

**Э.В. Батоева
Д.Ю. Мельников**

**ПРОЕКТИРОВАНИЕ ОПТИМАЛЬНОГО
МАРШРУТА ПРОЛЕГАНИЯ ЛИНЕЙНЫХ
ОБЪЕКТОВ**

*под редакцией
доктора экономических наук,
профессора Плотникова А.Н.*

2018

УДК 711
ББК 85.11
П 79

Рецензенты:

*Постюшков А.В., доктор экономических наук, профессор
Смирнова Т.В., доктор социологических наук, профессор*

Батоева Э.В., Мельников Д.Ю.

П 79 Проектирование оптимального маршрута пролегания линейных объектов / Э.В. Батоева, Д.Ю. Мельников – Саратов: изд-во ООО ЦПМ «Академия бизнеса», 2018. – 98 с.

ISBN 978-5-6041783-0-0

В монографии освещены основные направления теории и практики проектирования определения оптимального маршрута линейного объекта на примере прокладки трубопровода. Исследовано использование автоматизированных методов для построения маршрутов. Показаны особенности, достоинства и недостатки подходов проектирования линейных объектов.

Предназначена для студентов, исследователей и специалистов в сфере проектирования, строительства.

ISBN 978-5-6041783-0-0

УДК 711
ББК 85.11
П 79

© Батоева Э.В., 2018
© Мельников Д.Ю., 2018
© ЦПМ Академия бизнеса, 2018

Оглавление

ВВЕДЕНИЕ.....	5
1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МЕТОДОВ И АЛГОРИТМОВ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО МАРШРУТА ПРИ ПРОЕКТИРОВАНИИ ЛИНЕЙНЫХ ОБЪЕКТОВ.....	8
1.1. Понятие линейного объекта. Особенности кадастровой деятельности в отношении линейных объектов	8
1.1.1. Методические подходы к планировке территорий в части определения зон планируемого размещения линейных объектов	9
1.2. Современное состояние проблемы.....	13
1.2.1. Проектирование трубопроводов.....	14
1.2.2. Проблема определения оптимального маршрута	16
1.2.3. Обзор законодательных и нормативно-правовых документов	18
1.3. Теоретическое обоснование выбора методов и алгоритмов.....	21
1.3.1. Построение графа связности по данным растровой поверхности.....	21
1.3.2. Построение оптимального маршрута с использованием разных шаблонов.....	24
1.3.3. Анализ сложности алгоритмов	26
1.3.4. Алгоритмы поиска оптимального пути.....	33
1.3.5. Что используется в ArcGis.....	40
1.4. Анализ и выбор алгоритма для реализации.....	41
1.5. Выбор структуры данных.....	42
1.5.1. Хэш-таблица	44
1.5.2. Хеширование с цепочками.....	46
1.5.3. Хеширование с открытой адресацией.....	47
2. АНАЛИЗ ВХОДНЫХ ДАННЫХ И ВАРИАНТОВ ОПТИМИЗАЦИИ АЛГОРИТМА А*	49
2.1. Краткий анализ деятельности предприятия	49
2.2. Входные данные и их анализ	51
2.3. Анализ алгоритма А*	55
2.4. Оптимизация А*	59

3. РАЗРАБОТКА СЦЕНАРНОГО ЯЗЫКА ПРИМЕНЕНИЯ АЛГОРИТМА А* И ЕГО АПРОБАЦИЯ НА ПРИМЕРЕ НЕФТЕПРОВОДА «А МЕСТОРОЖДЕНИЕ – Б».....	61
3.1. Проектирование сценарного языка	61
3.2. Общие сведения о программной среде ArcGIS.....	64
3.3. Преобразование векторных объектов в растровые с помощью API ArcGIS.....	65
3.4. Анализ на возможность оптимизации нефтепровода «А месторождение – Б»	73
3.4.1. Показатели действующего трубопровода «А месторождение – Б» ...	73
3.4.2. Применение алгоритма оптимизации и анализ результатов	76
ЗАКЛЮЧЕНИЕ	85
Список использованной литературы.....	86
Приложение 1	92
Приложение 2	93
Приложение 3	94
Приложение 4	95
Приложение 5	96
Приложение 6	97

ВВЕДЕНИЕ

В России основным способом транспортировки энергоносителей являются трубопроводы. Поскольку применение трубопроводов экономически выгодно, а работают они в любую погоду и в любое время года, это средство транспортировки нефти действительно незаменимо – особенно для России, с ее огромными территориями и сезонными ограничениями на использование водного транспорта.

Трубопроводы по своему назначению делятся на 3 типа:

- Внутренние (внутрипромысловые). Соединяют различные объекты и установки на промыслах, НПЗ и нефтебазах.

- Местные (межпромысловые). По сравнению с внутренними имеют большую протяженность (до нескольких десятков км), соединяют нефтепромыслы или НПЗ с головной станцией магистрального нефтепровода.

- Магистральные. Характеризуются большой протяженностью (сотни и тысячи км), поэтому перекачка ведется несколькими станциями, расположенными на трассе.

Выбор трассы трубопроводов должен производиться по критериям оптимальности, при этом в качестве таких критериев принимают затраты на сооружение, техобслуживание и ремонт трубопровода при эксплуатации, учитывая и затраты по обеспечению охраны окружающей среды. Кроме того, учитывается металлоёмкость, безопасность, заданное время строительства и наличие дорог.

В нефтегазовой отрасли можно выделить 4 основных стадии строительных проектных работ:

- Обоснование инвестиций (ОИ) в строительство или реконструкцию, или технико-экономическое обоснование строительства (ТЭО-проект).

- Основные технические решения (ОТР).

- Разработка проектной документации (стадия «П»).

- Разработка рабочей документации (стадия «РД») для строительства (реконструкции) объекта.

Без обоснования инвестиций проект будет приостановлен в самом начале, но стоимость такой работы у подрядных организаций достаточно велика: порядка 5-20% стоимости разработки всей проектной документации, временные затраты при этом составляют от 4 до 16 недель.

Выбор маршрута прокладки — это первый существенный шаг в процессе проектирования и строительства линейного объекта, например, трубопровода, который может оказать значительное влияние на строительство и функционирование трубопровода в целом, поэтому оптимизация этого процесса может существенно повлиять на финансовые и материальные ресурсы, необходимые для строительства трубопровода. Таким образом, эффективный процесс выбора маршрута прокладки важен для минимизации экономических потерь.

В связи с этим становится очевидным, что определение оптимального маршрута прокладки трубопровода является актуальной экономической и научно-технической задачей. Использование автоматизированных методов для построения маршрутов фактически не распространено. Классические методики основываются на бумажно-аналоговых технологиях. При незначительном изменении входного набора данных (смена начальных и конечных пунктов, появление новой или просто другой информации) все операции по подготовке данных приходится делать заново, и, как правило, вручную.

Цель работы – уменьшение стоимости работ на этапе ТЭО проекта разработки месторождения углеводородов, за счет автоматизации процесса определения оптимального маршрута прокладки внутрипромыслового трубопровода.

Для достижения цели были решены следующие задачи:

- Исследована литература по предметной области, имеющееся программное обеспечение и инструменты по предметной области.
- Проведен анализ существующих методов преобразования массивов в графы;

- Проведен сравнительный анализ существующих алгоритмов построения оптимального расстояния;
- Реализовано преобразование векторных объектов в единую растровую карту со стоимостными оценками;
- Реализован алгоритм оптимального построения маршрута A star;
- Реализован экспорт полученных данных в табличном и графическом виде;
- Разработан сценарный язык для программного обеспечения ArcGIS;
- Проведена апробация приложения.
- Оценена эффективность проложенного пути в сравнении с уже существующим трубопроводом.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МЕТОДОВ И АЛГОРИТМОВ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО МАРШРУТА ПРИ ПРОЕКТИРОВАНИИ ЛИНЕЙНЫХ ОБЪЕКТОВ

1.1. Понятие линейного объекта. Особенности кадастровой деятельности в отношении линейных объектов

Вопрос о линейных объектах всегда был и является до сих пор одним из самых сложных в градостроительном и земельном законодательстве РФ. Отсутствует эффективное и универсальное нормативно-правовое регулирование, что существенно затрудняет отношения земельно-имущественного комплекса и градостроительства.

Согласно письму Министерства регионального развития Российской Федерации от 20 мая 2011г. №13137-ИП/08 «В соответствии с Градостроительным кодексом Российской Федерации к линейным объектам отнесены линии электропередачи, линии связи (в том числе линейно-кабельные сооружения), трубопроводы, автомобильные дороги, железнодорожные линии и другие подобные сооружения, расположенные в пределах красных линий - линий, которые обозначают существующие, планируемые (изменяемые, вновь образуемые) границы территорий общего пользования, границы земельных участков».

