots, p\}, i, j \in \{1, \dots, n\} \quad (1)$$



При этом нет пути от узла к самому себе (2):

$$x_{iik} \in \{0,1\} \quad \forall k \in \{1, \dots, p\}, i \in \{1, \dots, n\} \quad (2)$$

Параметр  $d_{ij}$  описывает расстояние между между  $i$ -м и  $j$ -м узлами.

Пусть  $n$  – количество узлов (депо = 1) и  $p$  – количество транспортных средств.

Тогда целевую функцию можно описать следующим образом (3):

$$\min \sum_{k=1}^p \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ijk} \quad (3)$$

В каждый узел нужно въезжать и выезжать из него один раз (кроме депо) и только на одном и том же транспортном средстве. Каждое транспортное средство должно выехать из депо и въезжать из него один раз.

Пусть  $q_i$  описывает требования каждого клиента,  $Q$  является вместимостью (грузоподъемностью) транспортных средств.

Сумма требований всех клиентов, которые будет обслуживать  $k$ -е транспортное средство, не должна превышать вместимость данного средства.

Все эти ограничения можно сформулировать следующим образом.

Ограничение 1. Транспортное средство покидает узел, в который заезжает (4):

$$\sum_{i=1}^n x_{ijk} = \sum_{i=1}^n x_{jik} \quad \forall k \in \{1, \dots, p\}, k \in \{1, \dots, p\} \quad (4)$$

Ограничение 2. В каждый узел транспортное средство заезжает один раз (5):

$$\sum_{k=1}^p \sum_{i=1}^n x_{ijk} = 1 \quad \forall j \in \{2, \dots, n\} \quad (5)$$

Ограничение 3. Каждое транспортное средство выезжает из депо (6):

$$\sum_{j=2}^n x_{1jk} = 1 \quad \forall k \in \{1, \dots, p\} \quad (6)$$

Ограничение 4. Ограничение на вместимость (7):

$$\sum_{i=1}^n \sum_{j=2}^n q_j x_{ijk} \leq 0 \quad \forall j \in \{1, \dots, p\} \quad (7)$$

Рассмотрим упрощенную математическую постановку задачи МТСВ по удовлетворению спроса множества клиентов, используя парк транспортных средств с минимальными расходами.

Это задача является задачей оптимизации маршрутов.

«Пусть  $V = \{0, \dots, n\}$  – набор узлов полного ненаправленного графа  $G = (V, E)$ .

Узел 0 представляет склад (депо), где имеются транспортные средства, каждое с грузоподъемностью  $D$ .

Остальные узлы представляют  $n$  клиентов, каждый из которых имеет неотрицательный спрос  $d_i$ ,  $i = 1, \dots, n$ .

Кроме того, для каждой дуги  $(i, j) \in E$  определен параметр  $c_{ij}$ , обозначающий стоимость переезда по этой дуге.

Задача сводится к определению ряда маршрутов с минимальной общей стоимостью таким образом, что каждый клиент посещается единственный раз единственным транспортным средством, которое начинает и заканчивает свой маршрут на складе» [2].

Целевую функцию можно описать следующим образом (8) [11]:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (8)$$

при условиях (9)-(14):

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \setminus \{0\} \quad (9)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in V \setminus \{0\} \quad (10)$$

$$\sum_{i \in V} x_{i0} = K \quad (11)$$

$$\sum_{j \in V} x_{0j} = K \quad (12)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \geq r(S), \quad \forall S \in V \setminus \{0\}, S \neq \emptyset \quad (13)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in V \quad (14)$$

В этой формулировке:

- $x_{ij}$  – двоичная переменная, которая равна 1, если дуга между  $i$ -м и  $j$ -м узлами находится на оптимальном маршруте 0 – в противном случае;
- $K$  – количество доступных транспортных средств;
- $r(S)$  – соответствует минимальному количеству транспортных средств, необходимых для обслуживания заданного количества

узлов.

Для решения данной задачи МТСВ используются алгоритмы, основанные на эвристических методах (таблица 1).

Таблица 1 – Эвристические алгоритмы для решения задачи маршрутизации транспорта

Метод	Алгоритм
«Конструктивные методы	алгоритм Кларка-Райта, алгоритм Мола-Джеймсона, алгоритм Кристофидеса-Мингоззи-Тосса
Двухфазные методы	алгоритм заметания, алгоритм Фишера-Джекумера, алгоритм Брамела-Симчи-Леви, алгоритм лепестков
Улучшающие методы	оптимизация отдельных маршрутов, улучшение нескольких маршрутов» [3]

Анализ известных программных аналогов показал, что наиболее популярной эвристикой для решения задачи оптимизации маршрутов транспортных средств, является алгоритм Кларка-Райта и его модификации [17].

Метод Кларка-Райта, также известный как метод экономии, основан на расчете экономии при объединении клиентов в один и тот же маршрут.

Алгоритм Кларка-Райта вычисляет все сбережения  $S_{ij}$  между клиентами  $i$  и  $j$ .

Предположим, что  $c_{i0}$  – это стоимость проезда из депо до клиента  $i$ , а  $c_{ij}$  – стоимость проезда от клиента  $i$  до клиента  $j$ .

Ниже приведены шаги по выполнению методов Кларка-Райта [20].

Шаг 1. Построить полуматрицу кратчайшего расстояния, состоящую из кратчайшего расстояния или наименьшего времени между каждой парой узлов, включая начальный узел.

Шаг 2. Разработать начальное распределение одного кругового пути от начального узла до каждого пункта назначения.

Шаг 3. Рассчитать чистую экономию для каждой пары узлов (исключая

начальный узел) и построить полуматрицу чистой экономии. Чистая экономия – это экономия, достигнутая за счет сопряжения узлов, по сравнению со стоимостью поездки туда и обратно к каждому парному узлу из депо или узла 1.

Шаг 4. Ввести специальный показатель  $I$  в соответствующие ячейки полуматрицы чистых сбережений. Этот индикатор покажет, связаны ли два рассматриваемых узла напрямую. Связь может быть либо от узла 1 к любому другому узлу  $j$ , либо между любой парой узлов  $i$  и  $j$ , когда ( $i \neq 1$  и  $j \neq 1$ ). Этот индикатор отключения может иметь одно из трех значений:

Шаг 4.1. Если транспортное средство перемещается из пункта отправления (узел 1) в узел  $j$  (кроме узла 1), а затем возвращается в пункт отправления, то  $I=2$ . То есть, поездка туда и обратно будет иметь  $I=2$ . В матрицу можно записать значение показателя  $I_{1j} = 2$  (где  $j \neq 1$ ). Это значение появится только в первой строке полуматрицы чистых сбережений.

Шаг 4.2. Если транспортное средство движется в одну сторону непосредственно между двумя узлами, то  $I_{ij} = 1$ .

Шаг 4.3. Значение индикатора поездки  $I_{ij} = 0$ , если транспортное средство не движется напрямую между двумя конкретными узлами, то между парой узлов нет поездки.

Шаг 5. Выбрать ячейку ( $i, j$ ) в полуматрице чистых сбережений, имеющую максимальную чистую экономию, и соединить  $i$  и  $j$ . Но прежде чем связать пару узлов, необходимо выполнить следующее условие:

Шаг 5.1.  $I_{1i}$  и  $I_{1j}$  должны быть больше 0.

Шаг 5.2. Узлы  $i$  и  $j$  уже не находятся на одном маршруте или кольце.

Шаг 5.3. Нет нарушения ограничения при связывании  $i$  и  $j$  (могут быть некоторые ограничения, такие как односторонний маршрут, который разрешен только между двумя улицами).

