

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ИРКУТСКИЙ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

На правах рукописи



НГУЕН ГУЙ ЛИЕМ

**ИНТЕЛЛЕКТУАЛЬНАЯ ТЕХНОЛОГИЯ РЕШЕНИЯ
ЗАДАЧ ОПТИМИЗАЦИИ ТРАНСПОРТНО–ЛОГИСТИЧЕСКИХ СИСТЕМ
НА ОСНОВЕ ФИЗИЧЕСКИХ АНАЛОГИЙ**

Специальность 05.13.18 – Математическое моделирование,
численные методы и комплексы программ

**Диссертация на соискание ученой степени
кандидата технических наук**

**Научный руководитель:
д.ф.-м.н. А.Л. Казаков**

Иркутск – 2016

СОДЕРЖАНИЕ

Введение.....	3
Глава 1. Состояние современных исследований в области математического моделирования в транспортной логистике.....	11
1.1. Применение математических моделей в транспортной логистике	11
1.2. Обзор численных методов для решения задачи маршрутизации, задачи об оптимальном размещении логистических объектов, задачи об упаковке и покрытии	16
1.3. Оптико-геометрический подход.....	33
Глава 2. Математические модели, численные методы для решения задач оптимизации транспортной логистики на основе оптико-геометрического подхода	36
2.1. О построении математических моделей	36
2.2. Задача построения оптимальных упаковок равных кругов.....	39
2.3. Задача построения оптимальных покрытий равными кругами	44
2.4. Задача размещения логистических объектов в условиях кооперации и конкуренции.....	47
2.5. Задача организации системы коммуникаций с учетом особенностей местности	52
2.6. Задача оптимизации ремонта автомобильных дорог в условиях ограниченного финансирования.....	56
Глава 3. Программное обеспечение для поддержки исследований транспортной логистики.....	61
3.1. Программная система «ОЛТП»	61
3.2. Интеллектуальная информационная система для поддержки исследований транспортной логистики (ИИС).....	69
Глава 4. Вычислительные эксперименты и прикладные задачи	80
4.1. Модельные примеры.....	80
4.2. Прикладные задачи	102
Заключение	114
Список литературы	115
Приложения	130

Введение

Актуальность исследования. Транспортно-логистическая система (ТЛС) является важной частью социальной и производственной инфраструктуры любого региона, поэтому проблема ее оптимизации с целью получения определенного экономического эффекта остается актуальной и в настоящее время.

Для решения транспортно-логистических задач (ТЛЗ) достаточно часто применяются методы теории графов, а также различные модификации задач линейного и дискретного программирования (Б.А. Аникин, Д.Т. Лотарев, А.П. Уздемир, В.С. Лукинский, В.В. Лукинский, Ю.В. Малевич и др.). В частности, при решении задачи об оптимальном размещении логистических центров традиционно применяют метод «центра тяжести» и метод кластеризации «к-средних», которые описаны в работах В.С. Лукинского и И.Д. Манделя, соответственно.

В последние десятилетия с развитием и усложнением структуры ТЛС появилась необходимость более широкого применения различных математических методов для решения возникающих задач оптимизации. Так, методы математического программирования для задач транспортно-складской логистики в своих работах использовали М.Я. Постан, В.Е. Николайчук, методы дискретной математики для задач планирования – Р. Борндорфер (R. Borndörfer), М. Гротшил (M. Grötschel), А. Лобеле (A. Lobe), а для задач размещения – Е.В. Алексеева, В.Л. Береснев, И.Л. Васильев, Москаленко А.И., Батурин В.А., Е.Н. Гончаров, А.В. Еремеев, А.А. Колоколов, Ю.А. Кочетов, Ю.Ю. Фикельштейн. Кроме этого, отметим следующих авторов: Н.Х. Гартнер (N.H. Gartner), Б.Л. Голден (B.L. Golden) и Р.Т. Вонг (R.T. Wong), которые в своих исследованиях, в частности, обращались к проблемам сетевого анализа, проектирования и управления сетью ТЛС.

Большинство из известных алгоритмов решения ТЛЗ работает с евклидовым расстоянием, из-за чего возникают трудности учета особенностей рельефа местности и наличия естественных преград (горы, овраги, водоемы и т.д.).

В диссертационной работе предлагается технология решения данной проблемы, основанная на аналогии между распространением света в оптически неоднородной среде и минимизацией интегрального функционала (оптико-геометрический подход), которая позволяет устранить указанные недостатки.

В рамках указанной технологии развиты математические модели ряда объектов и подсистем ТЛС и новые численные методы для решения возникающих при этом задач оптимизации: построение оптимальных упаковок и покрытий равными кругами (для задачи размещения логистических узлов при непрерывном распределении потребителей), размещение дополнительных логистических центров на области обслуживания в условиях кооперации и конкуренции, организация системы коммуникаций с учетом особенностей местности, ремонт автомобильных дорог в условиях ограниченного бюджета и т.д. Все вышеперечисленные алгоритмы были реализованы в виде программной системы. Отметим, что разработанный математический аппарат и программная система позволяют решать задачи, имеющие существенное прикладное значение.

Объект и предмет исследования. Объектом исследования является транспортно-логистическая система. Предмет исследования – математические модели ТЛС и численные методы решения возникающих при моделировании задач оптимизации.

Цель и задачи исследования. Целью исследования являются разработка математических моделей, численных методов и программных средств исследования оптимизационных задач транспортной логистики; разработка интеллектуальной системы поддержки научных исследований в области транспортной и инфраструктурной логистики.

Для достижения поставленной цели необходимо:

- 1) свести транспортно-логистические задачи к задачам вариационного исчисления с учетом возможных естественных условий, включая: дорожная сеть, населенные пункты, существующие логистические центры, особенности местности и т.д.;

2) разработать численные методы, основанные на использовании физических аналогий (оптико-геометрический подход: принципы Ферма и Гюйгенса), позволяющие решать задачи при наличии ограничений;

3) разработать программную систему, реализующую авторские численные алгоритмы;

4) в составе интеллектуальной информационной системы (ИИС) поддержки научных исследований в области транспортной и инфраструктурной логистики разработать информационную аналитическую подсистему (ИАС) для взаимодействия с пользователями, экспертную систему (ЭС) и методы интеграции между ИАС, ЭС и вычислительными модулями.

Методы исследования. При выполнении исследования применены методы: геометрической оптики, вычислительной геометрии, математического моделирования, непрерывной оптимизации. С другой стороны, для построения экспертных систем использовались методы приобретения знаний, представления знаний, управления процессом поиска решения, разъяснения принятого решения. Для реализации программной системы использована среда разработки Visual studio 2012 (язык программирования C#), система управления базами данных SQLServer Express 2008 и библиотеки CLIPSNet и CLIPSLib в соответствии с оболочкой поддержки CLIPS.

Научная новизна. Научная новизна исследования состоит в следующем.

1. Предложена многоэтапная технология исследования ТЛС регионального уровня.

2. Задачи оптимального размещения инфраструктурных логистических объектов сведены к специальным модификациям двух известных математических задач: об упаковке и покрытии равными кругами ограниченных множеств в двумерных метрических пространствах с неевклидовыми метриками.

3. Разработана модификация оптико-геометрического подхода, которая основывается на распространении световой волны из источников двух типов: то-

чечного и распределенного вдоль замкнутой кривой. В последнем случае волна может распространяться как во внутреннюю, так и во внешнюю область.

4. Предложены и реализованы на программном уровне новые методы построения оптимальных упаковок и покрытий равными кругами неодносвязных множеств в неевклидовой метрике.

5. Разработаны численные методы для решения задач об оптимальном размещении дополнительных логистических центров в условиях кооперации и конкуренции, об оптимизации системы коммуникаций с учетом региональных особенностей, об оптимизации ремонта автомобильных дорог в условиях ограниченного бюджета.

6. Создана и интегрирована в ИИС программная система «ОТЛП», в основу которой положены оригинальные численные алгоритмы, разработанные в рамках диссертационного исследования.

Достоверность и обоснованность. Достоверность и обоснованность научных результатов обеспечивается применением известных фундаментальных физических принципов для построения математических моделей и разработки численных алгоритмов, корректностью исходных данных для проведения проведенного эксперимента, согласованностью экспериментальных и теоретических данных.

Практическая значимость. Практическая значимость результатов исследования заключается в следующем.

Разработаны эффективные численные методы решения задач транспортной логистики при наличии ограничений различного рода.

Предложенный подход к построению и исследованию математических моделей может быть использован в других классах прикладных задач, таких, как задачи безопасности, управления движением, задачи выбора оптимальной мощности универсальных двигательных установок малой тяги, задачи проектировании энергоэффективной системы мониторинга протяженных объектов беспроводными сенсорными сетями и т.д.

Результаты диссертационного исследования использованы в учебном процессе при проведении занятий по дисциплинам «Системология» и «Методы оптимизации». Получены акты о внедрении результатов диссертационной работы в учебный процесс ФГБОУ ВО «ИРНИТУ», а также при проведении научных исследований в Институте технологии моделирования Технического университета им. Ле Куй Дона (Вьетнам).

Апробация результатов исследования. Работа выполнялась на кафедре автоматизированных систем ИРНИТУ. Основные результаты диссертационного исследования докладывались и обсуждались на следующих научных конференциях:

– Всероссийская молодежная научно-практическая конференция «Винеровские чтения» (ИРНИТУ, г. Иркутск. 2014, 2015);

– XIX и XX Байкальская Всероссийская конференция «Информационные и математические технологии в науке и управлении» (ИСЭМ СО РАН, г. Иркутск. 2014, 2015);

– Вьетнамо-российская международная научная конференция (ГТУ им Ле Куй Дона, г. Ханой, Вьетнам. 2015);

– III Российско-монгольская конференция молодых ученых по математическому моделированию, вычислительно-информационным технологиям и управлению (п. Ханх, Монголия. 2015);

– 8-я Всероссийская мультikonференция по проблемам управления (г. Геленджик. 2015);

– XVI Всероссийская конференция молодых ученых по математическому моделированию и информационным технологиям (ИВМ СО РАН, г. Красноярск. 2015);

– Международный семинар «ситуационное управление, интеллектуальные, агентные вычисления и кибербезопасность в критических инфраструктурах» (г. Иркутск. 2016);

– 5-я Международная конференция по анализу изображений, социальных сетей и текстов (г. Екатеринбург. 2016);

– Пятая межвузовская научно-практическая конференция молодых ученых «Проблемы информационного и математического моделирования сложных систем» (ИрГУПС, г. Иркутск. 2016).

Результаты диссертационного исследования неоднократно докладывались на научных семинарах кафедры автоматизированных систем (рук. – к.т.н. С.В. Бахвалов) Иркутского национального исследовательского технического университета, и лаборатории системного анализа и вычислительных методов ИДСТУ СО РАН (рук. – к.ф.-м.н. А.А. Лемперт).

Результаты диссертационного исследования опубликованы в 15 научных работах, из них 3 статьи в изданиях, входящих в Перечень ВАК. Получены 2 свидетельства о государственной регистрации программ для ЭВМ.

Основные результаты исследования опубликованы в следующих работах.

Издания, входящие в Перечень ВАК РФ:

1. Казаков А.Л. Оптимизация системы коммуникаций с учетом региональных особенностей: математическая модель и численный метод / А.Л. Казаков, А.А. Лемперт, Г.Л. Нгуен // Вестник Иркутского гос. технического ун-та. – 2014. – № 12 (95). – С. 17–22.

2. Казаков А.Л. Об одном алгоритме построения упаковки конгруэнтных кругов в неодносвязное множество с неевклидовой метрикой / А.Л. Казаков, А.А. Лемперт, Г.Л. Нгуен // Вычислительные методы и программирование. – 2016. – Т.17. – С. 177–188.

3. Казаков А.Л. Алгоритм построения оптимальных покрытий равными кругами невыпуклых многоугольников с неевклидовой метрикой / А.Л. Казаков, А.А. Лемперт, Г.Л. Нгуен // Вестник Иркутского гос. технического ун-та. – 2016. – № 5 (112). – С. 45–55.

Издания, включенные в РИНЦ:

4. Столбов А.Б. Информационная система поддержки исследований в инфраструктурной логистике / А.Б. Столбов, Г.Л. Нгуен // Программные системы: Теория и приложения. – 2015. – № 3 (26). – С. 3–20.

Свидетельства о государственной регистрации программы для ЭВМ:

5. Казаков А.Л. Программный модуль оптимального размещения логистических центров / А.Л. Казаков, Г.Л. Нгуен, А.А. Лемперт // Свидетельство о гос. регистрации программы для ЭВМ. № 2015616554 от 15 июня 2015 г. М.: Федеральная служба по интеллектуальной собственности. 2015.

6. Казаков А.Л. Программный модуль построения оптимальных покрытий и упаковок в оптически неоднородной среде / А.Л. Казаков, Г.Л. Нгуен, А.А. Лемперт // Свидетельство о гос. регистрации программы для ЭВМ. № 2016614997 от 13 мая 2016 г. М.: Федеральная служба по интеллектуальной собственности. 2016.

Прочие издания:

7. Казаков А.Л. Интеллектуальная информационная система поддержки комплексных исследований транспортных процессов и объектов / А.Л. Казаков, А.А. Лемперт, А.Б. Столбов, Г.Л. Нгуен // Информ. и матем. технологии в науке и управлении. Иркутск: ИСЭМ СО РАН, 2014. – Т. 3. – С. 57–61.

8. Столбов А.Б. Программная система поддержки исследований в инфраструктурной логистике на основе знаний / А.Б. Столбов, Г.Л. Нгуен // Информ. и матем. технологии в науке и управлении. – Иркутск: ИСЭМ СО РАН, 2015. – Т. 3. – С. 159–166.

9. Казаков А.Л. Оптимизация размещения дополнительных логистических центров в условиях конкуренции и кооперации / А.Л. Казаков, А.А. Лемперт, Г.Л. Нгуен, В.Н. Фам // Журнал по науке и технике (Journal of Science and Technology). – Ханой: ГТУ им Ле Куи Дона, 2015. – №. 171. – С. 20–31.

10. Столбов А.Б. О подходе к формализации знаний в транспортной предметной области / А.Б. Столбов, Г.Л. Нгуен // Винеровские чтения 2014. Материа-

лы Всерос. молодежной научно-практ. конф. Иркутск: Изд-во ИРНИТУ, 2014. – С. 9–11.

11. Нгуен Г.Л. О задаче размещения логистических объектов в условиях взаимодействия с существующими / Г.Л. Нгуен // Винеровские чтения 2015. Материалы Всерос. молодежной научно-практ. конф. Иркутск: Изд-во ИРНИТУ, 2015. – С. 48–49.

12. Казаков А.Л. Построение оптимальных покрытий и упаковок с неевклидовой метрикой с приложением к задачам логистики / А.Л. Казаков, А.А. Лемперт, Г.Л. Нгуен // Материалы 8-й Всероссийской мультikonференции по проблемам управления. Ростов-на-Дону: ЮФУ, 2015. – Т. 1. – С. 186–188.

13. Казаков А.Л. Построение оптимальных упаковок для компактных множеств в неевклидовой метрике / А.Л. Казаков, А.А. Лемперт, Г.Л. Нгуен // Тезисы Российско-монгольской конф. молодых ученых по матем. моделированию, вычисл. -информ. технологиям и управлению. Иркутск–Ханх: Изд-во ИДСТУ СО РАН, 2015. – С. 47.

14. Нгуен Г.Л. О задаче оптимального размещения логистических центров в неравномерно заселенной области / Г.Л. Нгуен // Материалы XVI Всерос. конф. молодых ученых по матем. моделированию и информ.технологиям. Новосибирск: ИВТ СО РАН, 2015. – С. 44–45.

15. Kazakov A.L. On an optimal packing problem of the equal circles in non-simply set / A.L. Kazakov, A.A. Lempert, H.L. Nguyen // Proceedings and program of the international Workshop «Contingency management, intelligent, agent-based computing and cyber security in critical infrastructures». Irkutsk, 2016. – P. 32–34.

Личный вклад. Все выносимые на защиту результаты получены лично автором или при его непосредственном участии.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, заключения и списка литературы из 144 наименований. Объем работы составляет 135 страниц, 68 рисунков и 15 таблиц.

Глава 1. Состояние современных исследований в области математического моделирования в транспортной логистике

1.1. Применение математических моделей в транспортной логистике

В области решения транспортно-логистических задач (ТЛЗ), в частности, задач оптимизации таких систем для повышения их производительности было проведено множество научных исследований. Можно утверждать, что современная транспортная логистика достигла такого уровня, когда применение математических методов стало не только возможным, но и необходимым.

Как правило, выделяют два типа задач оптимизации: по критерию стоимости (достижение минимума затрат на перевозку) или расстояния и по критерию времени (затрачивается минимум времени на перевозку), для которых можно выделить три уровня: стратегический (например, размещение пунктов обслуживания и организация систем коммуникаций), диспетчерский (оптимизация маршрутов доставки груза от поставщика до потребителей), оперативный (динамическая маршрутизация с учетом дорожного движения). При этом предметом транспортной логистики является множество задач планирования и управления, связанных с перемещением груза [51, 53], например, обеспечение технологического единства транспортно-складского хозяйства, совместное планирование производственного, транспортного и складского процессов; определение рациональных маршрутов; распределение транспортных средств по маршрутам; оценка качества транспортного сервиса и т.п.

В данной главе представлен обзор общего состояния исследований в транспортной логистике и рассмотрены некоторые математические модели, наиболее тесно примыкающие к тематике настоящего исследования.

Прежде всего отметим транспортную задачу, под которой понимается огромное количество постановок с единой математической моделью, относящейся к классу задач линейного программирования. Проблема была впервые формализована французским математиком М. Гаспаром в 1781 году. По сведениям А. Схрейвер [134] первым, кто изучал транспортную задачу математически, был

А.Н. Толстой из СССР. В 1930 г. вышла его работа о поиске минимального общего километража в железнодорожных перевозках, где использовались перераспределительные циклы [134]. Разработка общих методов решения задачи линейного программирования и их математическое исследование связано с именем советского математика, нобелевского лауреата Л.В. Канторовича [114]. Первой классической транспортной задачей в логистике можно считать задачу линейного программирования специального вида о поиске оптимального распределения однородных объектов от поставщика к потребителям с минимизацией затрат на перемещение [73, 89]. Основным методом для решения этой задачи – симплекс-метод, который был опубликован в 1949 г. американским ученым Дж.Б. Данцигом [19].

В последние десятилетия с развитием логистики математические методы исследования задач оптимизации ТЛС продолжают интенсивно развиваться. Можно отметить следующих авторов: Н.Х. Гартнер, Б.Л. Голден и Р.Т. Вонг в работе [106] фокусировались на методах оптимизации в анализе транспортных систем и, в частности, обсуждали проблемы сетевого анализа ТЛС, проблемы сетевого дизайна ТЛС, проблемы управления сетью ТЛС. М.Я. Постан, В.Е. Николайчук в своих работах [65, 68] использовали математические модели многоэтапной транспортной задачи, методы математического программирования для решения задач оптимизации в транспортно-складской логистике. Р. Борндорфер, М. Гротсчел, А. Лобель в [94] применяли методы дискретной математики для планирования структуры ТЛС. Л.Б. Миротин, А.Г. Некрасов, П.В. Степанов, П.Г. Трегубов [59] проводили исследования с целью повышения эффективности грузовых перевозок на основе создания устойчивой ТЛС модульного типа для высокоскоростной обработки и доставки грузов. В.М. Беляев, Л.Б. Миротин, А.Г. Некрасов, А.К. Покровский изучали задачу управления процессами в ТЛС [7].

Важным направлением развития математических методов исследования ТЛС является разработка методов прогнозирования. Например, Д.Н. Сидоров предложил оригинальный подход к построению прогноза по временным рядам с помощью преобразования Гильберта-Хуанга [72, 142]. О.М. Китурко, М.А. Матальщ-

кий в работе [33] исследовали стохастическую модель прогнозирования доходов в ТЛС «производитель – склады – пункты реализации товаров». Модель построена на использовании замкнутой НМ-сети (Howard – Matalytski) массового обслуживания с доходами и заявками одинакового типа. Для решения задач выполнена программная реализация на основе метода динамического программирования Беллмана. При этом если известны ожидаемые доходы систем сети в любой момент времени, можно решать широкий ряд различных задач оптимального управления [57]. Решению специальных задач оптимального управления посвящены работы [75, 74].

В работе [76] М.Е. Степанцова предлагает динамическую модель развития транспортной сети. При построении основной идеей модели является введение для каждого вида товаров величины, названной потенциалом. Потенциал численно характеризует потребность в данном товаре, существующую в данном узле транспортной сети. Именно разность потенциалов между узлами и создает в модели потоки товаров. Поэтому данный подход можно назвать «электромагнитной» моделью по аналогии с «гравитационными» моделями, уже достаточно давно использовавшимися в моделировании процессов с участием человека, а в последнее время применявшимися и для моделирования транспортных сетей [86].

В работе [81] И.Е. Агуреевым и В.М. Тропиной приведен вывод математической модели логистической системы, состоящей из двух складов и потока автомобилей, выполняющих перевозки между ними. Склады относятся к разным уровням логистической цепочки. Модель записывается в виде диссипативной системы обыкновенных дифференциальных уравнений (ОДУ). Показаны области применения модели и набор возможных решаемых задач. Приведены основные результаты предварительного исследования свойств модели.

Решению задач оптимизации, возникающих в ТЛС, посвящено большое количество работ отечественных и зарубежных ученых. Можно отметить следующих авторов: В.С. Лукинский [55–54], П.А. Козлов [37–38], А.С. Беленький [6],

В.С. Лубенцова [50], Н.А. Атрохов [4], А.П. Кожин, В.Н. Мезенцев [34], Д.Дж. Бауэрсокс [5] и др.

Одной из наиболее известных задач является задача о построении оптимального в том или ином смысле маршрута. Так, Э.Х. Гимади, Н.И. Глебовым, А.И. Сердюковым в работе [16] предложен вероятностный алгоритм решения классической задачи «коммивояжера». С.И. Сергеев также исследовал задачу коммивояжера [69–71], при котором предложен алгоритм точного решения задачи на основе метода ветвей и границ. В работе [84] приведено сравнение между двумя приближенными методами: жадным алгоритмом и алгоритмом с прогнозом, а в [85] разработаны модификации жадного алгоритма на основе перемещения в ближайший город; в [24] предложен эмпирический алгоритм решения задачи коммивояжера, основанный на последовательной декомпозиции задачи, приближенные алгоритмы отыскания на графе двух непересекающихся маршрутов с максимальным суммарным весом ребер [17], и т.д.

Как и задача об оптимальном маршруте, задача об оптимальном размещении логистических объектов вызывает большой интерес научного сообщества. Первой изученной задачей, по-видимому, была задача Лаунхардта о нахождении пункта оптимального размещения отдельного промышленного предприятия относительно источников сырья и рынков сбыта продукции. Для решения этой задачи В. Лаунхардт разработал метод весового (или локационного) треугольника. Следующий существенный шаг вперед сделал А. Вебер, введя в анализ новые факторы размещения производства в дополнение к транспортным издержкам и ставя более общую оптимизационную задачу по минимизации общих издержек производства, а не только транспортных [12]. На основе вышеупомянутых результатов В. Кристаллер создал теорию центральных мест, объясняющих количество, размер и местоположение населенных пунктов в городской системе [97].

Позднее появилось большое количество работ, посвященных данной задаче. Выделим работы М. Хаймовича и Т.Л. Маньянти [110], Ф. Моргана и Р. Болтона [128], В.С. Михалевича, В.А. Трубина, Н.З. Шора [60], В. Вертера [143].

Отметим, что при решении данной задачи необходимо учитывать ряд различного рода требований [103]: топография; расходы на строительство в данном районе; площадь земли, выделяемой под строительство объекта; расстояние от поставщиков сырья и материалов; расстояние от мест сбыта производимой продукции; учет конкурирующих и кооперирующих предприятий.

При решении задачи о размещении логистических объектов возникает необходимость организации средств коммуникаций (например, дорог, телекоммуникационных сетей и информационных каналов и т.п.), прокладка которых производилась бы с минимально возможными затратами. При этом проблема приводит к решению задачи о построении минимального остовного дерева или задачи Штейнера [113]. Построение минимального остовного дерева в вычислительном плане является относительно несложным, известны алгоритмы (Прима, Краскала, Борувки и др. [41]), а для решения задачи Штейнера в достаточно простых случаях можно применяться алгоритм Мелзака [124, 125] и его модификации [25].

Наибольшее развитие получило решение задачи Штейнера на графах. В работе [87] представлена задача Штейнера на ориентированных графах, при которых стоимостью всех дуг равна единице. В работе Лотарева Д.Т. [49] исследуется задача Штейнера на плоскости с евклидовой метрикой, в данной работе отмечено совпадение структуры деревьев на плоскости и на графе, поэтому решение задачи на графе принимается как решение задачи на плоскости. В статье [48] представлено решения задачи Штейнера, в которой для определения места расположения точек Штейнера производится локальный поиск.

Из представленного выше обзора исследований видна картина применения математических моделей в транспортной логистике. Ясно, что проблеме оптимизации транспортных перевозок, проектирования и организации функционирования транспортных сетей (оптимизации транспортно-логистических систем) уделяется большое внимание. Можно утверждать, что успешное применение построенных в рассматриваемых работах моделей доказывает необходимость использова-

ния математического моделирования при решении транспортно-логистических задач.

Несмотря на то, что к настоящему времени с помощью математического моделирования, решено большое количество проблем в транспортной логистике с высокой эффективностью, осталось еще достаточно много нерешенных задач, поскольку они имеют свою специфику и требуют существенных изменений в отлаженных алгоритмах. Нетрудно видеть, что транспортно-логистические задачи характеризуются незавершенностью, неполнотой, отсутствием глубоких прикладных разработок. Для того чтобы обозначить указанные недостатки, в следующем разделе более подробно рассмотрим наиболее известные численные методы, которые применяются при решении ТЛЗ.

1.2. Обзор численных методов для решения задачи маршрутизации, задачи об оптимальном размещении логистических объектов, задачи об упаковке и покрытии

1.2.1. Задача маршрутизации

Задачи маршрутизации – класс задач комбинаторной оптимизации ТЛС, сохраняющий свою актуальность до настоящего времени, который имеет большое значение и широкое применение.

Такая задача связана с задачей о нахождении кратчайшего пути от заданной начальной вершины до заданной конечной вершины. Следующие задачи являются непосредственными обобщениями поставленной: для заданной начальной вершины s найти кратчайшие пути между s и всеми другими вершинами; найти кратчайшие пути между всеми парами вершин.

Эти задачи применялись для решения различных прикладных задач [126], например,

- транспортное планирование: определение маршрута между местами, который обеспечит минимальное время передвижения с учетом скоростных и весовых ограничений;

- проектирование систем коммуникаций: например, прокладка линий связи (радио, телевидение, интернет, телефония и др.) между начальным коммутационным оборудованием и конечными пользователями, обеспечив минимальные затраты.

Обычно используются сетевые модели для решения задач маршрутизации [53], при этом сетевая модель – граф, состоящий из множества дуг и узлов. Каждая дуга может иметь направление и весовое значение, характеризующее расстояние, время, или стоимость доставки груза до соответствующего узла, а узлы характеризуют логистические центры (например, населенные пункты, перекрестки или остановочные пункты...).

С помощью этого метода, чтобы определить оптимальный маршрут, необходимо выбрать ту последовательность дуг, сумма весов которых будет минимальной. На рис. 1.1 показан пример сетевой модели, причем u_i ($i = \overline{1,6}$) – некоторые города, u_1 – поставщик, u_6 – потребитель, t_{ij} – время передвижения транспорта из i -го в j -й город.

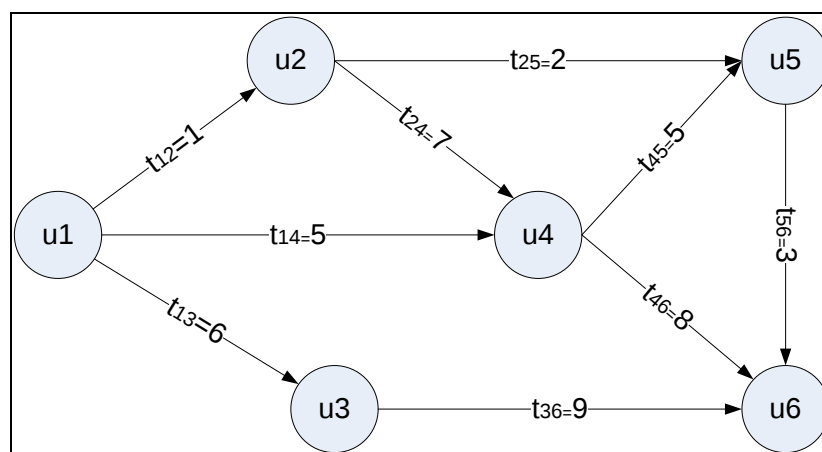


Рис 1.1. Сетевая модель (ориентированный граф)

Легко видеть, что время доставки груза будет наименьшим на маршруте $u_1 u_2 u_5 u_6$, который на первый взгляд является наиболее протяженным.

Рассмотрим наиболее популярные алгоритмы для решения предложенной задачи: алгоритм Дейкстры, метод Форда – Мура – Беллмана для случая общей матрицы весов, метод Флойда (для произвольной матрицы весов), метод ветвей и

границ, волновой алгоритм Ли. Кроме того, представим обзор современного состояния исследований.

Алгоритм Дейкстры [18,41] – алгоритм на графах, изобретенный нидерландским ученым Э. Дейкстрой в 1959 году [102], который находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без ребер отрицательного веса и широко применяется в программировании и технологиях передачи данных, например, его используют протоколы маршрутизации OSPF и IS-IS.

В общем случае этот метод основан на приписывании вершинам временных пометок, причем пометка вершины дает верхнюю границу длины пути от вершины s к текущей вершине. Эти пометки (их величины) постепенно уменьшаются с помощью некоторой итерационной процедуры, и на каждом шаге итерации точно одна из временных пометок становится постоянной. Последнее указывает на то, что пометка уже не является верхней границей, а дает точную длину кратчайшего пути от s к рассматриваемой вершине.

В соответствии с алгоритмом во множестве всех вершин V графа G выделяется подмножество вершин $S \in V$, до которых расстояние рассчитывается на данный момент.

Алгоритм поиска кратчайшего пути описывается следующим образом:

Шаг №1. Выбирается начальная вершина u_0 , именуемая «исток».

Шаг №2. Определяется вершина $u \in V - S$, до которой суммарное расстояние от «истока» u_0 является минимальным.

Шаг №3. Вершина u добавляется в подмножество $S \in V$.

Шаг №4. Если множество $Q = V - S$ является пустым, то дерево кратчайших путей построено и строится искомый маршрут. Если $Q \neq \emptyset$, то переходим к шагу №2.

Алгоритм Дейкстры является жадным алгоритмом (алгоритмом минимальной стоимости), на каждом шаге которого выбирается наименьшее расстояние из

всех возможных. В работе [41] доказано нахождение дерева кратчайших путей данным алгоритмом.

Для данного алгоритма, достоинством является гарантия нахождения оптимального маршрута. Но имеется недостаток, который является неспособностью адекватно решать задачу на графе, содержащем ребра с отрицательным весом.

Метод Форда – Мура – Беллмана для случая общей матрицы весов. Алгоритм Дейкстры применим лишь в том случае, когда вес дуги $C_{ij} \geq 0$ для всех i и j . Однако если матрица C является матрицей стоимостей, то дуги, приносящие доход, должны иметь отрицательные стоимости. В этом случае для нахождения кратчайших путей между вершиной s и всем другими вершинами можно воспользоваться методом, который был первоначально предложен в середине 50-х годов Фордом [105], Муром [127] и Беллманом [92].

Этот метод также является итерационным и основан на пометках вершин, причем в конце k -й итерации пометки равны длинам тех кратчайших путей (от s ко всем остальным вершинам), которые содержат не более $k+1$ дугу. В отличие от алгоритма Дейкстры никакая из пометок во время этого процесса не рассматривается как окончательная.

Метод Флойда (для произвольной матрицы весов). Алгоритм нахождения длин кратчайших путей между всеми парами вершин во взвешенном ориентированном графе, работает корректно, если в графе нет циклов отрицательной величины, а в случае, когда такой цикл есть, позволяет найти хотя бы один такой цикл.

Метод ветвей и границ. Общий алгоритмический метод для нахождения оптимальных решений различных задач оптимизации, особенно дискретной и комбинаторной. По существу, метод является вариацией полного перебора с отсеком подмножеств допустимых решений, заведомо не содержащих оптимальных решений.

Представим общую идею этого метода [58, 109] для рассматриваемой задачи. Пусть функция $T(X)$, $X \in H$, где H – некоторое конечное множество. Требу-

ется минимизировать функцию T . При этом для определения $\min T(X)$ необходимо выполнить две процедуры: ветвление и нахождение оценок (границ) функции $T(X)$.

1. Процедура ветвления (разбиение множества H на подмножества). Данная процедура состоит в разбиении множества допустимых значений переменной X на подобласти (подмножества) меньших размеров, в результате ветвления образуется дерево, узлами которого являются подмножества множества H . Множество $H^0 = H$ некоторым способом разбивается на конечное множество непересекающихся подмножеств $H_1^1, H_2^1, H_3^1, \dots, H_r^1$.

2. Процедура нахождения оценок заключается в поиске верхних и нижних границ для решения задачи на подобласти допустимых значений переменной X . Согласно [40] оценка подмножества H_1^1 не меньше оценки множества H^0 ($H_1^1 \subset H^0$), таким образом $\min_{X \in H_1^1} T(X) \geq \min_{X \in H^0} T(X)$. Тогда из всех подмножеств $H_1^1, H_2^1, H_3^1, \dots, H_r^1$ выбирается такое, оценка которого является наименьшей $\min\{\zeta(H_1^1), \dots, \zeta(H_r^1)\}$, другие подмножества исключаются из дальнейшего рассмотрения. Если X' принадлежит некоторому H_k^1 ($k = \overline{1, r}$) и $T(X') = \zeta(H_l^1) \leq \zeta(H_k^1)$, то X' является решением задачи. В противном случае поиск продолжается.

На любом j -ом шаге ($j \geq 1$) ветвления образуется дерево поиска решения (рис. 1.2).

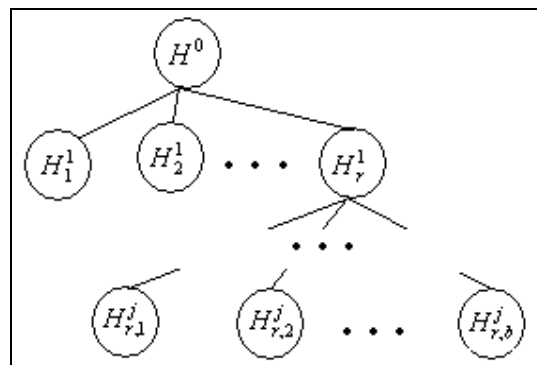


Рис. 1.2. Дерево поиска решения

При использовании этого метода можно получить наилучшее решение за конечное число итераций ветвления, но главная трудность заключается в том, что необходима разработка правил разбиения и нахождения оценок для каждой решаемой задачи.

Алгоритм Ли. Для решения задачи об оптимальном маршруте также можно использовать алгоритм Ли [62]. Алгоритм Ли (волновой алгоритм, алгоритм волновой трассировки) – алгоритм поиска пути, алгоритм поиска кратчайшего пути на планарном графе, принадлежит к алгоритмам, основанным на методах поиска в ширину.

Данный алгоритм разработан для трассировки печатных плат [20], поэтому он используется эффективно только для построения пути в евклидовой метрике (например, на плоскости).

Чтобы построить решение используется прямоугольная система координат, где каждый узел сетки изначально задается числом «0», обозначающее ноль Шагов от источника до данной точки. Также узел может обозначать некоторый непроходимый барьер.

На рис. 1.3 представлено «начальное состояние» некоторой местности, где серые квадраты – непроходимые барьеры.

0	0	0	0	0
0	0		0	0
0	0		0	0
0	0			0
0	0	0	0	0

Рис. 1.3. «Начальное состояние» местности

Работу алгоритма можно описать следующим образом:

Шаг №1. Выбирается определенный источник X_0 волны как точка в нужном узле, который не находится в области барьеров, и имеет значение «0».

Шаг №2. Из источника выпускается волна, распространение которой будет в четырех направлениях (вверх, вниз, влево, вправо). Производится проверка значения выбранного узла. Если данное значение равно нулю, то ему присваивается номер шага n . В других случаях, когда узел является барьером или значение которого имеет ненулевое число, то шаг в данном направлении не производится.

Шаг №3. На рассматриваемом шаге n проверяется возможность осуществления следующего шага для каждого узла, т.е. ищется «0» в соседних узлах. Если найден узел с нулевым числом, то выполняется шаг №2. Если не найден, осуществляется выбор следующего узла с номером шага n . Процесс распространения волны завершается, если после проверки всех узлов не удалось выполнить следующий шаг.

Шаг №4. Строится маршрут до выбранной конечной точки X_n (не совпадает с начальной точкой). Выполняет нахождение всех соседних узлов, значение которых меньше выбранного узла, и переходим в данный узел. Процесс повторяется до тех пор, пока не будет достигнута точка X_0 со значением «0» в данном узле.

Таким образом, найденный маршрут является оптимальным. В этом случае задача может иметь несколько решений, которые показаны на рис. 1.4 (например, если конечная точка располагается в правом нижнем узле со значением «8»), на котором представлен результат работы алгоритма.

0	1	2	3	4
1	2		4	5
2	3		5	6
3	4			7
4	5	6	7	8

Рис. 1.4. «Конечное состояние» местности

Существует модификация алгоритма, в котором «волна» распространяется не только по 4, но и по 8 направлениям, т.е. движение осуществляется, как по вертикали и горизонтали, но еще и по диагонали.

При использовании этого метода эффективно решается задача о прокладке пути в однородной среде (печатной плате) и в городской дорожной сети при условии однотипного покрытия дорог и пропускной способности, но не позволит адекватно решать задачу в неоднородной среде.

1.2.2. Задача об оптимальном размещении логистических объектов

Для построения решения этого класса задач разработаны и продолжают разрабатываться эффективные методы и подходы, например, эвристические алгоритмы, методы эволюционного моделирования, итеративные методы последовательного размещения, методы целочисленного и динамического программирования, методы искусственного интеллекта, комбинированные методы.

Теперь рассмотрим более подробно наиболее распространенные.

Метод «Хакими». Этот алгоритм позволяет найти точное расположение абсолютного центра и значение его разделения (радиус) [111]. Он очень прост и для неориентированного графа состоит всего из двух шагов.

Шаг №1: Для каждого ребра e_{ij} найти точку с наименьшим разделением.

Шаг №2: Из всех этих точек выбрать точку с наименьшим разделением — она и будет абсолютным центром.

Метод «определения центра тяжести». Чтобы решить проблему оптимального расположения одного логистического объекта, часто используется метод «определения центра тяжести» [13, 52]. Метод аналогичен определению центра тяжести физического тела, который основан на нахождении координат расположения склада.

$$M_x = \frac{\sum_{i=1}^n Q_i x_i}{\sum_{i=1}^n Q_i}, \quad M_y = \frac{\sum_{i=1}^n Q_i y_i}{\sum_{i=1}^n Q_i}, \quad (1.1)$$

где M_x, M_y — координаты логистического центра, Q_i ($i = \overline{1, n}$) — объем потребления i -го клиента, x_i, y_i — координаты расположения i -го клиента.

Кроме того, чтобы найти оптимальное место логистического центра используется модификация метода «центра тяжести» [52] на основе поиска центра равновесия транспортных издержек. В этом случае координаты склада определяются как

$$M_x = \frac{\sum_{i=1}^n T_i Q_i x_i}{\sum_{i=1}^n T_i Q_i}, \quad M_y = \frac{\sum_{i=1}^n T_i Q_i y_i}{\sum_{i=1}^n T_i Q_i}, \quad (1.2)$$

где M_x, M_y – координаты логистического центра, Q_i ($i = \overline{1, n}$) – объем потребления i -го клиента, T_i ($i = \overline{1, n}$) – транспортный тариф на доставку товара i -му клиенту, x_i, y_i – координаты расположения i -го клиента.

С помощью этого метода можно построить решение, но алгоритм работает только в евклидовой метрике, и не учитывает особенность рельефа местности и наличие естественных преград (т.е. расстояние от потребителей до места расположения логистического центра измеряется только по прямой), недостаточность развитой сети дорог, не обеспечивает информацией о точном расположении склада.