Федеральный закон от 27 июля 2010 г. N 210-ФЗ "Об организации предоставления государственных и муниципальных услуг" даёт пояснение что понимается под термином красные линии. «Красные линии — линии, которые обозначают существующие, планируемые (изменяемые, вновь образуемые) границы территорий общего пользования, границы земельных участков, на которых расположены линии электропередачи, линии связи (в том числе линейно-кабельные сооружения), трубопроводы, автомобильные дороги,

железнодорожные линии и другие подобные сооружения (далее — линейные объекты)»

Само понятие линейного объекта складывается из нескольких нормативно-правовых актов и складывается из нескольких характеристик:

- 1) Протяженность объекта (длина объекта) значительно преобладает над его шириной;
- 2) Объект - это всегда сооружение. Сооружение - это результат строительства, представляющий собой объемную, плоскую или линейную строительную систему, имеющую наземную, надземную и (или) подземную части, состоящую из несущих, а в отдельных случаях и ограждающих строительных конструкций и предназначенную для выполнения производственных процессов различного вида, хранения продукции, временного пребывания людей, перемещения людей и грузов;
- 3) Связь с землей. Можно выделить надземные (воздушные), наземные (поверхностные) и подземные ЛО;
- 4) Назначение линейного объекта. В зависимости от назначения линейные объекты подразделяются на: транспортные коммуникации (автомобильные дороги, троллейбусные линии, железные дороги); канализационные и ливневые коллекторы; электрические сети; линии связи; водоводы и водопроводы; газопроводы; нефтепроводы; каналы; мелиоративные каналы.

1.1.1. Методические подходы к планировке территорий в части определения зон планируемого размещения линейных объектов

Федеральный закон от 23 июня 2014 г. № 171-ФЗ «О внесении изменений в Земельный кодекс Российской Федерации и отдельные законодательные акты РФ», вступивший в силу с 1 марта 2015г., постановил строгий порядок действий по формированию и предоставлению земельных участков, который опирается на

систему документов территориального планирования и градостроительного зонирования. Узаконен приоритет системного комплексного планированию развития территорий, которое во многом связано со строительством инфраструктурных сооружений, представленных в значительной степени линейными объектами.

Предоставление земельного участка из земель, государственной или муниципальной собственности, осуществляется под функцию, определенную по градостроительным документам. Помимо этого, земельный участок может выделяться как для строительства, так и для эксплуатации построенного объекта.

Существующими проблемами планировки территорий для размещения линейных объектов посвящен ряд статей, например, О.А. Баевского, М.В. Бочарова, Д.В. Королева, О.А. Жарковой, Е.А. Семенченко. [13,18,20,33] Так же затрагиваемые вопросы рассматриваются в статьях С.Д. Митягина [26.27.28], а также в статье Т.В. Варгиной [19], в них затронуты методические вопросы подготовки документов территориального планирования, градостроительного зонирования, планировки территорий.

Подготовка необходимой документации по планировке территории размещения линейного объекта согласно с ч.2 ст. 41 Градостроительного кодекса РФ осуществляется для застроенных или предназначенных для застройки территорий. Обязательность подготовки документов по планировке территории (ППТ) для линейных объектов введено федеральным законом от 19 июля 2011 г. № 246-ФЗ. До этого для линейных объектов требовался лишь градостроительный план земельного участка.

Новая редакция Земельного кодекса упростила и упорядочила процедуру предоставления земельного участка для строительства линейного объекта. Согласно п. 5 ст. 11 этой редакции образование земельных участков или земельного участка для размещения линейных объектов федерального, регионального или местного значения может осуществляться исключительно в рамках утвержденного проекта межевания территории, который является частью градостроительной документации по планировке территории.

В свою очередь разработка документации по планировке территории для планируемого размещения объектов федерального, регионального или местного значения должна вестись строго в соответствии с документами территориального планирования и градостроительного зонирования на основании решения органа исполнительной власти РФ, субъекта РФ или органа местного самоуправления в зависимости от значения планируемого объекта.

Согласно положениям, ст. 39 новой редакции Земельного кодекса РФ на земельных участках государственной и муниципальной собственности на определенных условиях допускается размещение объектов капитального строительства, эксплуатация объектов и земельных участков или их частей на условиях публичного сервитута, в том числе и для линейных объектов.

В таблице 1 приведена классификация линейных объектов по специфике хозяйственной деятельности, способам прокладки и градостроительным условиям.

Таблица 1

Классификация линейных объектов

Виды линейных объектов	Способы прокладки	Потребность в земельном участке	Ограничения: зоны с особыми условиями использования территории (ЗОУ), Сервитуты (С).
Транспорт			
Автомобильные дороги	Наземный	Есть	ЗОУ
	Надземный	Под опорами	ЗОУ
	Подземный	Нет	ЗОУ, С
	Подводный	Нет	ЗОУ, С
	Надводный	Под опорами	ЗОУ, С
Железные дороги	Наземный	Есть	ЗОУ, С
	Наземный	Под опорами	ЗОУ, С
	Подземный	Нет	ЗОУ, С
	Подводный	Нет	ЗОУ, С
	Надводный	Под опорами, эстакадой	ЗОУ, С

Продолжение таблицы 1

Подъемный транспорт (канатные дороги, фуникулеры)	Наземный	Под опорами	ЗОУ, С
	Надводный		ЗОУ, С
Прочие виды транспорта (метрополитен, монорельс)	Наземный	Есть	ЗОУ, С
	Наземный	Под опорами	ЗОУ, С
	Подземный	Нет	ЗОУ, С
	Подводный	Нет	ЗОУ, С
	Надводный	Под опорами	ЗОУ, С
Трубопроводы	Наземный	Есть	ЗОУ, С
	Наземный	Под опорами, эстакадой	ЗОУ, С
	Подземный	Нет	ЗОУ, С
	Подводный	Нет	ЗОУ, С
	Надводный	Под опорами, эстакадой	ЗОУ, С
Энергетика и связь			
Линии электропередачи	Наземный	Под опорами	ЗОУ, С
	Подземный	Нет	ЗОУ, С
	Подводный	Нет	ЗОУ, С
	Надводный	Нет	ЗОУ, С
Инженерно-транспортное обеспечение территории: магистральные (общегородского значения) коммуникации			
Магистральные городские автомобильные дороги	Наземный	Есть	ЗОУ, С
	Наземный	Под опорами	ЗОУ, С
	Подземный	Нет	ЗОУ, С
	Подводный	Нет	ЗОУ, С
	Надводный	Нет	ЗОУ, С
Водоводы и канализационные коллекторы	Подземный	Нет	ЗОУ, С
Теплотрассы	Наземный	Есть	ЗОУ, С
	Надземный	Нет	ЗОУ, С
	Подземный	Нет	ЗОУ, С

Газопроводы	Надземный	Под опорами, эстакадой	ЗОУ, С
	Подземный	Нет	ЗОУ, С
	Подводный	Нет	ЗОУ, С

В зависимости от выбора способа прокладки линейного объекта (наземный, надземный, подземный, надводный и подводный) выделяются особенности зоны размещения данного объекта капитального строительства. Данная зона может представляться в виде или сплошной полосы единого земельного участка полностью для всего сооружения, или прерывистой полосы, состоящей из отдельных дробных участков под эстакадами и опорами, а также над инженерными сооружениями (многоконтурные участки). В определенных случаях земельные участки могут не существовать.

1.2. Современное состояние проблемы

С целью выбора и анализа литературы был проведён поиск научно-исследовательских трудов на темы, связанные с строительством и проектированием линейных объектов. Тематика оптимизации пути прокладки трубопроводов затрагивается в кандидатских диссертациях Рыльского И.А. «Геоинформационное обеспечение автоматизированного трассирования трубопроводов» и Кузнецова Р.М. «Определение оптимального маршрута прокладки трубопровода». [17]

На основании литературных источников, использовавшихся при написании данных диссертаций, а также нескольких других работ [16, 12, 13] и статей [11] можно выделить несколько существенных групп информационных источников необходимых для последующего анализа:

- статьи, касающиеся проектирования, строительства линейных инженерных сооружений или построения их моделей с учетом различных факторов;

- проектно-техническая документация и технические регламенты по проектированию инфраструктуры месторождений углеводородов;

- проектные технологические документы на разработку нефтяных месторождений;
- литература по теории проектирования и строительства трубопроводов;
- документы по оценке экономической эффективности проектов;
- технологии и организация трубопроводного строительства в условиях обводненной и заболоченной местности;
- законодательные и нормативно-правовые документы, регулирующие деятельность предприятий нефтедобывающей промышленности.

1.2.1. Проектирование трубопроводов

Алиев Р.А. и Бородавкин П.П. возможно являются ключевыми деятелями в теории проектирования и планирования трубопроводов в России. Помимо этого, Бородавкин П.П. является автором теории проектирования магистральных трубопроводов в сложных условиях [16, 17]. Наибольшая часть их работ направлена на создание и изучение рабочей и проектной документации для строительства и реконструкции объектов добычи углеводородов и промыслов.

В работах Алиева Р.А. и Бородавкина П.П. описывается теория проектирования и строительства магистральных трубопроводов. Авторы провели классификацию типов трубопроводов и типов местности, через которые они проходят. Так же приведены примеры выбора различных критериев оптимальности: аддитивных и неаддитивных. К аддитивным можно отнести критерии по капитальным или приведенным затратам, а также критерии по времени строительства, к неаддитивным – вероятность завершения строительства в заданный срок.