Если ячейка удовлетворяет всем вышеперечисленным условиям, то  $I_{ij} = 1$ , в противном случае:  $I_{ij} = 0$ , выбрать ячейку со следующей по величине чистой экономией и проверить условия, указанные в 5.1, 5.2 и 5.3.

Шаг 6. Если все узлы связаны одним маршрутом и ни одна другая ячейка не соответствует условиям шага 5, остановить алгоритм. В противном случае перейти к шагу 5.

Блок-схема алгоритма Кларка-Райта показана на рисунке 1 [5].



Рисунок 1 – Блок-схема алгоритма Кларка-Райта

Погрешность решения не превосходит в среднем 5–10%.

Достоинствами метода являются его простота, надежность и гибкость, что позволяет учитывать целый ряд дополнительных факторов, влияющих на конечное решение задачи.

С учетом вышеизложенного выбираем алгоритм Кларка-Райта в качестве алгоритма для решения задачи МТСВ и последующей программной реализации.

Сформулируем требования к разрабатываемому ПО.

Для разработки требований к программе используем методологию FURPS+.

FURPS+ - это метод проверки приоритетных требований к ПО после понимания потребностей клиента [18].

В методологии FURPS+ требования к ПО разделены на функциональные и нефункциональные требования. Функциональные требования объясняют, что должно быть сделано путем определения необходимой задачи, действия или деятельности, которые должны быть выполнены.

Нефункциональные требования к качеству ПО – это удобство использования, надежность, производительность и возможность поддержки.

В таблице 2 представлены основные требования ПО решения задачи МТСВ.

Таблица 2– Требования к ПО для решения задачи МТСВ

№	Требование	Статус	Полезность	Риск	Стабильность
Functionality — Функциональные требования					
1.	Онлайн-калькулятор для решения задачи МТСВ с помощью алгоритма Кларка-Райта	Одобренное	Критическая	Средний	Низкая
Usability— Требования к удобству использования					
2.	Дружественный интуитивный интерфейс	Одобренное	Критическая	Средний	Низкая
3.	Отсутствие функциональной избыточности	Одобренное	Критическая	Средний	Низкая
Reliability— Требования к надежности					
4.	Допустимая частота/периодичность сбоев: 1 раз в 300 часов	Одобренное	Важная	Средний	Средняя
5.	Среднее время сбоев: 1 раб. день	Одобренное	Важная	Средний	Средняя
6.	Возможность восстановления системы после сбоев: 1 раб. день	Одобренное	Важная	Средний	Средняя
7.	Режим работы: 7/24/365	Одобренное	Важная	Средний	Средняя

Продолжение таблицы 2

№	Требование	Статус	Полезность	Риск	Стабильность
Performance — Требования к производительности					
8.	Допустимое количество одновременно работающих пользователей: 10	Предложенное	Важная	Средний	Средняя
9.	Время реакции на возникновение аварийной ситуации: 10 с	Предложенное	Важная	Средний	Средняя
Supportability — Требования к поддержке					
10.	Время устранения критических проблем: в течение рабочего дня	Предложенное	Важная	Средний	Средняя
Проектные ограничения					
11.	Применение современных веб-технологий проектирования ПО	Предложенное	Критическая	Средний	Низкая

Представленный перечень требований является основой для разработки ПО для решения задачи МТСВ.

#### Выводы по главе 1

Первая глава посвящена постановке задачи на разработку ПО для решения задач комбинаторной оптимизации.

Результаты проделанной работы позволили сделать следующие выводы:

- задача МТСВ – это проблема маршрутизации транспорта, в которой транспортные средства с ограниченной грузоподъемностью должны забирать грузы из разных мест или доставлять их в разные места.
- анализ показал, что наиболее популярным алгоритмом для решения задачи МТСВ, является алгоритм Кларка-Райта и его модификации.

Представленный перечень требований является основой для разработки ПО для решения задачи оптимизации маршрутов транспортных средств.

## Глава 2 Анализ и выбор технологий разработки программного



## обеспечения для решения задачи оптимизации маршрутов транспортных средств

### 2.1 Анализ программного обеспечения для решения задачи оптимизации маршрута транспортных средств методом Кларка-Райта

#### 2.1.1 Программа VPR Solver

VPR Solver реализует рандомизированную версию алгоритма экономии Кларка-Райта для задач маршрутизации транспортных средств [21].

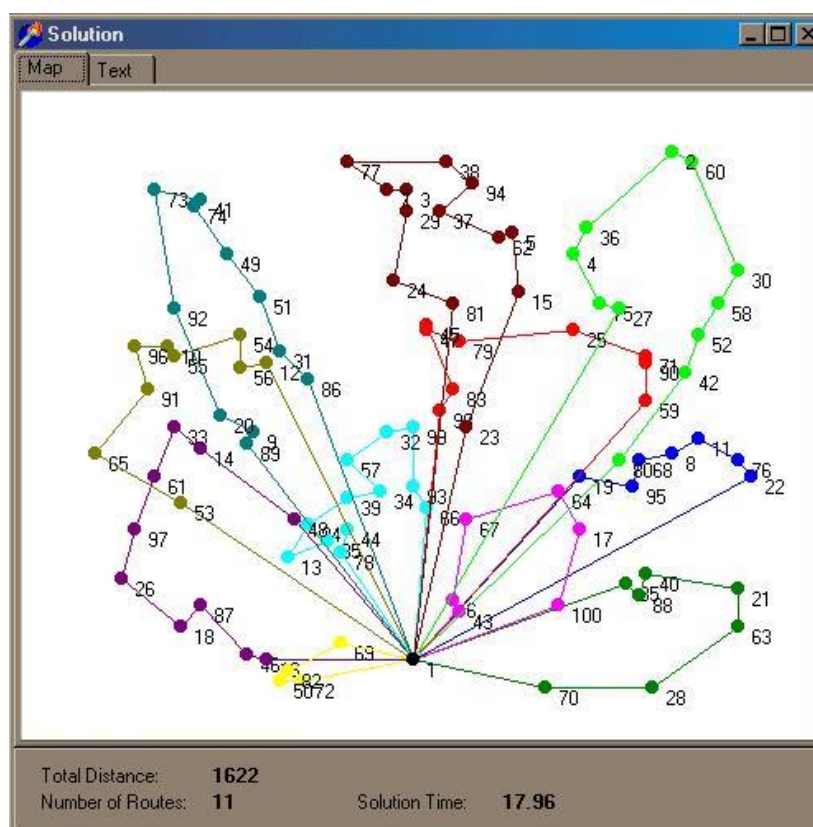


Рисунок 2 – Скриншот программы VPR Solver

Программа получает данные из текстового файла, в котором указывается местоположение каждого клиента (широта и долгота) и спрос.

Расстояния могут быть введены явно или рассчитаны автоматически с

использованием евклидовой метрики или метрики большого круга.

Программа строит маршруты транспортных средств, которые посещают каждый город ровно один раз и подчиняются заданным пользователем ограничениям по объему транспортных средств и расстоянию.

Результаты отображаются в графическом (карта) виде (рисунок 2) и в форме текста.

Программа разработана в среде разработки Delphi для ОС Windows и распространяется бесплатно.

### **2.1.2 Программа «1С: Транспортная логистика, экспедирование и управление автотранспортом КОРП»**

В программе «1С: Транспортная логистика, экспедирование и управление автотранспортом КОРП» реализована автоматическая маршрутизация заказов [6].