Метод k-средних. Для решения задачи о размещении нескольких логистических центров широко применяется метод k-средних, который является популярным методом кластеризации. Идея метода k-средних [56, 90] заключается в определении расположения каждого центра в каждой логистической зоне с помощью метода «определения центра тяжести» (количество которых задано заранее). При этом используя, например, формулы (1.1) или (1.2), минимизации целевой функции C , вычисляемой по формуле

$$C = \sum_{k=2}^n \sum_{i=0}^{m(k)} d(A(i, k), M(k)), \quad (1.3)$$

где n – количество логистических центров (также количество зон), $m(k)$ – количество потребителей в k -ой логистической зоне, $A(i, k)$ – i -ый потребитель k -ой логистической зоны, $M(k)$ – «центр тяжести» k -ой логистической зоны,

$d(A(i,k), M(k))$ – расстояние между точкой «центра тяжести» M и потребителем A в соответствующей логистической зоне.

Расстояние $d(A(i,k), M(k))$ вычисляется следующим образом:

$$d(A(i,k), M(k)) = \sqrt{(A_x(i,k) - M_x(k))^2 + (A_y(i,k) - M_y(k))^2}, \quad (1.4)$$

где $A_x(i,k)$ и $A_y(i,k)$ – координаты i -ого потребителя k -ой логистической зоны, $M_x(k)$ и $M_y(k)$ – координаты «центра тяжести» k -ой логистической зоны.

Алгоритм разбиения области на логистические зоны методом k -средних можно описать следующим образом:

Шаг №1. Выбирается расположение начальных точек $M(k)$, $k = \overline{2, n}$. Количество логистических зон n известно заранее. Выбор начальных центроидов может осуществляться следующим образом:

- выбор k наблюдений для максимизации начального расстояния;
- случайный выбор k наблюдений;
- выбор первых k наблюдений.

В результате каждый объект назначен определенному кластеру.

Шаг №2. Вычисляется расстояние между точками $M(k)$ и потребителями A .

Шаг №3. Определяется принадлежность потребителей A логистическим зонам обслуживания. Потребитель A принадлежит зоне k , если расстояние от A до $M(k)$ является наименьшим по сравнению с расстоянием от A до остальных $M(j)$, $j = \overline{2, n}$, $j \neq k$.

Шаг №4. Вычисляется «центр тяжести» в каждой логистической зоне, используя формулы (1.1) или (1.2). Определяется значение целевой функции (1.3). Процесс повторяется с шага №2 до тех пор, пока целевая функция не перестанет улучшаться.

Достоинствами этого метода являются простота и быстрота использования, понятность и прозрачность алгоритма.

К недостаткам метода можно отнести следующее: алгоритм очень чувствителен к выбросам, которые могут искажать среднее; при огромных базах данных алгоритм может работать медленно; необходимо задание количества логистических зон, что соответственно влияет на значение целевой функции; невозможно найти (без проведения сегментации) оптимальное количество логистических зон, при котором целевая функция достигает минимума.

Метод FOREL (Формальный элемент). Алгоритм кластеризации построен на идее объединения в один кластер объектов в областях их наибольшего сгущения, который основывается на переопределении «центра тяжести» для потребителей, попавших в окружность заранее заданного радиуса, и исключает их из дальнейшего рассмотрения, а также применяется для решения задачи размещения некоторых логистических центров [22].

Таким образом, к достоинствам метода можно отнести следующее: точность минимизации функционала качества (при удачном подборе параметров алгоритма), простота построения решения, сходимость алгоритма.

К недостаткам метода можно отнести неустойчивость алгоритма (зависимость от выбора начального объекта), невозможность определения (без проведения сегментации) необходимого радиуса, при котором определяется оптимальное разбиение, неприменимость алгоритма при плохой разделимости выборки на кластеры.

1.2.3. Задача об упаковке кругов с одинаковыми радиусами

Задача построения оптимальных упаковок [39] является одной из классических проблем комбинаторной геометрии. Она представляет интерес, как с теоретической точки зрения, так и в связи с очевидными многочисленными приложениями. Одним из важных ее приложений в транспортной логистике является размещение набора торговых либо логистических узлов, чтобы обеспечить максимальный охват области, где проживают потребители. При этом узлы, расположенные в центрах кругов упаковки, должны обслуживать потребителей, находящихся не только внутри «своего» круга, но и частично за его пределами.

История изучения данной проблемы весьма долгая, содержательная [132, 137, 144], достаточно указать, что одно из известных утверждений, относящихся к упаковке шаров в трехмерном евклидовом пространстве (о том, что среди всех упаковок шаров равного размера в трехмерном пространстве наибольшую среднюю плотность имеет гранецентрированная кубическая упаковка и упаковки, равные ей по плотности), носит название «гипотеза Кеплера» и было сформулировано более 400 лет назад.

В литературе встречаются различные постановки данной проблемы: от вариантов задачи о рюкзаке [116] до рассмотрения ее в абстрактном n -мерном пространстве [46]. Отметим, что практически всегда (кроме некоторых специальных случаев) задача об упаковке является NP-полной и, соответственно, проблема построения эффективных численных алгоритмов ее решения является чрезвычайно актуальной [23].

По-видимому, наиболее популярной из данного класса проблем является задача об оптимальной упаковке двумерных шаров (кругов) равного радиуса в ограниченное множество с гладкой или кусочно-гладкой границей. Так, в работах [96, 123, 138] рассматривалась задача о построении наилучшей упаковки в квадрат при числе кругов от 1 до 200. При этом в случае, когда количество элементов упаковки невелико (до 36 включительно), задача решена аналитически, т.е. строго доказано, что полученные результаты являются наилучшими. В работах [107, 108, 121] рассматривалась задача о наилучшей упаковке в круг, получены результаты при числе элементов упаковки до 81. В статье [93] исследована задача упаковки заданного числа кругов единичного радиуса в круг, прямоугольник, треугольник с минимизацией размеров этих фигур (радиуса, площади, периметра, ширины и т.п.). В статье Г.Н. Яськова [88] предлагается метод упаковки большого числа равных кругов на основе математической модели, в которой радиусы всех кругов являются переменными.

Задача об упаковке кругов разных заданных радиусов с максимизацией их числа или веса, либо минимизацией пустот исследована в [117, 118], предложена

эвристика на основе пространственного поиска для задачи упаковки кругов равного и разного радиусов в различные множества [119, 120]. Наконец, содержательна и оригинальна задача об упаковке кругов разных радиусов, когда меньшие могут располагаться внутри больших (telescoping), была исследована в статье [131].

В своей работе [108] Р.Л. Грехем, Б.Д. Любачевский, К.Дж. Нурмела, П.Р.Дж. Остергард рассмотрели проблему упаковки конгруэнтных кругов внутри большего круга с единичным радиусом. Авторами представлены два метода, основанные на попарной максимизации минимального расстояния между n точками, которые распространились в единичном круге с центром в начале координат. Обозначим это минимальное расстояние целевой функцией

$$d(S) = \min\{\|s_i - s_j\| : 1 \leq i < j \leq n\}, \quad (1.5)$$

где $S = \{s_1, s_2, \dots, s_n\}$ является множеством точек в единичном круге. Тогда задача упаковки может быть сформулирована как задача оптимизации

$$d(S) \rightarrow \max. \quad (1.6)$$

Первый из предложенных методов основан на оптимизации силами отталкивания. Сложность при попытке применить эффективный алгоритм локальной оптимизации заключается в том, что (1.5) не является гладкой и, кроме того, большинство из первых производных функций равны нулю почти во всей допустимой области.

Для преодоления таких трудностей авторы аппроксимировали исходную задачу (1.6) минимизацией целевой функции:

$$\sum_{1 \leq i < j \leq n} \left(\frac{\lambda}{\|s_i - s_j\|^2} \right)^m, \quad (1.7)$$

где λ является подходящим масштабным коэффициентом [98], которая может рассматриваться как функция потенциальной энергии, когда есть силы отталкивания между точками. Параметр m контролирует интенсивность сил отталкивания. Аппроксимация (1.7) будет оптимальной в том случае, когда m стремится к бес-

конечности, только тогда кратчайшее расстояние между точками становится наименьшим.

Функция стоимости (1.7) является гладкой всюду, кроме случая, где две точки совпадают. На практике ситуацию, когда две точки совпадают, практически получить невозможно, поскольку такие места в пространстве решений окружены очень высокими потенциальными барьерами.

Для формирования упаковки сначала находим локальный оптимум (1.7), используя относительно малое значение m , например, $m = 80$, после чего значение m увеличивается, и процесс оптимизации повторяется. В начале каждого этапа оптимизации авторы использовали простой метод скорейшего спуска с возвратом линии Гольдштейн–Армихо. Когда градиент становится достаточно малым, переходили к модифицированному методу Ньютона, чтобы получить большую (в данном случае квадратичную) скорость сходимости.

Вместо того чтобы делать расчеты в декартовых координатах, авторы преобразовали координаты для того, чтобы заменить исходную задачу задачей безусловной оптимизации. Обозначенные трансформированные координаты точки $x = (x_1, x_2)$ обозначены с помощью $x = (\tilde{x}_1, \tilde{x}_2)$. Преобразование определяется следующим образом:

$$x_1 = \sin(\tilde{x}_1)\sin(\tilde{x}_2), \quad x_2 = \cos(\tilde{x}_1)\sin(\tilde{x}_2).$$

Теперь получена задача безусловной оптимизации в преобразованных переменных. При этом можно использовать алгоритм условной оптимизации, но для того, чтобы ограничения были линейными, требуется выполнить некоторые дополнительные преобразования.

Во втором алгоритме упаковки авторами использована модель «жесткие диски» или «бильярд» из вычислительной физики. Рассмотрим множество S из n точек, которые распространились в единичном круге. Пусть $d = d(S)$ – минимальное попарное расстояние между точками, как определено в (1.5). Если мы проведем круги диаметром d вокруг каждой точки в S , то круги не пересекутся. Теперь конфигурация S ослабляется, и круги, которые также можно рас-

смаатривать как жесткие диски, двигаются хаотично, не накладываясь друг на друга (из-за ограничений), а также не выходят из большого круга радиуса $1 + d/2$ (центры кругов не выйдут из единичного круга).

В процессе движения дисков для каждой конфигурации подсчитывается значение (1.5), после чего определяется максимум (1.6). Если $d = 0$, то рассматривается пространство всех возможных конфигураций. Далее постепенно увеличивается d . Тогда подмножество конфигураций, постепенно сокращается. В результате, подмножество сводится к одной конфигурации (до преобразования эквивалентности вращения), и, таким образом, упаковка достигается в стационарном состоянии системы. Эта конфигурация является локальным максимумом функции (1.5).

Эффект хаотического движения получается, когда диски испытывают упругие столкновения при ударах, начиная со случайно выбранной начальной конфигурации $d = 0$. Ошибки округления в вычислениях работают в качестве дополнительного фактора рандомизации.

Вычисления осуществляются в эффективном, событийно-ориентированном виде, когда алгоритм переходит непосредственно от одного столкновения к другому, и не тратится время на вычисления для дисков, не участвующих в столкновении. Более подробное описание алгоритма дано в [122].

При рассмотрении задачи об упаковке ученые не ограничивались случаем, когда элементы последней являются кругами. Так, в работе [89] рассмотрена задача об упаковке прямоугольников (с возможностью их поворота) в треугольники, в [15] – задача об упаковке эллипсов в прямоугольную область.

Отметим, что библиографию можно было бы продолжить, существует огромное заслуживающих внимания публикаций. При этом, хотя большинство книг и статей посвящены исследованию задачи об упаковке в евклидовом пространстве, тем не менее, существуют и исключения. В работах [95, 101] представлено решение задачи расстановки сфер равных радиусов в двухмерных, трехмерных пространствах постоянной кривизны (гиперболическом и

эллиптическом) так, чтобы их плотность была наибольшей. Кроме этого, данная проблема исследовалась в цикле работ Дж. Жирмая (J. Szirmai): в [139] описан способ построения оптимальной упаковки шаров для черепицы Коксетера (Coxeter tiling) в гиперболическом трехмерном пространстве, в [140] продолжен метод, основанный на проективной интерпретации гиперболической геометрии, в [141] расширена задача нахождения наиболее плотной геодезической упаковки шаров для 3-мерных геометрий Терстона (Thurston geometries).

1.2.4. Задача о покрытии равными кругами

Задача о покрытии состоит в том, что в ограниченной области размещаются различные геометрические объекты так, чтобы покрываемая область полностью лежала внутри объединения последних. Данная задача находит широкое применение при решении прикладных проблем в различных отраслях деятельности человека: в технике, экономике, логистике и др. К типичным их представителям можно отнести задачи размещения станций различного назначения: сотовой связи, беспроводного Интернета, скорой помощи, банковских терминалов и т.д. [8, 14, 91, 104]; задачи построения сетей искусственных спутников Земли [61, 67]; задачи выбора оптимальной мощности универсальных двигательных установок малой тяги [9, 10]; задачи проектирования энергоэффективной системы мониторинга протяженных объектов беспроводными сенсорными сетями [1–3, 78] и т.д.

С точки зрения теории оптимизации можно выделить две основные постановки задачи о покрытии: минимизировать площадь перекрытий при заданном числе покрывающих объектов либо минимизировать количество объектов при заданных параметрах. При этом, пожалуй, наибольший интерес исследователей вызывает специальный класс задач, в которых в качестве покрывающих фигур в двухмерном пространстве выступают круги, а в трехмерном – шары равного радиуса. Так, в работах [8, 42] предложены алгоритмы нахождения оптимальных покрытий конгруэнтными кругами и указаны решения для покрытия квадрата n кругами при $n \leq 15$, а в [26] представлены оптимальные конфигурации покрытий единичных круга, квадрата и правильного треугольника для $n \leq 10$. В [45, 44] ис-

следовались задачи покрытия окружностями равного радиуса квадрата и круга с помощью метода построения наилучших чебышевских n -сетей, причем для малых n доказано, что представленные результаты являются оптимальными.

Для покрытия некоторых классов плоских односвязных множеств равными кругами предложены алгоритмы, основанные на использовании квазидифференцируемости целевой функции [112], эвристических [64] и метаэвристических методов [79, 82, 83], методов целочисленной [28] и непрерывной [30, 130] оптимизации, представлены результаты для небольшого количества кругов ($n \leq 36$). Для большего количества равных кругов в работе [136] разработана модификация метода возможных направлений для нахождения локального минимума и представлены оптимальные покрытия для $n \leq 72$, $n = 81$, $n = 100$.

Отметим, что большинство известных результатов получено для случая, когда покрываемая область является подмножеством евклидовой плоскости (пространства). В случае же неевклидовой метрики данная задача остается малоизученной.

Теперь рассмотрим конкретнее некоторые методы для решения ряда этих задач.

В 2010 году в своей работе [14] предложены авторами Ш.И. Галиевом, М.А. Карповой численные алгоритмы оптимизации многократного покрытия ограниченного множества G на плоскости P равными кругами. Рассмотрены варианты, когда G – связное ограниченное множество на P или конечное множество точек из P . Положения центров кругов могут быть в произвольных точках множества G или плоскости P либо в некоторых из заданного множества точек. Рассматриваются задачи минимизации радиусов заданного числа кругов и минимизации числа кругов заданного радиуса.

В статье В.Ю. Кузнецова [43] рассматривается модификация задачи покрытия выпуклой области равными кругами [80] с учетом обхода препятствий.

1.3. Оптико-геометрический подход

Из проведенного обзора можно видеть, что использование математического моделирования для решения оптимизационных задач, возникающих в ТЛС, показало высокую эффектность. При этом, во-первых, сложность таких моделей зачастую не позволяет осуществить аналитическое построение точного решения, а, во-вторых, большинство моделей работают только с евклидовым расстоянием. Следовательно, актуальной задачей является разработка методики их численного решения с целью преодоления указанных недостатков.

В диссертационной работе предлагается использовать оптико-геометрический подход, базирующийся на аналогии между распространением световой волны в оптически неоднородной среде и отысканием минимума интегрального функционала, которая в свою очередь, является следствием физических законов Ферма и Гюйгенса.

Напомним, что фронтом световой волны называется геометрическое место точек, которые достигнуты световой волной за определенное время.

Согласно принципу Ферма свет в своем движении выбирает минимальный по времени путь, а в соответствии с принципом Гюйгенса каждая точка, которой достигает фронт световой волны, становится самостоятельным источником света. Таким образом, выпустив световую волну из точки a , можно зафиксировать «фотон», который первым достигнет точки b . На рис. 1.5 представлена иллюстрация механизма работы вышесказанных принципов.

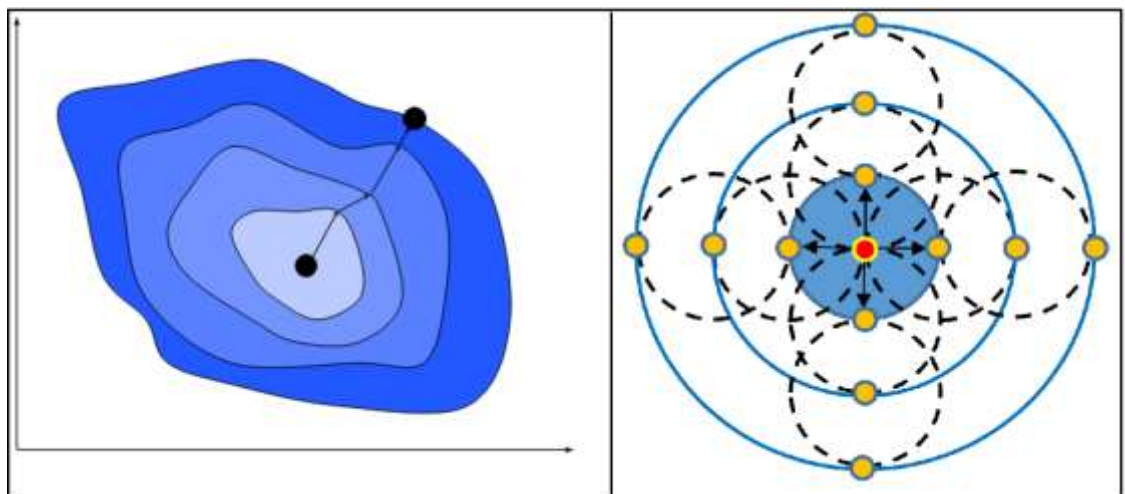


Рис. 1.5. Блок-схема алгоритма улучшения

Далее, двигаясь в попятном направлении по времени, можно восстановить траекторию его движения, которая и будет искомой кривой.

В работах [30, 31] предложен алгоритм конструирования волновых фронтов, инициированных из точки.

Недостатком данного подхода является отсутствие для него строгого математического обоснования, однако достоинства – естественность, простота, универсальность и удобство при решении прикладных задач.

В данной диссертационной работе продолжают исследования, представленные в [11, 26–28, 47, 115]. В указанных работах авторами на основе оптико-геометрического подхода предложены математические модели и численные методы решения следующих задач: определение оптимального (быстрейшего) маршрута между двумя точками; оптимальное размещение одного или нескольких логистических центров с сегментацией логистических зон обслуживания. Опишем их более подробно.

Пусть требуется определить оптимальный в некотором смысле маршрут передвижения из пункта A в пункт B , расположенных в области D . Тогда пункт A считается источником, и из него выпускается световая волна. При этом для каждой точки области фиксируется время ее достижения световой волной и фотон, который достиг данной точки первым. После того, как свет заполнил заданную область, из пункта B восстанавливается траектория движения волны, т.е. производится процесс распространения световой волны в обратном направлении по времени.

На основе вышеизложенного метода авторы предлагают алгоритмы решения задач оптимального размещения одного логистического центра и сегментации логистических зон обслуживания.

Для решения первой задачи используется алгоритм, суть которого заключается в определении точки в рассматриваемой области, суммарное время доставки товаров из которой до всех потребителей минимально.

Для второй задачи требуется произвести разбиение полигона обслуживания на зоны, каждая из которых обслуживается одним складом. При этом применяется метод, суть которого заключается в нахождении для каждого склада всех точек в рассматриваемой области, чтобы время достижения волны из соответственного склада до этих точек было минимальным. Тогда множество точек каждого склада образует его зону обслуживания.

При совместном использовании метода решения задачи оптимального размещения одного логистического центра и задачи сегментации логистических зон обслуживания можно решить задачу об оптимальном размещении нескольких логистических центров. Идея метода заключается в итеративном улучшении получаемого решения при помощи последовательной сегментации области на зоны, обслуживаемые одним центром, и нахождении оптимального расположения этого центра в соответствующей зоне. Генерация начальных положений повторяется несколько раз, в результате находятся локальные минимумы, среди которых выбирается лучший вариант.

Отметим, что в указанных алгоритмах расположение потребителей предполагалось точечным. В данном диссертационном исследовании предложенный подход перенесен на случай непрерывно распределенных потребителей.

Глава 2. Математические модели, численные методы для решения задач оптимизации транспортной логистики на основе оптико-геометрического подхода

В этой главе предложена многоэтапная технология исследования ТЛС. На первом этапе решаются задачи оптимального размещения инфраструктурных логистических объектов в предположении об отсутствии на территории исследуемого региона таких объектов (например, сотовые вышки определенного оператора, банкоматы конкретного банка и т.п.). Данные задачи посредством методов математического моделирования сведены к специальным модификациям двух известных математических задач: об упаковке и покрытии равными кругами ограниченных множеств в двумерных метрических пространствах с неевклидовыми метриками. На втором и третьем этапах решаются, соответственно, задачи об оптимальном размещении дополнительных логистических узлов в условиях кооперации и конкуренции и об организации системы коммуникаций между размещенными объектами. Предложены их математические модели, позволяющие учитывать особенности рельефа местности. Наконец, на последнем этапе рассматривается задача о поддержании коммуникаций в удовлетворительном состоянии на примере ремонта автомобильных дорог в условиях ограниченного бюджета. Далее опишем построенные математические модели и разработанные для их исследования численные методы.

2.1. О построении математических моделей

Пусть имеется некоторая ограниченная область, причем существуют отдельные зоны, строительство в которых и движение через которые запрещено. Предполагается, что потребителей достаточно много, поэтому по аналогии с механикой сплошной среды, при построении модели будем использовать феноменологический подход, в рамках которого исследуемый объект рассматривается целиком, без учета его внутренней структуры, и изучается как он реагирует на некоторую внешнюю силу в целом. Итак, пусть потребители распределены по территории непрерывным образом.

Требуется разместить заданное количество обслуживающих центров (ОЦ) так, чтобы, во-первых, обслужить максимально возможную долю рассматриваемой области, во-вторых, зоны обслуживания различных ОЦ не должны пересекаться, и, в-третьих, максимальное время доставки груза потребителю было одинаковым для всех ОЦ.

Построим математическую модель данной логистической задачи.

Пусть $X \subseteq \mathbb{R}^2$ – метрическое пространство; $D \subset X$ – заданная ограниченная область; $B_k \subset D$, $k=1, \dots, m$ – компактные множества, строительство в которых и движение через которые запрещено; n – количество ОЦ; $s_i = (x_i, y_i)$ – координаты ОЦ, $i=1, \dots, n$; $S_n = \{s_i\}$ – множество ОЦ; $0 \leq f(x, y) \leq \beta$ – непрерывная функция, задающая мгновенную скорость движения в каждой точке множества D . Причем $f(x, y) = 0 \Leftrightarrow (x, y) \in B_k$, $k=1, \dots, m$. Тогда вместо области D будем рассматривать замкнутое многосвязное множество

$$P = \text{cl} \left(D \setminus \bigcup_{k=1}^m B_k \right) \subset X \subseteq \mathbb{R}^2. \quad (2.1)$$

Здесь cl – оператор замыкания.

Расстояние между точками пространства X определим следующим образом:

$$\rho(a, b) = \min_{\Gamma \in G(a, b)} \int_{\Gamma} \frac{d\Gamma}{f(x, y)}, \quad (2.2)$$

где $G(a, b)$ – множество всех возможных непрерывных кривых, лежащих в X и соединяющих точки a и b . Иными словами, кратчайшим путем между точками будет кривая, на преодоление которой затрачивается наименьшее время.

Легко убедиться, что все аксиомы метрики выполнены. В частном случае, когда $f(x, y) \equiv 1$, имеем обычную евклидову метрику в пространстве размерности два, и тогда расстояние между точками a и b вычисляется, как:

$$\rho(a, b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}.$$

Требуется найти такое расположение $S_n^* = \{s_i^*\}$, при котором достигается максимальное значение

$$R^* = \min_{i=1,n} \min \left\{ \frac{\rho(s_i, (S \setminus \{s_i\}))}{2}, \tilde{\rho}(s_i, \partial P) \right\}. \quad (2.3)$$

Здесь ∂P – граница множества P , $\tilde{\rho}(s_i, \partial P) = \min_{x \in \partial P} \rho(s_i, x)$ – расстояние от точки до замкнутого множества.

Нетрудно видеть, что смысл выражения (2.3) состоит в том, чтобы максимизировать радиус равных кругов, которые можно разместить так, чтобы они пересекались друг с другом и с границей множества P только в точках ограничивающих их окружностей. Иначе говоря, решение поставленной логистической задачи эквивалентно решению задачи об упаковке кругов равного радиуса в многосвязное множество с метрикой (2.2).

Рассмотрим еще одну постановку: предположим, что требуется разместить заданное количество обслуживающих центров (ОЦ) так, чтобы, во-первых, все потребители были обслужены, во-вторых, максимальное время доставки груза потребителю было одинаковым для всех ОЦ, и, в-третьих, указанное время было минимальным.

Перейдем к математическому описанию данной логистической задачи.

Пусть $X \subseteq \mathbb{R}^2$ – метрическое пространство; $M \subset X$ – заданная ограниченная область с непрерывной границей; n – количество ОЦ; $S_i = (x_i, y_i)$ – координаты ОЦ, $i = 1, \dots, n$; $P_n = \{S_i\}$ – множество ОЦ; $0 < \alpha \leq f(x, y) \leq \beta$ – непрерывная функция, задающая мгновенную скорость движения в каждой точке множества M .

Требуется найти такое расположение $P_n^* = \{S_i^*\}$, при котором достигается минимальное значение

$$R_* = \max_{i=1,n} \tilde{\rho}(S_i, \partial M_i). \quad (2.4)$$

Другими словами, решение данной логистической задачи эквивалентно решению задачи о покрытии кругами равного радиуса области с метрикой (2.2).

2.2. Задача построения оптимальных упаковок равных кругов

Пусть имеются метрическое пространство X , равные круги C_i , $i=1, \dots, n$ с центрами $s_i = (x_i, y_i)$, радиусом R и замкнутое многосвязное множество (2.1). Необходимо найти такой вектор $s = (s_1, \dots, s_n) \in \mathbb{R}^{2n}$, который обеспечит размещение в области P заданного числа кругов максимального радиуса R .

В общем виде получим следующую задачу:

$$R \rightarrow \max \quad (2.5)$$

$$\rho(s_i, s_j) \geq 2R, \forall i = \overline{1, n-1}, \forall j = \overline{i+1, n} \quad (2.6)$$

$$\tilde{\rho}(s_i, \partial P) \geq R, \forall i = \overline{1, n} \quad (2.7)$$

$$s_i \in P, \forall i = \overline{1, n} \quad (2.8)$$

Здесь ∂P – граница множества P , $\tilde{\rho}(s_i, \partial P) = \min_{x \in \partial P} \rho(s_i, x)$ – расстояние от точки до замкнутого множества.

Целевая функция (2.5) максимизирует радиус упаковываемых кругов. Формула (2.6) обеспечивает непересечение внутренностей кругов. Формулы (2.7)–(2.8) гарантируют, что каждый круг полностью лежит внутри множества.

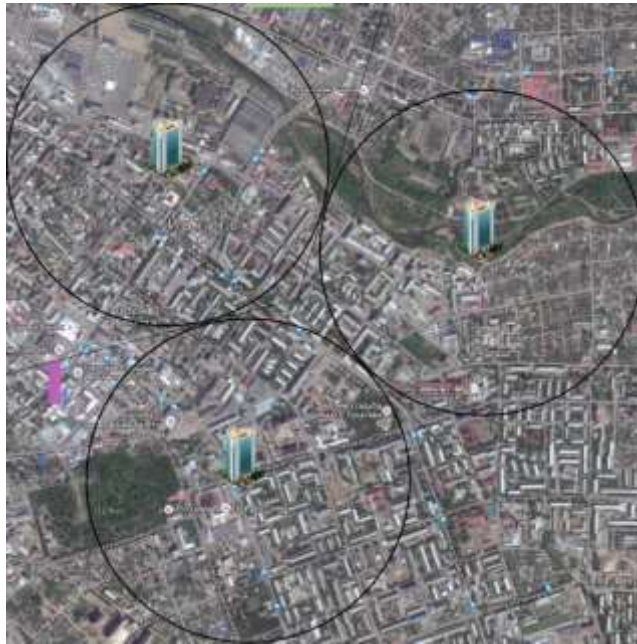


Рис. 2.1. Иллюстрация построения оптимальной упаковки равных кругов

Для любого вектора s , удовлетворяющего условиям (2.6)–(2.8) определим множества

$$P_i = \{p \in P : \rho(p, s_i) \leq \rho(p, s_j), \forall j = 1, \dots, n, i \neq j\}. \quad (2.9)$$

В литературе такие множества называют областями Дирихле [133] точек s_i во множестве P . Очевидно, что $P = \bigcup_{i=1}^n P_i$.

Решение поставленной задачи сводится к решению следующей последовательности подзадач:

1. Для каждого множества P_i найдем точку $\bar{s}_i \in P_i$ такую, что $\rho(\bar{s}_i, \partial P_i) = \max_{p \in P_i} \rho(p, \partial P_i)$.
2. Определим гарантированное значение радиуса, удовлетворяющего ограничениям (2.6)–(2.8), $R = \min_{i=1, \dots, n} \rho(\bar{s}_i, \partial P_i)$.
3. Для найденного вектора $\bar{s} = (\bar{s}_1, \dots, \bar{s}_n)$ переопределим множества P_i в соответствии с формулой (2.9).

Выполнение пунктов 1–3 производится до тех пор, пока изменяются координаты вектора $\bar{s}$.

На основе использования предложенного подхода для решения поставленной задачи разработаны следующие алгоритмы, при которых для решения первой подзадачи необходимо производить для каждой области P_i построение фронта световой волны, выпущенной с границы ∂P_i до момента времени, когда фронт вырождается в точку. Координаты последней и будут искомым решением $\bar{s}_i$. Для решения же третьей подзадачи требуется одновременно выпустить световые волны из точек $\bar{s}_i$ и определить те точки области P , которых две или более волны достигают одновременно. Такой алгоритм был разработан, и представлен ранее в работе [28].

Алгоритм «Волна с границы внутрь» (BorderWaveInside–BWI)

1. Граница области аппроксимируется замкнутой ломаной с узлами в точках $A_i, i = \overline{0, m}$ (исходные точки).

2. Из каждой пары точек A_i, A_{i+1} , вдоль нормали к соединяющему их отрезку откладываются отрезки $A_i B'_i$ и $A_{i+1} B''_i$ длиной $f(A_i)\Delta t$ и $f(A_{i+1})\Delta t$, соответственно. Здесь Δt – достаточно малый отрезок времени между соседними фронтами. Заметим, что новых точек получается в два раза больше, чем исходных. Пусть $\mathbf{B}$ – множество всех таких отрезков.

3. Если найдется пара отрезков $VW \in \mathbf{B}$, $YZ \in \mathbf{B}$ таких, что $W = Z$, то все точки начального фронта, лежащие между V и Y , исключаются из рассмотрения.

4. Строятся прямые $B'_i B''_i$, $i = \overline{0, m-1}$, проходящие через точки B'_i и B''_i .

5. Точки пересечения прямых $B'_i B''_i$ и $B'_{i+1} B''_{i+1}$, $i = \overline{0, m-2}$, образуют множество точек нового фронта.

6. Если найдется пара пересекающихся отрезков $VW \in \mathbf{B}$, $YZ \in \mathbf{B}$, то все точки, лежащие между V и Y на начальном фронте исключаются из рассмотрения, а точка пересечения принимается за точку нового фронта.

7. Если построенное множество является незамкнутой кривой, то решением является середина данной линии, т.е. точка, расстояния от которой до концов одинаково. Если остается единственная точка на строящемся фронте, то данная точка является решением. В противном случае построенный фронт принимается за начальный и выполняется пункт 1.

Отметим, что после определения начального фронта волны вся область вне замкнутой линии считается непроходимой и направление вектора нормали выбирается внутрь области.

Шаги 3 и 6 обеспечивают корректное построение фронта при образовании «ласточкиного хвоста». Явление «ласточкиного хвоста» возникает в случае, когда нарушается гладкость волнового фронта вследствие того, что в некоторую точку фронта одновременно приходят две или более волны.

На рис. 2.2 проиллюстрирована работа алгоритма: слева показан процесс конструирования первого фронта, по центру – момент распада фронта, справа – вписанный в заданное множество круг максимального радиуса.

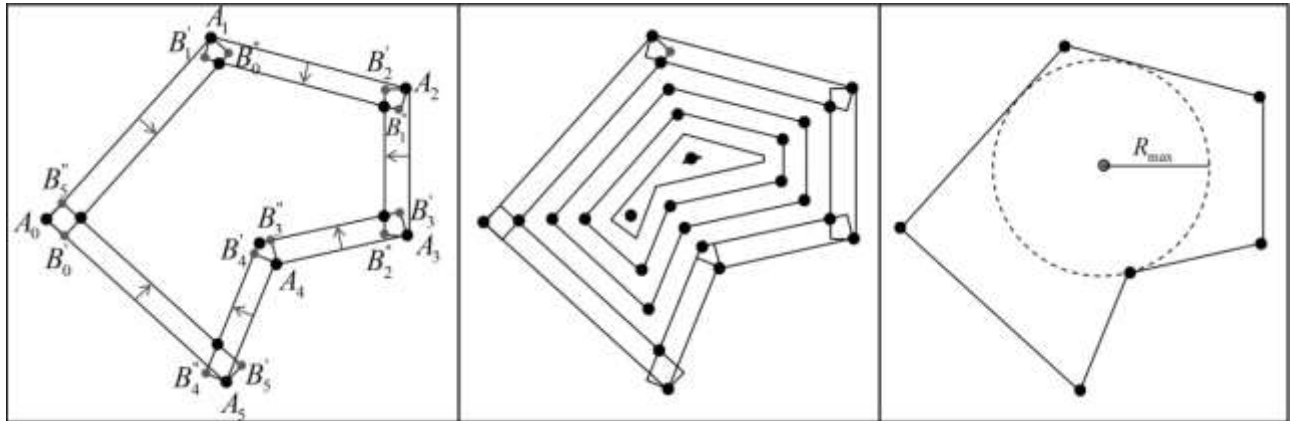


Рис. 2.2. Иллюстрация механизма работы алгоритма **BWI**

В случае, когда область P_i не является односвязной, иначе говоря, в ней присутствуют непроходимые для световой волны барьеры, для решения первой подзадачи необходим дополнительный алгоритм конструирования волновых фронтов, распространяющихся с границы барьера во внешнюю область (BorderWaveOutside–BWO).

Данный алгоритм отличается от BWI выбором направления нормали – здесь она направлена во внешнюю область. Тогда общий алгоритм решения подзадачи 1 с учетом возможной многосвязности имеет следующий вид.

Алгоритм (BorderWaveInside Multiply-connected domain–BWI–MCD)

1. С помощью алгоритма BWI строятся фронты световой волны, выпущенной с границы области.

С помощью алгоритма BWO строятся фронты световых волн, выпущенных с границ всех барьеров. Работа алгоритмов продолжается до первого соприкосновения BWI-волны с одной из BWO-волн. Построенные фронты сохраняются.

2. Множество точек, которых не достигла ни одна из волн, разбивается на максимальные односвязные подмножества S_j (сегменты), которые сохраняются в списке сегментов S . При этом сегменты, полученные на предыдущей итерации, удаляются.

3. Выполняется одна итерация алгоритмов BWI и BWO, начиная с сохраненных фронтов.

Процесс 2–3 продолжается до тех пор, пока множество точек, которых не достигла ни одна из волн, непусто.

4. Для всех элементов списка сегментов проводится анализ.

Если сегмент содержит единственную точку, то она является решением.

Если сегмент является незамкнутой кривой, то решением является ее «середина».

В остальных случаях решение находится с помощью алгоритма BWI, поскольку сегмент является односвязным множеством.

Все полученные решения q_j заносятся в список Q .

5. Анализируется список решений Q .

Если он содержит одну точку, то такая точка будет искомым решением. В противном случае для каждой точки $q_j \in Q$ с помощью алгоритма, предложенного в работе [30], вычисляется величина $r_j = \tilde{\rho}(q_j, \partial P_i)$, ∂P_i – граница рассматриваемого множества P_i . Искомым решением будет такая точка $q^* \in Q$, для которой $r^* = \max_j r_j$.

На рис. 2.3 представлены некоторые итерации алгоритма BWI-MCD: слева показан процесс конструирования первого фронта с границы множества – внутрь и с границы барьера – наружу, по центру – фронты на последней итерации, справа – вписанный круг максимального радиуса.

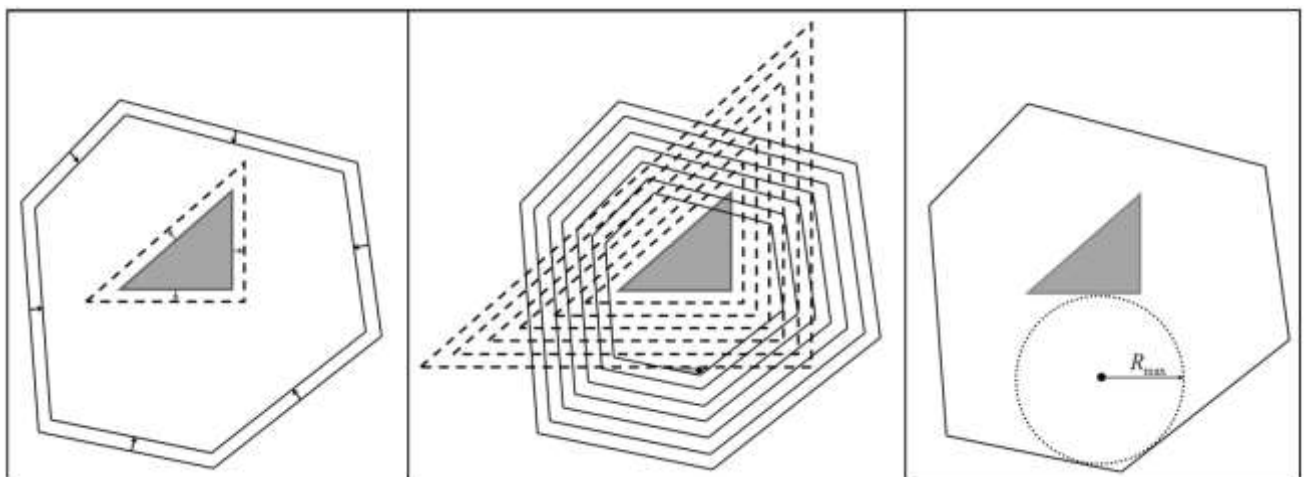


Рис. 2.3. Иллюстрация механизма работы алгоритма **BWI-MCD**

Таким образом, алгоритм решения задачи (2.5)–(2.8) имеет следующий вид.

Алгоритм (Algorithm of General Optimal Packing of Equal Circles – AGOPEC–MCS)

Шаг №1. Методом случайной генерации задается вектор $s = (s_1, \dots, s_n)$, удовлетворяющий ограничению (2.8) и определяющий начальное расположение центров упаковываемых кругов. Радиус R полагается равным нулю.

Шаг №2. Область P разбивается на подмножества P_i , $i = 1, \dots, n$ с учетом ограничения (2.8) с помощью алгоритма, предложенного в [30].

Шаг №3. Для каждого P_i , $i = 1, \dots, n$ с помощью алгоритма BWI–MCD решается подзадача 1. В результате для каждого P_i , $i = 1, \dots, n$ находятся координаты центра упакованного круга $\bar{s}_i$ и его максимально возможный радиус r_i .

Шаг №4. Вычисляется величина $R = \min_{i=1, \dots, n} \rho(\bar{s}_i, \partial P_i)$.

Шаги 2–4 повторяются до тех пор, пока R увеличивается, после чего текущий вектор $\bar{s}$ сохраняется в качестве следующего приближения.

Шаг №5. Значение счетчика количества генерации начальных положений $Iter$ увеличивается на единицу. Если значение $Iter$ достигло наперед заданной величины, то работа алгоритма завершается, в противном случае осуществляется переход к Шагу №1.

2.3. Задача построения оптимальных покрытий равными кругами

Пусть в некоторой ограниченной области $D \subseteq R^2$ имеется односвязное множество $M \subset D$ с непрерывной границей ∂M .

Рассматривается задача об оптимальном покрытии множества M заданным числом кругов равного радиуса $C_k(O_k, r)$, где O_k – центр k -го круга, r – радиус, $k = \overline{1, n}$.