В своей книге «Трубопроводный транспорт нефти и газа» [11] Алиев Р.А. обосновал выбор наиболее выгодного способа транспортировки нефтяных грузов, а также особенности каждого вида транспорта. Транспортировка с помощью трубопроводов имеет ряд неоспоримых преимуществ по сравнению с другими видами транспортировки, к числу которых можно отнести: трасса

трубопровода короче трасс других видов транспорта, трубопроводный транспорт, в отличие от других, непрерывный, меньше потери нефти и нефтепродуктов, трубопроводный транспорт наиболее механизированный и легче других поддается автоматизации.

Алиев Р.А. в своей работе отмечает, что выбор того или иного вида транспорта осуществляется технико-экономическим сравнением вариантов [11]. Важнейшими же показателями являются капитальные затраты и эксплуатационные расходы, так же приведены примеры расчётов этих показателей.

Большинство статей про оптимизацию маршрутов, написанных в последнее время, касаются изучения методологического обеспечения проектирования трасс. В статьях рассматриваются подходы к оптимизации прокладки различных линейных объектов строительства, а также разработка геоинформационного обеспечения.

Рыльский И.А. в статье «Оптимизация трассы трубопровода с помощью ГИС» [15] предложил метод, основанный на факторном анализе с использованием средств ГИС. Данный метод дает возможность проводить оптимизацию трассы не в узком коридоре вдоль линии, принятой на этапе обоснования инвестиций, а по всей территории региона строительства. При выполнении оптимизации используются данные о рельефе, углах наклона и экспозиции склонов, кривизне поверхности, высотах, характеристиках грунтов, сейсмичности и геологическом строении, растительности, мерзлотных процессах и по прочим характеристикам.

К сожалению, предложенная Рыльским И.А. методика не может использоваться на практике в полной мере. Причиной для этого послужил анализ предложенного в статье списка факторов и объектов, влияющих на стоимость строительства. Например, геологическое строение не оказывает существенного влияния на выбор маршрута строительства трубопровода. Также хочется отметить, что автор статьи в своей работе в качестве примера рассматривал

идеализированный вариант – наличие полного набора данных по всем показателям, на практике такой ситуации может не быть.

1.2.2. Проблема определения оптимального маршрута

В своей кандидатской диссертации «Определение оптимального маршрута прокладки газопровода» Кузнецов Р.Н. описывает возможность построения оптимального маршрута прокладки трубопровода с учетом ряда факторов, оказывающих влияние.

Автором рассматриваются две системы отображения данных в ГИС: растровую и векторную. При проведении анализа сетей комфортно применять векторную систему, так как это позволяет отображать данные в виде графа. Растровая система больше подходит для случаев, когда необходимо принимать во внимание большое количество факторов.

Р.Н. Кузнецов приводит ряд действий, которые необходимо осуществить, для того чтобы спланировать оптимальный маршрут: выявить факторы, влияющие на прокладку трубопровода, оценить весовые коэффициенты факторов, просчитать стоимостную карту и спланировать прокладку пути, соответствующий наименьшей стоимости.

Кузнецов Р.Н. в своей диссертации строит граф связи ячеек сетки стоимостной карты и приводит алгоритм расчета поверхности накопленной стоимости, разработанный на базе алгоритма Дейкстры.

К числу достоинств этой диссертации можно отнести то, насколько автор хорошо осветил тему, разработанные им алгоритмы и методики. Алгоритм при обходе итоговой поверхности накопленной стоимости одновременно учитывает данные о направлении прокладки и реализует целевую функцию минимальной стоимости прокладки газопровода.

К недостаткам данной работы относятся отсутствие обоснования выбора алгоритма Дейкстры в качестве базового алгоритма для модификации, а также то, что автор не стал затрагивать тему улучшения быстродействия алгоритма.

В настоящее время методы трассирования и выбора оптимального расположения трассы трубопровода (на стадии рабочего проектирования) разработаны достаточно хорошо, однако в этом случае трасса трубопровода не может выходить за пределы коридора, выбранного на стадии технико-экономического обоснования проекта (ТЭО). В то же время методология выбора трассы трубопровода на этапе ТЭО разработана недостаточно. Методики подготовки этих данных к использованию: сбору, анализу и цифровой интерпретации пространственного распределения характеристик окружающей среды были рассмотрены в диссертационной работе Рыльского И.А.

В своей кандидатской диссертации Рыльский И.А. «Геоинформационное обеспечение автоматизированного трассирования трубопроводов» описывает возможности более обширного использования ГИС в сочетании с методами, предложенными еще советскими авторами, например, Бородавкиным П.П. Автор делает упор на автоматизацию трассировки с учетом ранее разработанных научно-методологических подходов количественной оценки качества предполагаемых маршрутов трасс трубопроводов.

В результате проведения анализа выявлен целый ряд недостатков диссертации. Во-первых, автор присваивает себе разработку уникальной методики, которая основывается не на четком классифицировании территории (где стоимость строительства элементарного участка трубопровода одинакова в пределах выделенного типа территории), а на обобщенном анализе разных факторов, усложняющих трассирование и следующей за этим оценке их совокупного воздействия для каждой точки территории.

В своей кандидатской диссертации Басов Е.Д. описал все возможные технологии и организацию трубопроводного строительства при условиях заболоченной и обводненной местности, Басов Е.Д. проводит анализ строительства переходов через малые преграды, а также оценку распределения малых переходов по природно-климатическим зонам.

1.2.3. Обзор законодательных и нормативно-правовых документов

Обоснование инвестиций в строительство трубопровода позволяет получить достаточно достоверную информацию о возможности и приблизительной стоимости реализации проектируемого объекта. При этом сроки разработки ТЭО-проекта – от 2 до 6 недель (в зависимости от величины предполагаемого объекта), а стоимость разработки составляет 15-20% от стоимости разработки проектной документации.

ТЭО-проект позволяет определить основную технологическую схему, тип и примерную стоимость основного технологического оборудования, а соответственно и размеры необходимой под строительство площадки, что в свою очередь, позволяет достаточно точно оценить затраты на реализацию объекта целиком.

Основные технические решения разрабатываются до начала работ по проектной документации и на достаточно сложные технологически насыщенные объекты. В составе ОТР обычно разрабатываются основные схемы: технологическая схема, электроснабжения, автоматизации.

ОТР позволяет согласовать основные параметры технологического и вспомогательного оборудования, что значительно сокращает количество переделок при разработке проектной документации. Срок разработки ОТР 2-3 недели и стоимость не более 10% от стоимости проектных работ.

Разработка проектной документации всегда проходит государственную и ведомственную экспертизу, поэтому ее форма должна соответствовать требованиям законодательства. Данная стадия подразумевает разработку всех необходимых разделов проектной документации, согласно Постановлению РФ №87 «О составе разделов проектной документации и требованиях к их содержанию» и Градостроительного кодекса ст.48 п.13. Документация на линейные объекты капитального строительства включает в себя 10 разделов:

1. «Пояснительная записка» (ПЗ);
2. «Проект полосы отвода» (ППО);

3. «Технологические и конструктивные решения линейного объекта. Искусственные сооружения» (ТКР);
4. «Здания, строения и сооружения, входящие в инфраструктуру линейного объекта» (ИЛО);
5. «Проект организации строительства» (ПОС);
6. «Проект организации работ по сносу (демонтажу) линейного объекта» (ПОД);
7. «Мероприятия по охране окружающей среды» (ООС);
8. «Мероприятия по обеспечению пожарной безопасности» (ПБ);
9. «Смета на строительство» (СМ);
10. «Иная документация в случаях, предусмотренных федеральными законами».

Разработка рабочей документации является самым последним этапом. В ГОСТ Р 21.1001-2009 дается определение рабочей документации как совокупности текстовых и графических документов, обеспечивающих реализацию принятых в утвержденной проектной документации технических решений объекта капитального строительства, необходимых для производства строительных и монтажных работ, обеспечения строительства оборудованием, изделиями и материалами и/или изготовление строительных изделий. В состав рабочей документации входят основные комплекты рабочих чертежей, спецификации оборудования, изделий и материалов, сметы, другие прилагаемые документы, разработанные в дополнение к рабочим чертежам основного комплекта. [4]

В СНиП 2.05.06-85 приведены строительные нормы и правила по проектированию и реконструкции магистральных трубопроводов. Нормы не распространяются на проектирование трубопроводов, прокладываемых на территории городов и других населенных пунктов, в морских акваториях и промыслах, а также трубопроводов, предназначенных для транспортирования газа, нефти, нефтепродуктов и сжиженных углеводородных газов, оказывающих коррозионные воздействия на металл труб или охлажденных до температуры

ниже минус 40 °С. В тексте норм приведены категории трубопроводов в зависимости участков, по которым они проходят.