Алгоритм автоматической маршрутизации следующий. Настройка маршрутизации осуществляется в АРМ логиста через меню «Открыть настройки - вкладка Маршрутизация». Здесь можно использовать один из доступных способов маршрутизации:

- алгоритм Очередей;
- алгоритм Кларка-Райта.

Для проведения расчетов с помощью алгоритм Кларка-Райта устанавливаем необходимые флажки в настройках.

Устанавливаются по умолчанию понижающий коэффициент загрузки паллеты по объему - 90%, коэффициент загрузки кузова по объему и по весу - 95%. Это можно использовать в случае перевозки сыпучих грузов или если просто большой объем может не войти в машину, или же размер паллет не позволяет поместить их больше в машину. Заполняются также параметры маршрутного листа и адрес выезда и возвращения транспортного средства (рисунок 3).

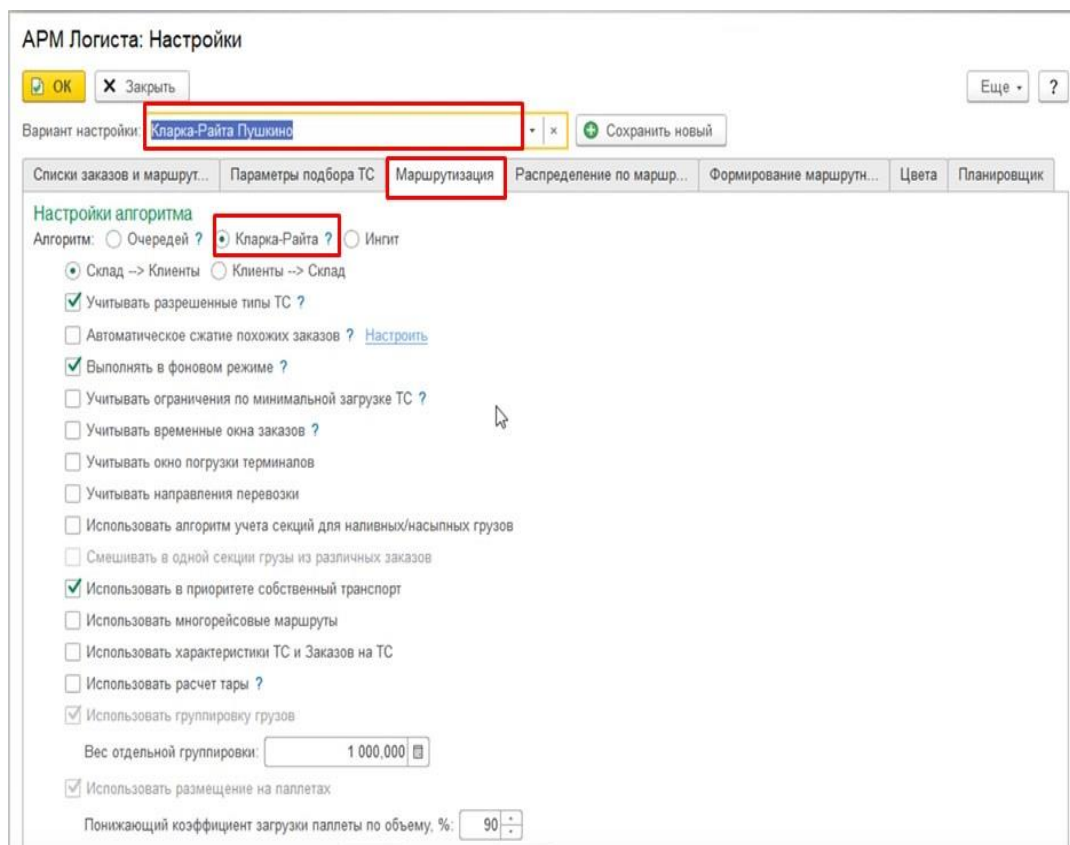


Рисунок 3 – Окно настройки алгоритма Кларка-Райта

Затем в АРМ логиста выполняется группировка по адресу отправления и выберем все нужные заказы на перевозку [4].

После этого нажимаем кнопку «Рассчитать – Маршрутизация».

В результате формируются маршрутные листы.

При автоматической маршрутизации автомобиль не будет перегружен по весу, объему и количеству упаковок.

Программа разработана на технологической платформе «1С: Предприятие 8».

### 2.1.3 Программа «Простые маршруты»

«Программа «Простые маршруты» предназначена для автоматизированной разработки оптимальных маршрутов с учетом временных окон доставки на основе полученных данных из учетной системы пользователя.

В программе используется онлайн-сервис Google Maps.

Вариант 1. Формирование количества маршрутов для заданного количества /транспорта.

Вариант 2. Определение оптимального количества маршрутов/транспорта для заданной максимальной грузоподъемности автомашины» [7].

Для автоматического распределения маршрутов используются следующие алгоритмы:

- алгоритм К-средних;
- алгоритм Кларка-Райта;
- алгоритм заметания.

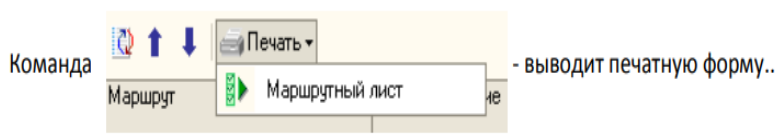
Решение задачи с помощью алгоритма Кларка-Райта происходит следующим образом:

Шаг 1. Указываем депо (склад отгрузки).

Шаг 2. Указываем грузоподъемность. Обязательное условие: грузоподъемность должна быть больше максимального веса любого из заказов на доставку.

Шаг 3. Выполняем команду «Рассчитать – Алгоритм Кларка-Райта».

В результате выполнения обработки формируется список маршрутов, как показано на рисунке 4.



2				
3	Документ	Адрес	Описание	Показатель
4	<i>Маршрут 1</i>			
5	Заказ покупателя ТДТ00021624 от 02.10.2015 9:23:04	194223, Санкт-Петербург г, Курчатова ул, дом 9	АкваПрофСнаб ООО	340,2
6	Заказ покупателя ТДТ00021829 от 01.10.2015 17:34:49	196084, Санкт-Петербург г, Заставская ул, дом 9	«Техникс» ООО	121,4
7	Заказ покупателя ТДТ00021831 от 01.10.2015 17:31:38	196158, Санкт-Петербург г, Звездная ул, дом 5,	Константинов М.Ю. ПБО	32,6
8	Заказ покупателя ТОП00002052 от 01.10.2015 17:09:57	194356 г. Санкт-Петербург пр. Луначарского д.52	МИСТЭК	147,6
9	<i>Маршрут 6</i>			
10	Заказ покупателя ТДТ00021945 от 05.10.2015 11:01:05	195248, Санкт-Петербург г, Энергетиков пр-кт, д	«Техникс» ООО	1 137,5
11	Заказ покупателя GR000005428 от 01.10.2015 10:16:01	190000, Санкт-Петербург г, Литовская ул, дом 8	ООО "Формат"	92,5

Рисунок 4 - Список маршрутов

При закрытии обработки посылается оповещение «ПростоеФормированиеМаршрутовНаКарте», с которым передаются данные о сформированных маршрутах. Эти данные можно получить в любой форме конфигурации, если для нее задана процедура-обработчик события ОбработкаОповещения (рисунок 5).