Определение 1. Покрытием $P_n(r)$ множества M из n кругов радиуса r называется объединение $C_1(O_1, r) \cup C_2(O_2, r) \cup \dots \cup C_n(O_n, r)$ такое, что

$$\forall k = \overline{1, n}: C_k(O_k, r) \subseteq D, O_k \subseteq M, P_n \cap M = M. \quad (2.10)$$

Иначе говоря, набор кругов P_n является покрытием множества M в том случае, когда их объединение содержит множество M .

Определение 2. Оптимальным покрытием P_n^* множества M будем называть покрытие, состоящее из кругов минимального радиуса.

Следовательно, при нахождении оптимального покрытия P_n^* множества M требуется найти такое разбиение M на n сегментов $M_k, k = \overline{1, n}$ и расположение центров покрывающих их кругов $O_k, k = \overline{1, n}$, чтобы радиус последних был минимальным, т.е. требуется решить задачу (2.4).

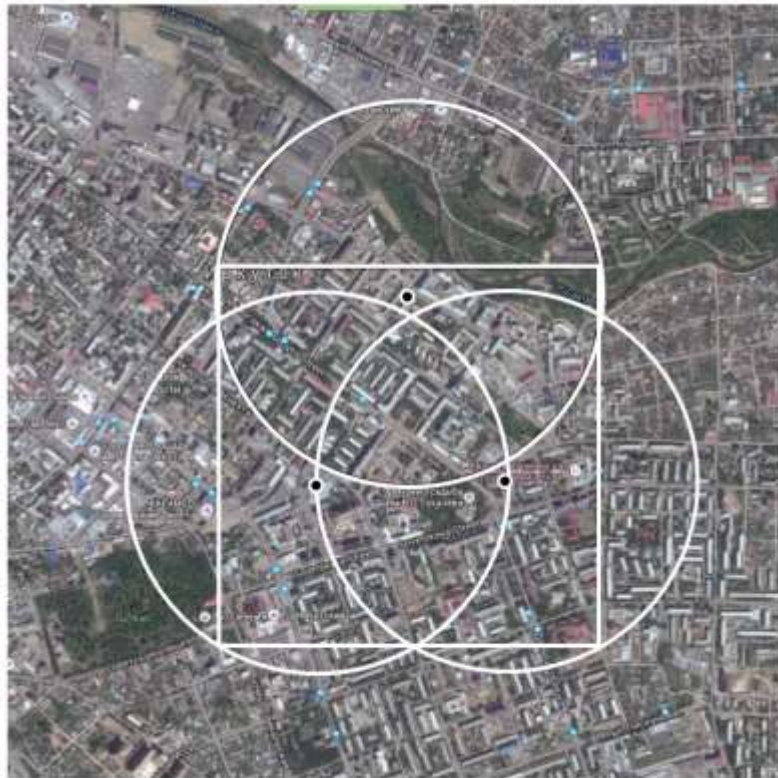


Рис. 2.4. Иллюстрация задачи построения оптимального покрытия равными кругами

Для решения задач (2.4), (2.10) предлагается алгоритм, суть которого заключается в последовательном разбиении покрываемого множества на сегменты относительно начального набора центров покрывающих кругов на основе диаграммы Вороного; нахождении наилучшего центра покрывающего круга для каждого сегмента; пересегментации относительно новых центров.

Алгоритм

Шаг №1. Определяются начальные координаты центров O_k кругов $C_k, k = \overline{1, n}$ в заданной области M методом случайной генерации положений, по координатные совпадения не допускаются.

Шаг №2. Из точек $O_k, k = \overline{1, n}$, выпускаются световые волны по алгоритму, предложенному в [30], производится разбиение области M на n сегментов M_k , и определяются границы сегментов $\partial M_k, k = \overline{1, n}$.

Шаг №3. Граница ∂M_k сегмента M_k аппроксимируется ломаной с узлами в точках $A_i, i = \overline{1, m}$.

Шаг №4. Из каждой точки $A_i, i = \overline{1, m}$, выпускаются световые волны по алгоритму из [30].

Шаг №5. Каждая точка $(x, y) \in M_k$, впервые достигнутая одной из световых волн, маркируется, фиксируется время ее достижения $T(x, y)$. Определяется точка $\bar{O}_k = \arg \max_{(x, y) \in M_k} T(x, y)$. Тогда минимальный радиус круга с центром $\bar{O}_k$, покрывающего область M_k , определяется как $R_{k \min} = \max_{i=1, m} \rho(O_k, A_i)$.

Шаги 3–5 выполняются независимо для каждого сегмента $M_k, k = \overline{1, n}$.

Шаг №6. Определяется $R_{\min} = \max_{k=1, \dots, n} R_{k \min}$, после чего осуществляется переход к Шагу №2, при этом в качестве начальных координат выступает найденный набор: $O_k = \bar{O}_k, k = \overline{1, n}$.

Шаги №2–6 повторяются до тех пор, пока уменьшается радиус $R_{\min}$, после чего текущее покрытие $P_n = \bigcup_{k=1}^n C_k(\bar{O}_k, R_{\min})$ сохраняется в качестве приближения к решению задачи.

Шаг №7. Значение счетчика количества генерации начальных положений $Iter$ увеличивается на единицу. Если значение $Iter$ достигло наперед заданной ве-

личины, то работа алгоритма завершается, в противном случае осуществляется переход к Шагу №1.

2.4. Задача размещения логистических объектов в условиях кооперации и конкуренции

Пусть на ограниченной территории в точках $B_i(x_i, y_i)$, $i = \overline{1, N}$, располагаются потребители с известным объемом потребления $b_i, i = \overline{1, N}$, а в точках $A_k(x_k, y_k)$, $k = \overline{1, m}$ – обслуживающие их логистические центры (предприятия, склады), при этом каждый потребитель входит в логистическую зону обслуживания того центра, товар из которого может быть доставлен за наименьшее время.

Необходимо определить расположение $Q_j(x_j, y_j)$, $j = \overline{1, M}$ новых центров, которые являются конкурентами для существующих (но не между собой), обеспечивающее захват максимально возможной доли рынка.

Пусть в области $D \subseteq R^2$ заданы точки $B_i(x_i, y_i)$, в которых располагаются потребители. Обозначим $T_i(Z)$ – минимальное время доставки товара из $Z \in D$ в B_i . Согласно (2.2) $T_i(Z) = \rho(B_i, Z)$.

Пусть также имеются M новых предприятий, координаты которых Q_j пока неизвестны. Обозначим $T_{i,k} = T_i(A_k)$ и $T_{i,m+j} = T_i(Q_j)$ – минимальное время доставки товара потребителю B_i со складов A_k либо Q_j , соответственно, $i = \overline{1, N}$, $k = \overline{1, m}$, $j = \overline{1, M}$. В этом случае минимально возможное время доставки товара до i -го потребителя определяется как $T_i = \min_{1 \leq p \leq m+M} T_{i,p}$, и он входит в логистическую зону предприятия $\hat{p} = \arg \min_{1 \leq p \leq m+M} T_{i,p}$.

Требуется найти оптимальное расположение дополнительных узлов $Q_j(x_j, y_j)$, $j = \overline{1, M}$ и осуществить новое разбиение множества потребителей на $m + M$ подмножеств, определив номера $I_k = \{i : T_{i,k} \leq T_{i,p}, p = \overline{1, m+M}\}$ потребителей, обслуживаемых складом $A_k(x_k, y_k)$, $k = \overline{1, m}$ и номера

$I_{m+j} = \{i : T_{i,m+j} \leq T_{i,p}, p = \overline{1, m+M}\}$ потребителей, обслуживаемых складом

$Q_j(x_j, y_j), j = \overline{1, M}$ таким образом, чтобы суммарный объем потребления объеди-

ненной логистической зоны $\bigcup_{j=1}^M I_{m+j}$ был максимальным:

$$\sum_{j=1}^M \sum_{i \in I_{m+j}} b_i \rightarrow \max. \quad (2.11)$$

Параметрами максимизации здесь являются координаты предприятий Q_j и состав подмножеств потребителей I_k, I_{m+j} . В случае, когда $T_{i,m+j} = T_{i,p}$, мы предполагаем, что потребитель не меняет обслуживающее предприятие.

Отметим, что случай, когда открытие новых предприятий направлено не на захват рынка, а на повышение качества обслуживания, и, соответственно, оно работает в кооперации с существующими, сводится к задаче об оптимальном размещении нескольких логистических центров [26, 29, 30]. При этом оптимальным расположением можно считать такое, при котором будет минимально суммарное время доставки товара до потребителей из ближайшего склада. Тогда функционал примет вид:

$$\sum_{k=1}^m \sum_{i \in I_k} T_i + \sum_{j=1}^M \sum_{i \in I_{m+j}} T_i \rightarrow \min. \quad (2.12)$$

Для решения задачи о размещении новых предприятия в условиях кооперации предлагается *модифицированный метод последовательного улучшения с мультистартом*. Идея метода заключается в итеративном улучшении получаемого решения при помощи последовательной сегментации области на логистические зоны обслуживания для каждого предприятия и нахождении оптимального расположения новых складов в соответствующих зонах [11, 26], при этом расположение существующих предприятий не изменяется. Генерация начальных положений повторяется несколько раз, в результате определяются локальные минимумы, среди которых выбирается

лучший вариант. Данный процесс описывается следующим модифицированным по сравнению с [11] алгоритмом.

Алгоритм ММПУМ

Шаг №1. Определяются начальные координаты новых предприятий $Q_j, j = \overline{1, M}$ методом случайной генерации положений. Если координаты какого-либо из размещаемых складов совпали с координатами одного из существующих складов, то производится регенерация его координат. Если начальное место расположения какого-либо из новых складов определено в непроходимой для световой волны области, то также производится регенерация его координат.

Шаг №2. Из точек $A_k, k = \overline{1, m}$ и $Q_j, j = \overline{1, M}$ выпускаются световые волны по алгоритму, предложенному в [31], и производится сегментация области D на $m + M$ зон. Для каждого потребителя $B_i, i = \overline{1, n}$ устанавливается принадлежность к сегменту $D_p, p = \overline{1, m + M}$.

Шаг №3. Внутри областей, обслуживаемых предприятием Q_j , определяется их оптимальное расположение. Из точек $B_i, i \in I_{m+j}$ выпускаются световые волны, скорость которых, в отличие от скорости в [11], пропорциональна b_i ; для каждой точки Z области, вычисляя норму (2.2), подсчитывается суммарное время ее достижения из всех источников $\sum_{i \in I_{m+j}} T_i(Z)$; среди всех точек $Z(x, y) \in D$ ищется точка с наименьшим значением времени, которая и будет определять оптимальное расположение Q_j .

Процесс переопределения координат (шаги №2,3) повторяется до тех пор, пока изменяются координаты хотя бы одного Q_j .

Шаг №4. Определяется суммарное время доставки груза со всех складов до «своих» потребителей с учетом их мощности. Запоминается конечное положение A_k и Q_j и соответствующее им суммарное время. Если заданное число запусков

случайной генерации начальных положений не достигнуто, переходим к Шагу №1.

Шаг №5. Из сохраненных на шаге №4 конечных положений A_k и Q_j выбирается то, которому соответствует минимально суммарное время.

И в результате работы алгоритма определяется такое положение складов Q_j , что суммарное время доставки груза со всех складов до «своих» потребителей будет минимальным.

Без использования многократного генерирования (мультистарта) возможна ситуация, когда задано такое начальное размещение складов, при котором после сегментации области D в зоне возможного обслуживания хотя бы одного склада не обнаруживается ни одного потребителя (некорректное начальное размещение). Использование мультистарта позволяет исключать из рассмотрения такие начальные размещения.

Отметим также, что в отличие от алгоритма из [11] здесь расположение существующих предприятий может оказаться неоптимальным относительно суммарного времени доставки товара внутри логистической зоны, поскольку их перенос не предусмотрен. Расположение же новых предприятий будет оптимальным относительно потребителей в своей зоне обслуживания.

Для решения задачи о размещении новых предприятий в условиях конкуренции предлагается *метод последовательного присоединения с мультистартом (МППМ)*. Идея метода заключается в последовательном улучшении получаемого решения при помощи смещения текущего расположения новых центров в направлении «наиболее перспективного» потребителя, не входящего в его текущую зону обслуживания, и повторном переопределении логистических зон. Данный процесс описывается следующим алгоритмом.

Алгоритм МППМ

Шаг №0. Параметр $j = 1$.

Шаг №1. Определяются начальные координаты одного предприятия Q_j методом случайной генерации положений.

Шаг №2. Из точек $A_k, k = \overline{1, m}$ и Q_j выпускаются световые волны, и производится сегментация D на $m + 1$ зон. Для каждого потребителя $B_i, i = \overline{1, n}$ устанавливается принадлежность к сегменту $D_p, p = \overline{1, m + j}$.

Шаг №3. Внутри области D_Q , обслуживаемой предприятием Q_j , подсчитывается число потребителей.

Шаг №4. Определяется потребитель B_s , ближайший к точке Q_j (здесь и далее близость определяется временем доставки груза), который не принадлежит области D_Q . Строится оптимальный (также в смысле минимального времени доставки груза) маршрут между точками B_s и Q_j .

Шаг №5. Производится смещение точки Q_j вдоль этого маршрута на величину, необходимую для вхождения B_s в область D_Q , и выполняется переопределение логистических зон. Если при этом число потребителей, попавших в область D_Q , увеличится, переходим к Шагу №4. В противном случае потребителя B_s исключаем из рассмотрения, точку Q_j возвращаем в исходное положение и переходим к Шагу №4. Шаги №4,5 повторяются до тех пор, пока множество рассматриваемых потребителей непусто.

Шаг №6. Сохраняется максимальное число потребителей, попавших в зону D_Q , и соответствующие координаты точки Q_j . Если заданное число запусков случайной генерации начальных положений не достигнуто, переходим к Шагу №1.

Шаг №7. Если $j < M$, то $j = j + 1$ и переходим к Шагу №1.

На рис. 2.5 показана иллюстрация работы данного алгоритма: на рис. 2.5.а показано начальное состояние (расположение двух существенных ОЦ, обозначающих черными звездами, и место жителей – черными кругами), на рис. 2.5.б показано определение зон обслуживания для существенных ОЦ, на рис. 2.5.в и 2.5.г показано расположение нового ОЦ и переопределение зон обслуживания для всех ОЦ, на рис. 2.5.е показан процесс захвата потребителей в новый ОЦ.

Таким образом, будет найдено локально оптимальное размещение новых предприятий. Использование многократной генерации начальных положений, как и в случае размещения в условиях кооперации, позволяет исключить некорректные начальные размещения.

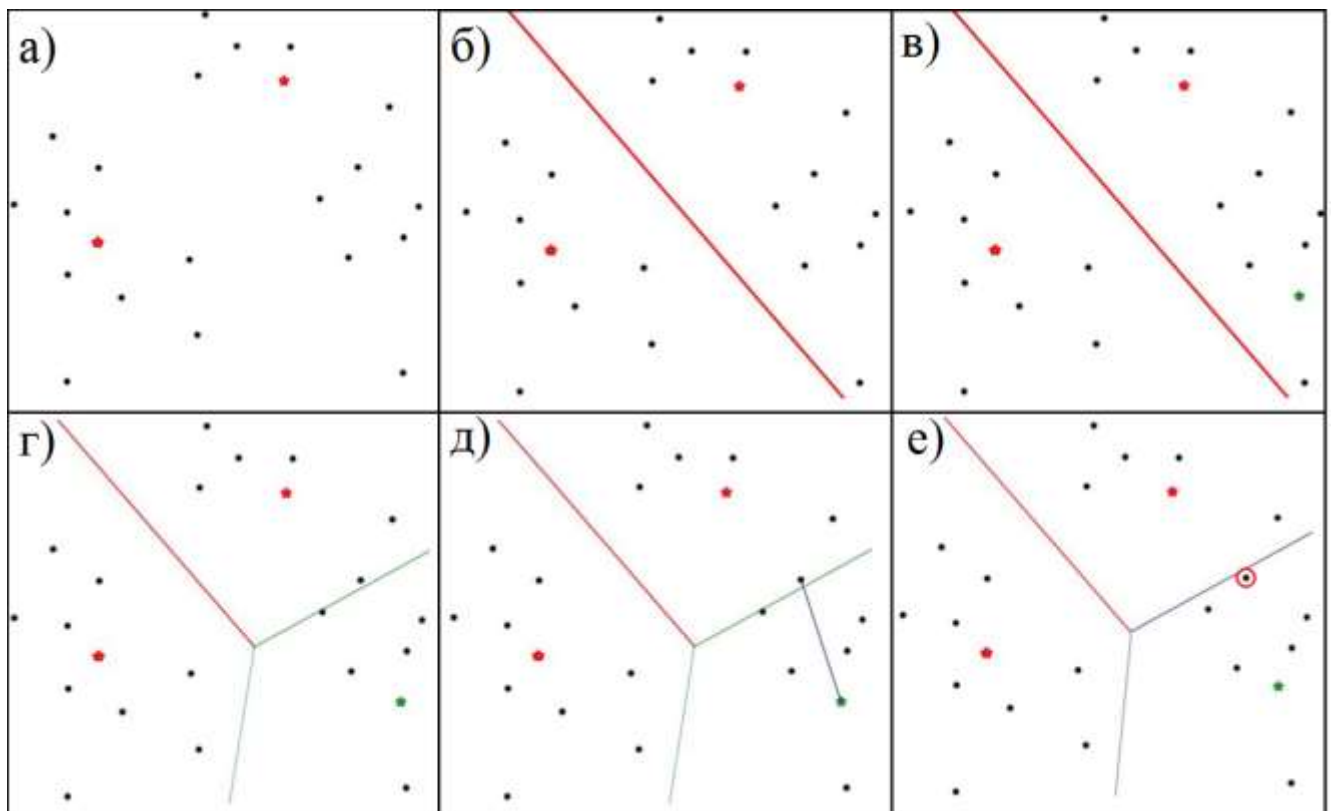


Рис. 2.5. Иллюстрация работы алгоритма МППМ

2.5. Задача организации системы коммуникаций с учетом особенностей местности

Пусть имеется m объектов $A_1, \dots, A_m$, которые необходимо обеспечить коммуникациями (дороги, трубопроводы, линии электропередачи и т.п.), при этом суммарная их стоимость должна быть минимально возможной. Отметим, что стоимость здесь величина условная и может определяться не только длиной ребер, но

и, к примеру, временем их преодоления, а также зависеть от выполнения определенных требований, в частности, включать плату за прохождение по определенным участкам.

Будем сначала предполагать, что точки ветвления (точки Штейнера) отсутствуют, т.е. необходимо построить минимальное остовное дерево. Такая постановка актуальна, например, при прокладке телекоммуникационных линий.

Здесь и далее мы полагаем, что вес участка между i -м и k -м объектом, в отличие от подобной задачи на графах [25], заранее не известен и зависит от выбранного маршрута (см. рис. 2.6), который в свою очередь напрямую связан с особенностями местности (наличие труднопроходимых участков, различных препятствий, экологических ограничений и т.п.). В такой постановке задачу можно разбить на две подзадачи: отыскание кратчайших (здесь и далее в смысле минимума по времени преодоления) маршрутов между всеми рассматриваемыми объектами и, собственно, построение минимального остовного дерева, например, методами теории графов [41].

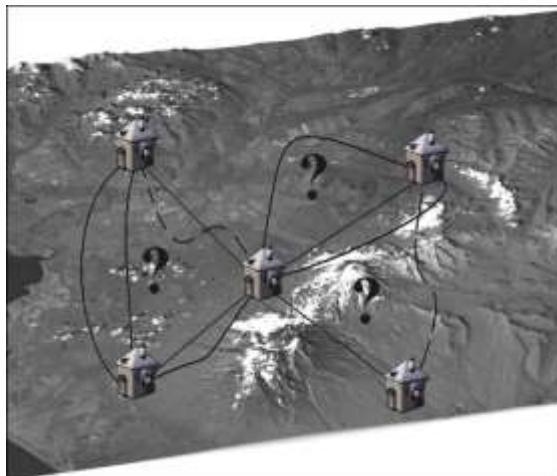


Рис. 2.6. Организация коммуникаций

Для решения первой подзадачи требуется решить серию задач (2.2), где в качестве точек a и b выступают $A_1, \dots, A_m$, при этом необходимо перебрать все их сочетания. В результате получим связный, неориентированный взвешенный граф $W(U, \Gamma^*)$, в котором $U = \bigcup_{i=1, m} A_i$ – множество вершин, а Γ^* – множество их воз-

можных попарных соединений (ребер), причем для каждого ребра (i, k) однозначно определен его вес $w(i, k) = \rho(A_i, A_k), i, k = \overline{1, m}, i \neq k$.

Задача состоит в нахождении такого связного ациклического подграфа $M \subset W$, содержащего все вершины, что суммарный вес его ребер будет минимален:

$$w(M) = \sum_{(i,k) \in M} w(i, k) \rightarrow \min. \quad (2.13)$$

Нетрудно видеть, что при $f(x, y) \equiv 1$ в формуле (2.2), задача (2.13) будет классической задачей о построении минимального остовного дерева в евклидовой метрике.

Отметим, что способ, при котором требуется отыскать кратчайшие маршруты между всеми парами объектов, является весьма затратным, поскольку для этого необходимо $m(m-1)/2$ раз вычислить (2.2).

Предлагаемый ниже алгоритм построения обобщенного минимального остовного дерева (Generalized Minimum Spanning Tree, GMST), опубликованный в [32], позволяет снизить вычислительную трудность до $m-1$ за счет специальной отбраковки «лишних» ребер.

Алгоритм GMST

Шаг №1. Из заданного множества объектов A_k ($k = \overline{1, m}$) выбирается один и маркируется.

Шаг №2. Из маркированного объекта выпускается световая волна. В момент времени t световая волна достигает ближайший немаркированный объект.

Шаг №3. Достигнутый волной объект маркируется и соединяется маршрутом с объектом-источником. Если одновременно достигается два или более немаркированных объекта, то маркируется один. Если немаркированный объект одновременно достигается двумя или более волнами, то выбирается любой из соответствующих объектов-источников и строится маршрут.

Шаг №4. Из каждого маркированного объекта выпускается световая волна. В некоторый момент времени t одна из световых волн достигает ближайший не-

маркированный объект. Если оставшихся немаркированных объектов больше двух, то выполняется Шаг №3. В противном случае достигнутый волной объект маркируется и соединяется маршрутом с объектом–источником.

Ниже на рис. 2.7 представлен механизм работы алгоритма.

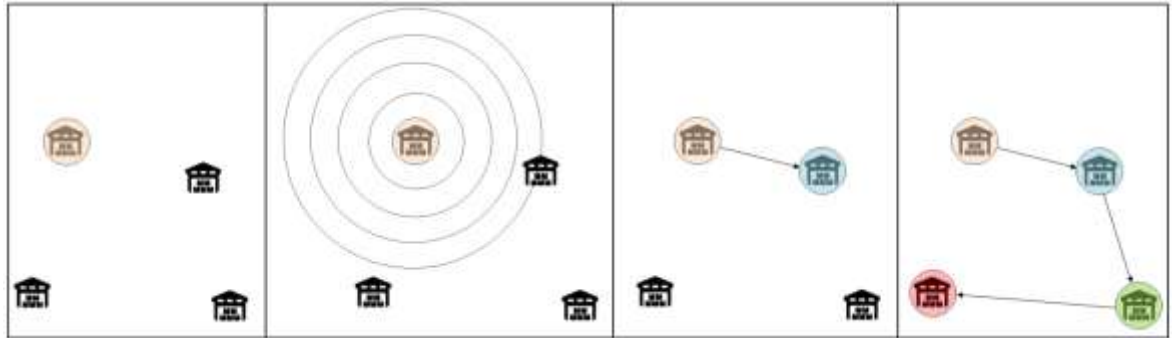


Рис. 2.7. Иллюстрация работы алгоритма GMST

В результате выполнения вышеизложенного алгоритма (за конечное число шагов) будет построено кратчайшее дерево, т.е. определено множество кривых, соединяющих заданные точки и доставляющих минимум функционалу (2.2). Построение фронтов волн основано на оптико-геометрическом подходе, представленном в работе [14]. Алгоритм построения GMST, фактически, является обобщением известного алгоритма Прима [41] для математической модели (2.2, 2.13).

Как уже упоминалось выше, в ряде случаев минимальное остовное дерево можно улучшить за счет введения дополнительных «перекрестков» – точек Штейнера. Иначе говоря, необходимо определить множества дополнительных вершин $B = \bigcup B_j$ и дополнительных ребер B^* такие, что $w(M^*) \leq w(M)$, где $M^* \subset W(U \cup B, \Gamma^* \cup B^*)$.

Алгоритм

Шаг №1. Для множества U строится минимальное остовное дерево M .

Шаг №2. Инициализируется новый подграф $S \equiv M$.

Шаг №3. Формируется множество Δ троек вершин A, B, C таких что, ребра AB и BC принадлежат M .

Шаг №4. Если множество $\Delta = \emptyset$, то переходим к шагу №7. Из множества Δ выбирается тройка d_r , из вершин которой выпускаются световые волны. Опреде-

ляется точка D_r , суммарное время достижения которой всеми волнами будет минимально возможным. Данная точка будет точкой Штейнера для d_r .

Шаг №5. Определяются ребра минимального веса, соединяющие точку D_r с вершинами d_r . Проверяется, произошло ли уменьшение суммарного веса. Если да, переходим к шагу №6, если нет, исключаем d_r из множества Δ и переходим к шагу №4.

Шаг №6. Точка D_r и выходящие из нее ребра добавляются к S , ребра $A_r B_r$ и $B_r C_r$ из S удаляются. Переходим к шагу №3, исключая $A_r B_r$ и $B_r C_r$ из M .

Шаг №7. Окончание работы алгоритма, подграф S – найденное решение.

Отметим, что в случае, когда задано ограничение на максимальное число дополнительных «перекрестков», работа алгоритма может быть досрочно завершена, если оно достигнуто.

В результате работы алгоритма будет построено дерево S , которое, по крайней мере, не хуже минимального остовного дерева M , хотя и далеко не всегда является деревом Штейнера. Однако указанное обстоятельство компенсируется тем, что вычислительная трудность не превышает $O(m^2)$, где m – число вершин в граф, и, соответственно, предложенный алгоритм находит решение за полиномиальное время.

2.6. Задача оптимизации ремонта автомобильных дорог в условиях ограниченного финансирования

Будем предполагать, что в первоочередном ремонте нуждается тот участок дороги, по которому труднее всего проехать, иначе говоря, на котором «концентрация дефектов» является максимальной.

Перейдем к строгой математической постановке задачи. Предполагается, что для некоторого участка дороги $D \subseteq R^2$ построена ограниченная неотрицательная непрерывная функция $f(x, y)$, характеризующая в точке $M(x, y)$ значимость дефекта (например, глубину: чем больше значение функции, тем серьезнее дефект).

Если $f(x, y) = 0$, дефект в точке M отсутствует и, соответственно, дорожное покрытие не нуждается в ремонте, наоборот, $f(x, y) > 0$, в точке $M(x, y)$ имеется дефект.

В случае, когда $f(x, y)$ является сеточной (табличной), для определения значений, которые лежат между узлами сетки, используется линейная интерполяция.

В рамках изучаемой модели будем пренебрегать изменением ширины дороги и считать последнюю константой. Мосты, тоннели и развязки также не рассматриваются. Таким образом, предполагается, что область D является криволинейным четырехугольником (вообще говоря, невыпуклым), граница которого состоит из двух параллельных кривых ℓ_1, ℓ_2 равной длины и двух отрезков прямой r_1, r_2 , соединяющих соответствующие концы кривых. Обозначим d – расстояние между ℓ_1 и ℓ_2 , которое соответствует ширине дорожного полотна. Будем полагать, что кривые ℓ_1, ℓ_2 являются гладкими и заданы параметрически

$$\ell_1: \begin{cases} x = x_1(t), \\ y = y_1(t), \end{cases} \quad \ell_2: \begin{cases} x = x_2(t), \\ y = y_2(t), \end{cases} \quad 0 \leq t \leq T. \quad (2.14)$$

Пусть $D(t_1, t_2)$, где $0 \leq t_1 < t_2 \leq T$ – подмножество множества D , которое ограничено фрагментами кривых ℓ_1, ℓ_2 , отвечающими изменению параметра t от t_1 до t_2 , и соединяющими их концы отрезками прямых.

Дефекты (ямы) рассматриваются как ограниченные односвязные множества, которые отделены друг от друга неповрежденными участками. Стоимость их заделки считаем пропорциональной объему необходимых для этого работ. Иначе говоря, если имеется дефект H_i , то его устранение обойдется в $\int_{H_i} f(x, y) dx dy$ условных рублей.

При бюджете C требуется найти участок дороги $D^* = D(t_*, t^*)$, стоимость ремонта которого равна C и который имеет минимальную длину, т.е. решить следующую задачу условной минимизации:

$$\int_{D^*} f(x, y) dx dy = M, \quad \int_{t_*}^{t^*} \sqrt{(x')^2 + (y')^2} dt \rightarrow \min_{t_*, t^*}. \quad (2.15)$$

Здесь $x'(t) = x'_1(t) = x'_2(t)$, $y'(t) = y'_1(t) = y'_2(t)$.

При рассмотрении небольших участков ремонтируемую трассу можно считать прямоугольником, так как в этом случае найдется конформное отображение, которое переводит рассматриваемый участок дороги в прямоугольник, однако, вообще говоря, такое предположение в определенной степени ограничивает общность исследования, поскольку при выполнении подобных отображений форма и размеры дефектов изменяются.

Ниже представлен разработанный алгоритм, позволяющий решить поставленную задачу непрерывной оптимизации.

Используются (за исключением 3-го шага, где волна пускается внутрь) криволинейные координаты: параметр t (см. 2.14) и расстояние z до «первой» обочины ℓ_1 . Как явствует из вышесказанного, $0 \leq t \leq T$, $0 \leq z \leq d$. В случае, когда трасса имеет форму прямоугольника, будем считать, что криволинейные координаты t, z совпадают с декартовыми x, y .

Алгоритм RBR (Repair Broken Road)

Шаг 1. Положим значения алгоритмических параметров $i = 0$, $j = 0$, $i = 0$, $\varepsilon_1 = 0$, $\varepsilon_2 = 0$, $U = \emptyset$ – множество участков–претендентов на ремонт, $r(t)$ – текущий отрезок границы, $r(0) = r_1$.

Будем перемещать отрезок $r(t)$ так, чтобы его концы двигались с одинаковой скоростью вдоль кривых ℓ_1 и ℓ_2 до тех пор, пока он не достигнет ближайшего дефекта.

Шаг 2. Определяем криволинейные t_i^H, z_i^H и декартовы x_i^H, y_i^H координаты «начала» ближайшего дефекта, останавливаем движение $r(t)$. Если $r(t)$ достиг противоположного конца рассматриваемого участка дороги, иначе говоря, $r(t) \equiv r_2$, то переходим к Шагу 6.

Шаг 3. Выпускаем волну w_i из точки x_i^H, y_i^H , за пределы дефекта волна выйти не сможет, так как на его границе $f(x, y) = 0$, определяем стоимость ее ремонта c_i , а также декартовы x_i^K, y_i^K и криволинейные координаты «конца» t_i^K, z_i^K . В случае необходимости (связанной с технологией ремонта) могут быть также зафиксированы координаты $x_i^{B_1}, y_i^{B_1}$ и $x_i^{B_2}, y_i^{B_2}$ точек границы рассматриваемого дефекта, которые являются ближайшими к «первой» ℓ_1 и «второй» ℓ_2 обочинам дороги, соответственно.

Шаг 4. Возобновляем движение $r(t)$, исключая из рассмотрения i -й дефект. Определяем координаты начала следующего дефекта t_{i+1}^H, z_{i+1}^H . Если $t_{i+1}^H < t_i^K$, то дефекты рассматриваются совместно, в соответствии с шагом 3 переопределяется стоимость ремонта c_i и «конец» дефекта t_i^K, z_i^K , в противном случае переходим к шагу 5.

Шаг 5. Если $\sum_{k=j}^i c_k < C - \varepsilon_1$, то возобновляем движение $r(t)$ из точки t_i^K и переходим к шагу 2 с параметрами $i = i + 1, j = j$; если $\sum_{k=1}^i c_k \in [C - \varepsilon_1, C + \varepsilon_2]$, то включаем участок $S_{j,i} = [t_j^H, t_i^K]$ длины $d_{j,i} = \int_{t_j^H}^{t_i^K} \sqrt{(x')^2 + (y')^2} dt$ в множество U претендентов на ремонт, возобновляем движение в $r(t)$ из точки t_i^K и переходим к шагу 2 с параметрами $i = i + 1, j = j + 1$.

Шаг 6. Если $U = \emptyset$ и $i = 0, j = 0$, то рассматриваемый участок не содержит дефектов.

Если $U = \emptyset$ и хотя бы один из параметров i, j отличен от нуля, то увеличиваем параметры ε_1 и ε_2 .

Если $U \neq \emptyset$, определяем $d_* = \min_{S_{i,j} \in U} d_{j,i}$, тогда соответствующий участок S_* является точным решением задачи (2.15), когда $\varepsilon_1 = 0$ и $\varepsilon_2 = 0$, и приближенным в случае $\varepsilon_1^2 + \varepsilon_2^2 > 0$

Таким образом, в результате работы алгоритма определяется участок, стоимость ремонта которого соответствует объему финансирования с учетом возможных отклонений, имеющий наименьшую длину.

Поскольку соблюдать строгое соответствие $\sum_{k=1}^i c_k = C$ зачастую не обязательно (а порой просто невозможно), то в алгоритм введены специальные параметры (допуски) $\varepsilon_1 \geq 0$, $\varepsilon_2 \geq 0$, благодаря чему во множество U включаются участки, стоимость ремонта которых лежит в пределах $C - \varepsilon_1$ до $C + \varepsilon_2$. Отметим, что здесь не оговаривается способ изменения ε_1 и ε_2 , которые, как правило, зависят от конкретной задачи (например, можно ли превышать значение C).

Замечание 1. Решение задачи (2.15), вообще говоря, может быть неединственно.

Замечание 2. Поскольку множество всех возможных решений задачи (2.15) конечно в силу того, что число дефектов конечно, то решение существует, если найдется хотя бы один дефект, стоимость ремонта которого не превосходит C .

Глава 3. Программное обеспечение для поддержки исследований транспортной логистики

При выполнении диссертационной работы автором создано программное обеспечение, которое состоит из двух программных систем. Первой из них является программная система «ОЛТП», предназначенная для решения оптимизационных задач, возникающих в ТЛС. Вторая – это интеллектуальная информационная система (ИИС) предназначена для поддержки исследования в транспортной логистике, которая использует систему «ОЛТП», включающую разработанные в диссертации модели и методы, как отдельный модуль. Опишем эти системы подробно.

3.1. Программная система «ОЛТП»

Программная система «ОЛТП», создана с помощью языка программирования C# в среде разработки Visual Studio 2008. В рамках данной системы реализованы как известные алгоритмы, представленные в работах [11, 26–28], так и оригинальные авторские алгоритмы, описанные во второй главе.

Архитектура программной системы «ОЛТП» показана на рис. 3.1.

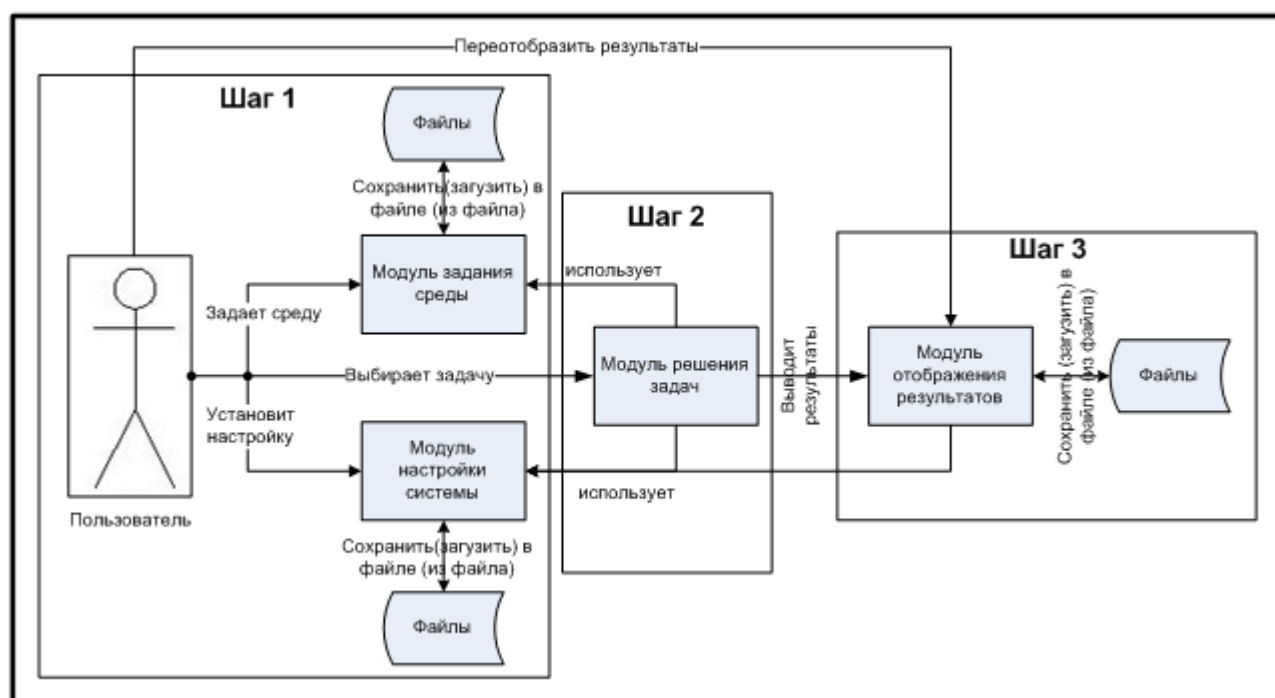


Рис. 3.1. Общая архитектура программной системы ОЛТП

3.1.1. Модуль «Задание среды»

Данный модуль представлен на рис. 3.2 в виде вкладки «Среда», предназначен для того, чтобы задать оптическую проницаемость среды (или мгновенную скорость распространения световой волны), характеризующую особенность местности, которая определяется функцией $f(x, y)$ в формуле (2.2).

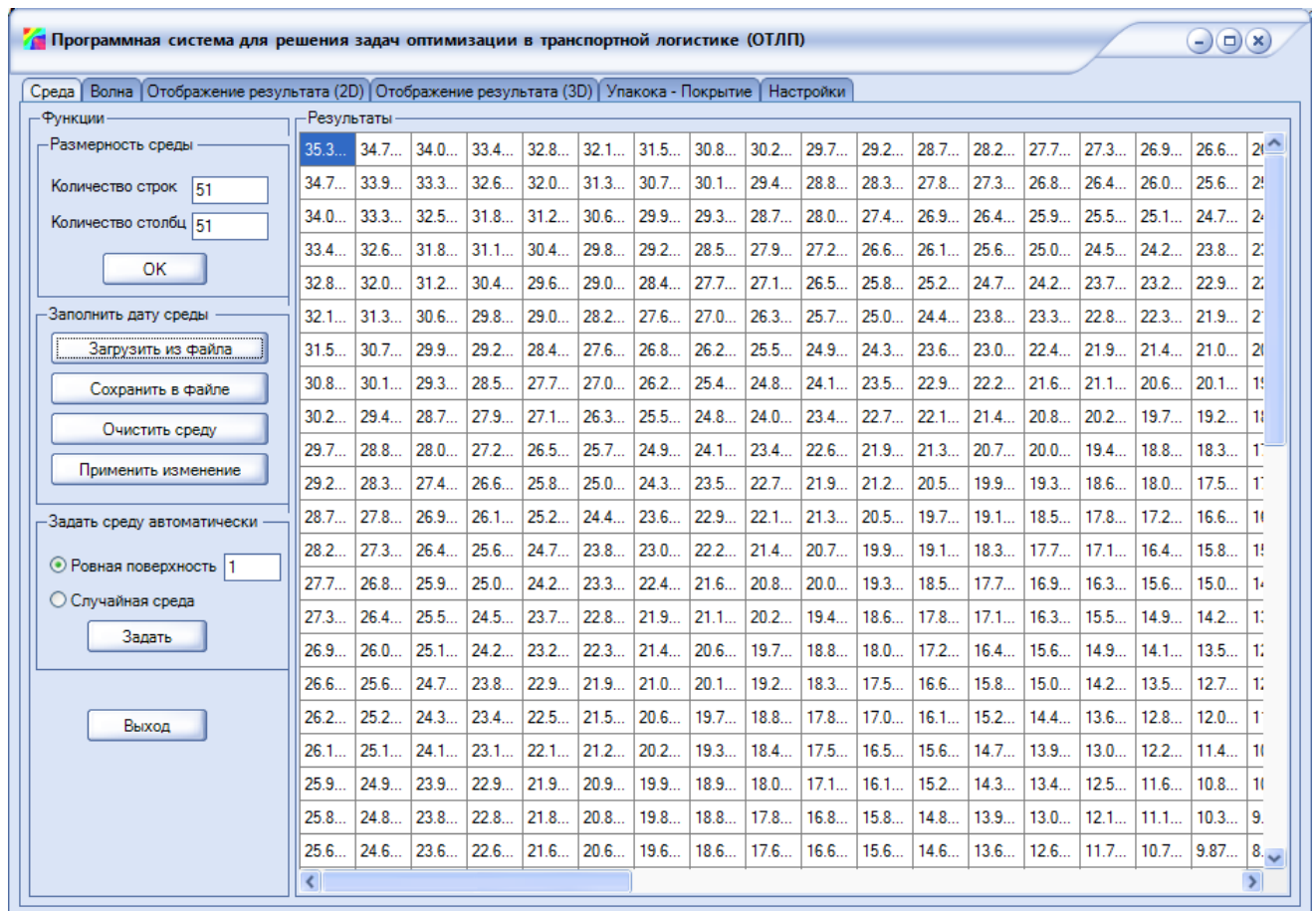


Рис. 3.2. Интерфейс модуля «Среда» для работы с метрикой

При этом можно задать среду тремя способами.