В СНиП III-42-80* даны общие положения по строительству трубопроводов. В документе приведена информация о строительстве переходов через естественные и искусственные препятствия, в особых природных условиях:

- прокладка через болота и обводненные участки;
- прокладка трубопроводов в горных условиях прокладка трубопроводов в тоннелях;
- прокладка трубопроводов в присадочных грунтах;
- прокладка трубопроводов в барханных песках, на поливных землях и при пересечении соров;
- прокладка трубопроводов в вечномёрзлых грунтах.

Анализ законодательных и нормативно-правовых документов, регулирующих деятельность предприятий нефтедобывающей промышленности, показал, что не все аспекты проектирования и строительства промысловых трубопроводов освещены должным образом, отсутствует четкая классификация всех возможных зон и коэффициентов удорожания по ним.

Однако, можно утверждать, что при прокладке трубопровода на территории Сибири особое внимание следует обратить на следующие участки:

- территории, расположенные в непосредственной близости от населенных пунктов;
- особо охраняемые зоны, военные части;
- заповедники, болота и обводненные участки;
- территории с просадочными грунтами;
- вечномёрзлые грунты.

1.3. Теоретическое обоснование выбора методов и алгоритмов

1.3.1. Построение графа связности по данным растровой поверхности

Существует два варианта возможного входа требуемых данных: карты и геологические параметры из базы данных. По причине того, что информация на картах является более обширной, то предлагается использовать эти данные с возможностью последующего внедрения в корпоративную базу.

При использовании растровой модели стоимости существует необходимость обозначить метод, по которому необходимо строить граф связности ячеек сетки. Обычно, граф связности создается, опираясь на следующие условия:

1. Центры ячеек сетки служат узлами графа;
2. Связи между соседними ячейками служат дугами графа. [24]

По причине того, что способом представления растровых данных служит регулярная сетка, граф связности следует строить как соединение четкого количества соседствующих ячеек сетки дугами графа. Существуют разные вариации данного метода:

1. Шаблон Ладьи
2. Шаблон Ферзя
3. Шаблон Коня

С целью определения оптимального шаблона проведем анализ каждого из них. Анализ осуществили на пробном растре, со стоимостью прохода по каждой ячейке была равной одной условной единице.

Стоимость веса ребра каждого графа будет определяться в зависимости от положения центральной и целевой ячейки. При этом необходимо принимать во внимание тот факт, что в действительности стоимость прохода каждой ячейки может быть различной.

По причине того, что по обозначенным выше условиям стоимость у всех ячеек равная, то принято решение перехода к одномерной системе координат, с

индексацией ячеек. Это преобразование избавит нас от необходимости проводить поиск по 2 значениям координат.

В итоге, будущие вершины графа получили название на основании индекса ячейки раstra. Следующим шагом будет являться исчисление веса ребер графа.

Вес ребра графа будет изменяться в зависимости от выбранного шаблона.

Шаблон ладьи

Шаблон ладьи является самым простым, максимально возможное количество «соседей» вершины будет равняться четырем, соответственно, ребер будет тоже четыре. Для упрощения, каждое ребро графа можно назвать в соответствии с вершиной, к которой он ведет. Образец представлен на рис. 1.1.

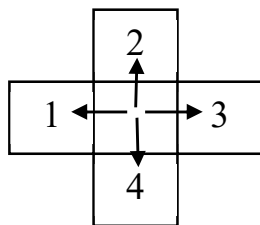


Рис. 1.1. Шаблон Ладьи

Вес каждого ребра при использовании шаблона ладьи в общем случае будет вычисляться по формуле:

$$E_{ij} = \frac{v_i + v_j}{2} (1)$$

Где i = от 1 до n , n – конечное число вершин, j = от 1 до 4 – соответствует количеству ребер.

Шаблон ферзя

В шаблоне ферзя некоторые ребра, а именно 2, 4, 6 и 8, будут считаться по той же формуле, что и в шаблоне ладьи. Образец шаблона изображен на рис. 1.2.

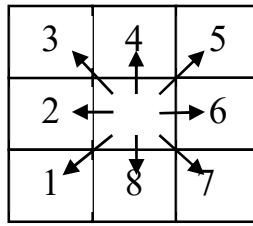


Рис. 1.2. Шабло́н Ферзя

Ребра 1, 3, 5 и 7 будут считаться как по формуле:

$$E_{ij} = \frac{\sqrt{2}}{2} (v_i + v_j)(2)$$

Где v_i – вес центральной ячейки, а v_{ij} – вес соседа.

Шабло́н коня

Пример шабло́на коня изображен на рис. 1.3.

Вычисление ребер в шаблоне проходят аналогично вычислению ребер в предыдущих шабло́нах, веса добавленных ребер (1, 3, 5, 7, 9, 11, 13 и 15) будут вычисляться по формуле:

$$E_3 = \sqrt{\left(\frac{a+a}{2} + a\right)^2 + \left(\frac{a+a}{2}\right)^2} = a\sqrt{5} \quad (3)$$

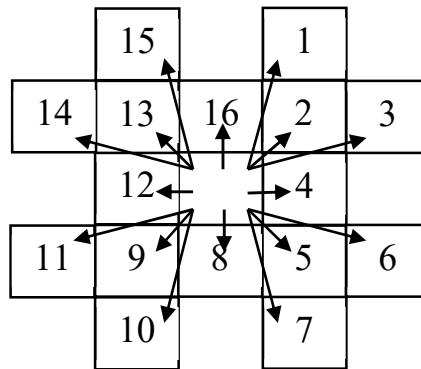


Рис. 1.3. Шабло́н коня

Так как вершиной каждого графа будет считаться центр раstra, то в ситуации, когда стоимость каждой ячейки одинакова, расчет будет сводиться к поиску расстояния от центра одной ячейки до центра другой.

В реальности формула должна учитывать вес всех ячеек, через которые проходит

$$E_3 = \sqrt{\left(\frac{a}{2} + b + \frac{c}{2}\right)^2 + \left(\frac{c+d}{2}\right)^2} \quad (4)$$

1.3.2. Построение оптимального маршрута с использованием разных шаблонов

Рассмотрим несколько частных случаев поиска оптимального пути предположив, что стоимость прохода между всеми ячейками условно равна единице:

1) Шаблон Ладьи.

Стоимость веса маршрута от начальной точки до конечной в примере «а» на рис. 1.4. составляет 10 условных единиц, а в примере «б» 8 условных единиц.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

а

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

б

Рис. 1.4. Поиск оптимального маршрута по шаблону Ладьи.

2) Шаблон Ферзя.

Стоимость веса маршрута от начальной точки до конечной в примере «а» (рис. 1.5) составляет $5\sqrt{2}$ условных единиц, в примере «б» $3\sqrt{2} + 2$ условных единиц.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Рис. 1.5. Поиск оптимального маршрута по шаблону Ферзя.

3) Шаблон Коня.

Стоимость веса маршрута от начальной точки до конечной в примере, «а» (рис. 1.6) составляет $5\sqrt{2}$ условных единиц, в примере «б» $2\sqrt{5} + \sqrt{2}$ условных единиц.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Рис. 1.6. Поиск оптимального маршрута по шаблону Коня.

Как можно заметить, шаблон ладьи является самым простым для построения, но его использование не есть самое универсальное. Применение шаблона коня наиболее универсально, но в тоже время построение графа по этому шаблону требует большего количества ресурсов. Исходя из вышесказанного, было принято решение использовать шаблон ферзя.

1.3.3. Анализ сложности алгоритмов

Сложность алгоритма - это способ измерить, насколько быстро алгоритм или программа работают. Анализ сложности позволяет объяснить, как будет вести себя алгоритм при возрастании входного потока данных.

Существуют несколько методов анализа сложности алгоритмов во времени, самые распространенные из них:

- Амортизационный анализ (Amortized Analysis);
- Анализ параллельных алгоритмов (Analysis of parallel algorithms);
- Асимптотический анализ сложности алгоритмов (Asymptotic computational complexity).

Amortized Analysis (амортизационный анализ) - метод анализа алгоритмов, позволяющий осуществлять оценку времени выполнения последовательности из n операций над некоторой структурой данных. Время выполнения усредняется по всем n операциям и оценивается среднее время выполнения одной операции в худшем случае. [25]

Analysis of parallel algorithms (анализ параллельных алгоритмов). При параллельном выполнении алгоритма разные части программы выполняются на разных вычислительных устройствах, и часто получается, что результат вычисления в одном устройстве является входом для другого устройства. Время, необходимое для обмена данными между вычислительными устройствами, зависит от топологии коммуникационной подсистемы и может оказаться сопоставимым (или даже превышать) с временем, необходимым для выполнения операций алгоритма. Поэтому для параллельных алгоритмов точный анализ сложности требует учета как количества шагов алгоритма (с учетом возможности параллельного выполнения), так и необходимых пересылок. [12]

Asymptotic computational complexity (асимптотический анализ сложности алгоритмов) – метод анализа алгоритмов, в котором рассматривается поведение алгоритма при $n \rightarrow \infty$ (рис. 2.9). При асимптотическом анализе вычислительной сложности алгоритма оценивают количество операций для следующих случаев:

наихудший случай (worst case) – максимальное количество операций, требуемых для обработки набора входных данных; средний случай (average case) – среднее количество операций, требуемых для обработки набора входных данных; наилучший случай (best case) – минимальное количество операций, требуемых для обработки набора входных данных. Для записи асимптотической сложности алгоритмов используются разные асимптотические обозначения, например, О-нотация (Big O).