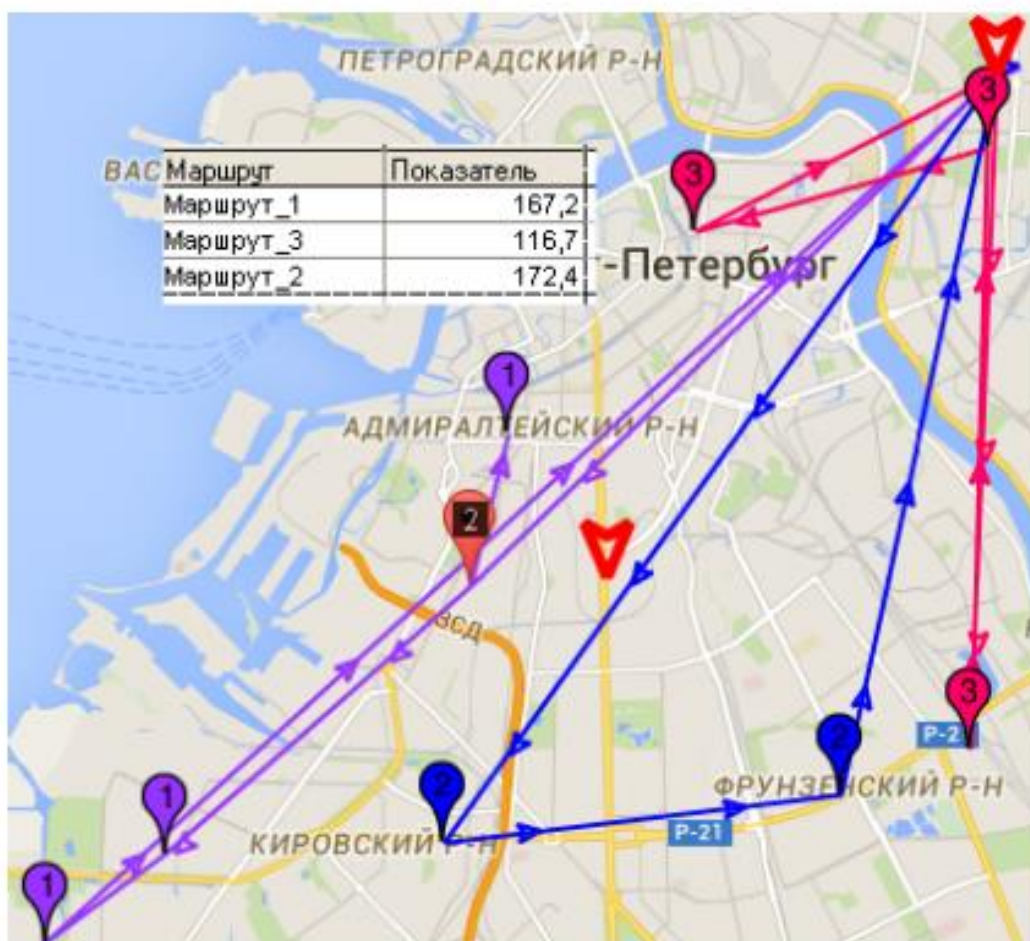


Рисунок 5 – Пример формирования маршрутов на карте

Программа разработана на технологической платформе «1С: Предприятие 8».

Для удобства анализа характеристик рассмотренных программ на предмет возможности использования в качестве основы для разработки ПО

для решения задачи МТСВ методом Кларка-Райта составлена таблица 3.

Таблица 3 – Сравнительный анализ ПО для решения задачи МТСВ методом Кларка-Райта

Характеристика/Оценка(+/-)	VRP Solver	1С:Транспортная логистика	Простые маршруты
Технология разработки ПО	IDE Delphi	Платформа «1С: Предприятие 8»	Платформа «1С: Предприятие 8»
Решение задачи МТСВ с помощью алгоритма Кларка-Райта	+	+	+
Дружественный интуитивный интерфейс	-	+	+
Онлайн-калькулятор	-	-	-
Функциональная избыточность	+	-	-

Как показал анализ, ни одна из представленных программ не соответствует всем требованиям, перечисленным в таблице 2.

Основным недостатком данных решений является невозможность использования в качестве онлайн-калькулятора.

Таким образом, разработка онлайн-калькулятора для решения задачи МТСВ методом Кларка-Райта с помощью современных веб-технологий представляет актуальность.

## 2.2 Анализ и выбор технологий для разработки программы

Для разработки веб-приложений используются системы управления контентом – CMS (Content Management System).

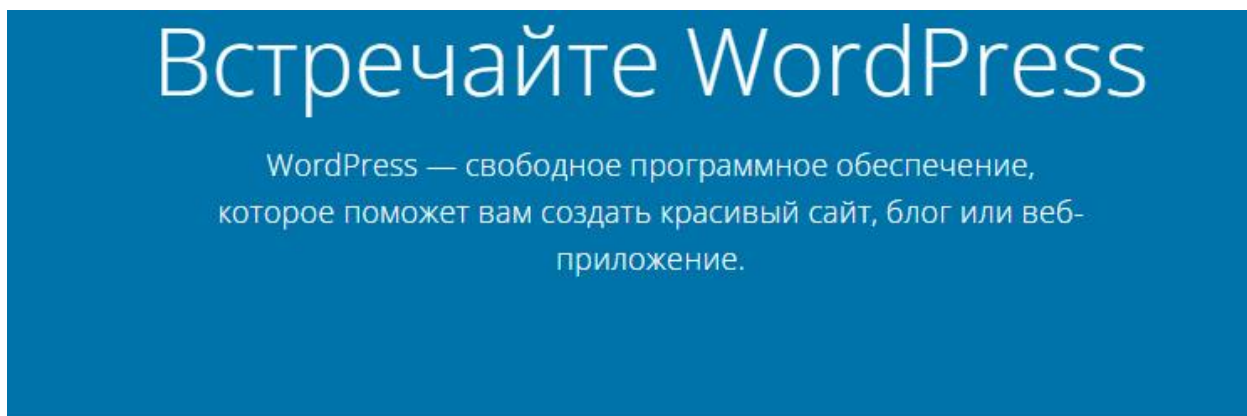
CMS обычно относится к веб-приложению, которое позволяет нескольким пользователям с разными уровнями привилегий управлять любым типом данных, контента или информации любого приложения веб-сайта, проекта на интернет-форуме.

Проанализируем популярные бесплатные CMS на предмет

использования в качестве платформы для разработки веб-приложения для решения задачи МТСВ методом Кларка-Райта.

### 2.2.1 CMS WordPress

WordPress — это программное обеспечение, используемое для создания, изменения и обслуживания веб-сайтов (рисунок 6) [16].



Красивый дизайн, впечатляющие возможности и свобода создавать всё, что вы захотите. WordPress одновременно бесплатен и бесценен.