- Ручным, указав конкретное число узлов, в которых устанавливается значение проницаемости после определения размера области (при этом значение «0» определяет непроходимый для света барьер). Скорость прохождения света тем выше, чем больше значение оптической проницаемости среды. После задания значений проницаемости необходимо нажать кнопку «Применить изменение».
- Автоматически сгенерировать среду с заданным (одинаковым для всех точек области) или случайным значением проницаемости (при нажатии кнопки

«Задать»). Отметим, что в этом случае также нужно определить в первую очередь размер области.

- Задать среду путем загрузки из Excel-файла (кнопка «Загрузить из файла»).

Во всех случаях можно изменить значение любой ячейки и сохранить среду в Excel-файле.

3.1.2. Модуль «Волна»

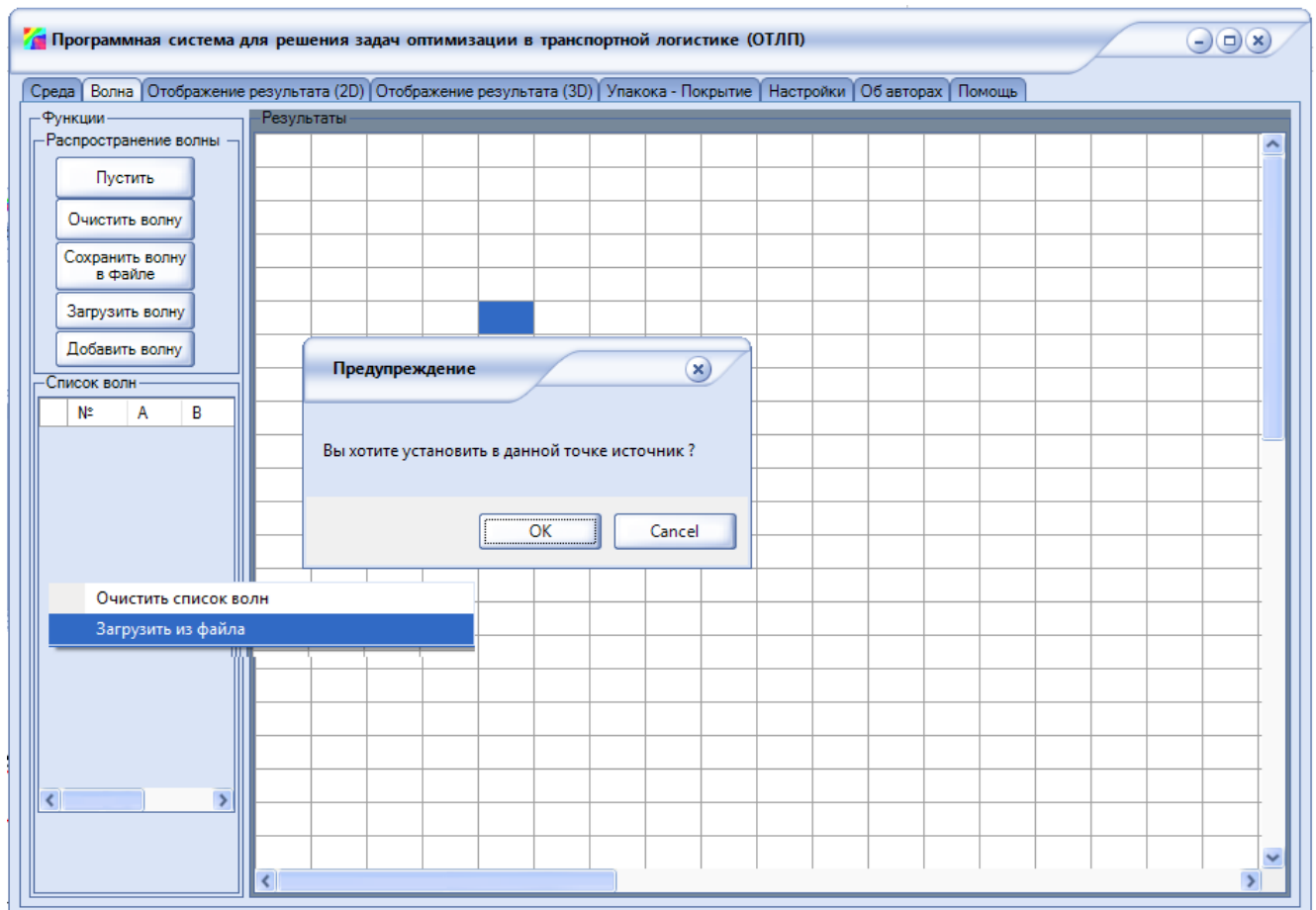


Рис. 3.3 – Вкладка «Распространение волны»

Данный модуль представлен на Рис. 3.3 в виде вкладки «Волна», который предназначен для реализации распространения световой волны в заданной метрике из одного или нескольких источников до границы, для которой необходимо сначала определить расположение источника. При этом модуль имеет два режима работы:

- дважды щелкнув на определенную ячейку таблицы (сетка) для выбора источника. После этого при нажатии кнопки «Пустить» осуществляется

распространение волны из указанного источника света на основе принципов Ферма и Гюйгенса;

- При нажатии правой клавиши мыши на область «Список волны», и загрузив из файла список координат источников модуль автоматически выпускает волны из всех источников (использование для решения некоторых задач, таких как сегментация логистических зон обслуживания, размещения логистических центров, и т.д. при точечном распределении потребителей).

Этот модуль также поддерживает возможность сохранения волны в Excel-файл (кнопка «Сохранить волну в файле»), загрузки волны из файла (кнопка «Загрузить волну»), очистки или добавления волны в список волн (кнопки «Очистить волну» и «добавить волну»).

Для построения световой волны в программной системе разработаны три алгоритма: однослойная аппроксимация (8 направлений), двухслойная аппроксимация (16 направлений), трехслойная аппроксимация (32 направления). Выбор алгоритма производится на вкладке «Настройки», в группе «Алгоритм распространения волны». При этом, если применяется алгоритм с большим количеством направлений, получаются более точные вычисления световых фронтов, но уменьшается скорость работы.

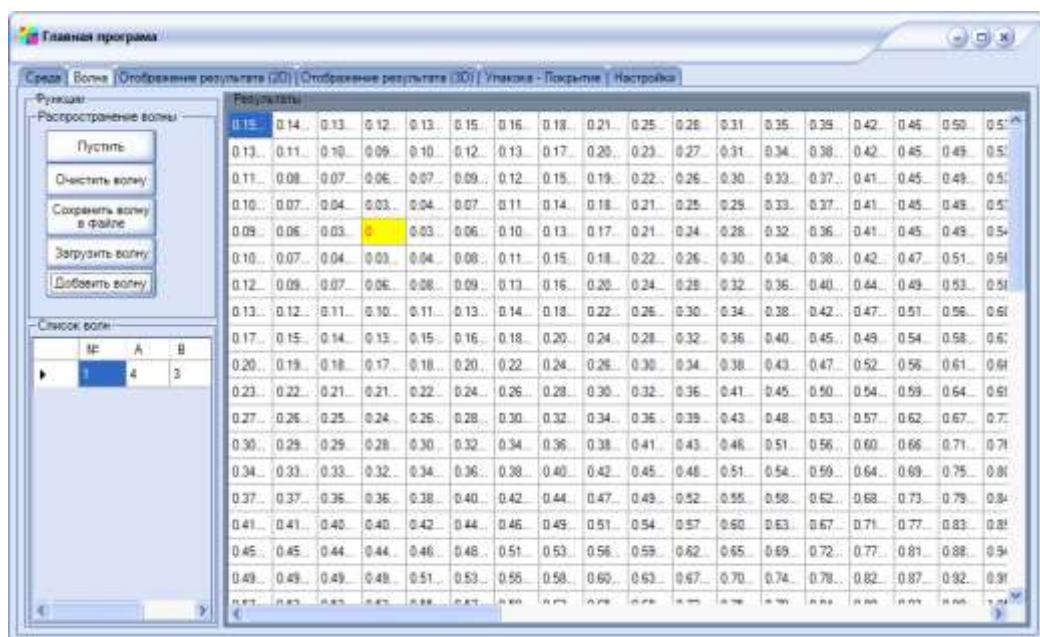


Рис. 3.4. Пустить волну и добавить в список

На рис. 3.4 показан результат распространения волны, и на рис. 3.7 представлены фронты волны, построенные с использованием алгоритма трехслойной аппроксимации в евклидовой метрике.

3.1.3. Модуль «Решение задач»

Это главный модуль, в котором реализованы все численные алгоритмы для решения предложенных задач, возникающих в транспортной логистике, интерфейсы которого представлены на рис. 3.5 и 3.6 в виде вкладки «Упаковка и покрытие» или контекстного меню функций.

Модуль «Решение задач» взаимодействует с модулями «Среда», «Настройка системы», «Волна» и получает от них информацию о настройках (в том числе, алгоритмах решения), о проницаемости среды и о времени распространения волны в узлах.

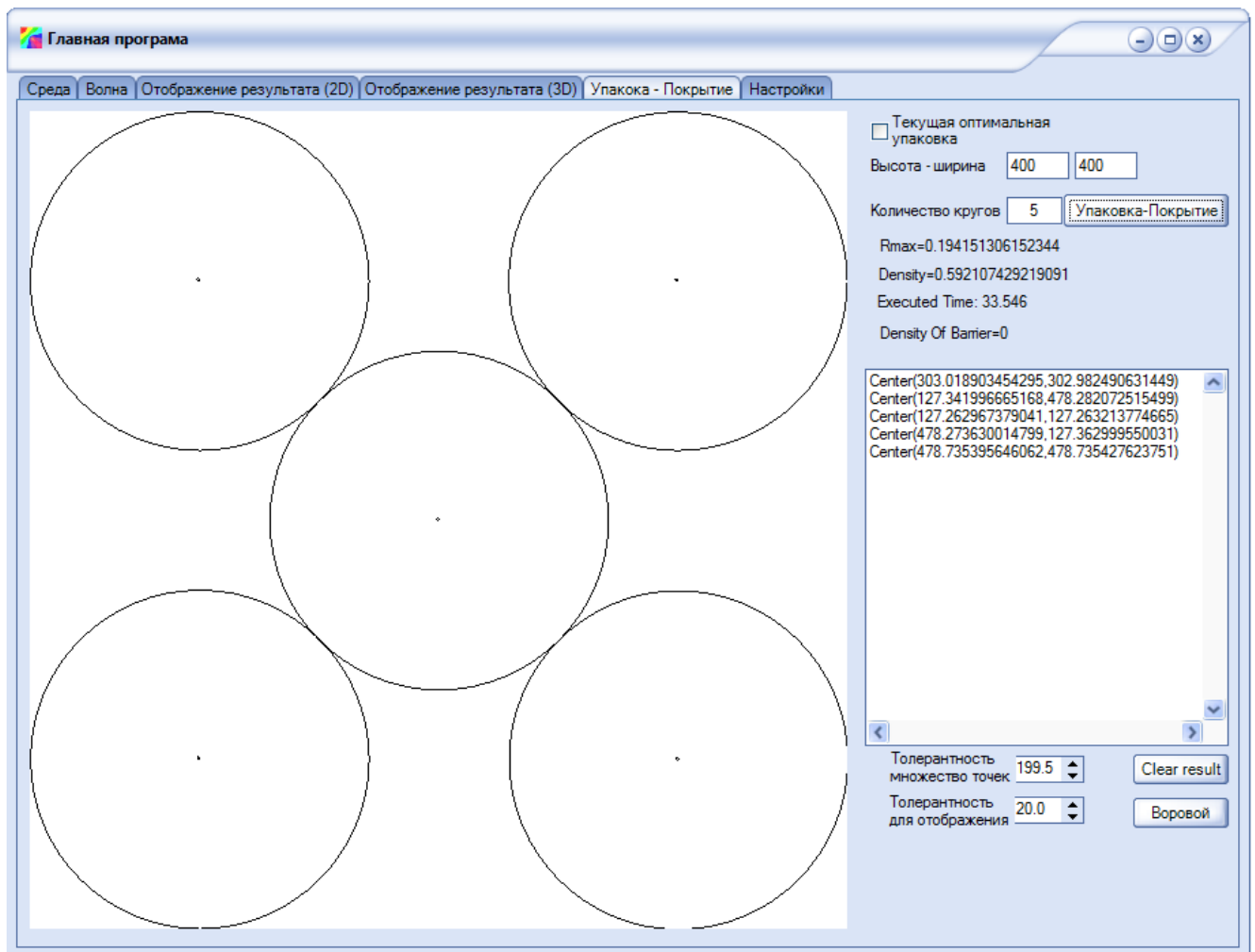


Рис. 3.5. Вкладка «Упаковка и покрытие»

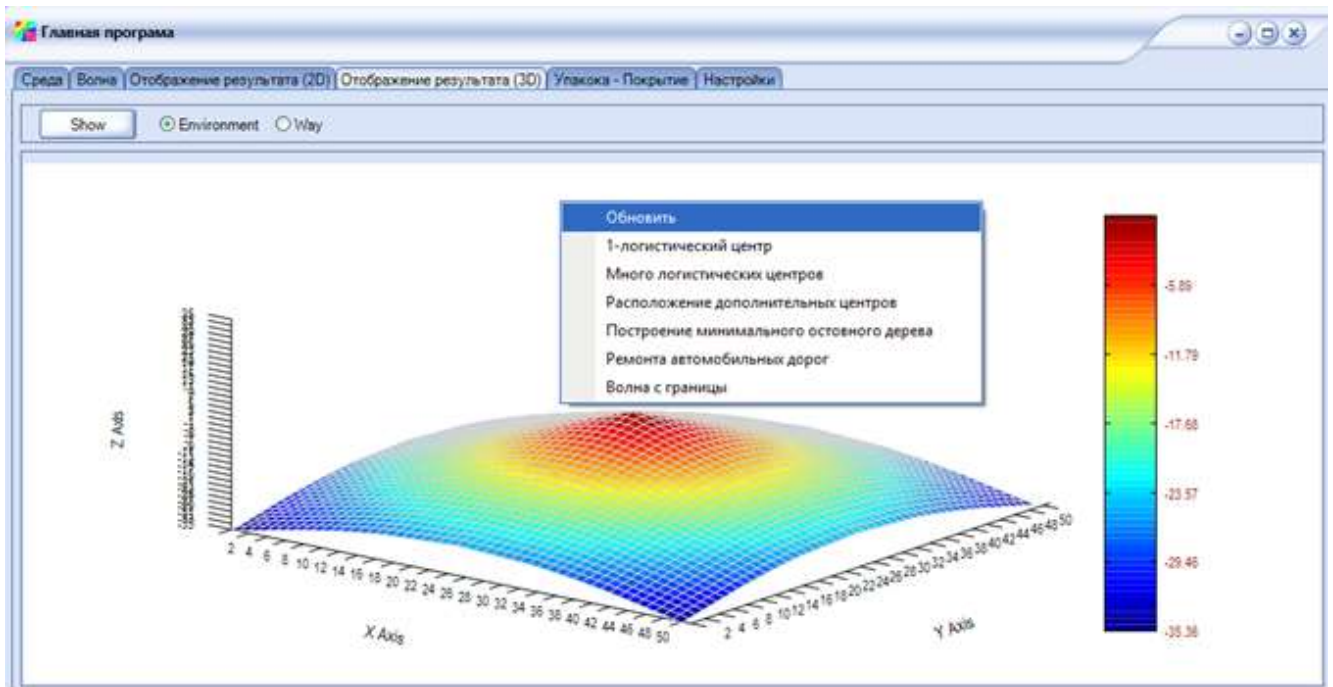


Рис. 3.6. Контекстное меню функций

Из каждого контекстного меню можно вызвать функцию решения одной или нескольких задач. Его можно вызвать из вкладок «Отображение результата (2D)» или «Отображение результата (3D)» при нажатии правой кнопки на области графического отображения.

Таким образом, модуль используется для решения следующих задач:

- построение оптимального маршрута между двумя точками;
- сегментация зон обслуживания;
- нахождение одного или нескольких логистических центров при точечных и непрерывно распределениях потребителей;
- нахождение оптимального расположение дополнительных логистических центров в условиях кооперации и конкуренции;
- построение минимального остовного дерева, в том числе, с точками Штейнера, между логистическими узлами;
- нахождение участка автомобильных дорог для ремонта при ограниченном бюджете;
- построение оптимальных упаковок и покрытий неодносвязных множеств равных кругов.

Результаты вычислительного эксперимента с использованием предложенных алгоритмов показаны в четвертой главе.

3.1.4. Модуль «Отображение результатов»

Модуль «Отображение результатов» предназначен для графического отображения полученных результатов решения задач в 2D и 3D видах, которые представлены на рис. 3.7 во вкладках «Отображение результата (2D)» и «Отображение результата (3D)». Данные модули взаимодействуют со всеми другими модулями и функциями, отображают результаты визуально.

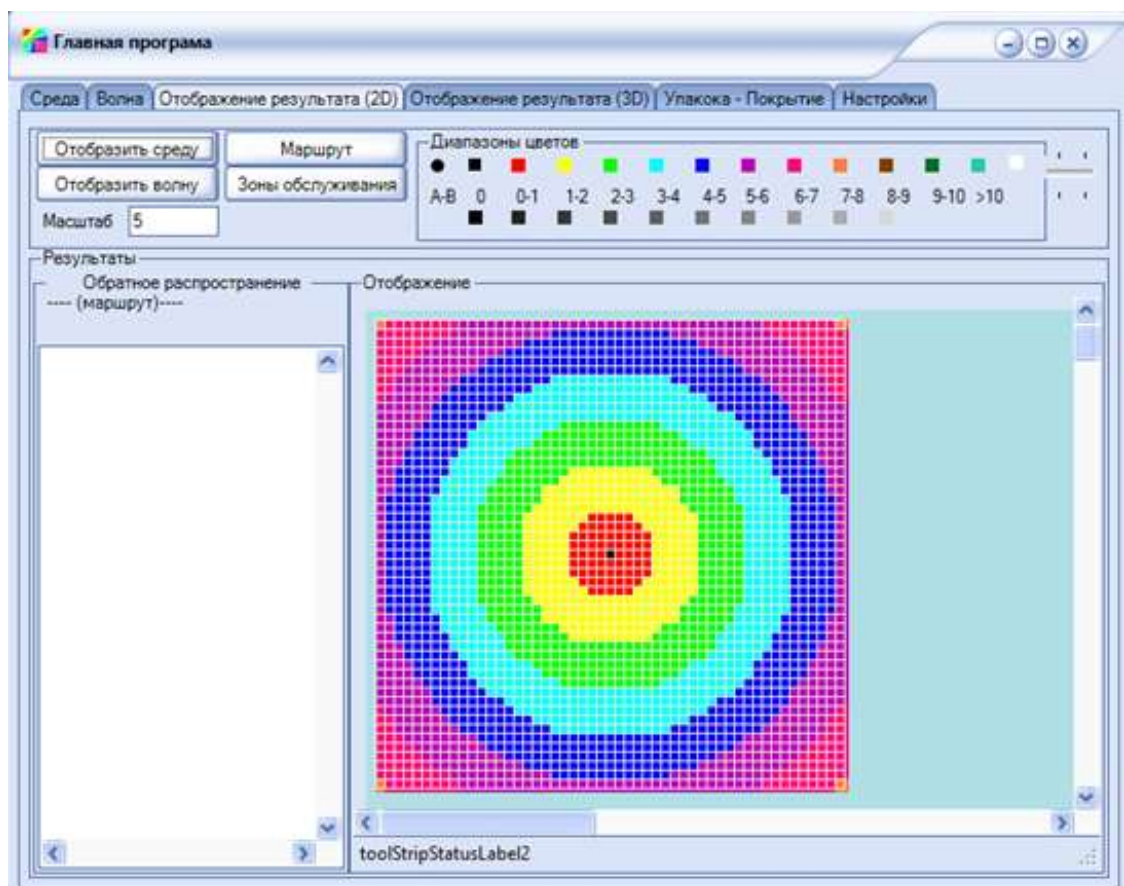


Рис. 3.7. Вкладка «Отображение результата (2D)» и «Отображение результата (3D)»

3.1.5. Модуль «Настройка системы»

Данный модуль предназначен для настройки системы, который повышает ее гибкость и удобство использования пользователем. Модуль представлен на рис. 3.8 в виде вкладки «Настройки», на которой доступны следующие группы действия:

- изменение режима отображения: переключение между режимами цветного и черно-белого рисования; переключение между режимами рисования с сеткой или без сетки;
- выбор (изменение) алгоритма распространения волны (однослойная, двухслойная, трехслойная аппроксимация);
- выбор (изменение) критерия оптимального расположения логистических центров (максимальное время минимально или суммарное время минимально);
- выбор (изменение) алгоритма построения оптимальных упаковок и покрытий;
- выбор способа распределения потребителей (точечное и непрерывное);
- настройка входных параметров для упаковки и покрытия кругами (граница покрывающего множества, барьеры).

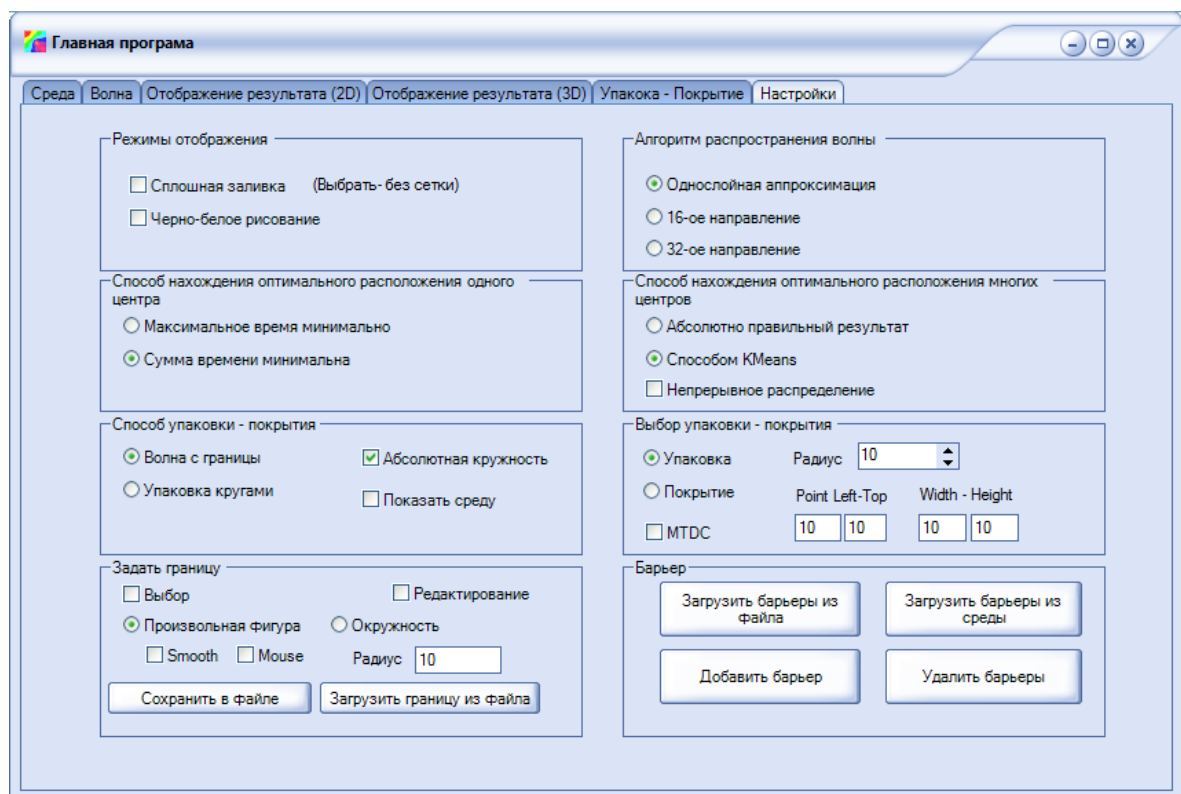


Рис. 3.8. Вкладка «Настройки»

3.2. Интеллектуальная информационная система для поддержки исследований транспортной логистики (ИИС)

3.2.1. Структура функционирования и механизм работы ИИС

Данная система [27], построена с целью прогнозирования транспортных потоков, решения ряд задач оптимизации транспортной логистики и диспетчерского управления и имеет общую архитектуру, показанную на рис. 3.9. Основным управляющим компонентом ИИС является экспертная система, использование которой позволяет в интерактивном режиме осуществлять выбор методов и средств, подходящих для решения конкретных задач. При этом для используемых в рамках ИИС модулей, методов и программных средств, реализующих необходимые вычисления, созданы формализованные описания, представленные в базе знаний фреймового типа.

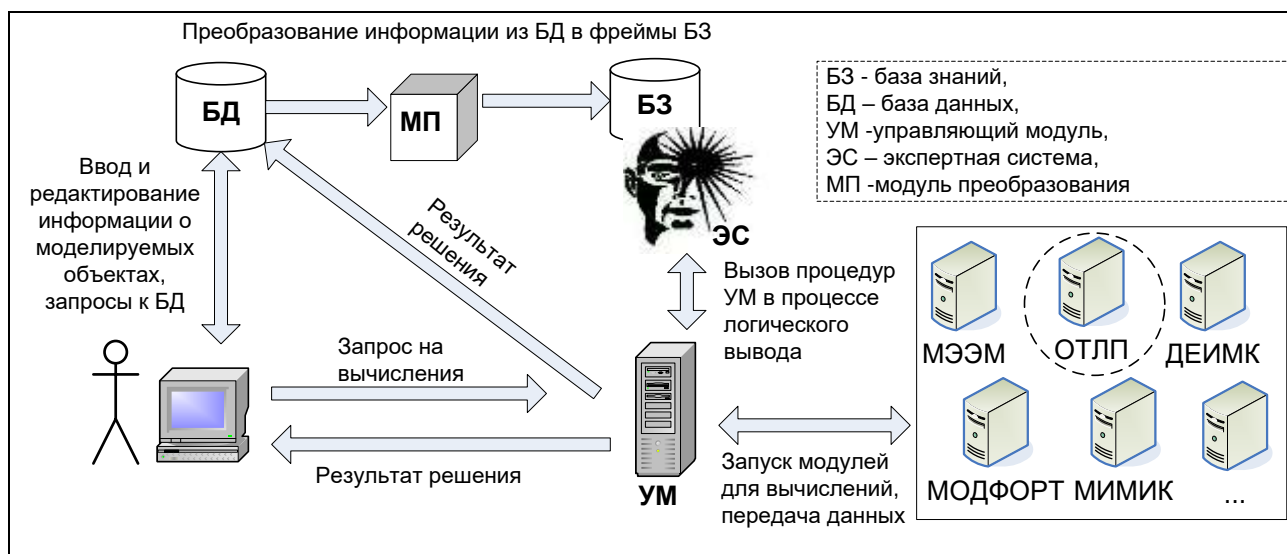


Рис. 3.9. Общая архитектура экспертной системы поддержки исследования транспортной логистики

Для удобства разработки и поддержки в актуальном состоянии база знаний (БЗ) экспертной системы (ЭС) разбита на три части. К первой части относятся онтологические модели предметных областей. Здесь задаются понятия объектов транспортно-логистической инфраструктуры, например, отрасль, регион, запасы, дорожная сеть, терминалы, и т.д. Ко второй части относится логическое описание взаимодействия уровней моделирования, оно включает такие термины, как модель, метод, прогноз развития отрасли, инфраструктурный объект и т.п. Данная

часть БЗ формализует способы использования предлагаемых моделей и методов в процессе решения задач. Наконец, в третьей части БЗ описываются спецификации программных модулей, решающих отдельные задачи.

Функционирование ИИС определяется следующей последовательностью операций. На первом шаге осуществляется выбор исследуемых объектов ТЛС, задачи, требующей решения, и выполняется запрос на вычисление от клиентского модуля к управляющему модулю ИИС. Далее запускается ЭС и производится логический вывод интерпретации задачи моделирования в терминах последовательности действий (плана), которую необходимо выполнить, чтобы получить решение задачи. В процессе вывода выполняется обмен служебной информацией между ЭС и управляющим модулем ИИС, осуществляется вызов необходимых программных процедур. Затем выполняется выбор вычислительных модулей, реализующих каждый этап синтезированного плана, при этом логический вывод останавливается и производится сохранение текущего состояния рабочей памяти; формируются входные данные для вычислительных модулей. После этого запускается полученная последовательность модулей, осуществляется передача результатов вычисления в управляющий модуль, который на их основе добавляет в БЗ новые факты в формате соответствующих фреймов БЗ. Наконец, производится запуск ЭС с обновленной базой знаний, и логический вывод продолжается по описанной схеме. После того, как работа ЭС завершается, осуществляется передача результатов решения от управляющего модуля к клиентскому программному обеспечению в предметном термине и в базу данных (БД) с привязкой к конкретным объектам транспортной логистики.

Для хранения актуальной информации о конкретных объектах транспортно-логистической инфраструктуры в ИИС используется БД, структура которой основана на используемых в БЗ онтологических моделях предметной области.

Для преобразования информации из БД в БЗ используется специальный модуль, в результате работы которого на основе данных из БД изменяются поля конкретных экземпляров фреймов БЗ.

Таким образом, процедурная часть подсистемы ИИС (Информационно-аналитическая подсистема), показана на рис. 3.10, 3.11, реализована на языке «C#», хранение данных организовано с использованием СУБД Microsoft SQL, и подробно описанная в [77]. Экспертная система реализована на основе оболочки CLIPS. Взаимодействие с пользователем осуществляется с помощью включения в базовую версию CLIPS диалогов Window Forms, которые реализуют следующие типовые элементы графического интерфейса: окно сообщений, выбор одного или нескольких вариантов, формы создания экземпляров понятий объектов транспортно-логистической инфраструктуры с автоматической генерацией полей ввода данных. Для интеграции CLIPS с ИАП применяются библиотеки CLIPSNet и CLIPSLib [129], которые позволяют реализовать механизм пользовательских функций CLIPS непосредственно в среде «.Net».

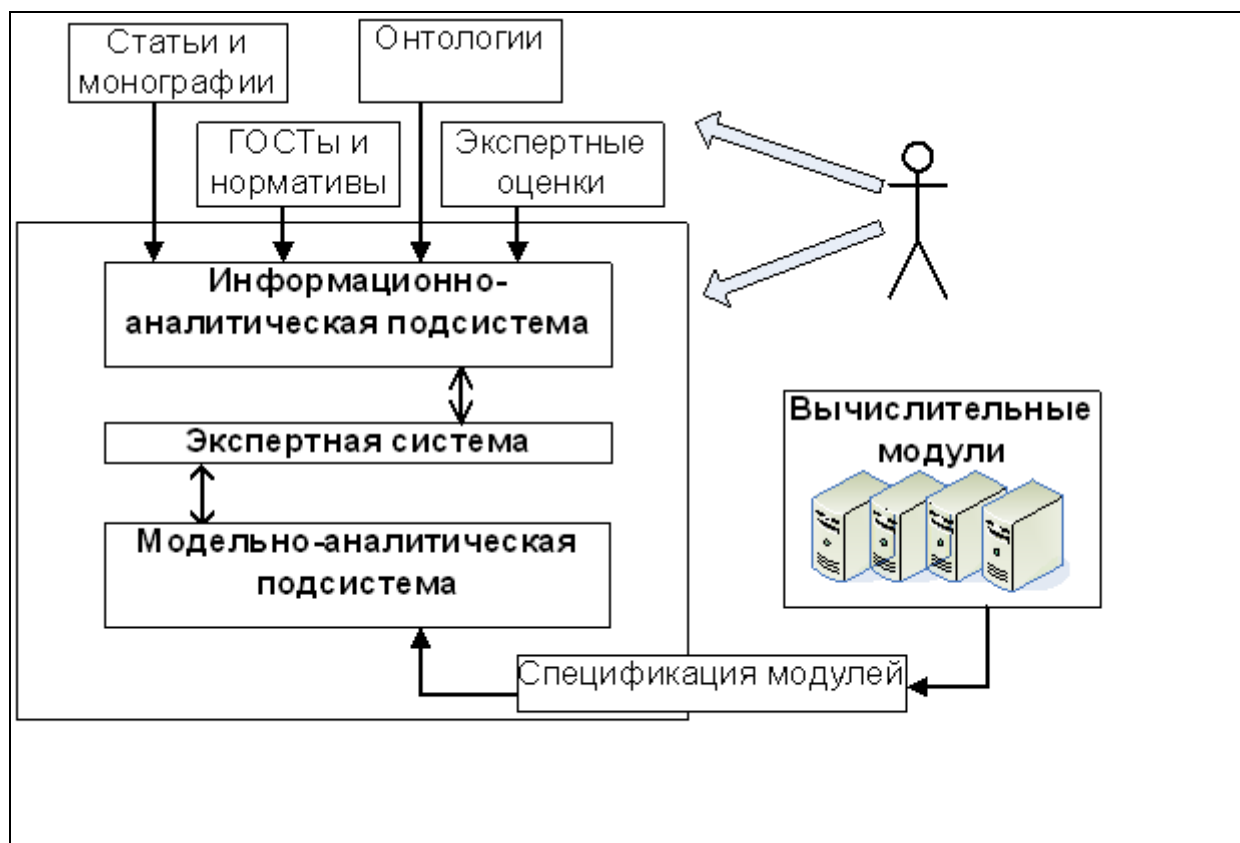


Рис. 3.10. Информационно-аналитическая подсистема в ИИС

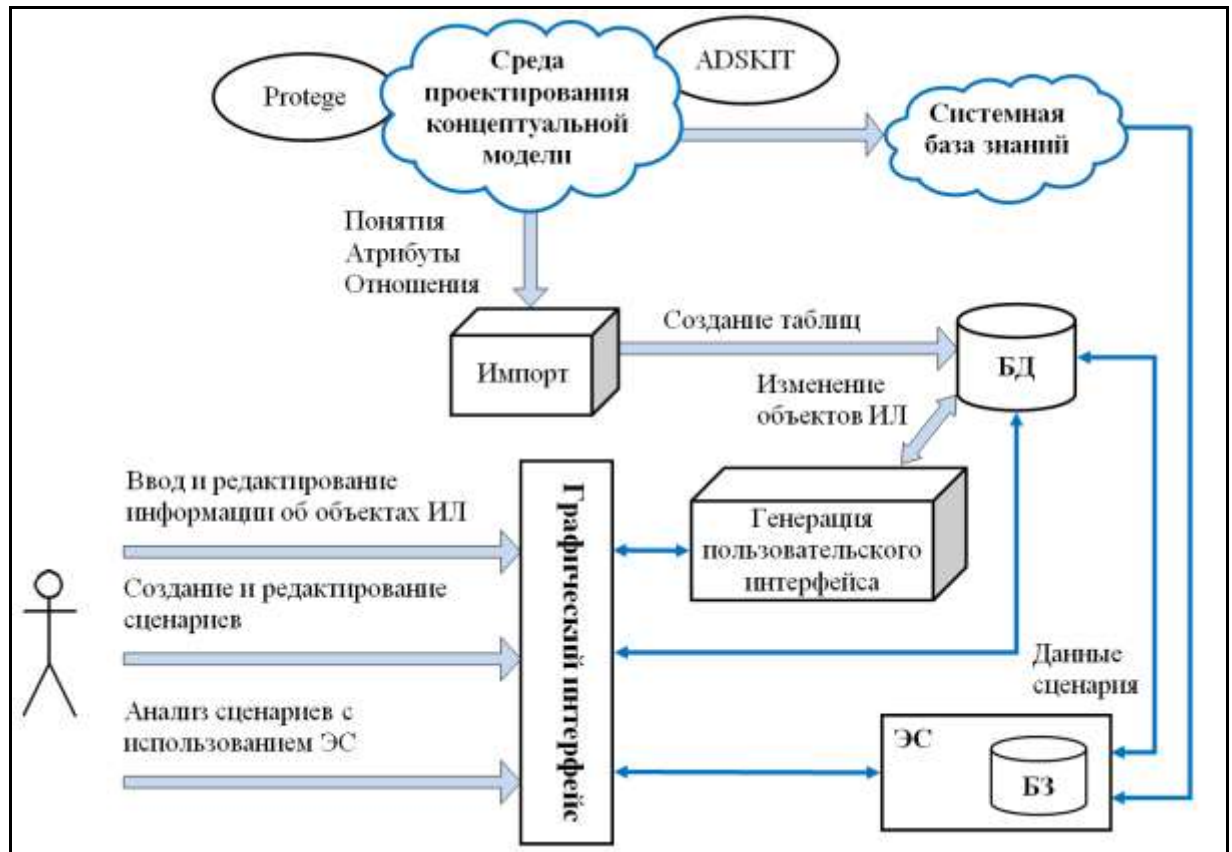


Рис. 3.11. Информационно-аналитическая подсистема

Далее представлены модули, разработанные автором для интеграции в ИИС программной системы ОТЛП и других вычислительных модулей.

3.2.2. Модуль импорта концептуальной модели во внутренние структуры данных

Этот модуль представлен на рис. 3.12 в виде вкладки «Импорт онтологии», который предназначен для импорта концептуальной модели инфраструктурной логистики (КМИЛ) во внутренние структуры данных. Для создания КМИЛ используется онтологический подход, была использована система ADSKIT [66], которая предоставляет доступный через браузер графический интерфейс для ввода элементов проектируемой модели, а также для разработки продукционных правил особого типа. Система ADSKIT позволяет представлять концептуальную модель в формате COOL, используемом оболочкой CLIPS [99] для описания фреймов и экземпляров базы знаний.

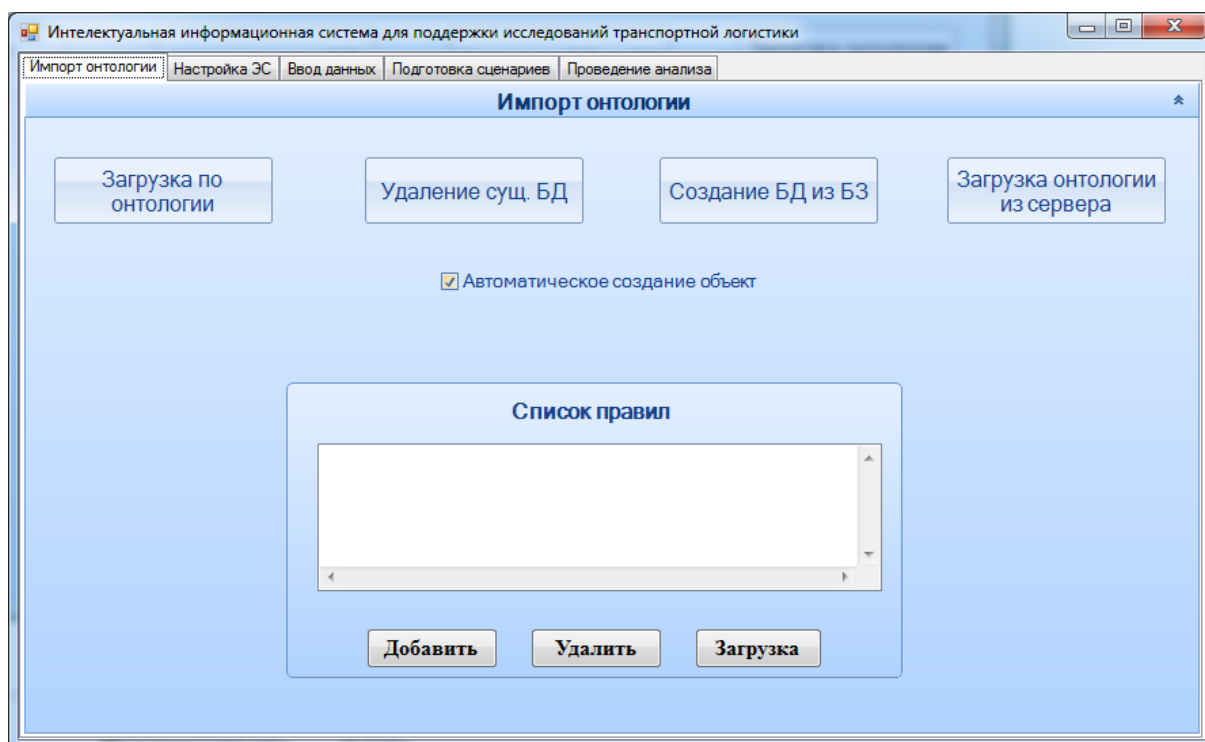


Рис. 3.12. Импорт онтологии

Информационная система обеспечивает два способа импорта концептуальной модели. Первый способ загружает модель в формате CLIPS из файла (нажав кнопку «Загрузка по онтологии»). Далее необходимо на основе понятий и отношений, содержащихся в загруженной БЗ, сформировать структуры БД, которые в последующем будут использованы для создания, редактирования и удаления информации об объектах инфраструктурной логистики. Отметим, что при загрузке нового варианта КМИЛ пользователь может как полностью удалить все связанные с предыдущей КМИЛ объекты (функция «Удаление сущ. БД»), так и сохранить те из них (функция «Создание БД из БЗ»), которые соответствуют спецификации нового варианта. Второй способ позволяет загрузить понятия и отношения КМИЛ непосредственно с сервера ADSKIT (нажать кнопку «Загрузка онтологии из сервера»).