Как правило, принято оценивать уровень сложности алгоритма или программы по объему используемых им основных ресурсов: процессорного времени и оперативной памяти. В связи с этим вводятся такие понятия, как временная сложность алгоритма (time complexity) и объемная сложность алгоритма (space complexity).

Параметр временной сложности имеет особое значение для задач, предусматривающих интерактивный режим работы программы, и для задач управления в режиме реального времени. Как правило, увеличение точности вычислений ведет к повышению времени вычислений.

Объемная сложность программы становится критической, когда объем обрабатываемых данных оказывается на пределе объема оперативной памяти ЭВМ. На современных компьютерах эта проблема не имеет такого большого значения.

На время выполнения алгоритма (программы) влияют:

- А) Размер входных данных;
- Б) Качество реализации алгоритма на языке программирования;
- В) Качество скомпилированного кода;
- Г) Производительность вычислительной машины.

У каждого алгоритма имеются параметры, определяющие размер его входных данных:

- А) Поиск наименьшего элемента в массиве: n – количество элементов в массиве;

Б) Алгоритм умножения двух матриц: количества строк m и столбцов n в матрицах;

В) Сравнения двух строк: s_1, s_2 - длина первой и второй строк;

Г) Поиск кратчайшего пути в графе между парой вершин: n, m - количество вершин и ребер в графе.

Количество операций алгоритма можно выразить как функцию от размера его входных данных: $T(n), T(s_1, s_2), T(n, m)$.

В качестве исполнителя используется модель однопроцессорной вычислительной машины с произвольным доступом к памяти (Random Access Machine—RAM):

- Машина обладает неограниченной памятью;
- Для выполнения арифметических и логических операций (+, −, *, /, %) требуется один временной шаг - такт процессора;
- Обращение к оперативной памяти для чтения или записи занимает один временной шаг;
- Выполнение условного перехода (if-then-else) требует вычисления логического выражения и выполнения одной из ветвей if-then-else;
- Выполнение цикла (for, while, do) подразумевает выполнение всех его итераций.

Существует три возможных случая:

- Лучший случай (best case);
- Худший случай (worst case);
- Средний случай (average case).

Лучший случай (best case) —это экземпляр задачи (набор данных), на котором алгоритм проводит наименьшее число операций.

$$T_{\text{лучш.}}(n) = 1$$

Худший случай (worst case) —это экземпляр задачи, на котором алгоритм выполняет наибольшее число операций.

$$T_{\text{худш.}}(n) = 1$$

Средний случай (average case) – это “средний” экземпляр задачи, набор “усреднённых” данных.

В среднем случае оценивается математическое ожидание количества операций, выполняемых алгоритмом. Не всегда возможно четко определить, какие входные данные можно считать «усреднёнными» для целей задачи. Пример анализа алгоритмов приведен на рис. 1.7.

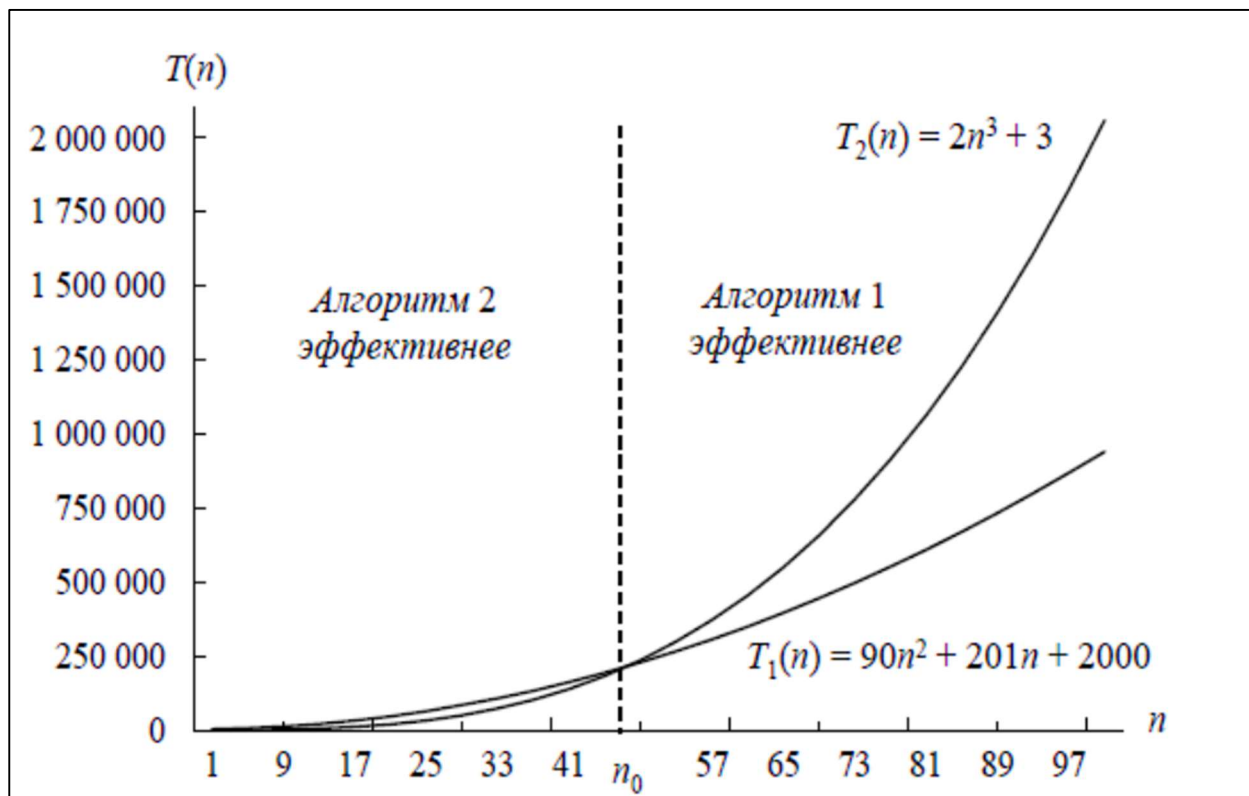


Рис. 1.7. Пример анализа алгоритмов.

Если известно, что на вход будут поступать данные небольших размеров, то вопрос о выборе эффективного алгоритма является первостепенным - можно использовать самый “простой” алгоритм. Вопросы, связанные с эффективностью алгоритмов, приобретают смысл при больших размерах входных данных – при $n \rightarrow \infty$.

Асимптотический анализ (asymptotic analysis), который позволяет оценивать скорость роста функций $T(n)$ при стремлении размера входных данных к бесконечности (при $n \rightarrow \infty$).

Мы можем найти такое значение n_0 при котором происходит пересечение функций $T_1(n)$ и $T_2(n)$, и на основе n_0 отдавать предпочтение тому или иному алгоритму.

Как правило, функция времени $T(n)$ выполнения алгоритма имеет большое количество локальных экстремумов – неровный график с выпуклостями и впадинами.

Проще работать с верхней и нижней оценками (границами) времени выполнения алгоритма (рис. 1.8)

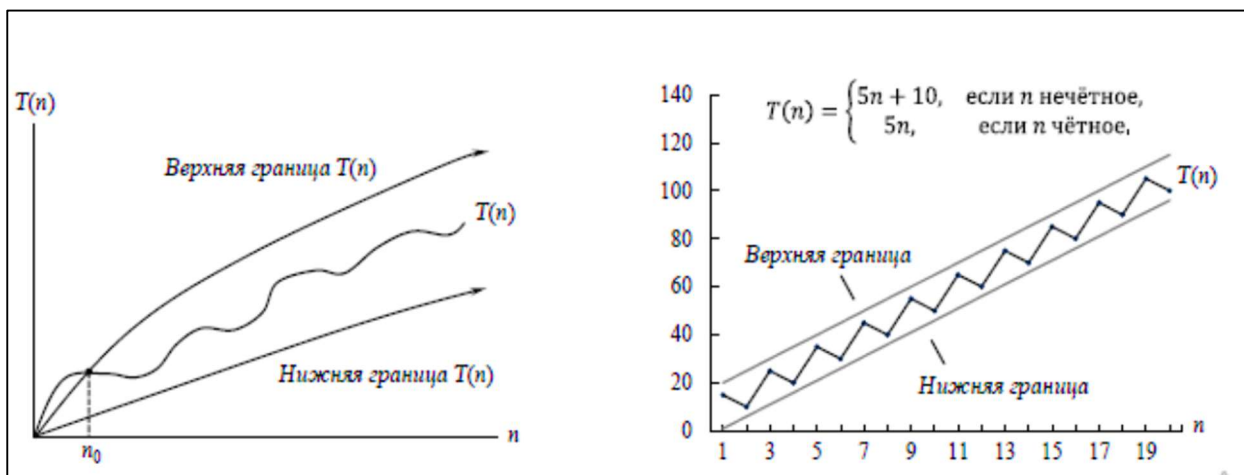


Рис. 1.8. - Верхняя и нижняя оценка времени выполнения алгоритма

В теории вычислительной сложности алгоритмов (computational complexity theory) для указания границ функций $T(n)$ используют асимптотические обозначения: O (о большое), Ω (омега большое), Θ (тета большое), а также o (о малое), ω (омега малое).