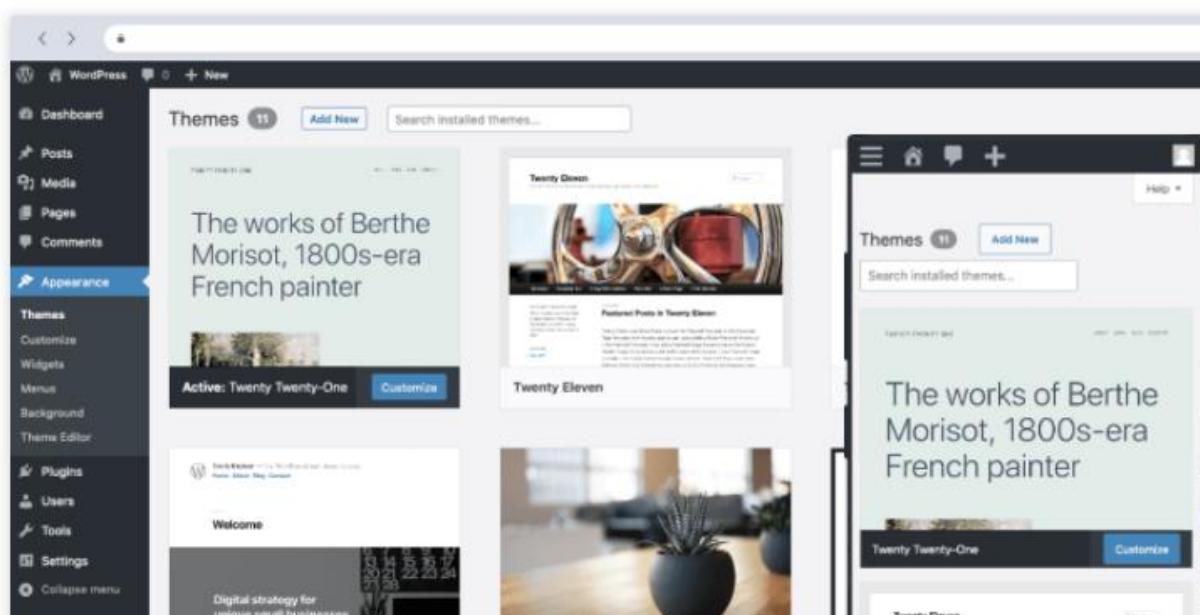


Рисунок 6 – Главная страница сайта WordPress.org

Преимущества WordPress:

- бесплатная лицензия. Хотя пользователям необходимо будет защитить веб-хостинг и доменное имя, прежде чем приступить к его использованию, сама платформа WordPress полностью бесплатна для загрузки и использования;
- гибкость и настраиваемость. WordPress имеет множество приложений — от ведения блога и создания онлайн-портфолио до создания магазина электронной коммерции. WordPress также приветствует разработчиков сообщества. В результате многие варианты тем и плагинов доступны в официальной библиотеке WordPress и сторонних хранилищах;
- масштабируемость. WordPress может обрабатывать веб-сайты любого размера, если в плане веб-хостинга есть необходимые ресурсы. Нет необходимости переходить на другую платформу после запуска веб-сайта;
- простота использования. По сравнению с ручным кодированием веб-сайта с нуля, WordPress требует значительно меньше технических знаний. Можно создавать красивые, полнофункциональные веб-сайты, используя только графический интерфейс пользователя.

WordPress — это самая популярная CMS на рынке, на которой работают 65,2% веб-сайтов, CMS которых известна.

Это соответствует 42,4% всех веб-сайтов, т.е. почти половине Интернета.

CMS WordPress работает в стеке технологий LAMP (Linux, Apache, MySQL и PHP) [8].

На уровне фронтенда используются HTML, CSS и JavaScript (jQuery).

WordPress использует принцип функционального программирования, основанный на вычислении путём вызова функции, которая не имеет состояния приложения.



## 2.2.2 CMS Drupal

Drupal — одна из самых популярных CMS в мире, на которой размещено более 7,2 миллиона веб-сайтов. Ее используют крупные международные организации, такие как The Economist, JYSK, Оксфордский университет и Pinteres [14].

Фрагмент страницы для разработчиков показан на рисунке 7.



Рисунок 7 – Фрагмент страницы Drupal для разработчиков

Основные характеристики Drupal:

- требования к квалификации разработчика. Drupal очень функциональна на базовом уровне, но, с другой стороны, это не самая удобная CMS, особенно по сравнению с WordPress. Например, у него есть разные темы, но многие из них представляют собой темы с пользовательским кодом. Это означает, что обычно понадобится

разработчик только для того, чтобы получить профессионально выглядящие результаты.

- возможность настройки. По определению это CMS с широкими возможностями настройки, которая включает более 16 000 модулей и плагинов. Это означает, что платформа позволяет изменять, настраивать и внедрять бесконечное множество дополнительных пользовательских функций на ваш веб-сайт, таких как CRM, безопасность, социальные сети и SEO;
- масштабируемость. CMS масштабируется для размещения самых насыщенных контентом сайтов и возможностей за счет интеграции ядра Drupal с множеством различных модулей Drupal. Это увеличивает возможности вашего сайта Drupal;
- служба поддержки. Огромное сообщество дополняет тысячи бесплатных плагинов. Это означает, что у вас есть еще больше возможностей предоставить своим клиентам потрясающие цифровые возможности. Кроме того, веб-сайт Drupal.org является важным источником поддержки, который позволяет вам изучать материал, быть в курсе предстоящих новостей или обращаться за поддержкой к участникам и специалистам.

Безопасность - это самая мощная и популярная функция Drupal.

Помимо безопасности на уровне предприятия, Drupal также публикует подробные отчеты о безопасности и, как правило, более прозрачен в отношении своей безопасности, чем WordPress.

### **2.2.3 CMS October**

October CMS — это система управления сайтом, построенная на компонентах Laravel [15].

Переведена на русский, позволяет гибко проектировать и масштабировать административную часть, без наследования проблем скорости ядра или избыточного, но не нужного проекту функционала.

Мощный фреймворк на php позволит решать сложные функциональные

задачи без значительных ограничений.

Использование October CMS даст базовую административную часть и возможность мелких расширений как плагинами из маркетплейса, так и собственными встраиваемыми страницами (рисунок 8).