БД используется для хранения информации об объектах инфраструктурной логистики, соответствующих загруженной БЗ. Структура БД разработана с учетом необходимости динамически в процессе исследования изменять КМИЛ. Схема структуры БД представлена на рис. 3.13.

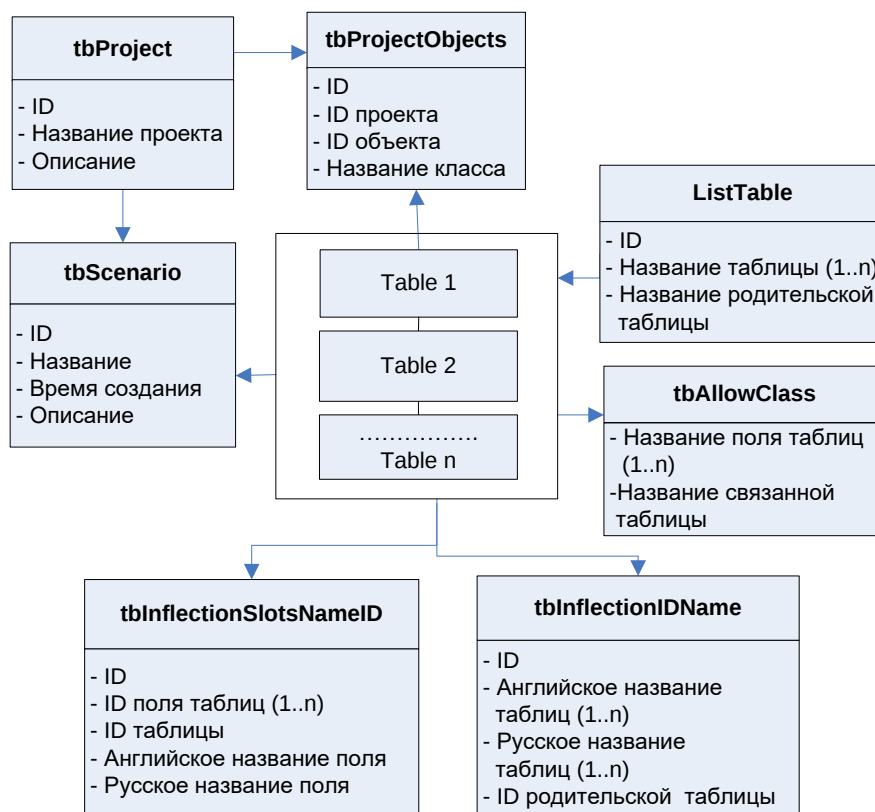


Рис. 3.13. Структура базы данных ИИС

Поясним обозначения на рис. 3.13. Под «Table 1», ..., «Table n» понимаются условные названия таблиц, в которых хранится информация об объектах инфраструктурной логистики, а также указана их связь с соответствующими классами БЗ. Все такие таблицы создаются автоматически в процессе импорта концептуальной модели. Название каждой таблицы соответствует названию класса, а название полей таблицы – слотам класса БЗ. Такое соответствие действительно можно установить, так как названия не являются произвольными. Система ADSKIT использует следующий вид для именования классов БЗ: символ «Т» или «S» и значения числового идентификатора понятия или атрибута КМ (например, имя класса «Т100» и имя слота «S10»). Информация о типе слота хранится в таблице «tbAllowClass». Название каждой таблицы и каждого поля таблицы на русском языке хранятся в таблицах «tbInflectionIDName» и «tbInflectionSlotsNameID». Информация о структурной иерархии классов (отношение потомок–родитель) хранится в таблице «tbProject».

Хранение информации для проведения сценарного анализа осуществляется в таблицах «tbProject», «tbProjectObjects» и «tbScenario». Таблица tbProject связывает объекты инфраструктурной логистики (ссылки на записи в таблицах Table 1», ..., «Table n») и сценарии (ссылки на записи в таблицах «tbScenario»).

3.2.3. Модуль редактирования информации об объектах инфраструктурной логистики

Объекты, соответствующие КМИЛ, отображаются в форме дерева. Для создания и редактирования информации формы ввода создаются автоматически согласно спецификациям объектов из КМИЛ.

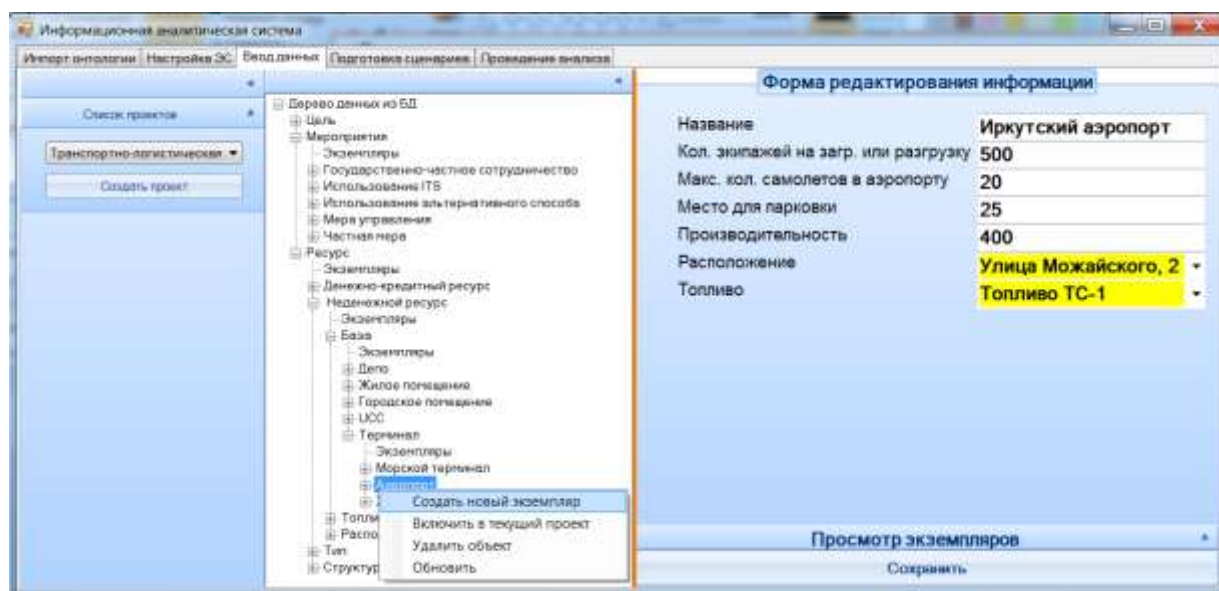


Рис. 3.14. Автоматически сгенерированный интерфейс для редактирования информации

На данный момент поддерживаются следующие элементы графического интерфейса: текстовое поле для ввода новой информации; комбинированный список и таблица для выбора имеющейся информации. Создание нового объекта, соответствующего некоторому понятию КМИЛ, инициируется нажатием по пункту контекстного меню «Создать новый экземпляр». Пример автоматически сгенерированной экранной формы редактирования информации об объекте представлен на рис. 3.14.

3.2.4. Модуль «Настройка ЭС»

Модуль предназначен для настройки экспертной системы. При этом можно снова загрузить или добавить правила, перезагрузить и запускать ЭС.

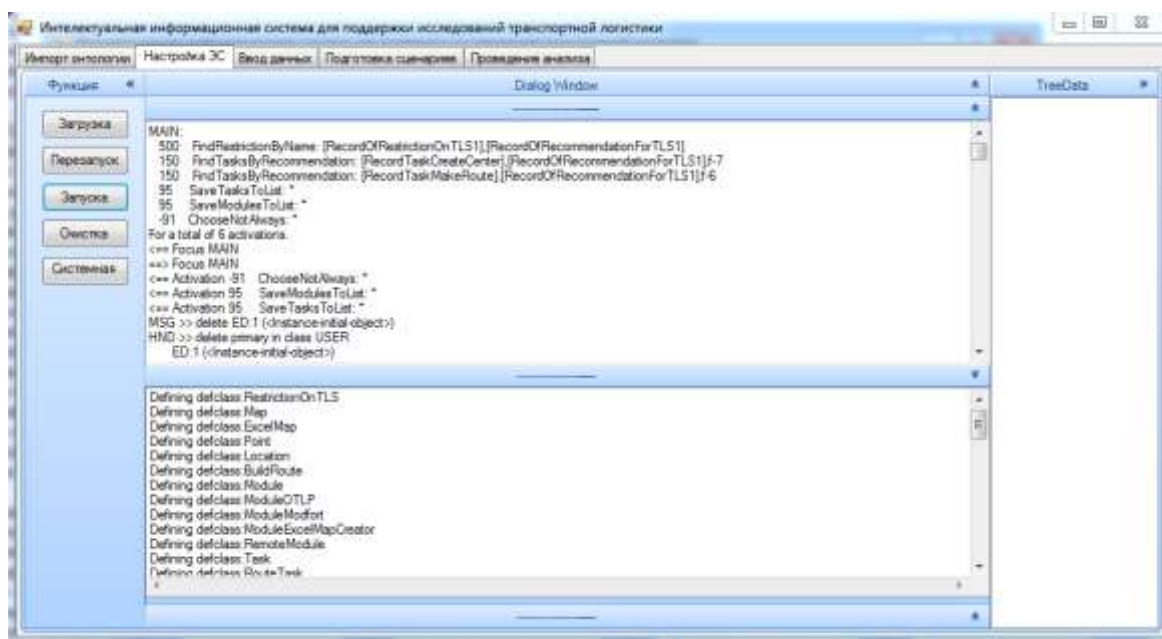


Рис. 3.15. Настройка экспертной системы

3.2.5. Модуль «Формирование базового сценария исследования»

При проведении сценарного анализа все сценарии сгруппированы по проектам, поэтому для начала анализа необходимо проект создать (рис. 3.16).

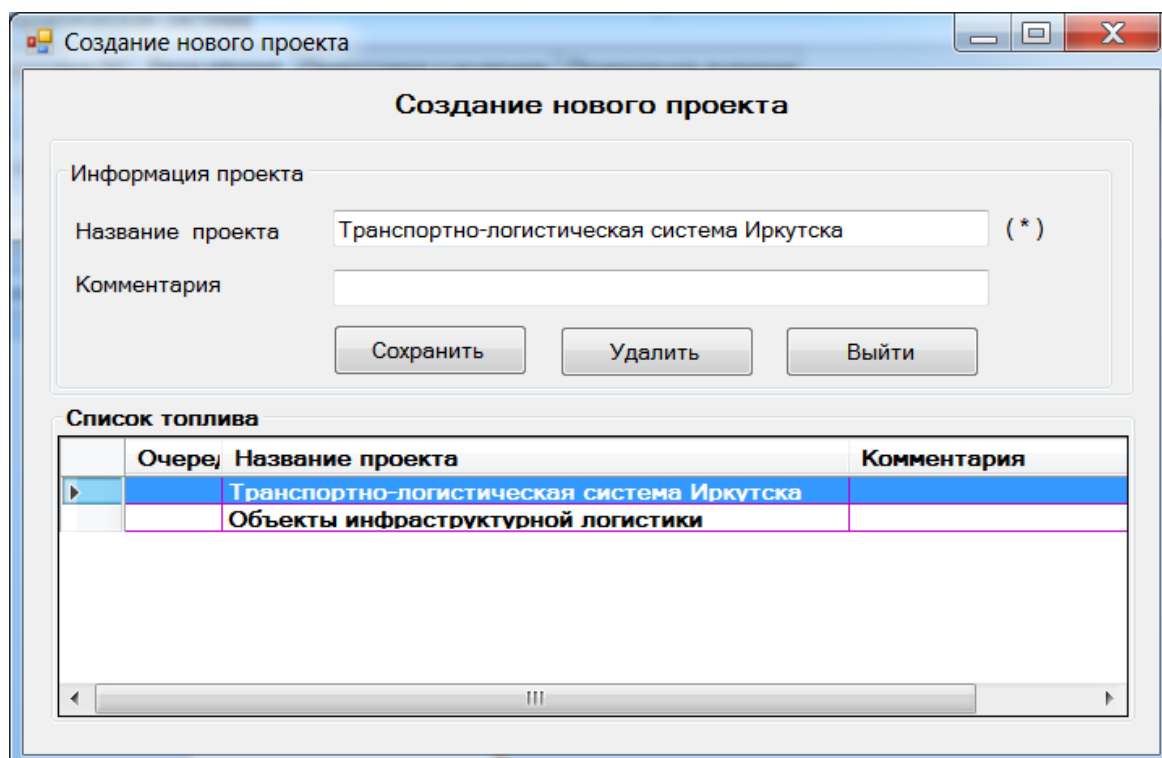


Рис.3.16. Создание нового проекта

В каждом проекте обязательно существует один «базовый» сценарий. Все другие сценарии проекта наследуются от базового и соответствующим образом модифицируются, в том числе могут быть созданы или удалены объекты ИЛ. Для

формирования базового сценария используется информация, представленная во вкладке «Ввод данных». Эта информация изначально была импортирована из КМИЛ, а затем отредактирована пользователями. Для включения конкретных объектов для последующего анализа необходимо, во-первых, выбрать текущий проект из списка (рис. 3.17а), во-вторых, добавить выбранный объект (рис. 3.17б).

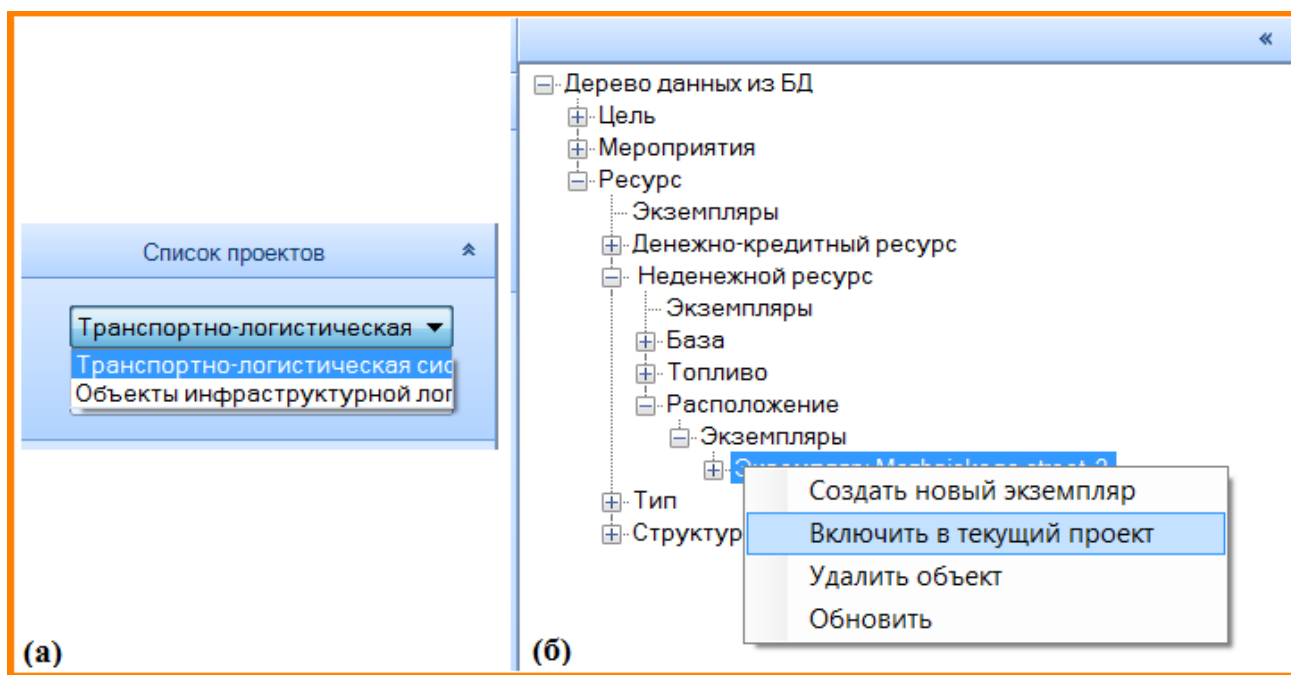


Рис. 3.17. (а) Включение объекта в текущий проект (б) Выбор текущего проекта

Выбранные объекты (рис. 3.17б) появятся в базовом сценарии текущего проекта (вкладка «Подготовка сценариев»). При этом объекты базового сценария тесно связаны с объектами главного дерева (вкладка «Ввод данных»), т.е. редактирование объектов базового сценария приведет и к изменению объектов из КМИЛ. Для создания «локальной» копии КМИЛ необходимо на основе базового сценария создать новый сценарий. Редактирование объектов нового сценария уже не будет отражаться на КМИЛ. В каждом сценарии существует два подмножества объектов: «исходные данные» и «результат». Объекты базового сценария изначально копируются в «исходные данные», а в подмножество «результат» сохраняются объекты после завершения сценарного анализа. Экранная форма для работы со сценариями показана на рис. 3.18.

3.2.7. Модуль «Анализ сценария с использованием системной базы знаний и интерактивного диалога с пользователем»

К настоящему времени реализовано несколько экранных форм для интерактивного диалога с пользователем в процессе сценарного анализа с использованием экспертной системы. Простые формы: сообщение, ввод данных, выбор одного или нескольких значений из списка (рис. 3.20); специализированные формы: редактирование информации об объекте ИЛ (аналогично автоматически генерируемому интерфейсу). При этом пользователь может задать запрос на решение проблемы ТЛС выбрать входные существующие или создать новые данные для ее решения. ЭС проверит и вызывает соответствующие вычислительные модули, которые решают полученные задачи из запроса, и передают результаты в ЭС. После этого ЭС сохраняет результат анализа сценария и выводит ответ (результаты решения конкретной задачи, рекомендации и т.д.) пользователю.

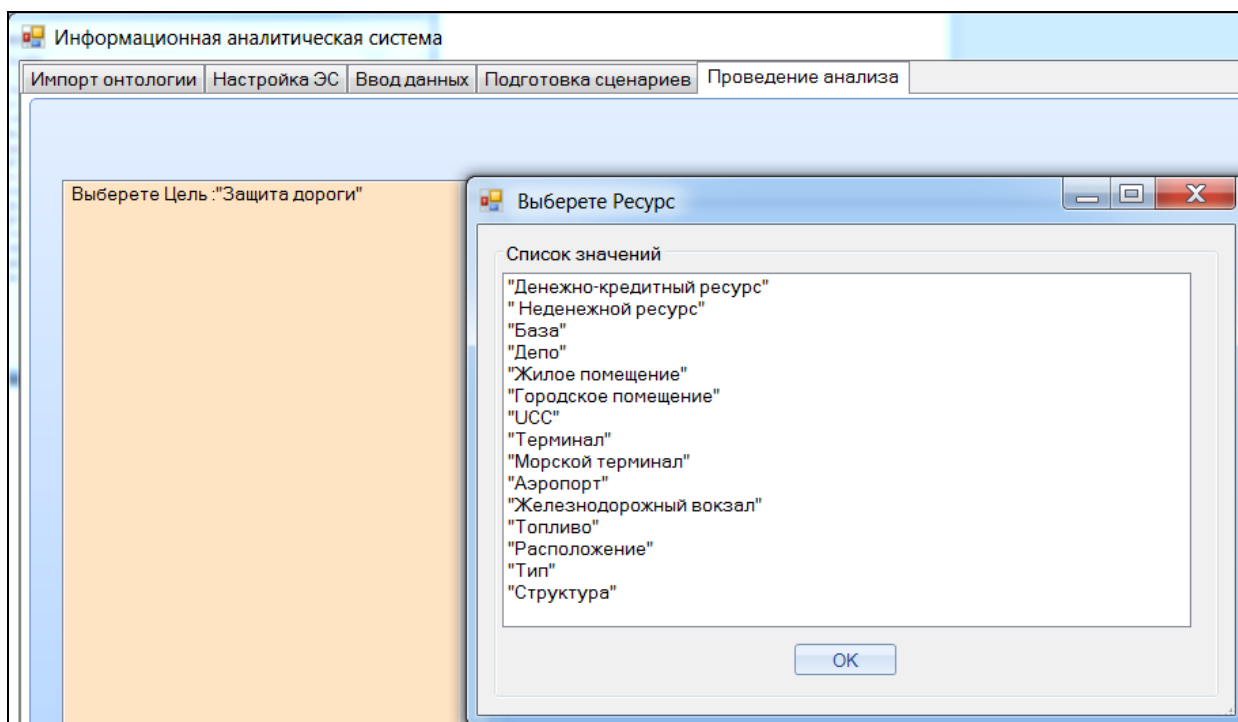


Рис. 3.20. Пример экранных форм диалога с пользователем для поддержки исследования транспортной логистики

Глава 4. Вычислительные эксперименты и прикладные задачи

4.1. Модельные примеры

Тестирование предложенных во второй главе алгоритмов, проведено с использованием ПК следующей конфигурации: Intel(R) Core(TM) i5–3570K (частота 3,4 ГГц, 8 Гб ОЗУ) и операционная система Windows 7. Алгоритмы реализованы на языке программирования C# с помощью пакета Visual Studio 2008.

4.1.1. Построение оптимальных упаковок равных кругов

Пример 1. В этом примере приведено сравнение полученных автором результатов с результатами из [135] для задачи упаковки в единичный квадрат равных кругов в евклидовой метрике $f(x, y) \equiv 1$ (табл. 4.1). Здесь n – количество упакованных кругов, R_{Known} – наилучший радиус упаковки из [135] для соответствующего числа кругов, R_{max} – наилучший радиус упаковки, найденный с помощью алгоритмов, $\Delta R = R_{Known} - R_{max}$ – отклонение радиуса, t – время работы алгоритмов FFS(119) (вычислены для ПК Intel Core 2 (частота 2,26 ГГц, 4 Гб ОЗУ)) и AGOPEC–MCS (в секундах). Число случайных генераций начальных положений $Iter = 25$ для количества кругов $n < 1000$ и $Iter = 10$ для $n \geq 1000$.

Отметим, что лучшие известные результаты, приведенные в [135], периодически обновляются и, фактически, представляют собой накопленные лучшие результаты за все годы исследований. Таким образом, можно считать, что мы сравним свои результаты с результатами всех предыдущих работ.

Пустые строки в табл. 4.1 означают отсутствие известных результатов для соответствующих n , а отрицательные отклонения означают, что представленный в настоящей работе алгоритм улучшает известные результаты.

Нетрудно видеть, что по сравнению с известными результаты, полученные при помощи предложенного алгоритма, несколько хуже, однако отклонение радиуса упаковочных кругов от оптимального не превосходит 0,12% (для $n < 50 - 0,001\%$). При этом общее время решения задачи относительно невелико даже для $n = 3000$ по сравнению с FSS-алгоритмом из [119]. Отсюда можно сде-

лать вывод, что предложенный алгоритм, несмотря на то, что он, строго говоря, напрямую не предназначен для решения задач об упаковке в евклидовой метрике, показывает и здесь достаточно хорошие результаты, особенно при достаточно больших значениях n .

Таблица 4.1.

Сравнение результаты упаковки единичного квадрата программы с лучшими известными результатами при евклидовой метрике

	Известные лучшие результаты	Результаты по FSS-алгоритму			Результаты по AGOPEC-MCS – алгоритму		
n	R_{Known}	$R_{\max}(C_i)$	ΔR	t (секунд)	$R_{\max}(C_i)$	ΔR	t (секунд)
50	0,07137710	0,071376623	0,000000481	276	0,070578606	0,000798498	88
55	0,06805536	0,068055219	0,000000141	286	0,067326392	0,000728968	105
60	0,06503041	0,065030109	0,000000304	350	0,064219475	0,000810937	154
65	0,06320396	0,063202599	0,000001358	487	0,062336383	0,000867574	178
70	0,06059669	0,060574482	0,000022212	517	0,059924009	0,000672684	196
75	0,05849454	0,058091304	0,000403232	621	0,057954653	0,000539883	217
80	0,05737068	0,057370675	0,000000009	750	0,056284924	0,001085760	264
85	0,05568018	0,055446138	0,000234044	978	0,054740839	0,000939342	290
90	0,05374995	0,053745827	0,000004121	1179	0,052814894	0,000935055	325
95	0,05242037	0,052302778	0,000117588	1423	0,051696657	0,000723709	350
100	0,05140107	0,051272763	0,000128308	1317	0,050269024	0,001132048	376
150	0,04214547	0,041976579	0,000168887	4107	0,041309389	0,000836076	895
200	0,03661280	0,025722283	0,010890516	8790	0,035969127	0,000643672	1457
250	0,03287632	0,030028915	0,002847403	18111	0,032102759	0,000773559	2246
300	0,03021956				0,029447787	0,000771768	3254
350	0,02795522				0,027246528	0,000708692	4294
400	0,02620235				0,025494351	0,000707994	5689
450	0,02479535				0,024040133	0,000755215	7199
500	0,02345550	0,000974943	0,022480556	133443	0,022846434	0,000609065	8725
750	0,01927449				0,018623105	0,000651382	13604
992	0,01587302				0,016236831	-0,00036382	19709
1000					0,016203402	0	25452
1500	0,01315790				0,013171861	-0,00001397	31713
1750					0,012198113	0	56750
2000	0,01172594				0,011516338	0,000209600	94265
2500					0,010224897	0	144541
3000	0,00967451				0,009243172	0,000431339	217489

Пример 2. Данный пример иллюстрирует работу алгоритма AGOPEC-MCS в случае, когда метрика задается формулой (2.2), где $f(x, y) = v_0(1 + ky)$, v_0 , k –

заданные константы. Такая метрика означает, что скорость распространения световых волн линейно возрастает с ростом координаты y . В работе [63] показано, что волновые фронты в этом случае так же, как и в евклидовой метрике имеют форму окружности, однако источник волны (центр круга) оказывается смещен. Результаты расчетов представлены в табл. 4.2. Здесь n – количество упакованных кругов; $R_{\max}(C_i)$, $d_{\max}(C)$ – наилучшие радиус и плотность упаковки, соответственно, найденные с помощью предложенного авторами алгоритма; $d(B)$ – плотность барьеров; t – время работы алгоритма (в секундах). Число случайных генераций начальных положений $Iter = 10$. На рис. 4.1 представлено графическое отображение результатов из табл. 4.2. Начало координат расположено в верхнем левом углу, ось Oy направлена вниз. Серым цветом показаны барьеры. Для каждого вписанного круга отмечены центр в заданной метрике (верхняя точка) и центр в евклидовой метрике (нижняя точка). Отметим, что в заданной метрике представленные на рис. 4.1 круги имеют одинаковый радиус.

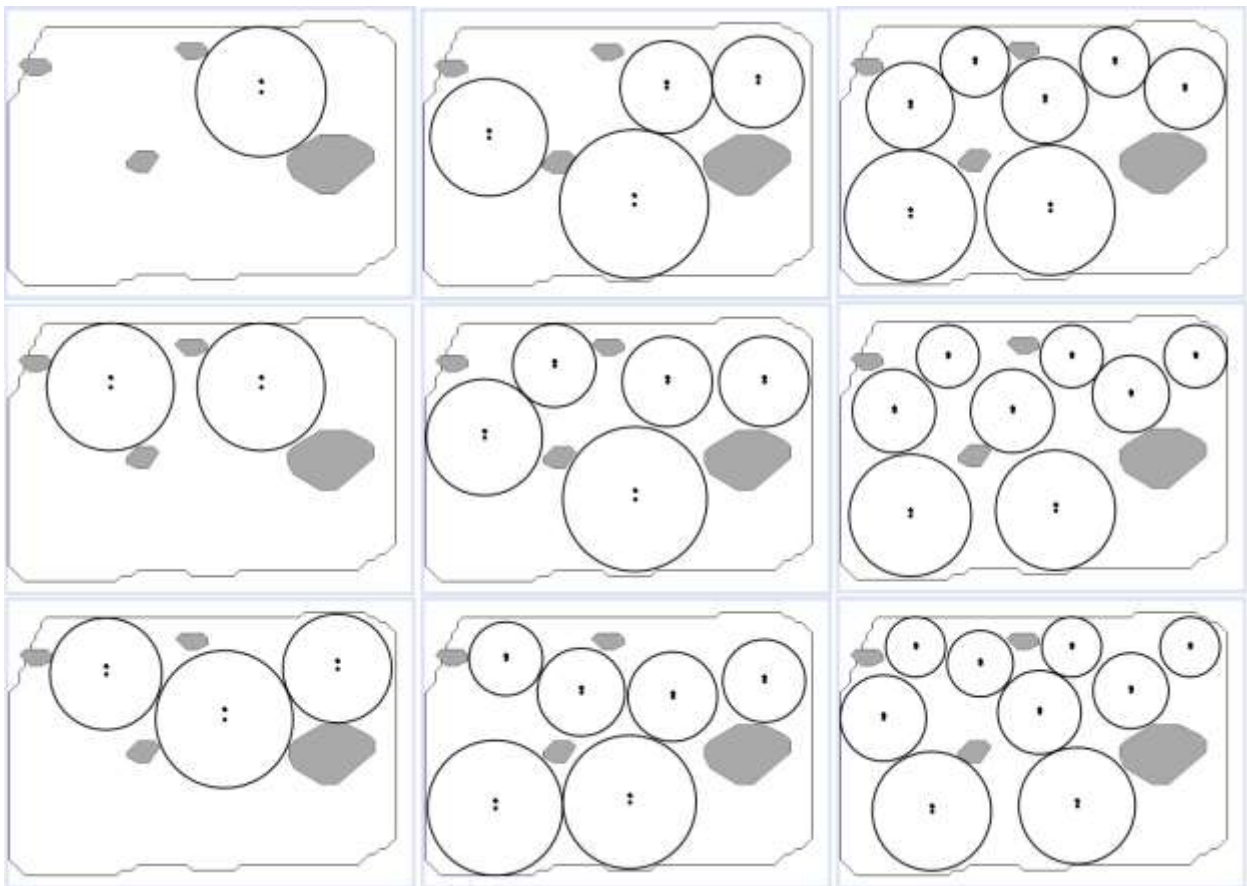


Рис. 4.1. Результаты работы алгоритма AGOPEC-MCS в метрике с линейной функцией $f(x, y)$

Результаты работы алгоритма AGOPEC-MCS в метрике с линейной функцией $f(x, y)$

n	$R_{\max}(C_i)$	$d_{\max}(C)$	$d(B)$	t_{ex} (секунд)
1	25,0243588672	0,16490486258	0,0274841438	3,478
2	21,9528824962	0,38054968288	0,0274841438	22,73
3	19,6029948175	0,47145877378	0,0274841438	35,412
4	16,4013415521	0,61057082452	0,0274841438	58,906
5	14,9186042515	0,57885835095	0,0274841438	77,049
6	13,6410597604	0,68344608879	0,0274841438	83,882
7	13,2273584546	0,5911205074	0,0274841438	95,644
8	12,0273463313	0,5682875264	0,0274841438	97,314
9	11,5874805655	0,58308668076	0,0274841438	108,42
10	11,4274463314	0,60253699789	0,0274841438	123,054
11	10,7273705875	0,58900634249	0,0274841438	127,578
12	10,2273694524	0,62156448202	0,0274841438	134,587

Пример 3. В данном примере метрика определяется по формуле (2.2) со следующей $f(x, y)$:

$$a(x, y) = \frac{(x - 3.5)^2 + (y - 3.5)^2}{1 + (x - 3.5)^2 + (y - 3.5)^2}, \quad f(x, y) = \begin{cases} 0.3, & a(x, y) \leq 0.3 \\ a(x, y) & \\ 0.8, & a(x, y) \geq 0.8 \end{cases}$$

На рис. 4.2 показана функция $f(x, y)$, значения которой увеличиваются от темного к светлому.

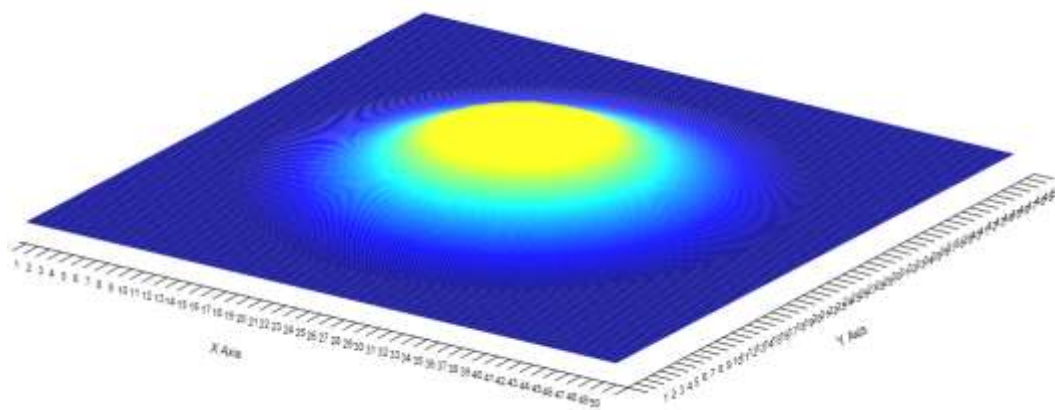


Рис. 4.2. 3D вид функции $f(x, y)$

Подобные метрики используются в инфраструктурной логистике, когда требуется разместить определенное количество однотипных обслуживающих объектов (магазинов, банкоматов и т.п.) на холмистой местности, предполагая,

что скорость движения зависит от угла подъема или спуска.

Таблица 4.3

Максимальные радиусы упаковочных кругов

n	$R_{\max}(C_i)$	$d_{\max}(C)$	$d(B)$	t_{ex} (секунд)
1	26,01653189	0,2841437632135	0,10792998017	3,557
2	20,87023745	0,2968287526427		35,116
3	17,773532697	0,3830866807611		41,122
4	14,415223918	0,3826638477801		53,321
5	13,394001606	0,4025369978858		60,279
6	12,3281904158	0,4097251585624		73,866
7	11,9309576301	0,3953488372093		79,092
8	11,3310576006	0,4452431289641		98,343
9	10,7652374503	0,4549682875264		109,435

Отметим, что форма волнового фронта зависит от расположения источника относительно начала координат и заранее не известна. Барьеры в данном случае являются кругами. Полученные результаты показаны на рис. 4.3 в табл. 4.3. Из рис. 4.3 можно видеть, что алгоритм AGOPEC-MCS показывает хорошие результаты даже для достаточно сложных метрик. Заметим, что, как и в предыдущем примере, в заданной метрике, представленные на рис. 4.2 упакованные фигуры являются окружностями с равными радиусами.

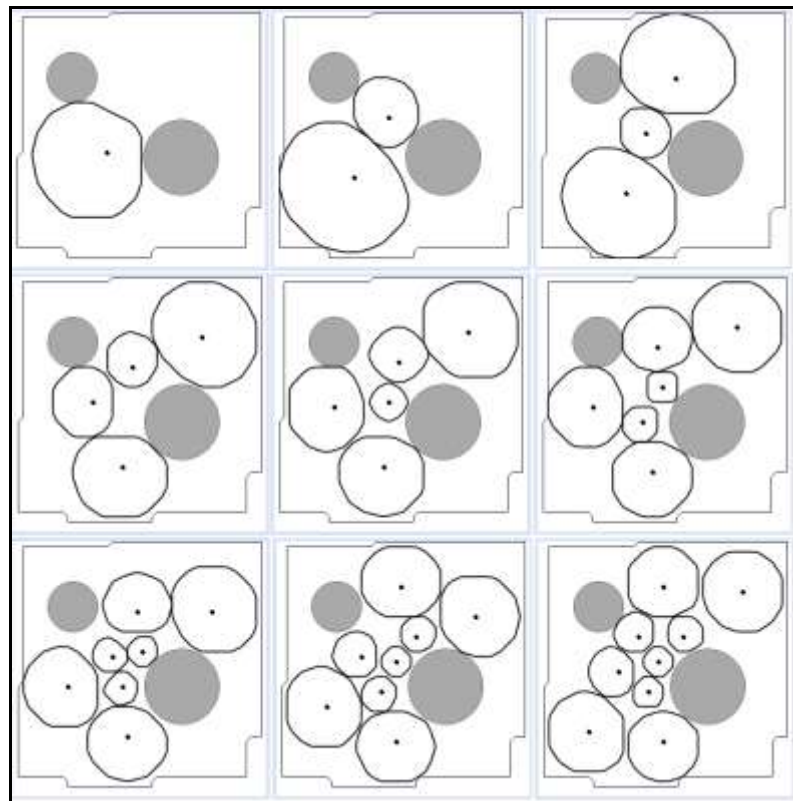


Рис. 4.3. Результаты работы алгоритма AGOPEC-MCS (пример 3)

Пример 4. Рассматриваем пример, при котором метрика определяется следующим образом:

$$a_1(x, y) = \frac{(x - 2.5)^2 + (y - 2.5)^2}{1 + (x - 2.5)^2 + (y - 2.5)^2}, f_1(x, y) = \begin{cases} 0, & a_1(x, y) \geq 0.8 \\ a_1(x, y) \end{cases}$$

$$a_2(x, y) = \frac{(x - 2.5)^2 + (y - 7.5)^2}{1 + (x - 2.5)^2 + (y - 7.5)^2}, f_2(x, y) = \begin{cases} 0, & a_2(x, y) \geq 0.8 \\ a_2(x, y) \end{cases}$$

$$a_3(x, y) = \frac{(x - 5.0)^2 + (y - 5.0)^2}{1 + (x - 5.0)^2 + (y - 5.0)^2}, f_3(x, y) = \begin{cases} 0, & a_3(x, y) \geq 0.8 \\ a_3(x, y) \end{cases}$$

$$a_4(x, y) = \frac{(x - 7.5)^2 + (y - 2.5)^2}{1 + (x - 7.5)^2 + (y - 2.5)^2}, f_4(x, y) = \begin{cases} 0, & a_4(x, y) \geq 0.8 \\ a_4(x, y) \end{cases}$$

$$a_5(x, y) = \frac{(x - 7.5)^2 + (y - 7.5)^2}{1 + (x - 7.5)^2 + (y - 7.5)^2}, f_5(x, y) = \begin{cases} 0, & a_5(x, y) \geq 0.8 \\ a_5(x, y) \end{cases}$$

$$F(x, y) = f_1(x, y) + f_2(x, y) + f_3(x, y) + f_4(x, y) + f_5(x, y)$$

$$f(x, y) = \begin{cases} 0.4, & 0 < F(x, y) \leq 0.4 \\ F(x, y) \\ 0.8, & F(x, y) > 0.8 \end{cases}$$

Граница области и барьеры показаны на рис. 4.4.