Далее будем считать, что областью определения функции $f(n)$ и $g(n)$, которые выражают число операций алгоритма, является множество неотрицательных целых чисел: $n \in \{0, 1, 2, \dots\}$

Функции $f(n)$ и $g(n)$ являются асимптотически неотрицательными – при больших значениях n они принимают значения большие или равные нулю.

O-обозначение (о большое)

Используется чтобы показать, что время работы алгоритма растёт не быстрее чем функция $g(n)$. Асимптотическая верхняя граница (asymptotic upperbound) для функции $f(n)$ показана на рис.1.9.

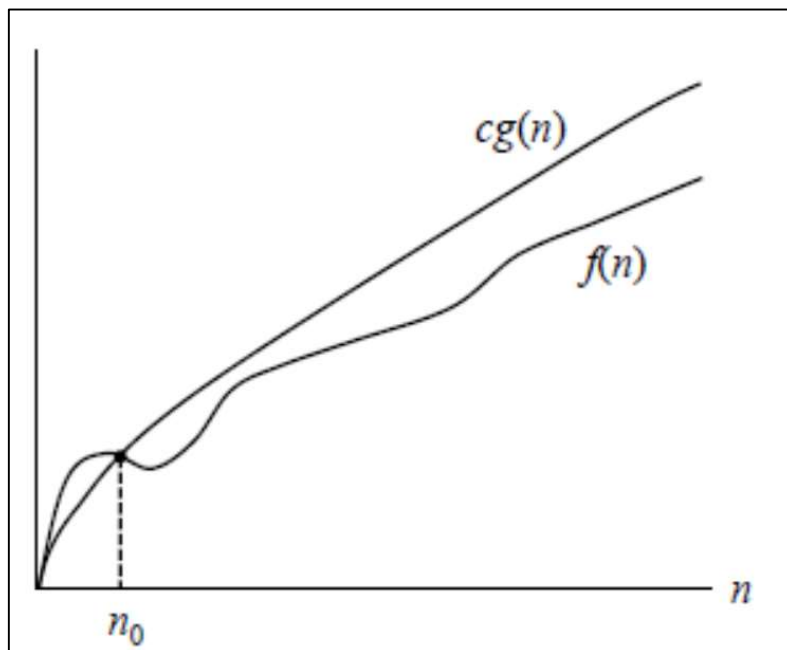


Рис. 1.9. Оценка O-обозначение

Ω-обозначение (омега большое)

Используется чтобы показать, что функция (время работы алгоритма) растет не медленнее чем функция $g(n)$. Асимптотическая нижняя граница (asymptotic lower bound) для функции $f(n)$ показана на рис.1.10.

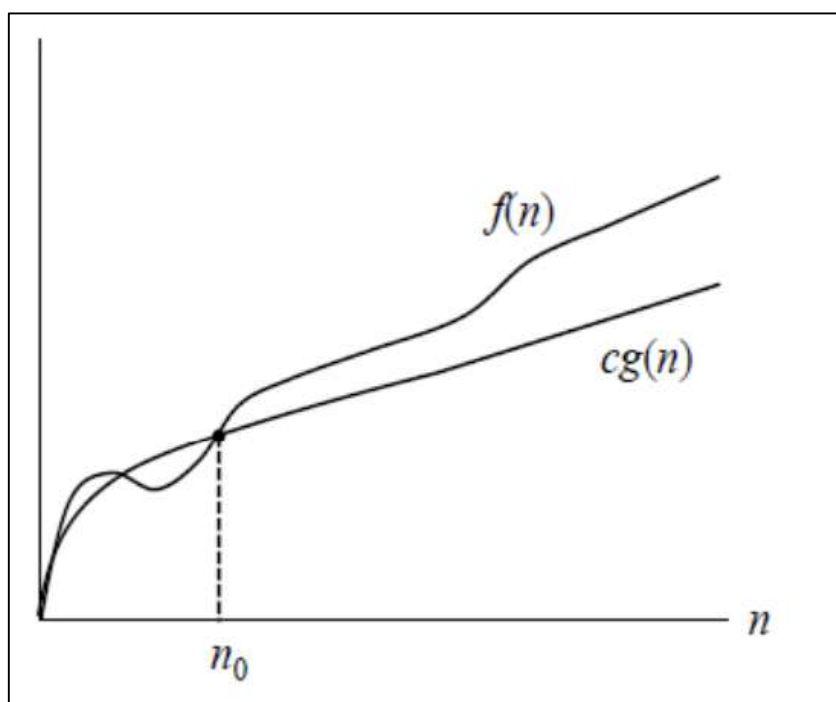


Рис. 1.10. Оценка Ω-обозначение

Θ-обозначения (тета большое)

Функция $f(n)$ ограничена снизу и сверху функцией $g(n)$ с точностью до постоянного множителя (рис.1.11).

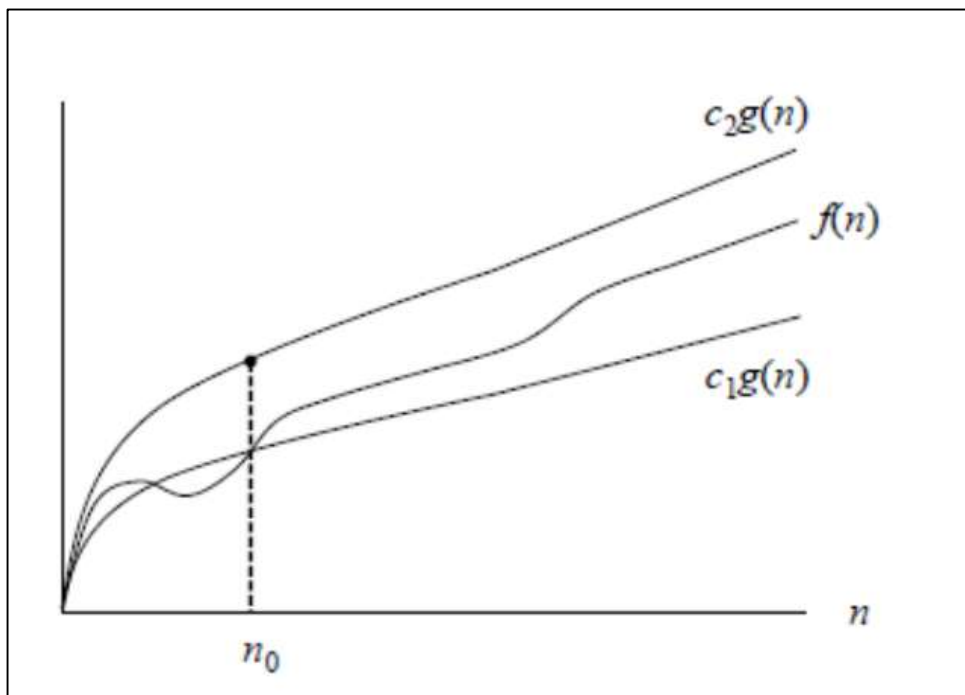


Рис 1.11. Оценка Θ -обозначения

Классификация алгоритмов по трудоемкости в порядке возрастания представлена в таблице 2.

Таблица 2.

Классификация алгоритмов по трудоемкости

Сложность	Нотация	Описание
Постоянная	$O(1)$	Постоянное количество операций, не зависящее от количества входящих данных
Логарифмическая	$O(\log n)$	Количество операций пропорционально $\log_2 n$
Линейная	$O(n)$	Количество операций пропорционально количеству входящих данных
Квадратичная	$O(n^2)$	Количество операций пропорционально квадрату входящих данных
Кубическая	$O(n^3)$	Количество операций пропорционально кубу входящих данных
Экспоненциальная	$O(2^n)$, $O(k^n)$, $O(n!)$	Количество операций очень быстро возрастает

1.3.4. Алгоритмы поиска оптимального пути

На сегодняшний день можно выделить следующие алгоритмы для построения оптимального маршрута линейных объектов:

- Алгоритм поиска A^* ;
- Алгоритм Дейкстры;
- Поиск в ширину (BFS, Breadth-First Search);
- Алгоритм поиска с итеративным углублением IDA* (Iterative-Deepening A^*);
- Jump Point Search;
- Поиск по первому наилучшему совпадению (Best-First Search).

1) Алгоритм Дейкстры

Алгоритм Дейкстры подразумевает под собой n итераций, для каждой из которых избирается непомеченная вершина с минимальной величиной $d[v]$, данная вершина помечается, и после рассматриваются все рёбра, исходящие из этой вершины, и вдоль всех ребер делается попытка максимизировать значение $d[v]$ на другом конце ребра.