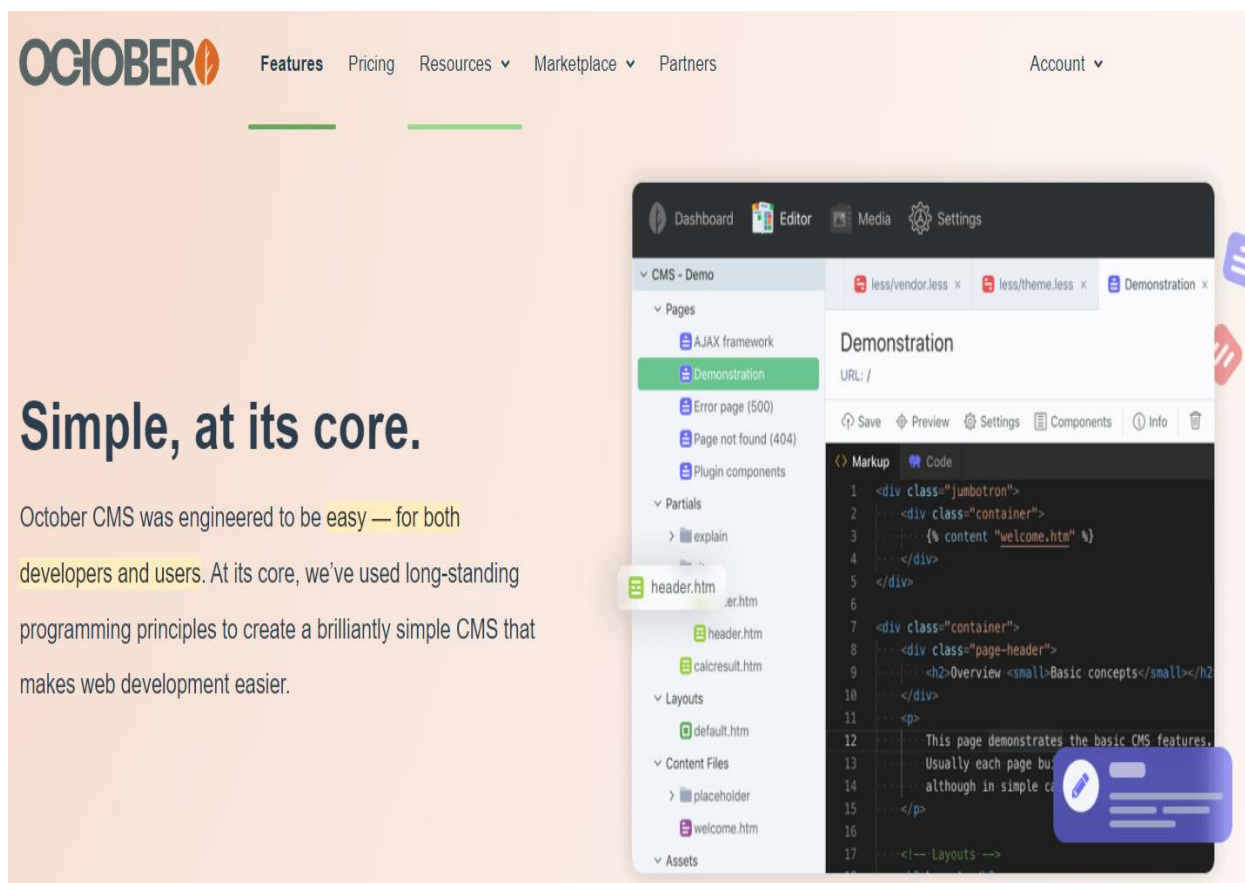


Рисунок 8 – Главная страница сайта octobercms.com

Разработчики и агентства любят эту CMS за простоту, как и их клиенты. Тысячи разработчиков и цифровых студий с такими клиентами, как Toyota, KFC и Nestle, уже используют October CMS в качестве своей основной платформы.

Для сравнения характеристик платформ составлена таблица 4.

Таблица 4 – Сравнительный анализ CMS WordPress, Drupal и October

Характеристика/балл (0-3)	WordPress	Drupal	October
Требования к квалификации разработчика	3	2	2
Применение шаблонов тем и плагинов	3	2	2
Масштабируемость	3	3	3
Предпочтение разработчика	3	1	1
Итого:	12	8	8

Таким образом, на основании сравнительного анализа в качестве платформы для реализации онлайн-калькулятора используем CMS WordPress.

### Выводы по главе 2

Вторая глава посвящена анализу и выбору технологий разработки ПО для решения задачи оптимизации маршрутов транспортных средств.

Результаты проделанной работы позволили сделать следующие выводы:

- как показал анализ, ни одна из представленных программа не соответствует всем требованиям, предъявляемым к онлайн-калькулятору для решения задачи МТСВ методом Кларка-Райта. Таким образом, разработка онлайн-калькулятора для решения задачи МТСВ методом Кларка-Райта с помощью современных веб-технологий представляет актуальность;
- для разработки веб-приложений используются системы управления контентом – CMS.

На основании сравнительного анализа в качестве платформы для реализации онлайн-калькулятора выбрана CMS WordPress.

### Глава 3 Реализация и тестирование онлайн-калькулятора для решения задачи оптимизации маршрутов транспортных средств

Для изображения программной архитектуры онлайн-калькулятора используем диаграмму компонентов UML.

Диаграммы компонентов используются для визуализации, спецификации и документирования систем, основанных на компонентах, а также для создания исполняемых систем путем прямого и обратного проектирования.

Диаграммы компонентов обычно содержат компоненты, интерфейсы и зависимости, обобщение, ассоциацию и отношения реализации. Они также могут содержать примечания и ограничения.

Главное преимущество диаграммы компонентов UML заключается в повторном использовании компонентов, особенно для разработки сложных приложений [19].

Программная архитектура онлайн-калькулятора показана на рисунке 9.

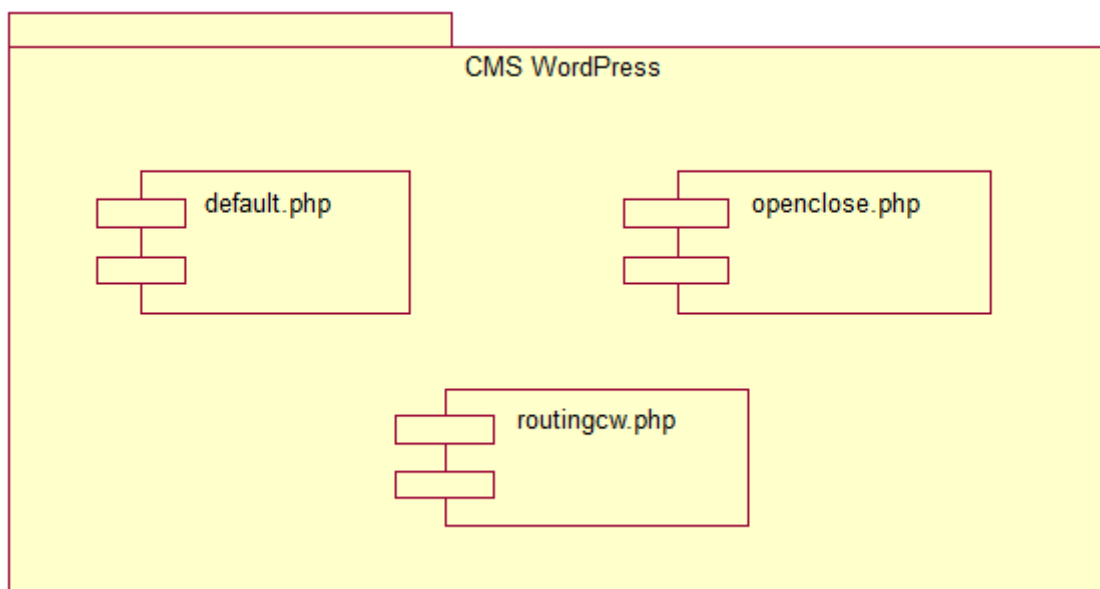


Рисунок 9 – Программная архитектура онлайн-калькулятора

Диаграмма классов онлайн-калькулятора показана на рисунке 10.

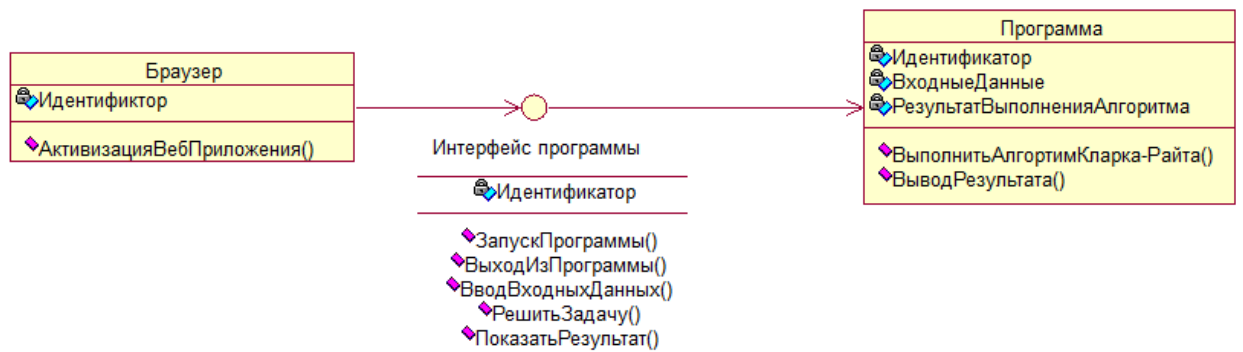


Рисунок 10 – Диаграмма классов онлайн-калькулятора

Основные компоненты веб-приложения онлайн-калькулятора написаны на языке PHP [9].