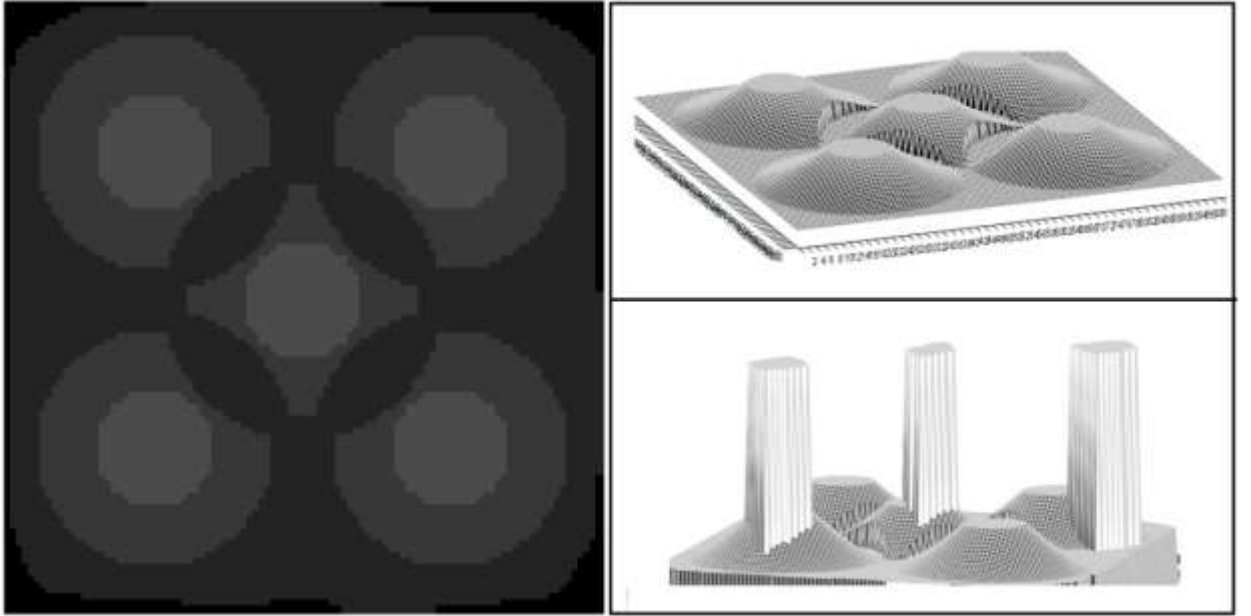


Рис. 4.4. Отображение среды, границы, и барьеры в области

Результаты вычислений показаны на рис. 4.5. и в табл. 4.4.

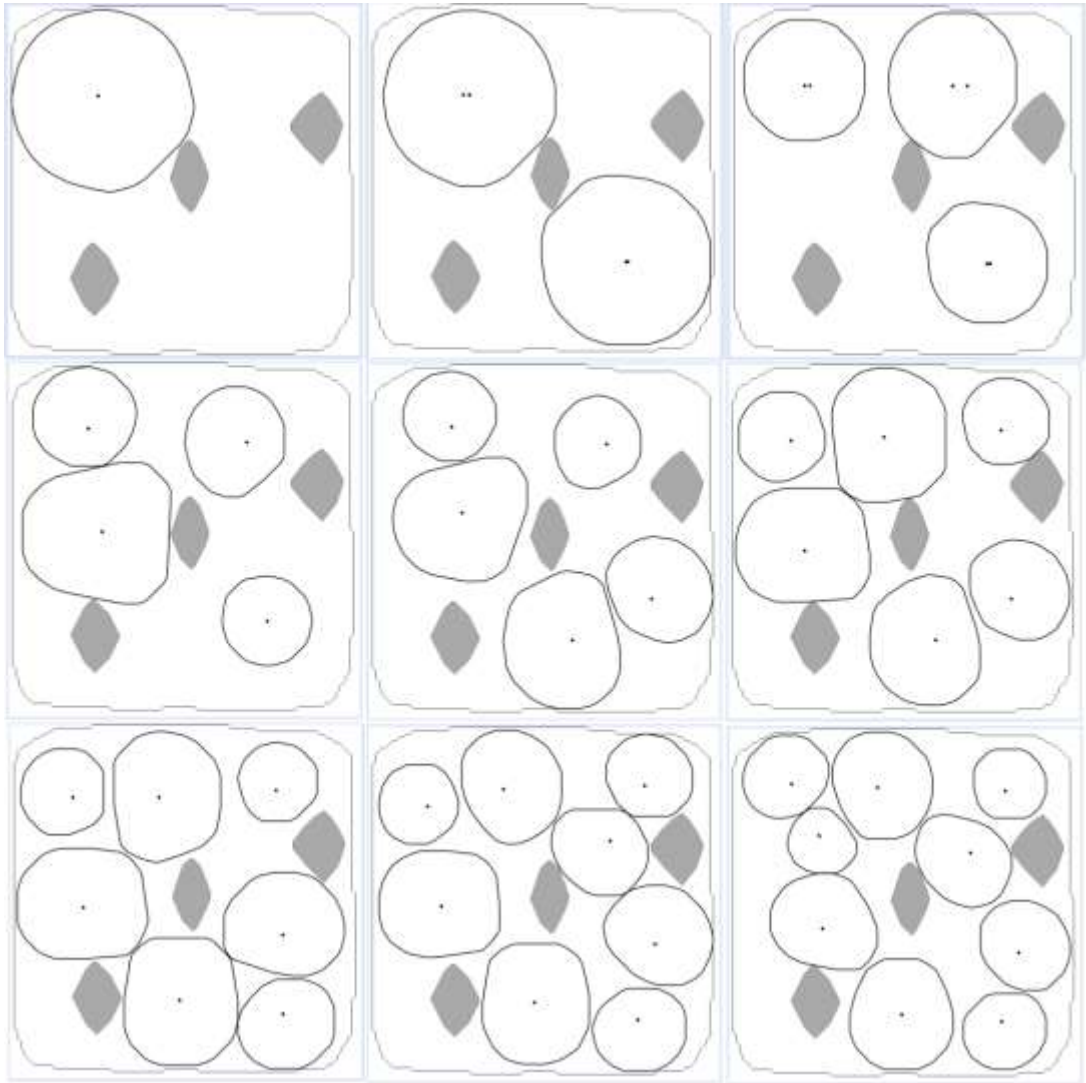
Рис. 4.5. Упаковка в заданное множество n кругов ($n=1-9$)

Таблица 4.4

Максимальные радиусы упакованных кругов

n	$R_{\max}(C_i)$	$d_{\max}(C)$	$d(B)$	t_{ex} (секунд)
1	47,8437209425	0,2215776081	0,0547582697	12,012
2	46,4525834620	0,4090585242		107,75
3	36,8639545920	0,3391348601		119,17
4	29,9909202978	0,3540966921		152,569
5	26,8089520229	0,4181170483		214,221
6	25,8675733791	0,5111450382		243,829
7	24,8690265518	0,5726208652		278,743
8	23,2032542107	0,5640712468		316,37
9	21,1387736935	0,5173536896		341,314

Из приведенных выше примеров можно видеть, что предложенный алгоритм показал хорошее решение не только в известных случаях, но во всех рассмотренных задачах.

4.1.2. Построение оптимальных покрытий равных кругов

Пример 5. В этом примере приведено сравнение полученных автором результатов с результатами из [130] для задачи покрытия единичного квадрата равными кругами в евклидовой метрике $f(x, y) \equiv 1$ (табл. 4.5). Здесь n – количество покрывающих кругов, R_{Known} – наилучший радиус покрытия из [130] для соответствующего числа кругов, R_{min} – наилучший радиус покрытия, найденный с помощью предложенного автором алгоритма, $\Delta R = R_{Known} - R_{min}$, t – время работы алгоритма. Число случайных генераций начальных положений $Iter = 25$.

Таблица 4.5

Покрытие единичного квадрата в евклидовой метрике

n	R_{Known}	R_{min}	ΔR	t (секунд)
10	0,218233512793	0,218233693441	0,000000180648	43,477
11	0,212516016493	0,213107846578	0,000591830085	44,175
12	0,202275889208	0,202747599284	0,000471710076	45,132
13	0,194312371432	0,194342244670	0,000029873238	46,687
14	0,185510547260	0,185635185242	0,000124637981	47,143
15	0,179661759933	0,180281054179	0,000619294246	47,611
16	0,169427051598	0,169711748759	0,000284697161	48,516
17	0,165680929571	0,166406071981	0,000725142410	49,593
18	0,160639663597	0,160864969889	0,000225306292	50,463
19	0,157841981747	0,158309478760	0,000467497013	50,81
20	0,152246811233	0,152426892598	0,000180081365	51,683
21	0,148953789551	0,149133934021	0,000180144470	52,853
22	0,143693177122	0,144275792440	0,000582615318	53,165
23	0,141244822388	0,142205886841	0,000961064453	53,914
24	0,138302883283	0,139246317546	0,000943434263	54,522
25	0,133548706561	0,134470667521	0,000921960960	54,835
26	0,131764875615	0,132809473674	0,001044598059	56,441
27	0,128633534503	0,129617869059	0,000984334556	57,05
28	0,127317553466	0,128024566650	0,000707013185	57,892
29	0,125553507964	0,126029658000	0,000476150036	58,469
30	0,122036868819	0,123001449585	0,000964580766	58,984
35		0,115444819132		62,884
40		0,108376286825		67,221
45		0,102320378621		71,605
50		0,095904051463		73,508
75		0,078877824148		88,141
100		0,068332659403		128,342
150		0,05554107666		144,831
500		0,03082452774		408,707

Пустые строки в табл. 4.5 означают отсутствие известных результатов для соответствующих n . Несложно увидеть, что по сравнению с известными (оптимальными), результаты, полученные автором, несколько хуже, однако отклонение радиуса покрывающих кругов от оптимального не превосходит 0,1% (для $n < 10$ – 0,0001%). При этом общее время решения задачи относительно невелико даже для $n = 500$.

Пример 6. Введем следующую («линейную») метрику: $f(x, y) = v_0(1 + kx)$, где v_0, k постоянны, т.е. скорость движения растет линейно с ростом координаты x ; покрываемое множество является выпуклым многоугольником. Результаты расчетов для $v_0 = 1/2$; $k = 1/10$ показаны на рис. 4.6 и в табл. 6.

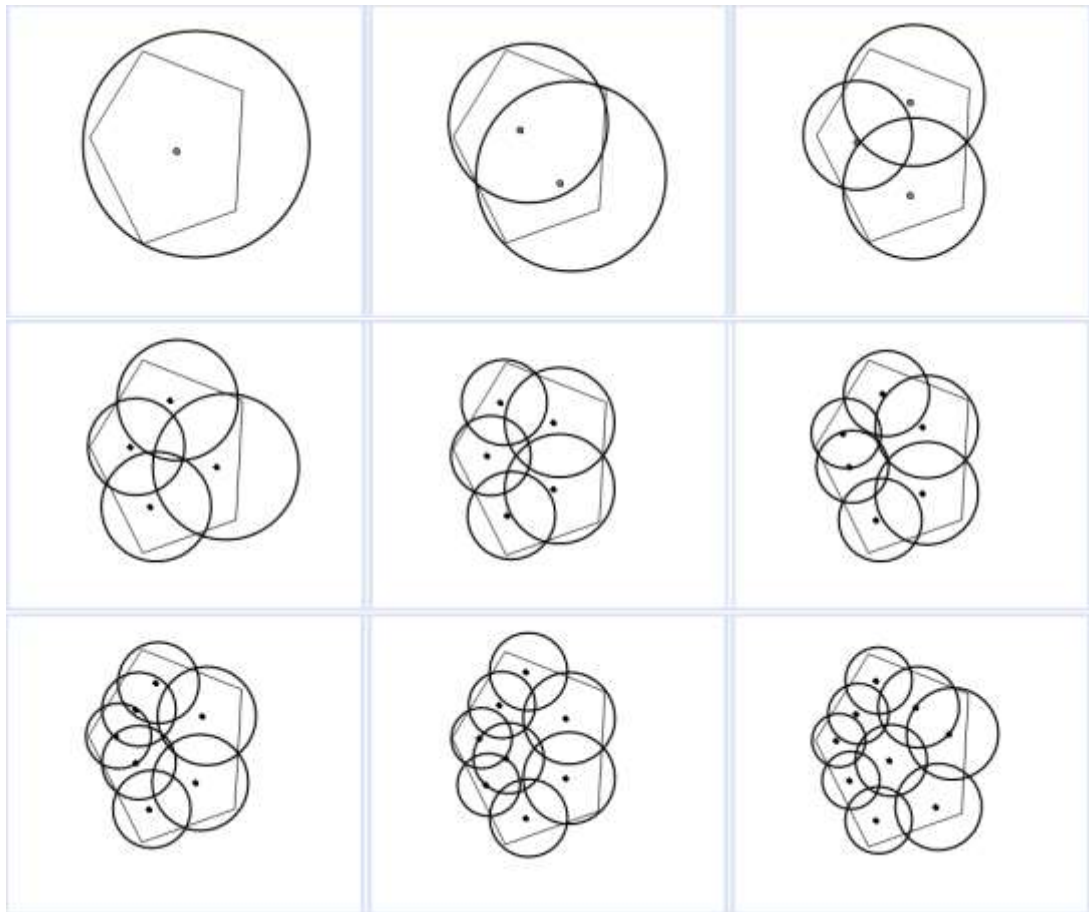


Рис. 4.6. Оптимальные покрытия равными кругами для многоугольника

В работе [63] установлено, что круги в «линейной» метрике для евклидовой метрики также являются кругами, но со смещенным центром. Из рис. 4.6 можно видеть, что результаты численного решения хорошо соотносятся с теоретическими.

Покрывание выпуклого многоугольника в неевклидовой метрике

n	$R_{\min}$	t (секунд)
1	10,514174027378	1,014
2	8,229119718281	5,226
3	6,549338964147	12,797
4	5,774430168742	20,561
5	4,826911112031	27,566
6	4,431735800433	30,499
7	4,167997532613	36,146
8	3,888225133265	41,418
9	3,579772457365	45,287

Кроме того, отметим, что в «линейной» метрике радиусы покрывающих кругов одинаковы (см. табл. 4.6), а при переходе к евклидовой метрике они зависят от расположения центров: чем ближе к оси Oy , тем меньше радиус.

Пример 3. Введем следующую метрику:

$$\begin{aligned}
 a_1(x, y) &= \frac{(x-2.5)^2 + (y-2.5)^2}{1 + (x-2.5)^2 + (y-2.5)^2}, \quad f_1(x, y) = \begin{cases} 0, & a_1(x, y) \geq 0.8, \\ a_1(x, y), & a_1(x, y) < 0.8, \end{cases} \\
 a_2(x, y) &= \frac{(x-2.5)^2 + (y-7.5)^2}{1 + (x-2.5)^2 + (y-7.5)^2}, \quad f_2(x, y) = \begin{cases} 0, & a_2(x, y) \geq 0.8, \\ a_2(x, y), & a_2(x, y) < 0.8, \end{cases} \\
 a_3(x, y) &= \frac{(x-7.5)^2 + (y-2.5)^2}{1 + (x-7.5)^2 + (y-2.5)^2}, \quad f_3(x, y) = \begin{cases} 0, & a_3(x, y) \geq 0.8, \\ a_3(x, y), & a_3(x, y) < 0.8, \end{cases} \\
 a_4(x, y) &= \frac{(x-7.5)^2 + (y-7.5)^2}{1 + (x-7.5)^2 + (y-7.5)^2}, \quad f_4(x, y) = \begin{cases} 0, & a_4(x, y) \geq 0.8, \\ a_4(x, y), & a_4(x, y) < 0.8, \end{cases} \\
 F(x, y) &= f_1(x, y) + f_2(x, y) + f_3(x, y) + f_4(x, y), \\
 f(x, y) &= \begin{cases} 0.4, & 0 < F(x, y) \leq 0.4, \\ F(x, y), & 0.4 < F(x, y) < 0.8, \\ 0.8, & F(x, y) = 0.8. \end{cases}
 \end{aligned}$$

Данная метрика может применяться, к примеру, при описании движения по «холмистой» местности, когда скорость зависит от угла подъема или спуска либо по «городу», когда средняя скорость в пригороде постоянна, затем уменьшается при движении от окраин к центру и снова становится постоянной (но меньшей, по сравнению с пригородом) в районе центра и т.п.

Покрываемое множество в данном случае является невыпуклым много-

угольником.

Таблица 4.7

Покрывание невыпуклого многоугольника в неевклидовой метрике

n	$R_{\min}$	t (секунд)
1	48,3509591310017	3,375
2	39,6535062156294	25,093
3	30,9690548988679	48,109
4	25,3337283136898	71,562
5	22,3412999182734	88,641
6	21,479024599642	108,734
7	19,4344397313216	126,078
8	18,2625236493321	142,922
9	17,6052158181218	168,688

На рис. 4.7 представлены результаты численного решения.

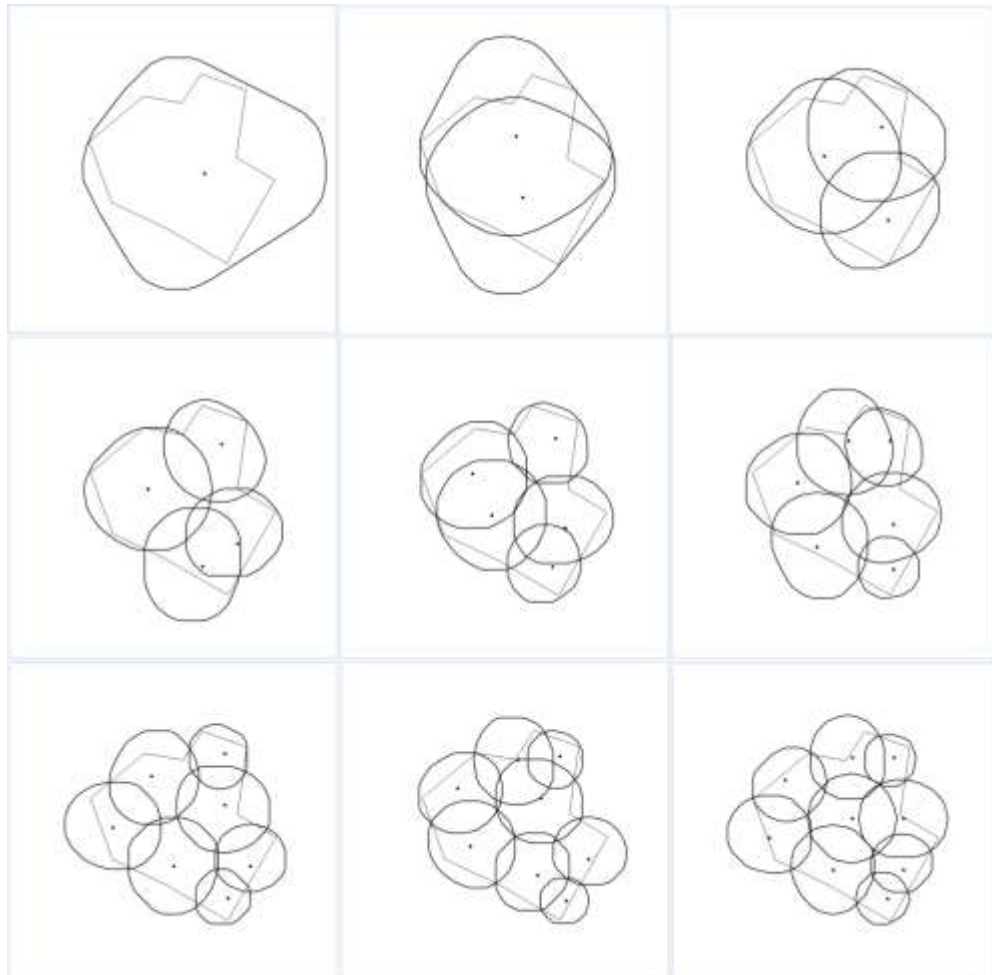


Рис. 4.7. Покрывание невыпуклого многоугольника в неевклидовой метрике

Можно убедиться, что площадь построенных покрытий во всех представленных случаях ненамного превосходит площадь покрываемого множества. Кроме того, с ростом количества кругов в покрытии уменьшается площадь перекры-

тий. Отметим также, что, как и в предыдущем примере в заданной метрике радиусы покрывающих кругов одинаковы (см. табл. 4.7). Из табл. 4.7 можно видеть, что время выполнения относительно невелико.

Вышесказанное позволяет сделать вывод о применимости предложенного автором подхода к решению задач о покрытии невыпуклых односвязных множеств со сложной неевклидовой метрикой.

4.1.3. Размещение логистических объектов в условиях кооперации и конкуренции

Пример 7. Данный пример иллюстрирует работу алгоритмов размещения дополнительных логистических центров в условиях кооперации и конкуренции при $f(x, y) \equiv 1$ и наличии непроходимых барьеров.

Пусть в некотором районе расположены два обслуживающих центра с координатами (6;13) и (55;18) (обозначены бело-черными кругами), в точке (7;14) расположены 25 клиентов, в (20;37) – 20 (обозначены домиками), в остальных точках уровень потребления равен единице (всего 58). На рис. 4.8а они отмечены черными точками, кроме того, выделены цветом соответствующие логистические зоны и отмечены кратчайшие маршруты доставки товара, черным показаны непроходимые участки. Начало координат расположено в левом нижнем углу. Результаты решения задач в кооперативной и конкурентной постановках представлены в табл. 4.8 и на рис. 4.8. Расположение новых узлов обозначено звездами.

Таблица 4.8

Результаты решения

	Суммарное время доставки (в у.е.)	Количество клиентов новых центров	Координаты новых центров
Начальное состояние	6831	—	—
Кооперация (Рис.4.8б)	798	42	(39;12); (20;37)
Конкуренция (Рис.4.8в)	2862	55	(37;22); (78;9)

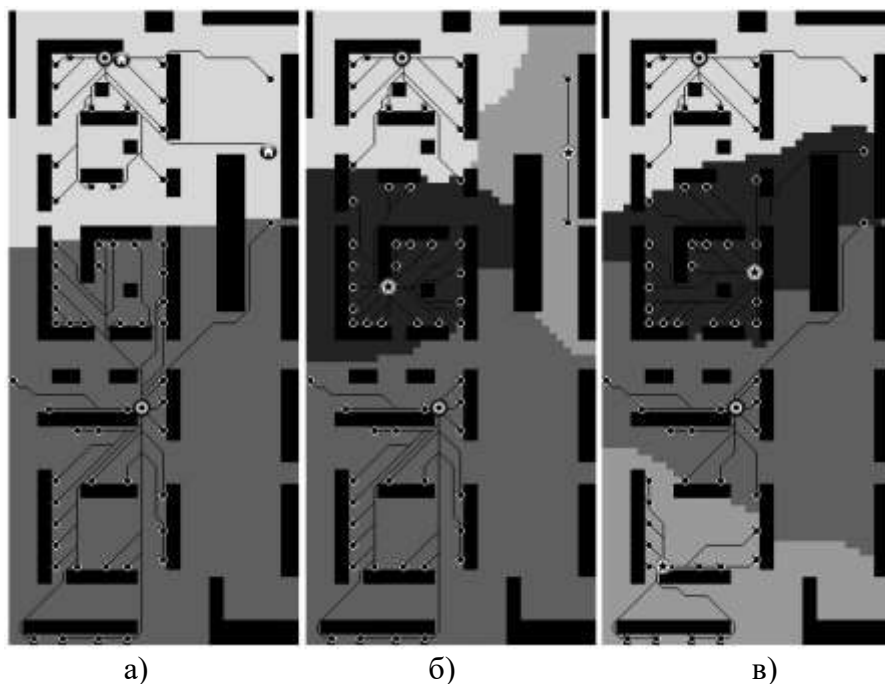


Рис. 4.8. а) Начальное расположение логистических зон; б) расположение при кооперативном вхождении; в) расположение при конкурентном вхождении

Отметим, что в первом случае расположение новых обслуживающих узлов совпало с наиболее крупным потребителем, что позволило значительно уменьшить суммарное время доставки (порядка 88,31%). Во втором случае был осуществлен захват достаточно большой доли рынка (более 53,4% клиентов), однако уменьшение суммарного времени обслуживания оказалось незначительным (более чем на 58,1%) по сравнению с первым случаем.

Пример 8. Данный пример иллюстрирует работу алгоритмов в неевклидовой метрике, порожденной функцией $f(x, y)$, которая определяется следующим образом.

Обозначим

$$a(x, y) = \frac{(x - 3.5)^2 + (y - 3.5)^2}{1 + (x - 3.5)^2 + (y - 3.5)^2}, \quad f(x, y) = \begin{cases} 0.2, & a(x, y) \leq 0.2 \\ a(x, y), & 0.2 < a(x, y) < 0.8 \\ 0.2, & a(x, y) \geq 0.8 \end{cases}$$

Кроме того, имеются непроходимые области (на рис. 4.9 показаны в виде ям).

Обслуживающие центры расположены в точках (47;17) и (36;59). Расположение потребителей показано на рис. 4.9, причем в точке (11;61) находятся 15 клиентов из 46.

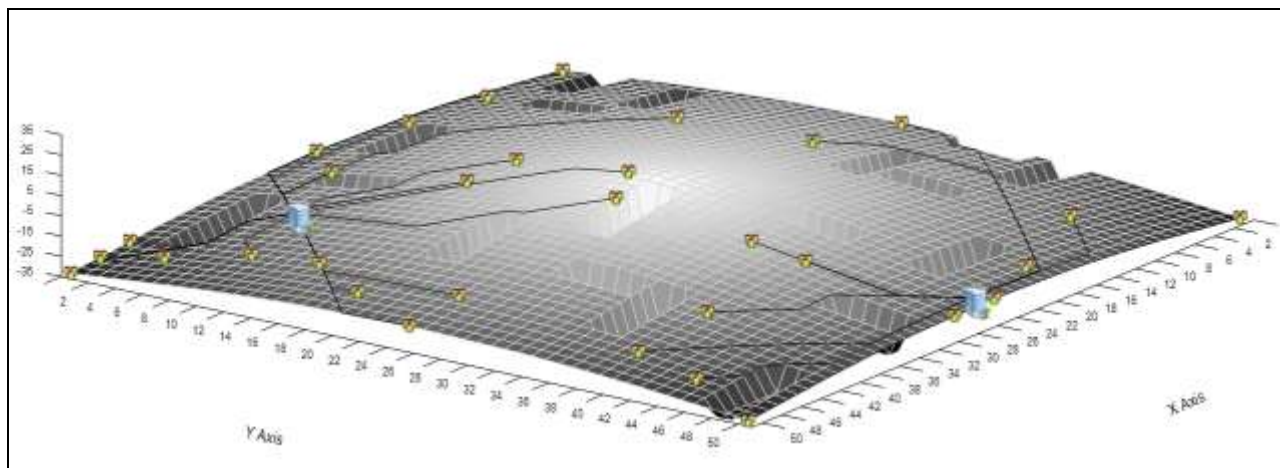


Рис. 4.9. Начальное расположение логистических зон для неевклидовой метрики

Таблица 4.9

Результаты решения

	Суммарное время доставки (в у.е.)	Количество клиентов новых центров	Координаты новых центров
Начальное состояние	138.92	—	—
Кооперация	25.86	16	(25;23)
Конкуренция	116.37	25	(25;9)

Из табл. (4.9) можно видеть, что в первом случае произошло значительное (более чем в 5 раз) уменьшение суммарного времени обслуживания, при этом логистическая зона нового центра оказалась сравнимой с существующими (около 34% клиентов). Это объясняется существенно неоптимальным начальным расположением обслуживающих центров в своих логистических зонах. В случае конкурентного размещения логистическая зона нового центра включает более 54% потребителей.

4.1.4. Решение задачи построения системы коммуникаций

Пример 9. Данный пример иллюстрирует работу алгоритма GMST в неевклидовой метрике, которая порождается функцией $f(x, y) = (x^2 + y^2) / (1 + x^2 + y^2)$ в области $D = \{-2 \leq x \leq 2; -2 \leq y \leq 2\}$ (рис. 4.10).

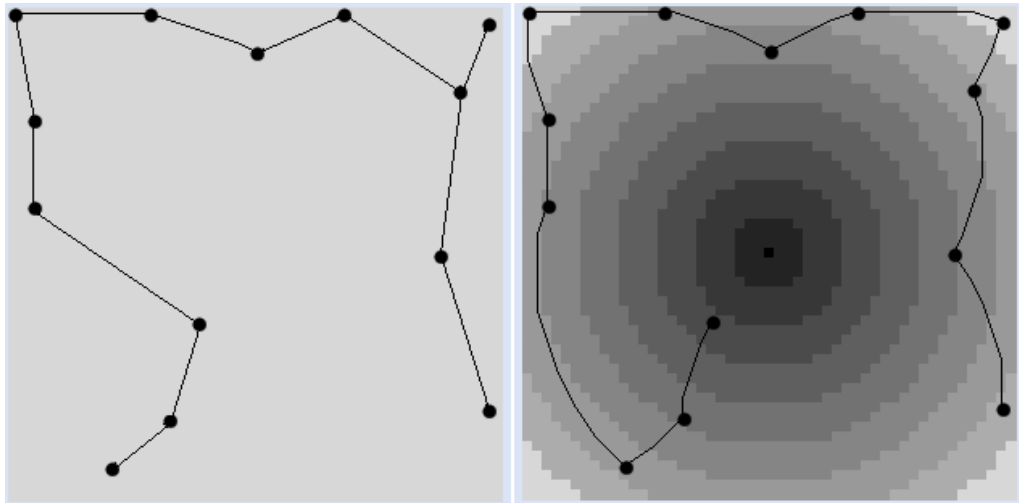


Рис. 4.10. Минимальное остовное дерево: слева – в евклидовой метрике, справа – в метрике, порожденной функцией $f(x, y)$

Нетрудно видеть, что построенные деревья существенным образом отличаются друг от друга, как по составу ребер, так и по их геометрии, поскольку в первом случае все ребра, очевидно, являются отрезками прямых, а во втором – часть ребер искривлена.

Пример 10. Здесь решается задача из примера 1 в случае, когда в систему можно вводить неограниченное число перекрестков. Результаты решения представлены на рис. 4.11. Длина минимального остовного дерева составляет 6437, дерева с одним «перекрестком» – 65998, с двумя – 65885, с тремя – 65702, и с четырьмя – 65698 условных единиц. Введение пятой и последующих точек Штейнера в соответствии с предложенным алгоритмом не приводит к дальнейшему сокращению длины дерева, что, впрочем, не означает неулучшаемость полученного решения с помощью каких-либо других алгоритмов.

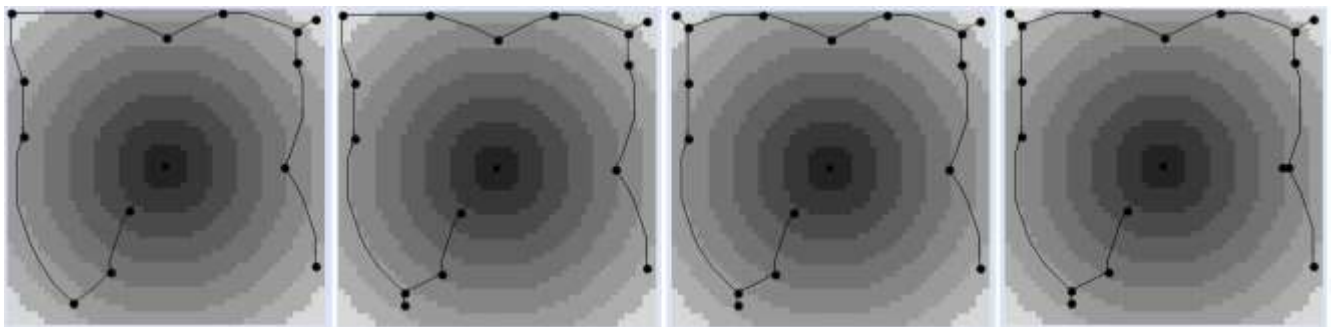


Рис. 4.11. Дерево с одной, двумя, тремя и четырьмя точками Штейнера

4.1.5. Решение задачи оптимизации ремонта автомобильных дорог в условиях ограниченного финансирования

В представленных ниже иллюстративных примерах дорога имеет форму прямоугольника (рис. 4.12, 4.13) с началом координат в левом нижнем угле. Как уже отмечалось выше, это допустимо, поскольку рассматривается небольшой участок. Все размеры указаны в условных единицах.

Пример 11. На рис. 4.12 представлена маска покрытия фрагмента реально существующей дороги, подлежащей ремонту, которая получена с помощью методов распознавания и классификации дефектов на основе статистического анализа и машинного обучения (Random Forests) (авторы Сидоров Д.Н., Жуков А.В.) из набора данных, предоставленного организацией Centre for Telecommunications and Multimedia, INESC TEC, Portugal.

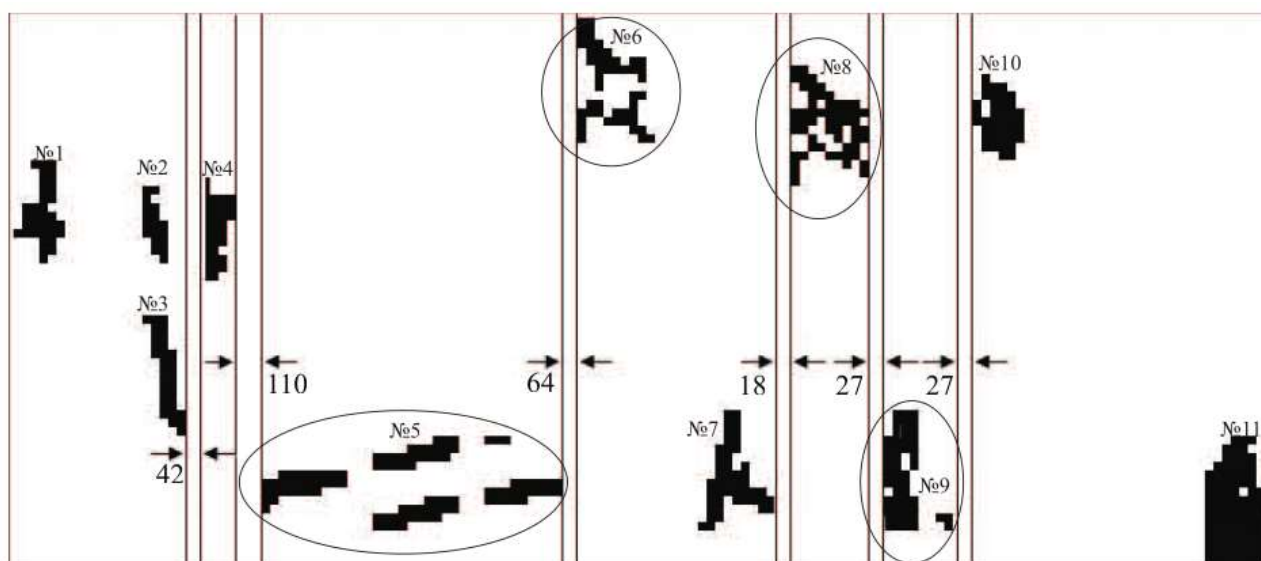


Рис. 4.12. Маска дорожного покрытия

Здесь дефекты обозначены черным цветом, протяженные области, не содержащие дефектов, удалены, их расположение и размеры обозначены двойной вертикальной чертой. Характеристики отдельных дефектов представлены в табл. 4.10. Поскольку предложенный метод на настоящий момент не позволяет определять глубину дефекта, в данном примере предполагается, что стоимость ремонта пропорциональна площади.

Таблица 4.10

Характеристики дефектов

Номер дефекта	Границы (левая, правая, верхняя, нижняя)	Площадь	тип
1	10,15,35,46	34	Выбоина
2	25,29,15,28	30	Вертикальная трещина
3	25,27,35,43	18	Вертикальная трещина
4	71,79,33,45	85	Выбоина
5	189,223,4,14	97	Горизонтальная трещина
6	287,295,49,62	47	Аллигатор
7	301,309,4,17	47	Выбоина
8	327,335,44,57	58	Аллигатор
9	362,370,4,17	52	Выбоина
10	397,402,47,56	39	Выбоина
11	424,430,0,14	89	Выбоина

В табл. 4.11 представлено решение задачи методом RBR, описанном в разделе 3, для различных вариантов бюджета. Требуется использовать не менее 90% бюджета.

Таблица 4.11

Результаты решения

Бюджет	Номер дефекта	Длина участка	Стоимость ремонта
50	2,3	5	48
100	5	20	97
150	8,9,10	76	149
200	7,8,9,10	102	196
250	8,9,10,11	104	238

Из табл. 4.11 можно видеть, что при небольшом бюджете, с точки зрения предложенного в разделе 2.5 критерия оптимальности и без учета глубины повреждения в первоочередном порядке следует устранять дефекты типа «вертикальная трещина», что вполне согласуется с рекомендациями экспертов. При увеличении финансирования в план ремонта попадают горизонтальные трещины, затем «аллигаторы» и выбоины, и, наконец, их комбинации.

Пример 12. Данный пример иллюстрирует работу алгоритма RBR, когда характеристикой состояния дорожного покрытия, наряду с расположением и

площадью, является и глубина дефектов (табл. 4.12). Стоимость восстановления принимается пропорциональной объему дефекта. Требуется использовать не менее 90% бюджета. На рис. 4.13 представлена модель покрытия фрагмента дороги, подлежащего ремонту.

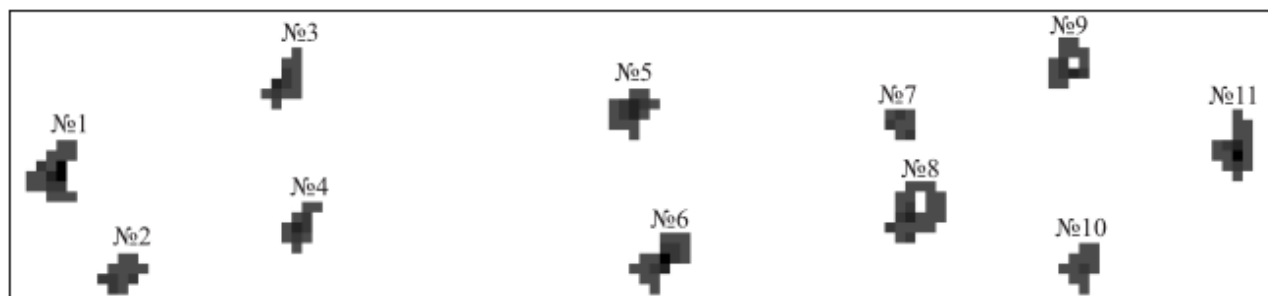


Рис. 4. 13. Модель участка дороги

Таблица 4.12

Характеристики дефектов

Номер дефекта	Границы (левая, правая, верхняя, нижняя)	Площадь	Объем
1	1,5,9,14	19	30
2	8,12,0,3	12	14
3	24,27,18,23	13	18
4	26,29,4,8	11	16
5	58,62,15,19	15	22
6	60,65,0,5	18	27
7	85,90,5,10	23	29
8	85,87,15,17	8	11
9	101,104,20,24	14	19
10	102,105,0,4	12	13
11	117,120,11,17	17	24

В табл. 4.13 представлено решение задачи для различных вариантов бюджета.

Таблица 4.13

Результаты решения

Бюджет	Номер дефекта	Длина участка	Стоимость ремонта
50	5,6	8	49
60	9,10,11	20	56
70	9,10,11	20	56
80	7,8,9,10	21	72
90	5,6,7,8	33	89
100	7,8,9,10,11	36	96

Из табл. 4.13 можно видеть, что при использовании критерия оптимальности из раздела 2.5, в план ремонта зачастую не попадают отдельно стоящие глубокие дефекты, кроме того, не всегда удается использовать дополнительное финансирование (например, для 60 и 70 у.е. ремонтируемые участки совпадают). Однако нетрудно убедиться в том, что найденные решения действительно являются оптимальными с точки зрения рассматриваемого критерия.

4.1.6. Применение ИИС для поддержки исследования транспортной логистики

С использованием функций, которые показаны в разделе 3.2, выполнены загрузка онтологии, подготовлены БД, БЗ и др. Рассмотрим простой пример использования ИИС для поддержки исследования транспортной логистики.

Во-первых, загружаем нужные правила базы знаний, которые предназначены для получения новых фактов (результаты) из известных фактов (рис. 4.14) (выходных параметров). Пользователь может использовать существующие БЗ для решения некоторых поддерживающих задач или сами создать свои БЗ (для исследователей).