Период работы алгоритма будет складываться из последующих действий:

- n раз поиск вершины с наименьшей величиной $d[v]$ среди всех непомеченных вершин, т.е. среди $O(n)$ вершин;
- m раз производится попытка релаксаций.

Во время простейшей реализации данных операций на поиск вершины будет тратиться $O(n)$ операций, а на одну релаксацию - $O(1)$ операций, и конечная асимптотика алгоритма составит: $O(n^2+m)$.

В ориентировочном взвешенном графе $G = (V, E)$, вес ребер которого нулевой или положительный и определяется весовой функцией $w: E \rightarrow R$, данный алгоритм находит протяженности кратчайших путей из обозначенной вершины s до всех прочих.

Алгоритм Дейкстры поддерживает множество вершин « U », с вычисленными длинами кратчайших путей до них из « s ». При каждой итерации

главного цикла выбирается вершина « u », которой на данный момент цикла соответствует наименьшая оценка кратчайшего маршрута. Вершина « u » включается во множество « U » и происходит релаксация исходящих из неё рёбер.

Псевдокод алгоритма Дейкстры:

```
func dijkstra(s):
    for v in V
        d[v] = ∞
        used[v] = false
    d[s] = 0
    for i in V
        v = null
        for j in V // найдем вершину с минимальным расстоянием
            if !used[j] and (v == null or d[j] < d[v])
                v = j
        if d[v] == ∞
            break
        used[v] = true
        for e : исходящие из v рёбра // произведём релаксацию по всем рёбрам,
            исходящим из v
            if d[v] + e.len < d[e.to]
                d[e.to] = d[v] + e.len
```

2) Модифицированный метод Дейкстры

- n – количество узлов сети распределения;
- $D(k)$ – вес маршрута от корневого узла 1 до k ;
- $l(m,k)$ – вес ребра меж m и k узлами;
- N – множество, элементы которого - номера узлов, добавленные на каждой ступени алгоритма по оптимизации основываясь на вычислении маршрутов с наименьшим весом.

Модификация алгоритма заключается в редактировании целевой функции и выражений с целью вычисления обновляемых весов маршрутов. Сравнение модифицированного алгоритма и алгоритма Дейкстры представлено в таблице 3.

Сравнение алгоритма Дейкстры и модифицированного алгоритма
Дейкстры

Модифицированный алгоритм	Алгоритм Дейкстры
Целевая функция	
Определяется суммарным весом ребер всех маршрутов	Определяется весом каждого из маршрутов, следующих из корневого узла
Критерий оптимальности	
Минимум суммарного веса ребер	Минимум веса одного пути
Обновление весов путей	
$D_i(k) = D_{i-1}(m_k) + l(m_k, k)$, $m \in N, k \notin N$, где $m_k = \operatorname{argmin}[l(m, k)]$ - номер узла сети, при котором функция $l(m, k)$ принимает минимальное значение	$D_i(k) = \min[D_{i-1}(k), D_{i-1}(m) + l(m, k)]$, где $D_i(k)$ – вес пути из корневого узла в k узел на i шаге процедуры оптимизации

3) Алгоритм поиска A^*

Алгоритм A^* ("A star") — представляет собой алгоритм поиска, находящий в взвешенном графе путь минимальной стоимости от первоначальной вершины до конечной. Данный алгоритм впервые описан в 1968 году Нильсом Нильсоном, Питером Хартом и Бертрамом Рафаэлем.

В процессе выполнения алгоритма для вершин рассчитывается функция $f(v) = g(v) + h(v)$, где

$g(v)$ — минимальная стоимость маршрута в v из начальной вершины;

$h(v)$ — приближение стоимости маршрута от v до конечной цели.

$h(v)$ должна быть допустимой, например, для целей маршрутизации $h(v)$ должна представлять расстояние до целевой точки по прямой линии, потому что это физически минимально возможная протяженность между двумя точками.

При меньшем h , вершина будет раньше открыта и исследована. Следовательно, открытые алгоритмом вершины содержатся в очереди с приоритетом по значению $f(v)$. Алгоритм A^* подобен алгоритму Дейкстры и

просматривает среди всех остальных возможных путей, которые ведут к цели, сначала те, которые по имеющейся информации в данный момент времени являются наилучшими.

На всех этапах работы алгоритма A^* из множества Q выбирается вершина с минимальным значением функции и просматриваются её соседние вершины. Для каждой из соседних вершин обновляется расстояние, значение функции и добавляется в множество.

Псевдокод:

```
bool A*(start, goal):
    U = ∅
    Q = ∅
    Q.push(start)
    g[start] = 0
    f[start] = g[start] + h(start)
    while Q.size() != 0
        current = вершина из Q с минимальным значением  $f$ 
        if current == goal
            return true // нашли путь до нужной вершины
        Q.remove(current)
        U.push(current)
        for v : смежные с current вершины
            tentativeScore = g[current] + d(current, v) // d(current, v) —
стоимость пути между current и v
            if v ∉ U and tentativeScore >= g[v]
                continue
            if v ∉ U or tentativeScore < g[v]
                parent[v] = current
                g[v] = tentativeScore
                f[v] = g[v] + h(v)
                if v ∉ Q
                    Q.push(v)
    return false
```

В приведённой реализации алгоритма в псевдокоде:

- Q — множество рассматриваемых вершин;
- U — множество вершин, которые уже рассмотрены;
- $f[x]$ — функция ("расстояние + стоимость") для вершины x ;
- $g[x]$ — цена пути от первоначальной вершины до x ;
- $h(x)$ — расстояние от вершины x до конечной вершины.

4) Алгоритм поиска с итеративным углублением (IDA*)

Алгоритм поиска с итеративным углублением – усовершенствованная версия алгоритма A*, которая использует меньшее количество памяти за счет сокращения развертываемых узлов. Результат выполнения - кратчайший маршрут.

Главное отличие между алгоритмом IDA* и нормальным алгоритмом итеративного углубления заключается в том, что используемым условием для остановки развертывания является f-стоимость (g+h), а не глубина; для всякой итерации этим остановочным значением является наименьшая f-стоимость любого узла, превышающая остановочное значение, достигнутое в предыдущей итерации. Алгоритм IDA* является практически применимым для решения многих задач с единичными стоимостями этапов и позволяет избежать существенных издержек, связанных с поддержкой отсортированной очереди узлов.

Псевдокод:

```
procedure ida_star(root, cost(), is_goal(), h())
  bound := h(root)
  loop
    t := search(root, 0, bound)
    if t = FOUND then return FOUND
    if t = ∞ then return NOT_FOUND
    bound := t
  end loop
end procedure
function search(node, g, bound)
  f := g + h(node)
  if f > bound then return f
  if is_goal(node) then return FOUND
  min := ∞
  for succ in successors(node) do
    t := search(succ, g + cost(node, succ), bound)
    if t = FOUND then return FOUND
    if t < min then min := t
  end for
  return min
end function
```

node – текущий узел

g – стоимость начала решения $root..node$

f – оценка стоимости минимального пути через $node$

$h(node)$ – оценка стоимости остатка пути $node..goal$

$cost(node, succ)$ – функция стоимости пути

$is_goal(node)$ – функция проверки цели

$successors(node)$ – функция развёртывания узла $node$

5) Поиск в ширину (BFS, Breadth-First Search)

Поиск в ширину (англ. breadth-first search, BFS) — метод обхода графа и поиска пути в графе. Поиск в ширину является одним из неинформированных алгоритмов поиска.

Алгоритм был разработан независимо Муром и Ли для разных приложений (поиск пути в лабиринте и разводка проводников соответственно) в 1959 и 1961 годах. Этот алгоритм можно сравнить с поджиганием соседних вершин графа: сначала мы зажигаем одну вершину (ту, из которой начинаем путь), а затем, огонь за один элементарный промежуток времени перекидывается на все соседние с ней не горящие вершины. В последствии то же происходит со всеми подожженными вершинами. Таким образом, огонь распространяется "в ширину". В результате его работы будет найден кратчайший путь до нужной клетки.

Поиск в ширину работает путём последовательного просмотра отдельных уровней графа, начиная с узла-источника u .

Рассмотрим все рёбра (u, v) , выходящие из узла u . Если очередной узел v является целевым узлом, то поиск завершается; в противном случае узел v добавляется в очередь. После того, как будут проверены все рёбра, выходящие из узла u , из очереди извлекается следующий узел u , и процесс повторяется.

Ниже приведён псевдокод алгоритма для случая, когда необходимо лишь найти целевой узел. В зависимости от конкретного применения алгоритма, может потребоваться дополнительный код, обеспечивающий сохранение нужной информации (расстояние от начального узла, узел-родитель и т. п.)