Программный код, реализующий алгоритм Кларка-Райта, представлен в листинге 1.

Листинг 1 – PHP-код алгоритма Кларка-Райта.

```

<?php
/*
parametric run execution of cw model
*/
set_time_limit(2400);
ini_set('error_log','error_log.txt');
include('matrixcw.php');
include('data.php');
$time_start = microtime(true);
$min_unstisfied = count($client_nodes);
$saved_routes;
$saved_unstisfied;
$saved_total_cost;
$saved_reassign_cost;

```

```

$min_total_cost=0;
$max = 100;
for($n=0 ; $n<$max ; $n++){
    $cw = new cw();
    $cw-
>loadData($client_nodes,$client_demands,$distance_matrix,$truck_capacity);
    if($n==0){
        $cw->cwParametricSavings(1,0); //no parametric run case for comparison
    }
    else{
        $cw->cwParametricSavings();
    }
    $cw->sortSavings();
    $cw->cwroutes();
    $cw->finishingTouch();
    if(count($cw->unsatisfied)<$min_unstisfied || (count($cw-
>unsatisfied)==$min_unstisfied && $min_total_cost>$cw->total_cost)){
        $min_unstisfied = count($cw->unsatisfied);
        $min_total_cost = $cw->total_cost;
        $saved_routes = $cw->routes;
        $saved_total_cost = $cw->total_cost;
        $saved_unstisfied = $cw->unsatisfied;
        $saved_reassign_cost = $cw->reassing_cost;
    }
}

$time_end = microtime(true);
echo "<h1>Clarke & Wright savings Hueristics with assymmetric costs and time
windows : Parametric Run $max iterations</h1><br>";
foreach ($saved_routes as $key => $value) {

```

```

echo "<h2>Route number {$key}</h2><br>";
echo "<b>Truck: </b>". $value['truck']. "<br>";
echo "<b>Path: </b>". implode('-', $value['road']). "<br>";
echo "<b>Total Distance of Path: </b>". $value['total_distance']. "<br>";
echo "<b>Total Time of Path: </b>". $value['total_time']. "<br>";
echo "<b>Time back at depot: </b>". $value['time_back_at_depot']. "<br>";
echo "<b>Weight capacity of truck : </b>". $value['truck_max_weight']. "
Kg<br>";
echo "<b>Total weight of order : </b>". $value['total_weight']. " Kg<br>";
echo "<b>Volume capacity of truck : </b>". $value['truck_max_volume']. "
M3<br>";
echo "<b>Total volume of order : </b>". $value['total_volume']. " M3<br>";
echo "<br><br>";
}
if(count($saved_unstisfied)>1){
echo "The following clients weren't visited : <br>";
?>
<table>
    <thead>
        <th>Client</th><th>Reason</th>
    </thead>
    <tbody>
        <?php foreach($saved_unstisfied as $client=>$reason){ ?>
<tr><td><?=$client?></td><td><?=$reason?></td></tr>
        <?php } ?>
    </tbody>
    <table>
        <br>
    <?php
}

```



```

else echo "all clients were visite! <br>";
echo "The total cost for the routing problem is : ".$saved_total_cost." seconds
<br>";
echo "The total excution time of the Hueristics was : ".$time_end-$time_start."
seconds "."<br>";
echo "The total reassign cost of the process was :
".$saved_reassign_cost."<br>".<br>";
if(count($conversion)>0){
    echo "the conversion for duplicate nodes was as follow : <br>";
    foreach ($conversion as $key => $value) {
        list($from,$to) = $value;
        $num = $key+2;
        echo "$from -> $to <br>";
    }
}
}

```

Для проверки работоспособности онлайн-калькулятора использован метод функционального тестирования.

«Функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного ПО исходным функциональным требованиям заказчика.

Проведение функционального тестирования – процесс, позволяющий проверить способность информационной системы работать в определенных условиях и решать задачи, нужные пользователям» [10].

Работа с онлайн-калькулятором выполняется в следующей последовательности:

Шаг 1. Выберите количество клиентов (узлов) (рисунок 11);



Выбрать количество клиентов:

Рисунок 11 – Окно выбора количества клиентов

Шаг 2. Введите требования каждого клиента (рисунок 12).

Требования *Клиент*

Клиент	1	2	3
Требование (U)	<input type="text" value="40"/>	<input type="text" value="15"/>	<input type="text" value="10"/>

Рисунок 12 – Окно ввода требований клиентов

Шаг 3. Укажите стоимость (расстояние) между депо и клиентом (рисунок 13).

Стоимость *Депо. - Клиент*

$c(0,i)$	1	2	3
0	<input type="text" value="4"/>	<input type="text" value="6"/>	<input type="text" value="5"/>

Рисунок 13 – Окно ввода стоимости между депо и клиентом

Шаг 4. Укажите максимальную грузоподъемность (вместимость) транспортного средства (рисунок 14). Обратите внимание, что она должна быть указана в тех же единицах, что и требования.

### Вместимость транспорт / маршрут

<b>Маршрут</b>	
<b>Вместимость (U)</b>	<input type="text" value="80"/>

Рисунок 14 – Окно ввода максимальной вместимости транспортного средства

Шаг 5. Рассчитайте матрицу стоимости (расстояния) (рисунок 15).

### Стоимость Клиент-Клиент







$c(i,j)$	 1	 2	 3
 1	<input type="text" value="0"/>	<input type="text" value="4"/>	<input type="text" value="2"/>
 2	<input type="text"/>	<input type="text" value="0"/>	<input type="text" value="3"/>
 3	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>

Рисунок 15 – Окно расчета матрицы стоимости

Шаг 6. Рассчитайте матрицу экономии (рисунок 16).

### Матрица экономии

$s(i,j)$	1	2	3
1	0	6	7
2	0	0	8
3	0	0	0

Проверить значения массива

Показать правильные значения

Рисунок 16 – Окно расчета матрицы экономии

Шаг 7. Решить задачу, нажав на кнопку «Алгоритма Кларка-Райта» (рисунок 17).

### Назначение Клиентов на Маршруты и порядок посещения

Маршрут Клиент	1	2	3
1	3	-	-
2	1	-	-
3	2	-	-

(U)

Расходы

Расходы Итого

Clarke & Wright

Рисунок 17 – Окно запуска алгоритма решения

Сразу после этого вы можете увидеть матрицу расстояний, автоматически вычисляемую алгоритмом.

Таким образом, при решении задачи онлайн-калькулятором будут получены следующие результаты:

- матрица спроса (U): входные требования;
- матрица расстояний: отображает расстояние между складом и каждым клиентом;
- матрица экономии: отображает экономию, полученную по разработанному маршруту;
- матрица маршрутов, назначений и посещений: отображает распределение грузовиков по разным адресам. Это означает, пункты назначения, в которые должен доставить каждый грузовик и в каком порядке. Обратите внимание, что пользователь может изменить эту матрицу.
- матрица (U): отображает общую загрузку каждого грузовика после назначения.
- матрица расходов: отображает общее расстояние, пройденное каждым грузовиком, что может помочь пользователю рассчитать транспортные расходы.

Как показало функциональное тестирование, разработанный онлайн-калькулятор позволяет решить задачу оптимизации методом Кларка-Райта, путем автоматического вычисления оптимальных маршрутов и матрицы экономии.

### Выводы по главе 3

Третья глава посвящена реализации и тестированию онлайн-калькулятора для решения задачи оптимизации маршрутов транспортных средств.