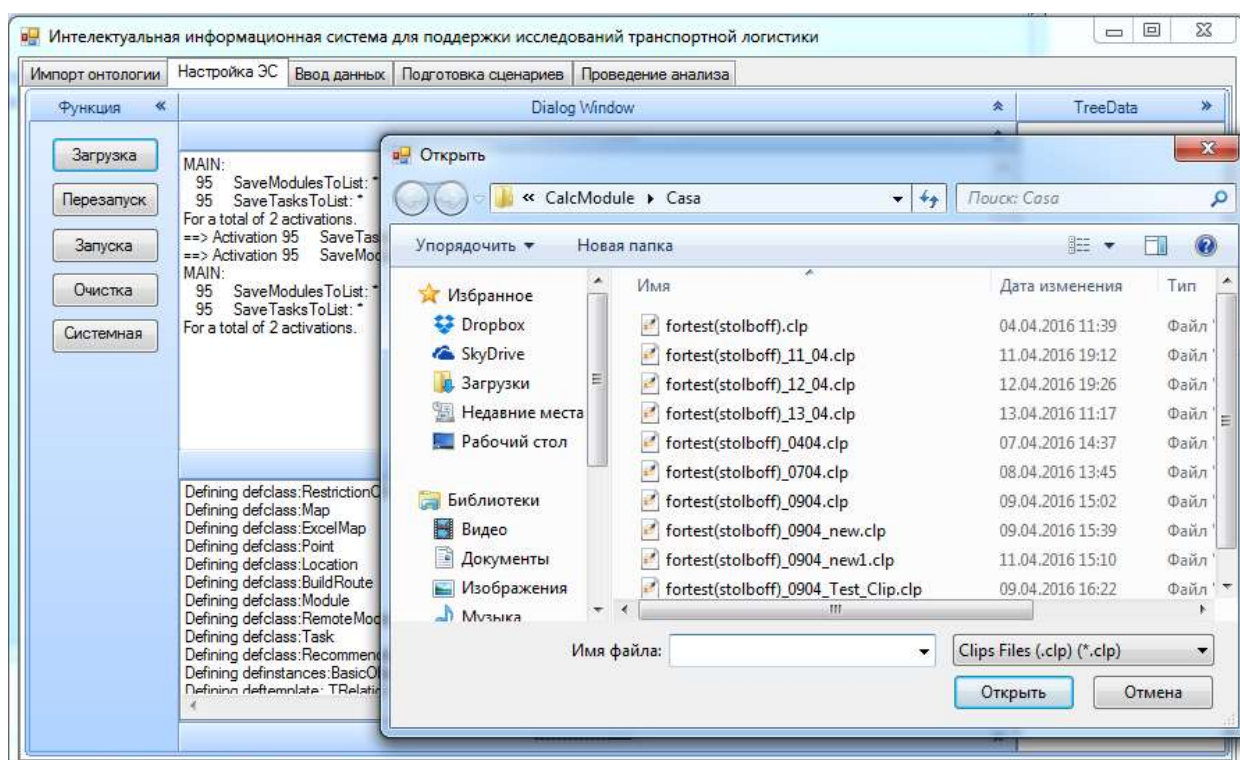


Рис. 4.14. Загрузка правил базы знаний

После этого запускается ЭС, и ЭС автоматически выводит сообщение или варианты для пользователя. По шагу выполнения пользователь будет получать результаты из начальных запросов. Ниже показано несколько форм интерактивных диалогов между ИИС и пользователем (рис. 4.15–4.20).

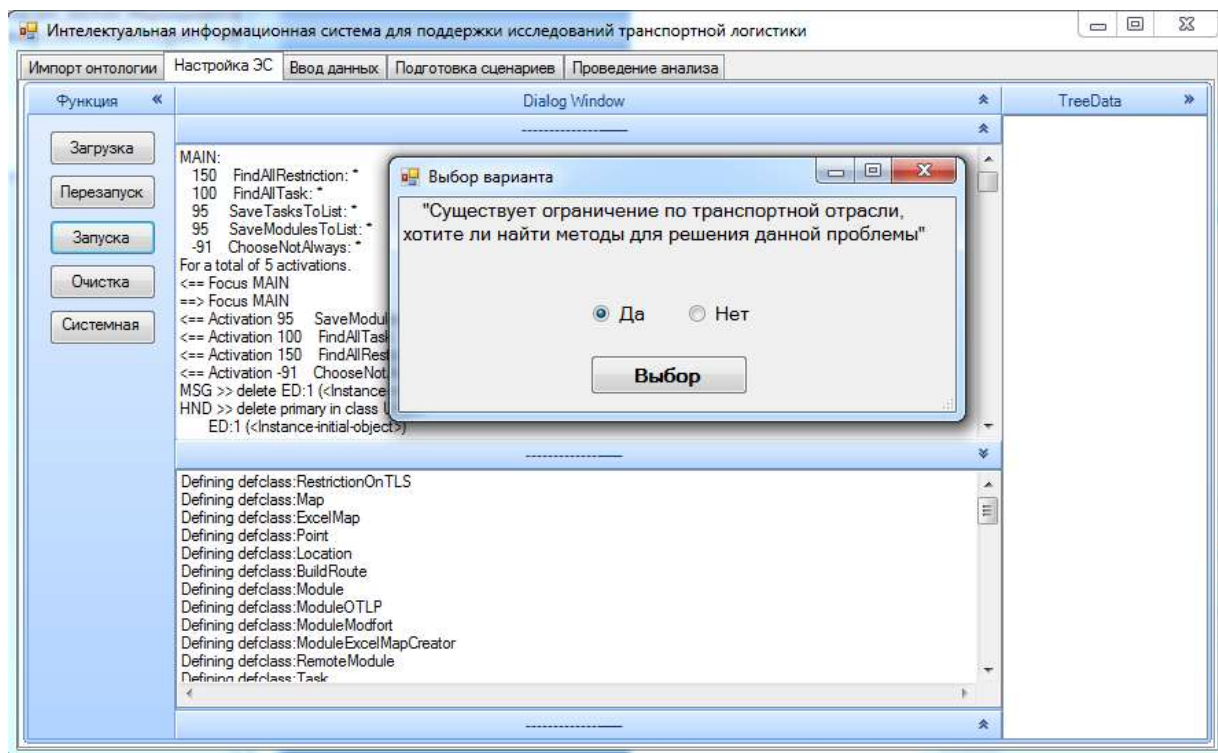


Рис. 4.15. Запуска ЭС и выполнять работу

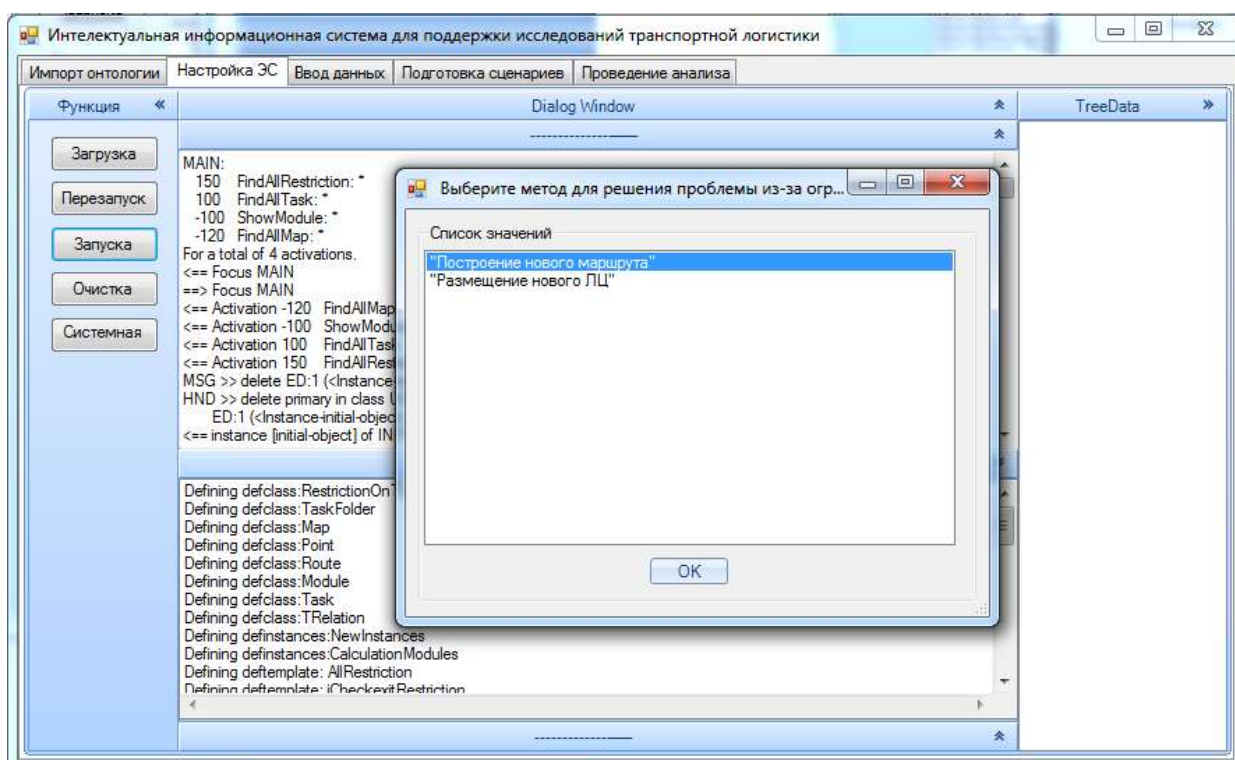


Рис. 4.16. Выбор задачи

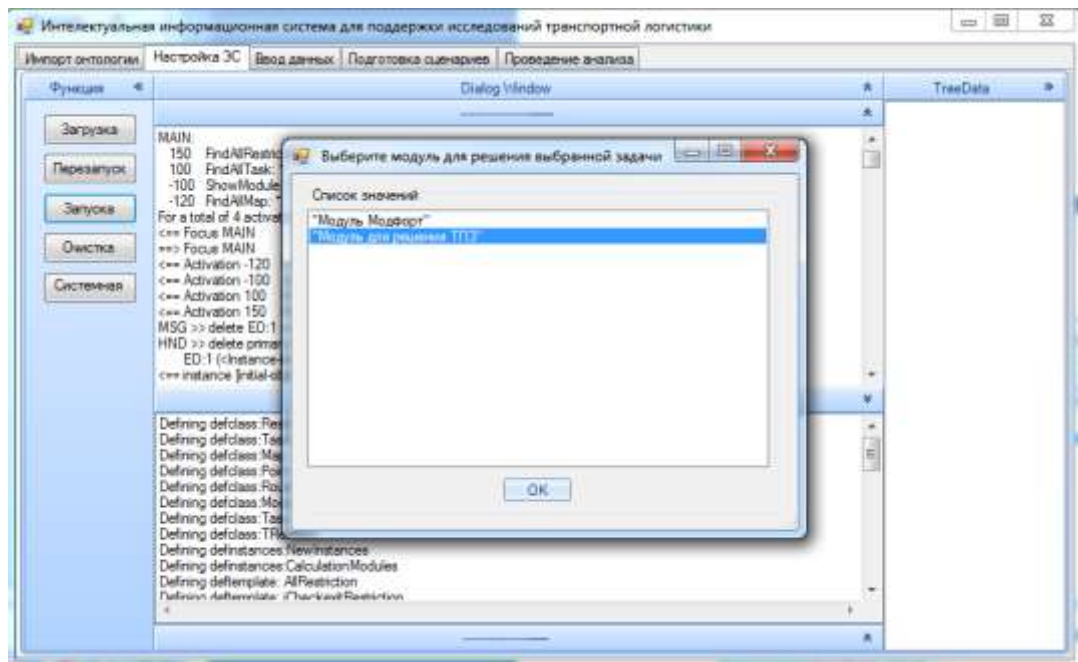


Рис. 4.17. Выбор модуля для решения задачи

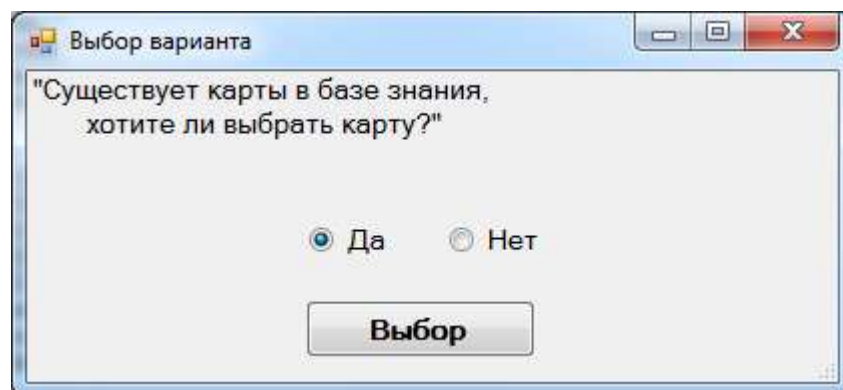


Рис. 4.18. Использование существующей карты или создание новой карты

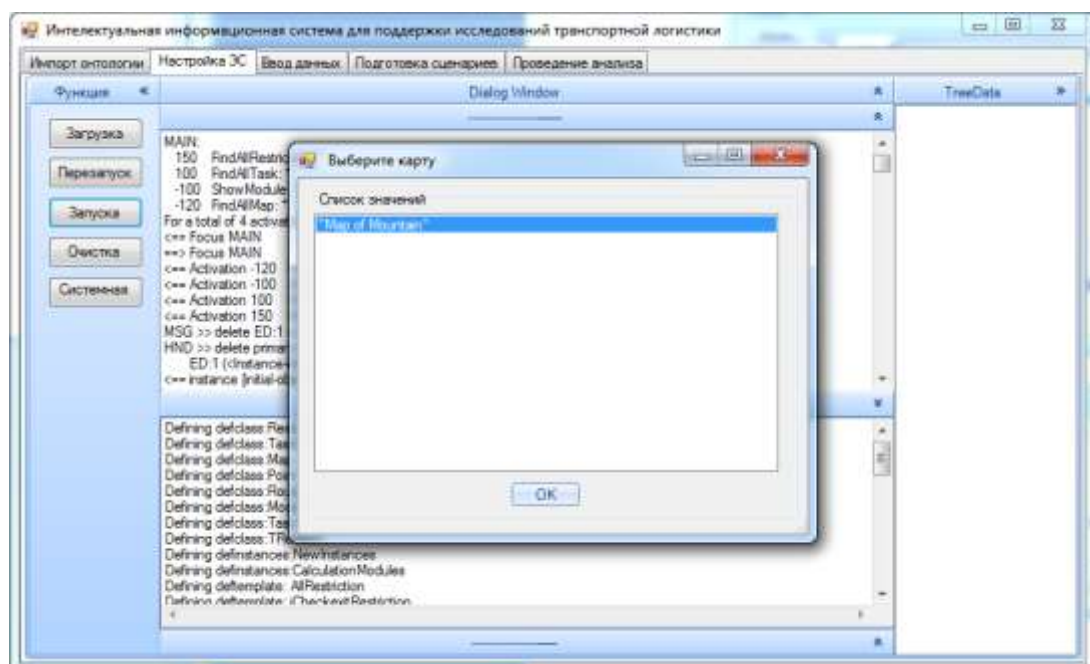


Рис. 4.19. Выбор существующей карты

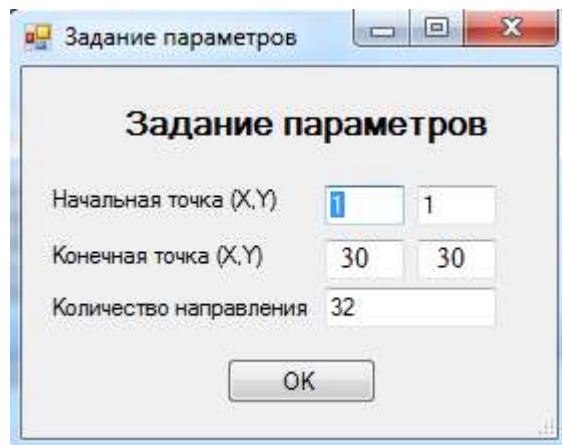


Рис. 4.20. Настройка параметров

После завершения результаты возвращаются в БЗ, сохраняются в файлах, сообщаются и показываются на формах (рис. 4.21 и рис. 4.22).

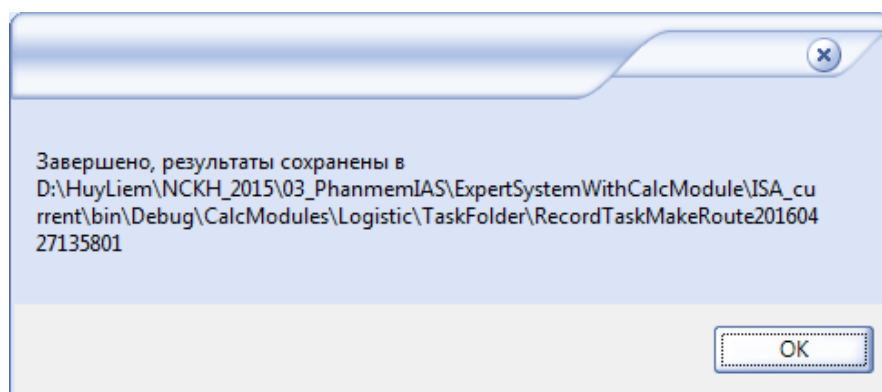


Рис. 4.21. Сообщение о выполнении работы

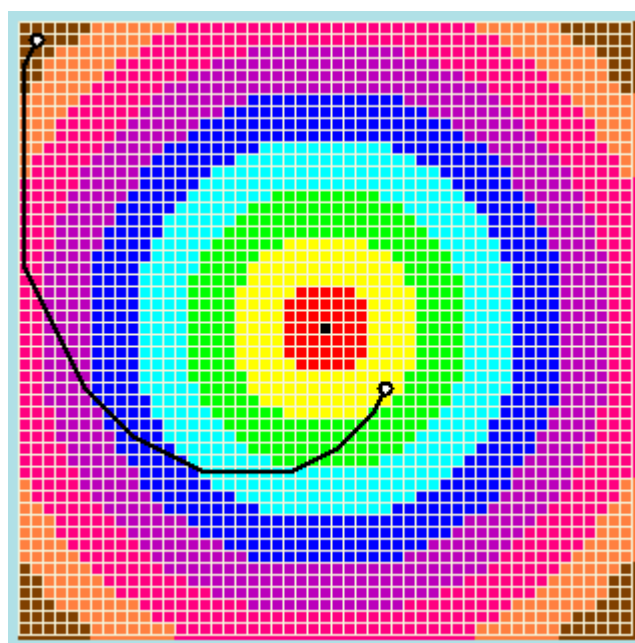


Рис. 4.22. Результат построения нового маршрута по запрос пользователя

4.2. Прикладные задачи

С использованием результатов исследования можно решить прикладные задачи в разных областях. В данной диссертации представлены решения некоторых прикладных проблем в транспортной логистике.

4.2.1. Определение оптимального расположения дополнительных больниц в районе Нам Дан (провинции Нге Ан, Вьетнам) в условиях кооперации и конкуренции

Рассматривается район Нам Дан в провинции Нге Ан (Вьетнам), данные по административной структуре и населению предоставляются Отделом обследования и картографии Министерства природных ресурсов и охраны окружающей среды Вьетнама (URL: <http://www.dosm.gov.vn>), Общим статистическим бюро Вьетнама (URL: <https://gso.gov.vn>) и корпорацией «Информатика - Карта Вьетнама» (<http://maps.vietbando.com>). На рис. 4.23 показана карта района Нам Дана, находящийся в прямоугольнике с шириной 22,4 км и длиной 26,79 км.



Рис. 4.23. Карта района Нам Дана, Нге Ана (Вьетнам)

На рис. 4.24 и 4.25.в показано отображение района в 2D и 3D виде.

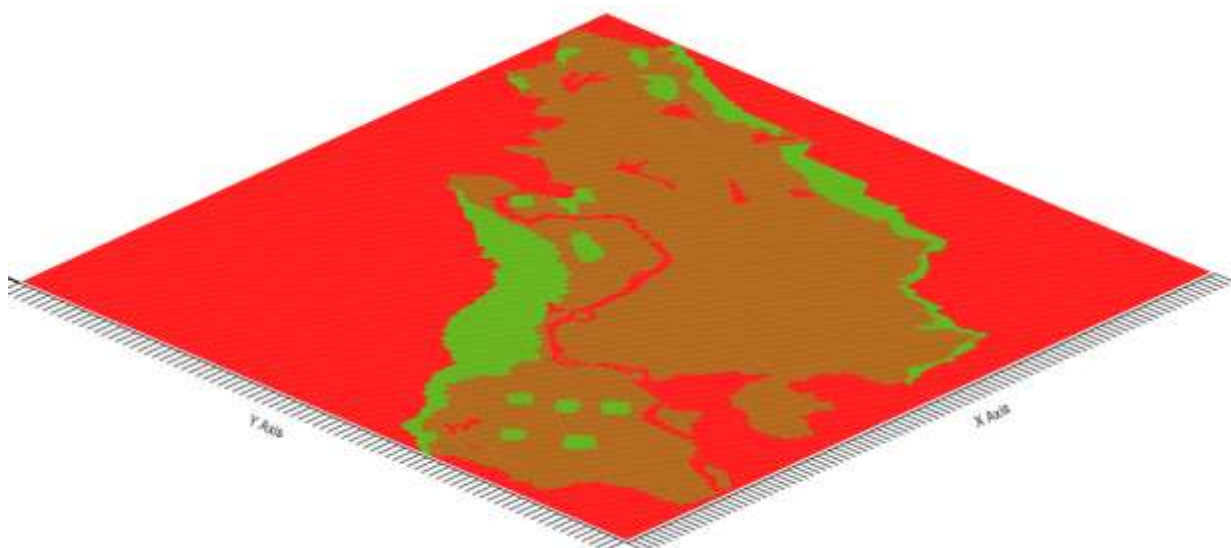


Рис. 4.24. Отображение района Нам Дана в 2D виде

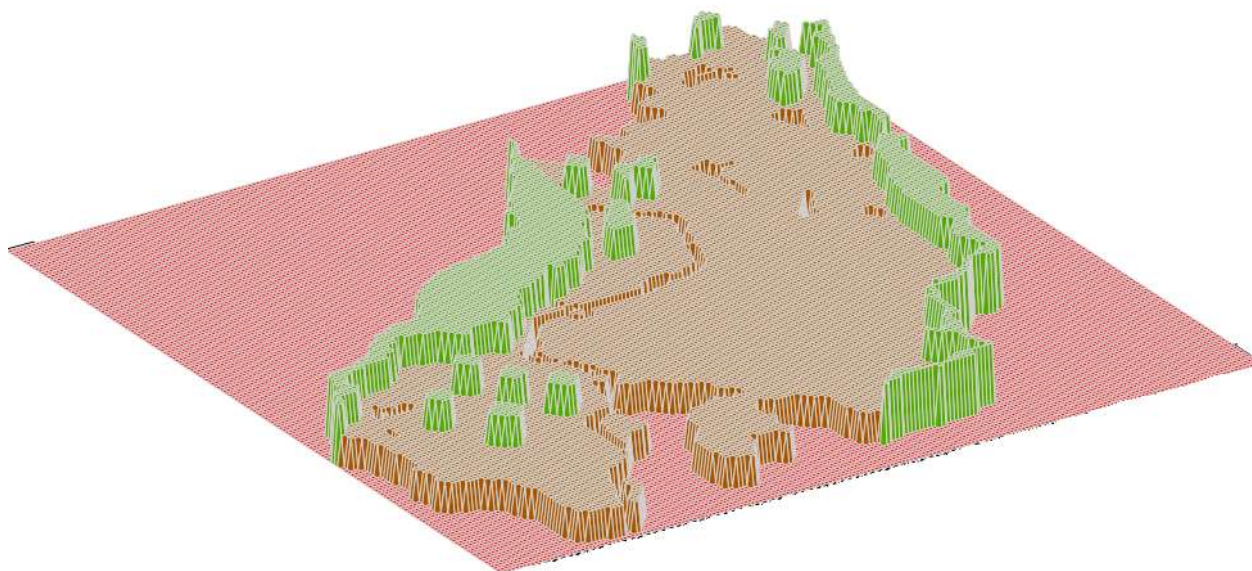


Рис. 4.25. Отображение района Нам Дана в 3D виде

Известно, что в районе расположены два обслуживающих центра (больницы) с координатами (71;83) и (167;112). На рис. 4.26. они обозначены звездами.

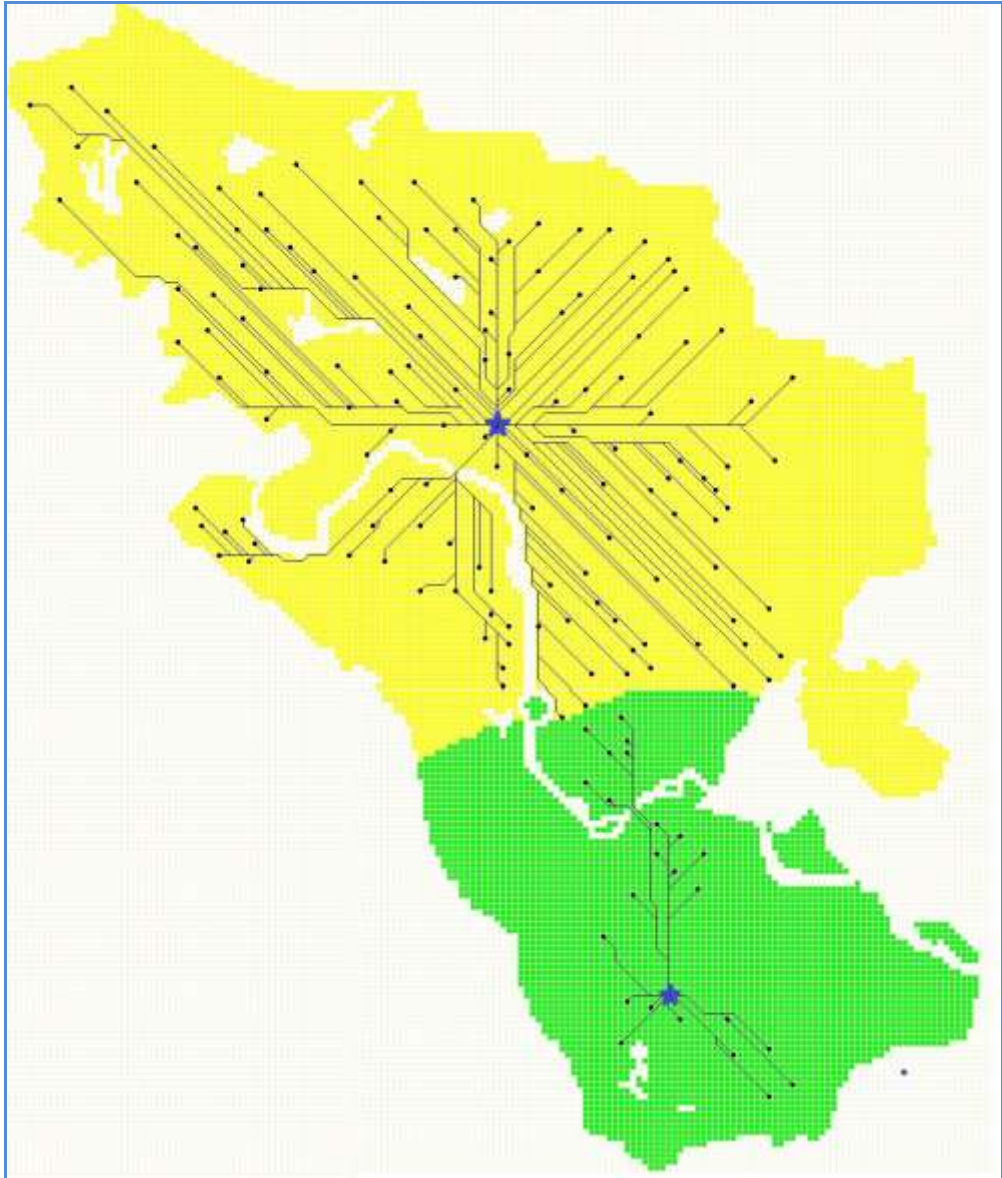


Рис. 4.26. Начальное расположение логистических зон и распределение населения

Черными точками отображены места проживания населения (всего 116016). При этом каждая точка соответствует одному кварталу (административный район на уровне деревни), который характеризуется количеством жителей.

На рис. 4.26 цветом выделены соответствующие логистические зоны и отмечены кратчайшие маршруты от каждого квартала до своего обслуживающего центра (в данном случае больницы), белым показаны непроходимые участки. Начало координат расположено в левом верхнем углу.

Результаты решения задач в кооперативной и конкурентной постановках представлены в табл. 4.14 и на рис. 4.27 и рис. 4.28. Расположение новых узлов обозначено красными звездами.

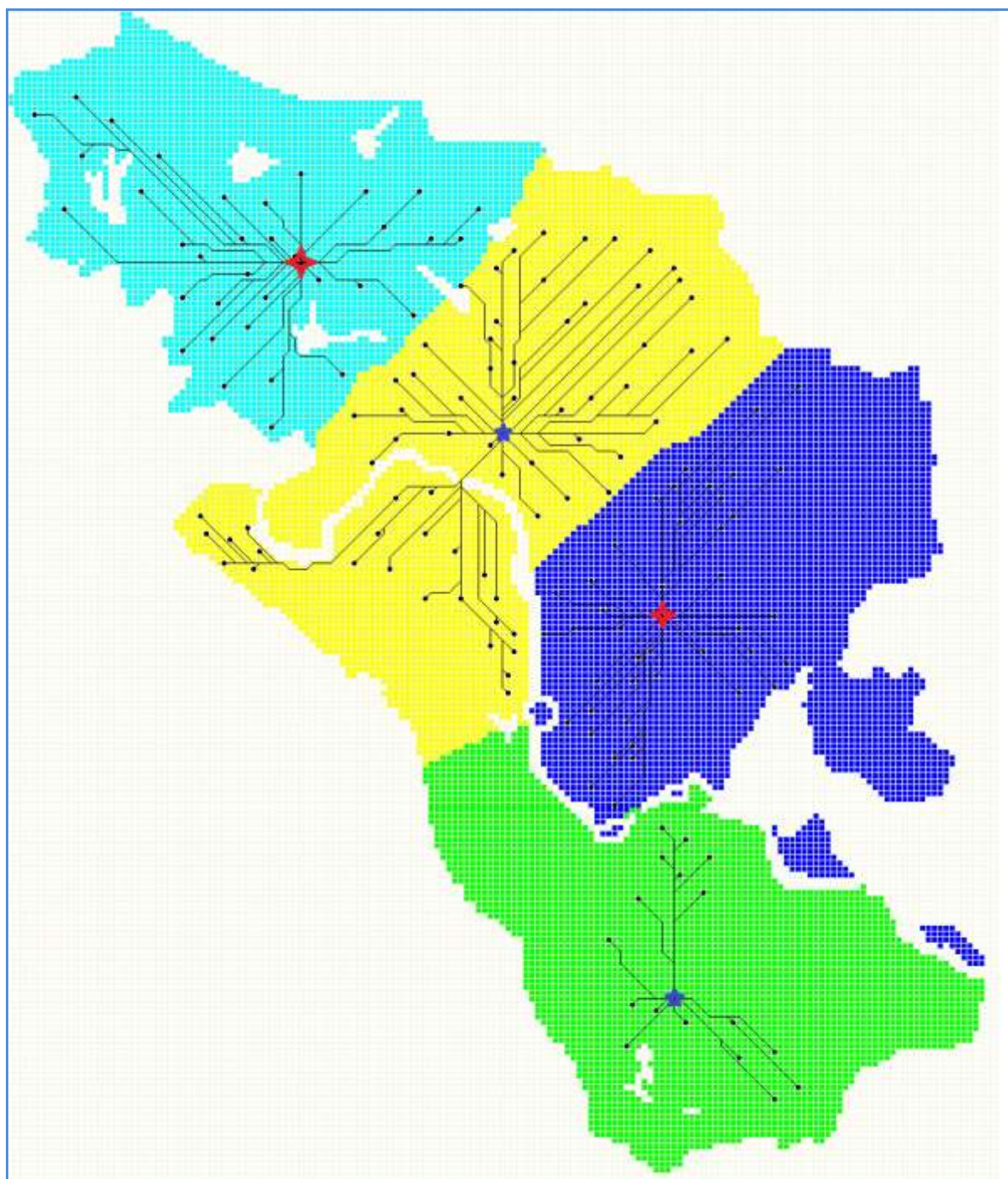


Рис. 4.27. Расположение при кооперативном вхождении

Таблица 4.14

Результаты решения

	Суммарное время доставки (в у.е.)	Количество клиентов новых центров	Координаты новых центров
Начальное состояние	41282626	—	—
Кооперация (рис. 4.27)	26940115	41266	(102;110); (42;49)
Конкуренция (рис. 4.28)	34082641	55361	(100;110); (58;971)

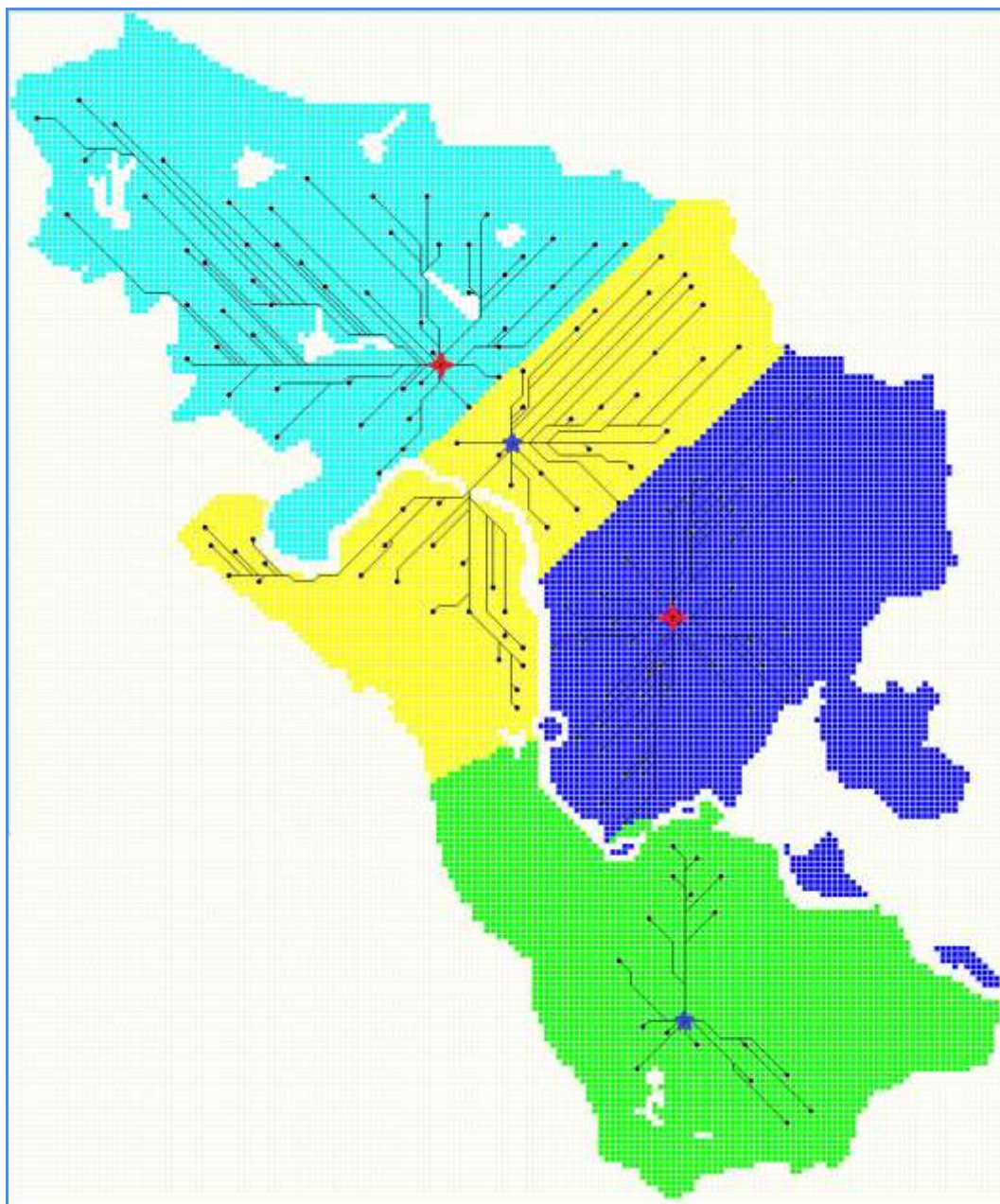


Рис. 4.28. Расположение при конкурентном вхождении

Отметим, что в первом случае произошло значительное (более чем на 34,74%) улучшение качества обслуживания, при этом логистическая зона новых центров оказалась сравнимой с существующими (около 35,57% клиентов).

Во втором случае был осуществлен захват достаточно большой доли рынка (более 47,72% клиентов). Однако уменьшение суммарного времени обслуживания оказалось незначительным (менее чем на 17,44%).

Легко видеть, что при использовании предложенного метода, мы можем решить ряд практических задач планирования в целях повышения качества обслуживания или экономической эффективности.

4.2.2. Построение системы автобусных маршрутов между населенными пунктами в районе Нам Дан минимальной суммарной длины

В районе Нам Дан (описание которого представлено в разделе 4.2.1), как во многих других районах во Вьетнаме, имеются населенные пункты (коммуны), расположенные в горной местности, с трудным условием жизни и неразвитой транспортно-логистической инфраструктурой. Поэтому построение системы общественного транспорта в целом и, в частности, маршрутной сети автобусов имеет огромное практическое значение в повышении уровня жизни, обеспечении спроса на поездки, а также будет способствовать улучшению порядка и безопасности дорожного движения.



Рис. 4.29. Оптимальное расположение остановок автобусов в каждой коммуне района Намдан

Кроме того, оптимизация транспортной сети имеет большое значение для обеспечения долгосрочного планирования развития ТЛС в ограниченных экономических условиях.

Метод, предложенный автором, позволит выработать рекомендации по решению данной проблемы.

Во-первых, определяется такое расположение остановочного пункта в каждой коммуне, до которого общее (суммарное) время поездки всех жителей в данном районе будет минимальным. При этом применяется алгоритм для решения основной задачи оптимального размещения одного или нескольких логистических центров при точечном распределении потребителей. Результат этого этапа показан на рис. 4.29.



Рис. 4.30. Построение оптимальной коммуникации между остановками автобусов

Во-вторых, построить оптимальную систему коммуникаций между ними с помощью метода, описанного на разделе 2.6, и полученный результат представляется на рис. 4.30. Построенная система автобусных маршрутов между населенными пунктами в районе Нам Дан имеет минимальную суммарную длину, равную 65,19 (км).

4.2.3. Определение оптимального расположения заданного количества радиолокационных станций с наименьшей мощностью, покрывающих г. Ханой

Управление воздушным пространством является важным требованием для всех стран в целях обеспечения государственной безопасности и эксплуатации воздушной транспортной деятельности.

При этом строится сеть радиолокационных станций, чтобы объединение их зон обслуживания покрывало воздушное пространство рассматриваемой области.

Здесь рассматриваем ситуацию, в которой нужно построить систему радиолокаторов с заданным количеством одинаковых мощных радаров (соответственно зоне контроля радара по расстоянию), покрывающую воздушное пространство города Ханой (Вьетнам).

Ханой – столица Вьетнама, является главным политическим, образовательным и культурным центром страны, которая имеет географические координаты $21^{\circ}02'$ с. ш. $105^{\circ}51'$ в. д. и карту, отображающую на рис. 4.31 с шириной 76,814км, длиной 91,25км.

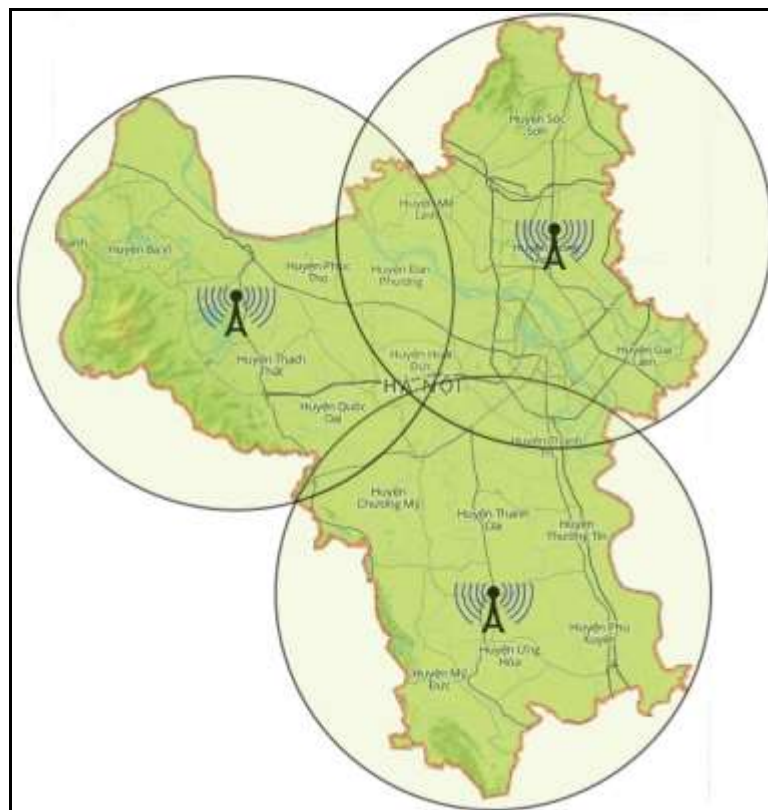


Рис. 4.34. Покрытие города Ханой тремя радиолокационными станциями с минимальным радиусом зоны обслуживания 25,266 км

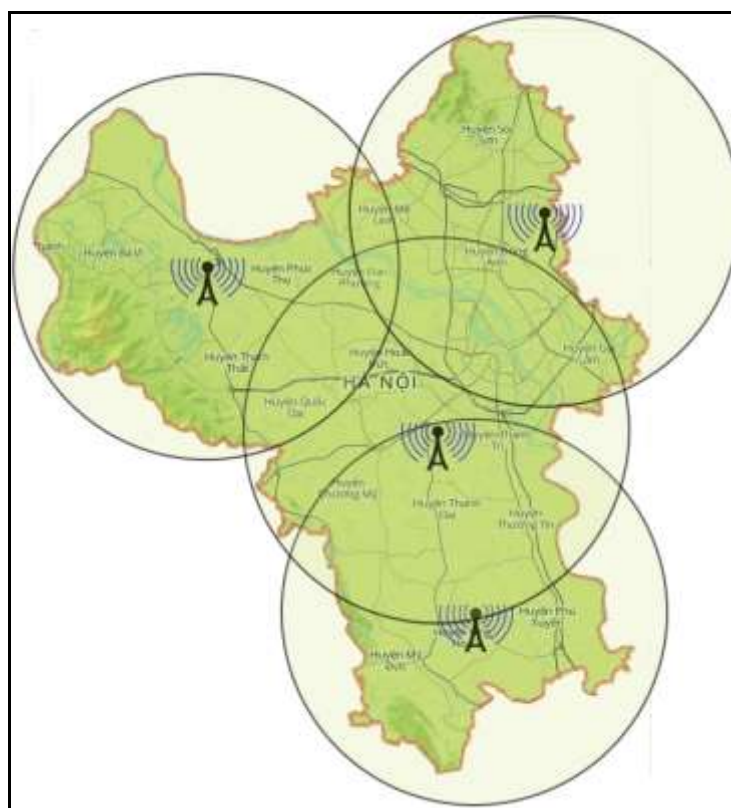


Рис. 4.35. Покрытие города Ханой четырьмя радиолокационными станциями с минимальным радиусом зоны обслуживания 23,707 км

Заключение

В ходе проведенного диссертационного исследования получены следующие основные результаты.

1. Предложена многоэтапная технология исследования ТЛС.
2. Задачи оптимального размещения инфраструктурных логистических объектов сведены к специальным модификациям двух известных математических задач: об упаковке и покрытии равными кругами ограниченных множеств в двумерных метрических пространствах с неевклидовыми метриками.
3. Для построения оптимальных упаковок равных кругов и покрытий равными кругами многосвязных множеств в неевклидовой метрике созданы новые вычислительные алгоритмы.
4. На основе оригинальных модификаций оптико-геометрического подхода разработаны вычислительные алгоритмы для решения задач об оптимальном размещении дополнительных логистических центров в условиях кооперации и конкуренции, об оптимизации системы коммуникаций с учетом региональных особенностей, об оптимизации ремонта автомобильных дорог в условиях ограниченного бюджета.
5. Создана и интегрирована в ИИС программная система «ОТЛП», в основу которой положены численные алгоритмы, разработанные в рамках диссертационного исследования.
6. Проведена экспериментальная проверка на ряде модельных и прикладных задач программной системы «ОТЛП». Так, решены следующие задачи: найдено оптимальное расположение дополнительных больниц в районе Нам Дан (провинции Нге Ан, Вьетнам) в условиях кооперации и конкуренции; установлено оптимальное расположение заданного количества радиолокационных станций с наименьшей мощностью, покрывающих г. Ханой; построена система автобусных маршрутов между населенными пунктами в районе Нам Дан минимальной суммарной длины.

Список литературы

1. Алдыноол, Т.А. Покрытие плоской области случайно распределенными сенсорами / Т.А. Алдыноол, А.И. Ерзин, В.В. Залюбовский // Вестн. НГУ. Сер. Математика, механика, информатика. – 2010. – Т. 10, № 4. – С. 7–25.
2. Астраков, С.Н. Сенсорные сети и покрытие плоскости кругами / С.Н. Астраков, А.И. Ерзин, В.В. Залюбовский // Дискрет. анализ и исслед. операций. – 2009. – Т. 16, № 3. – С. 3–19.
3. Астраков, С.Н. Сенсорные сети и покрытие полосы эллипсами / С.Н. Астраков, А.И. Ерзин // Вычислительные технологии. – 2013. – Т. 18, № 2. – С. 3–11.
4. Атрохов, Н.А. О задаче коммивояжера / Н.А. Атрохов // Повышение эффективности и качества автотранспортного обслуживания. – М.: МАДИ, 1989. – С. 72–74.
5. Бауэрсокс, Д. Логистика: интегрированная цепь поставок / Д. Бауэрсокс, Д. Клосс. – М.: «Олимп – Бизнес», 2001. – 640 с.
6. Беленький, А.С. Исследование операций в транспортных системах: идеи и схемы методов оптимизации планирования / А.С. Беленький. – М.: Мир, 1992. – 582 с.
7. Беляев, В.М. Управление процессами в транспортных логистических системах: учеб. пособие / В.М. Беляев, Л.Б. Миротин, А.Г. Некрасов, А.К. Покровский; под общ. ред. А.Г. Некрасова. – М.: МАДИ, 2011. – 127 с.
8. Брусов, В.С. Вычислительный алгоритм оптимального покрытия областей плоскости / В.С. Брусов, С.А. Пиявский // Журн. вычисл. математики и мат. физики. – 1971. – Т. 11, № 2. – С. 304–313.
9. Брусов, В.С. Двигательная установка малой тяги, универсальная для двумерного диапазона / В.С. Брусов, С.А. Пиявский // Изв. АН СССР. Космич. исслед. – 1970. – № 4. – С. 542–546.

10. Брусов, В.С. Применение теории оптимальных покрытий к задачам выбора мощности двигателя малой тяги / В.С. Брусов, С.А. Пиявский // Изв. АН СССР. Механ. твердого тела. – 1969. – № 2. – С. 10–14.
11. Бухаров, Д.С. Программная система «ВИГОЛТ» для решения задач оптимизации, возникающих в транспортной логистике / Д.С. Бухаров, А.Л. Казаков // Вычислительные методы и программирование: новые вычислительные технологии. – 2012. – Т. 13, № 2 (26). – С. 65–74.
12. Вебер, А. Теория размещения промышленности / А. Вебер. – М.: Л.: Книга, 1926.
13. Гаджинский, А.М. Современный склад. Организация, технология, управление и логистика: учебное пособие / А.М. Гаджинский. – М.: ТК Велби, 2005. – 176 с.
14. Галиев, Ш.И. Оптимизация многократного покрытия ограниченного множества кругами / Ш.И. Галиев, М.А. Карпова // Журн. вычисл. математики и мат. физики. – 2010. – Т. 50, № 4. – С. 757–769.
15. Галиев, Ш.И. Численные методы оптимизации упаковок равных ортогонально ориентированных эллипсов в прямоугольную область / Ш.И. Галиев, М.С. Лисафина // Журн. вычисл. математики и мат. физики – 2013. – Т. 53, № 11. – С. 1923–1938.
16. Гимади, Э.Х. Алгоритм для приближенного решения задачи коммивояжера и его вероятностный анализ / Э.Х. Гимади, Н.И. Глебов, А.И. Сердюков // Сибирский журнал исследования операций. – 1994. – № 2. – С. 8–17.
17. Гимади, Э.Х. Приближенные алгоритмы решения задачи о двух коммивояжерах на максимум / Э.Х. Гимади, Е.В. Ивонина // Дискретный анализ и исследование операций. – 2012. – №1. – С. 17–32.
18. Гудман, С. Введение в разработку и анализ алгоритмов / С. Гудман, С. Хидетниemi. – М.: Мир, 1981. – 368 с.
19. Данциг, Дж.Б. Линейное программирование, его применения и обобщения (Linear Programming and Extensions, 1963) / Дж.Б. Данциг. – Перевод с ан-

- глийского Г.Н. Андрианова, Л.И. Горькова, А.А. Корбута, А.Н. Ляпунова. Общая редакция и предисловие Н.Н. Воробьева. – М.: Изд-во «Прогресс», 1966. – 600 с.
20. Деньдобренко, Б.Н. Автоматизация конструирования РЭА: Учебник для вузов / Б.Н. Деньдобренко, А.С. Малика. – М.: Высш. школа, 1980. – 384 с.
 21. Джонс, М.Т. Программирование искусственного интеллекта в приложениях / М.Т. Джонс. – М.: ДМК Пресс, 2006. – 312 с.
 22. Загоруйко, Н.Г. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко. – Новосибирск: Изд-во Ин-та математики, 1999. – 270 с.
 23. и труднорешаемые задачи / М.Р. Гэри, Д.С. Джонсон. – М.: Мир, 1982.
 24. Иванко, Е.Е. Метод масштабирования в приближенном решении задачи коммивояжера / Е.Е. Иванко // Автоматика и телемеханика. – 2011. – № 12. – С. 115–129.
 25. Иванов, А.О. Теория экстремальных сетей / А.О. Иванов, А.А. Тужилин. – М.–Ижевск: Институт компьютерных исследований, 2003. – 424 с.
 26. Казаков, А.Л. Вопросы сегментации логистических платформ в условиях становления региональной логистики / А.Л. Казаков, М.А. Журавская, А.А. Лемперт // Транспорт Урала. – 2010. – № 4. – С. 17–20.
 27. Казаков, А.Л. Интеллектуальная информационная система поддержки комплексных исследований транспортных процессов и объектов / А.Л. Казаков, А.А. Лемперт, А.Б. Столбов, Г.Л. Нгуен // Информационные и математические технологии в науке и управлении. – 2014. – Т. 3. – С. 57–61.
 28. Казаков, А.Л. К вопросу о сегментации логистических зон для обслуживания непрерывно распределенных потребителей / А.Л. Казаков, А.А. Лемперт, Д.С. Бухаров // Автоматика и телемеханика. – 2013. – № 6. – С. 87–100.
 29. Казаков, А.Л. О методе решения задачи оптимальной прокладки высокоскоростных железнодорожных магистралей с учетом региональных особенностей / А.Л. Казаков, М.А. Журавская, А.А. Лемперт, Д.С. Бухаров // Транспорт: наука, техника, управление. – 2012. – № 2. – С. 41–44.

30. Казаков, А.Л. Об одном подходе к решению задач оптимизации, возникающих в транспортной логистике / А.Л. Казаков, А.А. Лемперт // Автоматика и телемеханика. – 2011. – № 7. – С. 50–57.
31. Казаков, А.Л. Об одном численном методе решения некоторых задач оптимизации, возникающих в транспортной логистике / А.Л. Казаков, А.А. Лемперт, Д.С. Бухаров // Вестник ИрГТУ. – 2011. – № 6. – С. 6–12.
32. Казаков, А.Л. Оптимизация системы коммуникаций с учетом региональных особенностей: математическая модель и численный метод / А.Л. Казаков, А.А. Лемперт, Г.Л. Нгуен // Вестник ИрГТУ. – 2014. – № 12(95). – С. 17–22.
33. Китурко, О.М. О решении новых задач математической логистики / О.М. Китурко, М.А. Маталыцкий // Международный конгресс по информатике: информационные системы и технологии. Ч 2. Минск: БГУ, 2011. – С. 291–296.
34. Кожин, А.П. Математические методы в планировании и управлении грузовыми автомобильными перевозками: Учеб. для вузов. / А.П. Кожин, В.Н. Мезенцев. – М.: Транспорт, 1994. – 304 с.
35. Козлов, П.А. Имитационная экспертиза проекта развития транспортного узла устьлуга / П.А. Козлов, И.П. Владимирская, И.В. Иванов, О.В. Осокин // Транспорт Российской Федерации. – СПб: Т-Пресса, 2009. – Т. 25, № 6. – С. 18–21.
36. Козлов, П.А. Оптимальное управление работой вагонов разных собственников / П.А. Козлов, И.П. Владимирская, Н.А. Тушин // Вестник научно-исследовательского ин-та железнодорожного транспорта. – М: ВНИИЖТ, 2010. – № 4. – С. 7–10.
37. Козлов, П.А. Оптимизация структуры транспортных потоков в динамике при приоритете потребителей / П.А. Козлов, С.П. Миловидов // Экономика и математические методы. – М: ЦЭМИ, 1982. – Т. XVIII, вып.3. – С. 521–531.
38. Козлов, П.А. Построение систем автоматизированного управления потоками вагонов разных собственников / П.А. Козлов, И.П. Владимирская // Вестник

- научно-исследовательского ин-та железнодорожного транспорта. – М: ВНИИЖТ, 2009. – № 6. – С. 8–11.
39. Конвей, Дж. Упаковки шаров, решетки и группы / Дж. Конвей, Н. Слоэн. – В 2-х томах. Пер. с англ. – М.: Мир, 1990. – 791с.
 40. Корбут, А.А. Дискретное программирование / А.А. Корбут, Ю.Ю. Финкельштейн. – М.: Наука, 1969. – 368 с.
 41. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – М.: Вильямс, 2005. – 1296 с.
 42. Кротов, В.Ф. Достаточные условия оптимальности в задачах об оптимальных покрытиях / В.Ф. Кротов, С.А. Пиявский // Изв. АН СССР. Техн. кибернетика. – 1968. – № 2. – С. 10–17.
 43. Кузнецов, В.Ю. Задачи покрытия ортогональных многоугольников с запретными участками / В.Ю. Кузнецов // Вестник УГАТУ. – 2008. – Т. 10, № 2(27). – С. 177–182.
 44. Лебедев, П.Д. Алгоритмы наилучшей аппроксимации плоских множеств объединениями кругов / П.Д. Лебедев, А.А. Успенский, В.Н. Ушаков // Вестник Удмуртского ун-та. Математика. Механика. Компьютерные науки. – 2013. – Вып. 4. – С. 88–99.
 45. Лебедев, П.Д. Аппроксимация многоугольников наилучшими наборами кругов / П.Д. Лебедев, Д.С. Бухаров // Известия Иркутского гос. ун-та. Математика. – 2013. – № 3. – С. 72–87.
 46. Левенштейн, В.И. О границах для упаковок в n -мерном евклидовом пространстве / В.И. Левенштейн // Доклады АН СССР. – 1979. – Т. 245, №6. – С. 1299–1303.
 47. Лемперт, А.А. Математическая модель и программная система для решения задачи размещения логистических объектов / А.А. Лемперт, А.Л. Казаков, Д.С. Бухаров // Управление большими системами. – 2013. – Вып. 41. – С. 270–284.

48. Лотарев, Д.Т. Локальная оптимизация в задаче Штейнера на евклидовой плоскости / Д.Т. Лотарев, А.П. Уздемир // Автоматика и телемеханика. – 2004. – №7. – С. 60–70.
49. Лотарев, Д.Т. Преобразование задачи Штейнера на евклидовой плоскости к задаче Штейнера на графе / Д.Т. Лотарев, А.П. Уздемир // Автоматика и телемеханика. – 2005. – №10. – С. 80–92.
50. Лубенцова, В.С. Математические модели и методы в логистике: учебное пособие / В.С. Лубенцова. Под редакцией В. П. Радченко. – Самара: Гос. техн. ун-т, 2008. – 158 с.
51. Лукинский, В.С. Логистика автомобильного транспорта. Концепция, методы, модели / В.С. Лукинский, В.И. Бережной, Е.В. Бережная, И.А. Цвиринько. – М.: Финансы и статистика, 2000. – 280с.
52. Лукинский, В.С. Логистика автомобильного транспорта: Учеб. Пособие / В.С. Лукинский, В.И. Бережной, Е.В. Бережная. – М.: Финансы и статистика, 2004. – 368 с.
53. Лукинский, В.С. Модели и методы теории логистики / В.С. Лукинский. – СПб.: Питер, 2008. – 448 с.
54. Лукинский, В.С. Оценка влияния размещения складской сети на транспортные расходы / В.С. Лукинский, А.А. Бочкарев, О.Ю. Пеховский, И.А. Цвиринько // Экономика и менеджмент на транспорте: Сб. научн. тр.:– СПб.: СПбГИЭУ, 2002. – Вып.2. – С.99 –106.
55. Лукинский, В.С. Прогнозирование надежности автомобилей / В.С. Лукинский, Е.И. Зайцев. – Л.: Политехника, 1991. – 220 с.
56. Мандель, И.Д. Кластерный анализ / И.Д. Мандель. – М.: Финансы и статистика, 1988. – 176 с.
57. Маталыцкий, М.А. Решение задачи оптимального управления функционированием логистической транспортной системы методом полного перебора стратегий / М.А. Маталыцкий, О.М. Китурко // Вестник ГрГУ. Сер. 2. – 2011. – № 3(118). – С. 136–48.

58. Мелихов, А.Н. Применение графов для проектирования дискретных устройств / А.Н. Мелихов, Л.С. Берштейн, В.М. Курейчик. – М.: Наука, 1974. – 304 с.
59. Миротин, Л.Б. Повышение эффективности грузовых перевозок на основе создания устойчивой транспортно-логистической системы модульного типа для высокоскоростной обработки и доставки грузов / Л.Б. Миротин, А.Г. Некрасов, П.В. Степанов, П.Г. Трегубов // URL:http://www.logistikaprim.ru/sites/default/files/Povyshenie_effektivnosti_gruzovyh_perevozok.pdf. – С. 1–10.
60. Михалевич, В.С. Оптимизационные задачи производственно–транспортного планирования: Модели, методы, алгоритмы / В.С. Михалевич, В.А. Трубин, Н.З. Шор. – М.: Наука, 1986. – 264 с.
61. Можаяев, Г.В. Задача о непрерывном обзоре поверхности Земли и кинематически правильные спутниковые системы / Г.В. Можаяев // Косм. исслед. – 1972. – Т.10, № 6. – С. 833–840.
62. Мозговой, М.В. Занимательное программирование: Самоучитель / М.В. Мозговой. – СПб.: Питер, 2005. – 208 с.
63. Москаленский, Е.Д. О нахождении точных решений двумерного уравнения эйконала для случая, когда фронт волны, распространяющийся в среде, является окружностью / Е.Д. Москаленский // Сиб. журн. вычислит. математики. – 2014. – Т. 17, № 4. – С. 363–372.
64. Мухачева, А.С. Простые эвристики для решения двумерной задачи максимального покрытия: Принятие решений в условиях неопределенности / А.С. Мухачева // Межвузовский научный сборник. – Ч. 1.– Уфа: УГАТУ. –2005. – Вып. 2 – С. 24–32.
65. Николайчук, В.Е. Транспортно–складская логистика: Учебное пособие / В.Е. Николайчук. – М.: Издательско–торговая корпорация «Дашков и Ко», 2005. – 452 с.

66. Павлов, А.И. Архитектура системы поддержки проектирования агентов для имитационных моделей сложных систем / А.И. Павлов, А.Б. Столбов // Программные продукты и системы. – 2015. – № 1. – С. 12–16.
67. Пиявский, С.А. Об оптимизации сетей / С.А. Пиявский // Изв. АН СССР. Техн. кибернетика. – 1968. – № 1. – С. 68–80.
68. Постан, М.Я. Экономико–математические модели смешанных перевозок / М.Я. Постан. – Одесса: Астропринт, 2006. – 376 с.
69. Сергеев, С.И. Алгоритмы решения минимаксной задачи коммивояжер. II. Двойственный подход / С.И. Сергеев // Автоматика и телемеханика. – 1995. – №8. – С. 124–141.
70. Сергеев, С.И. Использование методов теории оптимального управления для решения некоторых задач дискретной оптимизации II. Статическая задача коммивояжера / С.И. Сергеев // Автоматика и телемеханика. – 2006. – №6. – С. 106–112.
71. Сергеев, С.И. Использование методов теории оптимального управления для решения некоторых задач дискретной оптимизации III. Динамическая задача коммивояжера / С.И. Сергеев // Автоматика и телемеханика. – 2006. – №7. – С. 27–40.
72. Сидоров, Д.Н. Методы анализа интегральных динамических моделей. Теория и приложения / Д.Н. Сидоров. – Иркутск: Изд-во ИГУ, 2013. – 293 с.
73. Словарь по кибернетике / Под редакцией академика В. С. Михалевича. – Киев: Главная редакция Украинской советской энциклопедии имени М. П. Бажана, 1989. – 751 с.
74. Срочко, В.А. Задачи оптимального управления для билинейной системы специальной структуры / В.А. Срочко, Е.В. Аксеньюшкина // Известия Иркутского гос. ун-та. Серия: Математика. – 2016. – Т.15. – С. 78-91.
75. Срочко, В.А. К численному решению некоторых задач минимаксного управления / В.А. Срочко // Автоматика и телемеханика. – 2013. – №6. – С. 17-25.

76. Степанцов, М.Е. Динамическая модель развития транспортной сети / М.Е. Степанцов // Препринты ИПМ им. М. В. Келдыша. – 2008. – С. 79–91.
77. Столбов, А.Б. Информационная система поддержки исследований в инфраструктурной логистике / А.Б. Столбов, Г.Л. Нгуен // Программные системы: Теория и приложения. – 2015. – № 3(26). – С. 3–20.
78. Тахонов, И.И. О некоторых задачах покрытия плоскости кругами / И.И. Тахонов // Дискрет. анализ и исслед. операций. – 2014. – Т. 21, №1. – С. 84–102.
79. Телицкий, С.В. Комплексный подход к решению задачи покрытия области заготовками неопределенных размеров / С.В. Телицкий, А.С. Филиппова // Научно–технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – 2012. – № 2(145). – С. 61–67.
80. Тот, Л.Ф. Расположения на плоскости, на сфере и в пространстве / Л.Ф. Тот – М.: Физматлит, 1958. – 364 с.
81. Тропина, В.М. Динамика логистической системы в транспортных цепях постановок / В.М. Тропина, И.Е. Агуреев // Известия Тульского гос. ун-та. Технические науки. – 2011. – № 4. – С. 158–167.
82. Филиппова, А.С. Задачи о минимальном покрытии ортогональных многоугольников с запретными участками / А.С. Филиппова, В.Ю. Кузнецов // Информационные технологии. – 2008. – № 9 (145). – С. 60–65.
83. Фроловский, В.Д. Оптимизация раскроя материалов на оборудовании с ЧПУ: модели, методы, алгоритмы / В.Д. Фроловский. – Saarbrücken: LAP LAMBERT Academic Publishing, 2011. – 124 с.
84. Чеблоков, И.Б. Об одной задаче маршрутизации с внутренними работами / И.Б. Чеблоков, А.Г. Ченцов // Вестник удмуртского ун-та. Математика. Механика. Компьютерные науки. – 2012. – Вып. 1. – С. 96–119.
85. Ченцов, А.А. Экстремальная задача маршрутизации «на узкие места» с ограничениями в виде условий предшествования / А.А. Ченцов, А.Г. Ченцов // Труды института математики и механики. – 2008. – №2. – С. 129–142.

86. Швецов, В.И. Математическое моделирование загрузки транспортных сетей / В.И. Швецов, А.С. Алиев. – М.: УРСС, 2003. – 64 с.
87. Щербакова, В.А. Классы ориентированных градуированных графов с полиномиально разрешимой мощностной задачей Штейнера / В.А. Щербакова // Дискретная математика. – 1997. – Вып. 4. – С. 73–85.
88. Яськов, Г.Н. Размещение максимального числа равных кругов в круге / Г.Н. Яськов // Журнал вычислительной и прикладной математики. – 2013. – № 2. – С. 60–67.
89. Andrade, R. Symmetry-breaking constraints for packing identical orthogonal rectangles within polyhedral / R. Andrade, E.G. Birgin, J.N. Tavares // Optimization Letters 7. – 2013. – P. 375–405.
90. Anil, K.J. Algorithms for clustering data / K.J. Anil. – New Jersey: Prentice Hall, 1988. – 320 p.
91. Balazs, B. Optimal circle covering problems and their applications / B. Balazs, P. Endre, L.L. Balazs // CEJOR. – 2015. – V. 23. – P. 815–832.
92. Bellman, R. On kth-best policies / R. Bellman, R. Kalaba // J. of SIAM 8. – 1960. – 582 pp.
93. Birgin, E.G. New and improved results for packing identical unitary radius circles within triangles, rectangles and strips / E.G. Birgin, T. Csendes // Computers and Operations Research. – 2010. – V. 37, № 7. – P. 1318–1327.
94. Borndorfer, R. Optimization of Transportation Systems / R. Borndorfer, M. Grotschel, A. Lobel // The future of mobility & transportation in a moving world. – 1998. – 18p. URL: <https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/352>.
95. Boroczky, K. Packing of spheres in spaces of constant curvature / K. Boroczky // Acta Mathematica Academiae Scientiarum Hungarica. – 1978. – V. 32. – P. 243–261.
96. Casado, L.G. Packing equal circles in a square II: New results for up to 100 circles using the TAMSASS-PECS stochastic algorithm / L.G. Casado, I. Garcia, P.G.

- Szabó, T. Csendes // Giannessi F., Pardalos P., Rapcsák T. (eds) Optimization Theory: Recent Developments from MATrahBza. Kluwer Academic Publishers, Dordrecht Boston London. – 2001. – P. 207–224.
97. Christaller, W. Central places in southern Germany / W. Christaller. – N.J.: Prentice–Hall, 1966. – 230 p.
 98. Clare, B.W. The closest packing of equal circles on a sphere / B.W. Clare, D.L. Keper // Proc. Roy. Soc. London Ser. A 405. – 1986. – P. 329–344.
 99. CLIPS: A Tool for Building Expert Systems [Электронный ресурс]. URL: <http://clipsrules.sourceforge.net>.
 100. Conway, J.H. Sphere Packing. Lattices and Groups / J.H. Conway, N.J.A. Sloane. – N.Y.: Springer science and Business Media, 1999. – 706 p.
 101. Coxeter, H.S.M. Arrangements of equal spheres in non–Euclidean spaces / H.S.M. Coxeter // Acta Mathematica Academiae Scientiarum Hungarica. – 1954. – № 3. – P. 263–274.
 102. Dijkstra, E.W. A note on two problems in connexion with graph / E.W. Dijkstra // Numerische Mathematik. – 1959. – V. 1. – P. 269–271.
 103. Dileep, R.S. Logistics of facility location and allocation / R.S. Dileep. – N. Y.: Marcel Dekker, Inc., 2001. – 459 p.
 104. Drezner, Z. Facility location: A survey of applications and methods / Z. Drezner. – NY: Springer–Verl, 1995. – 571 p.
 105. Ford L.R. Jr. Network flow theory / L.R. Jr. Ford. – Rand Corporation Report, 1946. – 923 p.
 106. Gartner, N.H. Modeling and Optimization for Transportation Systems Planning and Operations / N.H. Gartner, B.L. Golden, R.T. Wong. – Massachusetts Institute of Technology, Operations Research Center, 1976. – 25 p.
 107. Goldberg, M. Packing of 14, 16, 17 and 20 circles in a circle / M. Goldberg // Math. Mag. – 1971. – T. 44. – P. 134–139.

108. Graham, R.L. Dense packings of congruent circles in a circle / R.L. Graham, B.D. Lubachevsky, K.J. Nurmela, P.R.J. Östergard // *Discrete Mathematics*. – 1998. – № 181. – P. 139–154.
109. Gutin, G. The traveling salesman problem and its variations / G. Gutin, A.P. Punnen. – Dordrecht: Kluwer Academic Publishers, 2004. – 830 p.
110. Haimovich, M. Extremum properties of hexagonal partitions and uniform distribution in Euclidean location / M. Haimovich, T.L. Magnanti // *SIAM Journal of Discrete Mathematics*. – 1988. – № 1. – P. 50–64.
111. Hakimi, S.L. Optimum location of switching centres and the absolute centres and medians on a graph / S.L. Hakimi // – *Ops. Res.*, 1965. – V. 12. – 450 p.
112. Heppes, A. Covering a planar domain with sets of small diameter / A. Heppes // *Periodica Mathematica Hungarica*. – 2006. – V. 53. – P. 157–168.
113. Hwang, F.K. The Steiner tree problem / F.K. Hwang, D.S. Richards, P. Winter. – Amsterdam: Elsevier Science Publishers B.V., 1992. – 352 p.
114. Kantorovich, L. On the translocation of masses / L. Kantorovich // *C. R. (Doklady) Acad. Sci. URSS (N. S.)*, – 1942. – V. 37. – P. 199–201.
115. Kazakov, A.L. On Mathematical Models for Optimization Problem of Logistics Infrastructure / A.L. Kazakov, A.A. Lempert // *International Journal of Artificial Intelligence*. – 2015. – V. 13, № 1. – P. 200–210.
116. Kellerer, H. Knapsack problems / H. Kellerer, J.M. Gentil. – Berlin: Springer–Verlag, 2004. – 548 p.
117. Litvinchev, I. Integer Programming Formulations for Approximate Packing Circles in a Rectangular Container / I. Litvinchev, E.L. Ozuna // *Mathematical Problems in Engineering*. – 2014. – P. 1–6. – URL: <http://dx.doi.org/10.1155/2014/317697>.
118. Litvinchev, I. Packing circles in a rectangular container / I. Litvinchev, E.L. Ozuna // *Proceedings of the International Congress on Logistics and Supply Chain*. – 2013. – P. 24–25.

119. Lopez, C.O. A heuristic for the circle-packing problem with a variety of containers / C.O. Lopez, J.E. Beasley // *European Journal of Operational Research*. – 2011. – V. 214. – P. 512–525.
120. Lopez, C.O. Packing unequal circles using formulation space search / C.O. Lopez, J.E. Beasley // *Computers and Operations Research*. – 2013. – V. 40. – P. 1276–1288.
121. Lubachevsky, B.D. Curved hexagonal packings of equal disks in a circle / B.D. Lubachevsky, R.L. Graham // *Discrete Comput Geom*. – 1997. – V. 18. – P. 179–194.
122. Lubachevsky, B.D. How to simulate billiards and similar systems / B.D. Lubachevsky // *J. Comput. Phys*. – 1991. – V. 94. – P. 255–283.
123. Markót, M.Cs. A new verified optimization technique for the «packing circles in a unit square» problems / M.Cs. Markót, T.A. Csendes // *SIAM Journal on Optimization*. – 2005. – V. 16. – P. 193–219.
124. Melzak, Z.A. Companion to concrete mathematics / Z.A. Melzak // Wiley–Interscience, N. Y., 1973.
125. Melzak, Z.A. On the problem of Steiner / Z.A. Melzak // *Canad. Math. Bull*, 1961. – V. 4. – P. 143–148.
126. Mitsuo, G. Network models and optimization: Multiobjective genetic algorithm approach / G. Mitsuo, C. Runwei, L. Lin. – London: Springer, 2008. – 692 p.
127. Moore, E.F. The shortest path through a maze / E.F. Moore. – *Proc. Intern. Symp. on the Theory of Switching*. – 1957. – Part II. – 285 p.
128. Morgan, F. Hexagonal economic regions solve the location problem / F. Morgan, R. Bolton // *The American Mathematical Monthly*. – 2002. – V. 109, № 2. – P. 165–172.
129. NET library for embedding CLIPS in to .NET applications [Электронный ресурс]. – URL: <http://sourceforge.net/projects/clipsnet>.
130. Nurmela, K.J. Covering a square with up to 30 equal circles / K.J. Nurmela, P.R.J. Ostergard. – *Res. rept A62. Lab. Technol. Helsinki Univ*. – 2000. – 20 p. – URL:

http://www.ymmf.hu/sites/talata.istvan.ymmf.hu/files/2015_osz/sikgeom_bemut1_segedanyagok/covering_a_square_with_circles.pdf

131. Pedroso, J.P. Recursive circle packing problems / J.P. Pedroso, S. Cunha, J.N. Tavares // International Transactions in Operational Research. – 2014. – V. 23, №1. – P. 355–368.
132. Peikert, R. Packing Circles in a Square: A Review and New Results / R. Peikert, D. Wiirtz, M. Monagan, C. Groot // System Modelling and Optimization, Lecture Notes in Control and Information Sciences. – Springer–Verlag, Berlin. – 1992. – V. 180. – P. 45–54.
133. Preparata, F.P. Computational Geometry. An Introduction / F.P. Preparata, M.I. Shamos. – N.Y.: Springer– Verlag, 1985. – 390 p.
134. Schrijver, A. Combinatorial optimization / A. Schrijver. – Springer–Verlag Berlin Heidelberg, 2003. – V. 24. – 1879 p.
135. Specht, E. Packomania [Электронный ресурс]. – 2010. – URL: <http://www.packomania.com>
136. Stoyan, Y.G. Covering a compact polygonal set by identical circles / Y.G. Stoyan, V.M. Patsuk // Comput. Optim. Appl. – 2010. – V. 46. – P. 75–92.
137. Szabó, P.G. Packing Equal Circles in a Square I.: Problem Setting and Bounds for Optimal Solutions / P.G. Szabo, T. Csendes, L.G. Casado, I. Garcia // Optimization Theory: Recent Developments. – 2001. – V. 59. – P. 191–206.
138. Szabó, P.G. Packing up to 200 equal circles in a square / P.G. Szabó, E. Specht // Models and Algorithms for Global Optimization. – 2007. – V. 4. – P. 141–156.
139. Szirmai, J. The optimal ball and horoball packings of the Coxeter tilings in the hyperbolic 3–space / J. Szirmai // Beitr. Algebra Geom. – 2005. – V. 46, №. 2. – P.545–558.
140. Szirmai, J. The optimal ball and horoball packings to the Coxeter honeycombs in the hyperbolic d–space Beitr / J. Szirmai // Algebra Geom. – 2007. – V. 48, № 1. – P. 35–47.

141. Szirmai, J.A. Candidate for the densest packing with equal balls in Thurston geometries / J.A. Szirmai // Beitr. Algebra Geom. – 2014. – V. 55. – P. 441–452.
142. Tomin, N. Random Forest Based Model for Preventing Large-scale Emergencies in Power Systems / N. Tomin, A. Zhukov, D. Sidorov, V. Kurbatsky, D. Panasetsky, V. Spiryaev // International J. of Artificial Intelligence. – 2015. – V.13, № 1. – P. 211-228.
143. Verter, V. Uncapacitated and capacitated facility location problems / V. Verter // In Foundations of location analysis. H.A. Eiselt, V. Marianov eds. – N.Y.: Springer, 2011. – P. 25–37.
144. Wiirtz, D. The history of packing circles in a square / D. Wiirtz, M. Monagan, R. Peikert // Maple Technical Newsletter 0. – 1994. – P. 35–42.

Приложения

Приложение 1. Свидетельства о государственной регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО
о государственной регистрации программы для ЭВМ
№ 2015616554

**Программный модуль оптимального размещения
логистических центров**

Правообладатель: *федеральное государственное бюджетное
образовательное учреждение высшего профессионального
образования «Иркутский национальный исследовательский
технический университет» (RU)*

Авторы: *Казаков Александр Леопидович (RU), Нгуен Лиен Гуй
(RU), Лемперт Анна Ананьевна (RU)*

Заявка № **2015613368**
Дата поступления **23 апреля 2015 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ **15 июня 2015 г.**



Врио руководителя Федеральной службы
по интеллектуальной собственности



Л.Л. Кирий

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2016614997

**Программный модуль построения оптимальных покрытий
и упаковок в оптически неоднородной среде**

Правообладатель: *федеральное государственное бюджетное
образовательное учреждение высшего образования «Иркутский
национальный исследовательский технический университет»
(RU)*

Авторы: *Казаков Александр Леонидович (RU), Нгуен Лиен Гуй
(RU), Лемперт Анна Ананьевна (RU)*



Заявка № 2016611131

Дата поступления 15 февраля 2016 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 13 мая 2016 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Иалиев

Приложение 2. Акты о внедрении программного продукта

**АКТ**

О внедрении программного продукта «Программный модуль оптимального размещения логистических центров»
(автор Нгуен Гуй Лиём)

Комиссия в составе: Бахвалов С.В., к.т.н., зав. кафедрой автоматизированных систем ИРНИТУ (председатель), Григорьев С.В., доцент кафедры автоматизированных систем, Засядко А.А., профессор кафедры автоматизированных систем, составила настоящий акт о том, что программный продукт «Программный модуль оптимального размещения логистических центров», Автором которого является Нгуен Гуй Лиём, внедрен в учебный процесс и использован на кафедре автоматизированных систем в учебных курсах «Методы оптимизации» и «Системология».

Использование созданной Нгуен Гуй Лиём программы для ЭВМ ««Программный модуль оптимального размещения логистических центров» (Свидетельство о государственной регистрации № 2015616554 от 15 июня 2015), позволяет решить задачи построения кратчайшего маршрута, идентификации логистических зон обслуживания, а также оптимального размещения логистических центров в условиях конкуренции и кооперации. Результаты расчетов представляются в виде интерактивных 2D и 3D иллюстраций, а также в виде таблиц.

Программный модуль имеет простой, удобный и интуитивно понятный интерфейс. Указанный программный модуль предназначен для исследователей, аспирантов и магистрантов, научные интересы которых лежат в области методов оптимизации, может оказаться полезным специалистам по планированию развития транспортно-логистической инфраструктуры, а также студентам при изучении соответствующих дисциплин.

Председатель комиссии

Члены комиссии

Бахвалов С. В.

Григорьев С. В.

Засядко А. А.



LE QUY DON TECHNICAL UNIVERSITY

236 Hoang Quoc Viet Str., Hanoi, Vietnam

Tel/Fax: 84.4.37555706

e-mail: phonghtqt@mta.edu.vn

website: www.lqdtu.edu.vn

Dated 19/5/ 2016

CERTIFICATE ON SOFTWARE IMPLEMENTATION

«The software module for optimal placement of logistic centers»

(Authors Kazakov A.L., Lempert A.A., Nguyen Huy Liem)

The Commission of Ta Huu Vinh, Ph.D., Deputy Director of the Institute of Simulation Technology (Commission Chairman), Bui The Truyen, Ph.D., Deputy Director of the Institute, Nguyen Trung Kien, Ph.D., Head of the Department "GIS and System's Modeling" (Commission members), certifies that the software product «The software module for optimal placement of logistic centers», which is developed by Mr. Kazakov Alexander Leonidovich, Ms. Lempert Anna Ananievna, Mr. Nguyen Huy Liem, is implemented in the educational process and used in the activities of the Institute of Simulation Technology, Le Quy Don Technical University, Vietnam.

Using the computer program "The software module for optimal placement of logistics centers", (The certificate of state registration № 2015616554 dated June 15th 2015) allows us to solve the problem of constructing the shortest route, the identification of logistic service areas, as well as optimal placement of logistics centers in the conditions of competition and cooperation. The calculation results are presented in the form of 2D, 3D interactive illustrations and tables.

The software module has a simple, user-friendly and intuitive interface. The software module is intended for researchers, graduate students and undergraduates, who have the scientific interests in the field of optimization methods. It may be useful to specialists in planning the development of transport and logistics infrastructure, as well as students in the study of relevant disciplines.

Chairman of the Commission

 Ta Huu Vinh, Ph.D.

The Commission members

 Bui The Truyen, Ph.D.

 Nguyen Trung Kien, Ph.D.



Vice-President of Le Quy Don Technical University
Assoc. Prof. Le Ky Nam



УТВЕРЖДАЮ
 Директор института
 Кибернетики им. Е.И.Попова
 О. В. Дударева
 2016 г.

АКТ

О внедрении программного продукта «Программный модуль построения оптимальных покрытий и упаковок в оптически неоднородной среде»
 (авторы Казаков А.Л., Нгуен Г.Л., Лемперт А.А.)

Комиссия в составе: Бахвалов С.В., к.т.н., зав. кафедрой автоматизированных систем ИРНТУ (председатель), Григорьев С.В., доцент кафедры автоматизированных систем, Засядко А.А., профессор кафедры автоматизированных систем, составила настоящий акт о том, что программный продукт «Программный модуль построения оптимальных покрытий и упаковок в оптически неоднородной среде», Авторами которого являются Казаков Александр Леонидович, Нгуен Гуй Лиен, Лемперт Анна Ананьевна внедрен в учебный процесс и использован на кафедре автоматизированных систем в учебных курсах «Системология».

Использование созданной Казакова А.Л., Нгуен Г.Л., Лемперта А.А. программы для ЭВМ «Программный модуль построения оптимальных покрытий и упаковок в оптически неоднородной среде» (Свидетельство о государственной регистрации 2016614997 от 13 мая 2016), позволяет решить задач построения оптимальных упаковок и покрытия равными кругами неодносвязных множеств в неевклидовой метрике, и оптимального размещения логистических центров разного назначения (сотовой связи, беспроводного Интернета, скорой помощи, банковских терминалов и т.д.) при непрерывном распределении потребителей. Результаты расчетов представляются в виде интерактивных 2D и 3D иллюстраций, а также в виде таблиц.

Программный модуль имеет простой, удобный и интуитивно понятный интерфейс. Указанный программный модуль предназначен для исследователей, аспирантов и магистрантов, научные интересы которых лежат в области методов оптимизации, может оказаться полезным специалистам по планированию развития транспортно-логистической инфраструктуры, а также студентам при изучении соответствующих дисциплин.

Председатель комиссии

Члены комиссии

Бахвалов С. В.

Григорьев С. В.

Засядко А. А.

Приложение 3. Дипломы