Результаты проделанной работы позволили сделать следующие выводы:

- главное преимущество диаграммы компонентов UML заключается в повторном использовании компонентов, особенно для разработки сложных приложений;
- основные компоненты веб-приложения онлайн-калькулятора написаны на языке PHP;
- для проверки работоспособности онлайн-калькулятора использован метод функционального тестирования;
- функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного ПО исходным функциональным требованиям заказчика.

Как показало функциональное тестирование, разработанный онлайн-калькулятор позволяет решить задачу оптимизации методом Кларка-Райта, путем автоматического вычисления оптимальных маршрутов и матрицы экономии.

## Заключение

Выпускная квалификационная работа посвящена актуальной проблеме разработки программного обеспечения для решения задач комбинаторной оптимизации.

Следует констатировать отсутствие универсального программного обеспечения, предназначенного для решения конкретных задач комбинаторной оптимизации.

Вместе с тем многие компании, особенно занятые в сфере перевозок грузов, нуждаются в таких программах для формирования оптимального плана для их доставки.

В этой связи разработка программного обеспечения для решения задач комбинаторной оптимизации представляет актуальность и научно-практический интерес.

Для достижения данной цели в процессе работы над бакалаврской работой решены следующие задачи:

- выполнена постановка задачи на разработку программного обеспечения для решения задачи комбинаторной оптимизации. Как показал анализ, задача МТСВ – это проблема маршрутизации транспорта, в которой транспортные средства с ограниченной грузоподъемностью должны забирать грузы из разных мест или доставлять их в разные места. Наиболее популярным алгоритмом для решения задачи МТСВ, является алгоритм Кларка-Райта и его модификации. Разработан перечень требований, который является основой для разработки ПО для решения задачи оптимизации маршрутов транспортных средств;
- проанализированы и выбраны технологии для разработки программного обеспечения для решения задач комбинаторной оптимизации. Как показал анализ, ни одна из известных программ не соответствует всем требованиям, предъявляемым к онлайн-

калькулятору для решения задачи МТСВ методом Кларка-Райта. Таким образом, разработка онлайн-калькулятора для решения задачи МТСВ методом Кларка-Райта с помощью современных веб-технологий представляет актуальность. Для разработки веб-приложений используются системы управления контентом – CMS. На основании сравнительного анализа в качестве платформы для реализации онлайн-калькулятора выбрана CMS WordPress;

- выполнены реализация и тестирование онлайн-калькулятора. Разработана программная архитектура онлайн-калькулятора в виде диаграммы компонентов UML. Основные компоненты веб-приложения онлайн-калькулятора написаны на языке PHP. Для проверки работоспособности онлайн-калькулятора использован метод функционального тестирования. Как показало функциональное тестирование, разработанный онлайн-калькулятор позволяет решить задачу оптимизации методом Кларка-Райта, путем автоматического вычисления оптимальных маршрутов и матрицы экономии.

Результаты бакалаврской работы представляют научно-практический интерес и могут быть рекомендованы для разработки программного обеспечения для решения задач комбинаторной оптимизации.



## Список используемой литературы

1. Комбинаторная и линейная оптимизация [Электронный ресурс]. URL: <https://veeroute.ru/combinatorial-and-linear-integer-optimization/> (дата обращения: 19.03.2022).
2. Косяков С.В., Гадалов А.Б., Жидовинов К.А. Оптимальное планирование грузоперевозок на базе ГИС-технологий // Вестник ИГЭУ, 2010.
3. Мартынова Ю. А., Мартынов Я. А. Формализация задачи организации маршрутных сетей городского пассажирского транспорта // Вестник евразийской науки. 2014. №6 (25). URL: <http://naukovedenie.ru/PDF/124TVN614.pdf> (дата обращения: 21.03.2022).
4. Метод Кларка-Райта. Оптимальное планирование маршрутов грузоперевозок [Электронный ресурс]. URL: <https://infostart.ru/1c/articles/443585/> (дата обращения: 21.03.2022).
5. Никоноров В. М. Реализация усовершенствованного метода Кларка-Райта для решения задачи маршрутизации автомобильных мелкопартионных перевозок // Научно-технические ведомости СПбГПУ. 2012. 2-2'. С. 210-215.
6. Программа «1С: Транспортная логистика, экспедирование и управление автотранспортом КОРП» [Электронный ресурс]. URL: <https://efsol.ru/manuals/1s-transportnaya-logistika-ekspedirovanie-i-upravlenie-avtotransportom-korp/> (дата обращения: 21.03.2022).
7. Программа «Простые маршруты» [Электронный ресурс]. URL: <https://infostart.ru/public/635798/> (дата обращения: 21.03.2022).
8. Стек против LAMP: плюсы и минусы [Электронный ресурс]. URL: <https://triu.ru/stek-protiv-lamp-plyusy-i-minusy/> (дата обращения: 21.03.2022).
9. Флloyd К. С. Введение в программирование на PHP5 [Электронный ресурс]: учебное пособие. Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. 280 с. URL: <https://www.iprbookshop.ru/101998.html> (дата обращения: 30.03.2022).
10. Функциональное тестирование [Электронный ресурс]. URL:

<https://www.appline.ru/services/testing/functionalnoe-testirovanie> (дата обращения: 30.03.2022).

11. Achuthan N.R., Caccetta L., Hill S.P. A new subtour elimination constraint for the vehicle routing problem. *European Journal of Operational Research*. 1996; 91(3):573–86.

12. Alameen M., Aljamal R. and Damrah S. A Clarke and Wright Improved Algorithm to Solve the Vehicle Routing and Traveling Salesman Problem, *Global Journal of Enterprise Information System*, Volume 8, Issue 1, P. 35-37.

13. Capacitated Vehicle Routing Problem formulation [Электронный ресурс]. URL: <https://how-to.aimms.com/Articles/332/332-Formulation-CVRP.html> (дата обращения: 19.03.2022).

14. CMS Drupal [Электронный ресурс]. URL: <https://drupal.org> (дата обращения: 21.03.2022).

15. CMS October [Электронный ресурс]. URL: <https://octobercms.com/> (дата обращения: 21.03.2022).

16. CMS Wordpress [Электронный ресурс]. URL: <https://ru.wordpress.org/> (дата обращения: 21.03.2022).

17. Pichpibul T., Kawtummachai R. A Heuristic Approach Based on Clarke-Wright Algorithm for Open Vehicle Routing Problem, *The Scientific World Journal*, 2013. URL: <https://www.hindawi.com/journals/tswj/2013/874349/> (дата обращения: 21.03.2022).

18. Software Engineering-FURPS [Электронный ресурс]. URL: <https://www.1000sourcecodes.com/2012/05/software-engineering-furps.html> (дата обращения: 21.03.2022).

19. The component diagram [Электронный ресурс]. URL: <https://developer.ibm.com/articles/the-component-diagram/> (дата обращения: 21.03.2022).

20. Valenton G. et al. A Comparison Of Clarke-Wright Method And Branch & Bound Method For Vehicle Routing Problem In Pangasinan, Southeast Asian Journal of Science and Technology, 2021, 6(2 (SI), 1-9.

21. VRP Solver [Электронный ресурс]. URL: <https://coral.ise.lehigh.edu/larry/software/vrp-solver/> (дата обращения: 21.03.2022).

22. What is Combinatorial Optimization [Электронный ресурс]. URL: <https://www.igi-global.com/dictionary/optimization-approaches-for-a-home-healthcare-routing-and-scheduling-problem/4533> (дата обращения: 21.03.2022).

Вып. 4. С. 1-6.