Рекурсивная формулировка:

```

BFS(start_node, goal_node) {
    return BFS'({start_node}, ∅, goal_node);
}
BFS'(fringe, visited, goal_node) {
    if(fringe == ∅) { // Целевой узел не найден
        return false;
    }
    if (goal_node ∈ fringe) {
        return true;
    }
    return BFS'({child | x ∈ fringe, child ∈ expand(x)} \ visited, visited ∪ fringe,
goal_node);
}

```

Итеративная формулировка:

```

BFS(start_node, goal_node) {
    for(all nodes i) visited[i] = false; // изначально список посещённых узлов пуст
    queue.push(start_node); // начиная с узла-источника
    visited[start_node] = true;
    while(! queue.empty() ) { // пока очередь не пуста
        node = queue.pop(); // извлечь первый элемент в очереди
        if(node == goal_node) {
            return true; // проверить, не является ли текущий узел целевым
        }
        foreach(child in expand(node)) { // все преемники текущего узла, ...
            if(visited[child] == false) { // ... которые ещё не были посещены
                queue.push(child); // ... добавить в конец очереди...
                visited[child] = true; // ... и пометить как посещённые
            }
        }
    }
    return false; // Целевой узел недостижим
}

```

б) Алгоритм поиска пути Jump Point Search

Jump Point Search - это алгоритм поиска пути, представленный в 2011 году Дэниэлом Харбором в серии статей «Fast Pathfinding via Symmetry Breaking», «Rectangular Symmetry Reduction», «Jump Point Search». Алгоритм является улучшением алгоритма A*. JPS ускоряет поиск пути, “перепрыгивая” многие места, которые должны быть просмотрены. В отличие от подобных алгоритмов JPS не требует предварительной обработки и дополнительных затрат памяти.

Алгоритм работает на гридах, у которых стоимость прохода везде одинакова, но существуют «препятствия» - ячейки с бесконечно большой

стоимостью. Каждое поле карты имеет ≤ 8 соседей, которые могут быть проходимы или же нет. Каждый шаг по направлению (по вертикали или по горизонтали) имеет стоимость 1; шаг по диагонали имеет стоимость $\sqrt{2}$. Движения через препятствия запрещены. “Прыжковые точки” позволяют ускорить алгоритм поиска пути, рассматривая только “необходимые” точки. Такие точки могут быть описаны двумя простыми правилами выбора соседей при рекурсивном поиске: одно правило для прямолинейного движения и другое – для диагонального. В обоих случаях необходимо доказать, что, после исключения полей из набора ближайших соседей точки, всегда найдётся оптимальный путь из предка текущей точки до каждого из соседей, и этот путь не будет содержать в себе посещенную точку.

1.3.5. Что используется в ArcGis

Механизмы расчета задач выбора маршрута в дополнительном модуле. Дополнительный модуль ArcGIS Network Analyst - Маршрут (Route), Ближайший пункт обслуживания (Closest Facility) и Матрица Источник - Назначение (OD Cost Matrix) основаны на известном алгоритме поиска кратчайших путей Дейкстры. Каждый из этих трех механизмов расчета реализует два типа алгоритма поиска пути. Первый тип – поиск точного кратчайшего пути, а второй – иерархический поиск пути для повышения быстродействия. Классический алгоритм Дейкстры решает задачу поиска кратчайшего пути при помощи неориентированного, неотрицательного, взвешенного графа. Для использования в контексте реальных данных о транспортных перевозках этот алгоритм модифицируется с учетом пользовательских параметров, таких как ограничения, связанные с односторонним движением и поворотами, импеданс узлов, барьеры и ограничения для стороны улицы, при этом минимизируется заданный пользователем стоимостный атрибут. Быстродействие алгоритма Дейкстры улучшается при использовании более эффективных структур данных, таких как

d-кучи. Кроме того, алгоритм должен иметь возможность моделировать местоположения в любой точке ребра, а не только на узлах пересечения.

Классический алгоритм Дейкстры решает задачу поиска кратчайшего пути с одной исходной точкой на взвешенном графе. Для поиска кратчайшего пути из начального местоположения s до конечного местоположения d алгоритм Дейкстры использует набор узлов S , окончательный кратчайший путь от которых до s уже вычислен. Алгоритм многократно выполняет в наборе поиск узла с минимальной оценкой кратчайшего пути, добавляет его в набор узлов S и обновляет оценки кратчайшего пути всех соседей этого узла, которые не входят в S . Алгоритм повторяется, пока узел назначения не будет добавлен в набор S .

В ArcGis используется модифицированный алгоритм Дейкстры[41] для двух направлений с целью вычисления маршрута между исходной и конечной точками, а также применяются иерархии по предпочтительности путей. Поскольку поиск ограничен иерархией верхнего уровня, он затрагивает меньшее количество ребер, что повышает быстродействие. Следует отметить, что это эвристический алгоритм; его цель – высокое быстродействие и хорошие решения, но он не гарантирует, что будет найден кратчайший путь. Для успешной работы эвристического алгоритма необходимо подключить иерархию высшего уровня, поскольку он не спускается на уровень ниже при достижении тупикового конца.

1.4. Анализ и выбор алгоритма для реализации

Рассмотрим и проанализируем ключевые факторы, представленные в таблице 4 с целью выбора наиболее подходящего для реализации алгоритма.

Сравнение алгоритмов

Алгоритм	Стоимость прохода через ячейки	Худшее время	Наибольший объем памяти
Алгоритм Дейкстры (Dijkstra)	любая	$O(n^2)$	-
Поиск в ширину (BFS, Breadth-First Search)	1 или ∞	$O(bd+1)$	$O(bd)$
A*	любая	$O(bd+1)$, зависит от метрики	$O(bd)$
IDA* (Iterative Deeping A*, A* с итеративным углублением)	1 или ∞	$O(bd)$	$O(bd)$
Jump Point Search	1 или ∞	-	-

Стоимость прохода через ячейки определяет пригодность данного алгоритма для задачи поиска оптимального пути на карте. Некоторые алгоритмы разрабатывались для поиска пути в лабиринте, поэтому стоимость прохода через ячейки в них может принимать значение 1 или ∞ , как в алгоритме «Поиск в ширину», а подобный алгоритм не подходит для решения поставленной задачи, в силу узкой направленности.

Проведённый анализ выявил, что два алгоритма из рассматриваемых возможно использовать: алгоритмы Дейкстры и A*. Из данных алгоритмов самые лучшие параметры по времени показал A*, поэтому принято решение дальнейшие исследования посвятить оптимизации алгоритма A*.

1.5. Выбор структуры данных

Существует достаточно большое количество разнообразных структур данных, наиболее распространенные из них проиллюстрированы на рис. 1.12.

Классические структуры данных

Линейные структуры	Нелинейные структуры
<ul style="list-style-type: none"> • Массив • Стэк • Связанный список 	<ul style="list-style-type: none"> • Дерево • Хэш-таблица • Графы

Рис. 1.12. Классические структуры данных

Каждая из предложенных структур используется для своих целей, в таблице 5 продемонстрированы результаты временной сложности выполнения операций: поиска, доступа, удаления и вставки в среднем и в худшем случае.

Таблица 5

Время выполнения стандартных операций

Структура данных	Время выполнения							
	В среднем				В худшем случае			
	Доступ	Поиск	Вставка	Удаление	Доступ	Поиск	Вставка	Удаление
Массив	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
Стэк	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$
Связанный список	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$
Бинарное дерево	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Хэш-таблица	-	$O(1)$	$O(1)$	$O(1)$	-	$O(n)$	$O(n)$	$O(n)$

В таблице 6 представлены результаты анализа структур данных по объемной сложности.

Анализ структуры данных по объемной сложности

Структура данных	Объем сложности
Массив	$O(n)$
Стек	$O(n)$
Связанный список	$O(n)$
Бинарное дерево	$O(n)$
Хэш-таблица	$O(n)$

По результатам анализа структур данных было принято решение использовать хэш-таблицу как структуру данных, требующую минимального времени для выполнения обычных операций.

1.5.1. Хэш-таблица

Хэш-таблица представляет собой обобщение обычного массива. Однако в то время как ключом массива может быть только число, для хэш-таблицы им может быть любой объект, для которого можно вычислить хэш-код. Интерфейс хэш-таблиц предоставляет следующие операции:

- Добавление новой пары ключ-значение;
- Поиск значения по ключу;
- Удаление пары ключ-значение по ключу.

Среднее время поиска значения по ключу в хэш-таблице равно $O(1)$. Это значит, что в среднем, поиск не будет зависеть от количества элементов в хэш-таблице, и будет равен некоторому константному значению. В зависимости от самой внутренней реализации хэш-таблицы, время поиска для наихудшего случая может быть $O(n)$, то есть линейно зависеть от количества элементов в таблице, либо же оставаться $O(1)$.

Идея хеширования основана на распределении ключей в обычном массиве $N[0..m-1]$. Распределение осуществляется вычислением для каждого ключа элемента некоторой хэш-функции h . Эта функция на основе ключа вычисляет целое число n , которое служит индексом для массива N . Сложность

заключается в том, что необходимо придумать такую хеш-функцию, которая бы давала различный хеш-код для различных объектов. На рис. 1.13 изображена хеш-таблица, индексами ключей в ней является результат хеш-функции h , применённой к ключу.

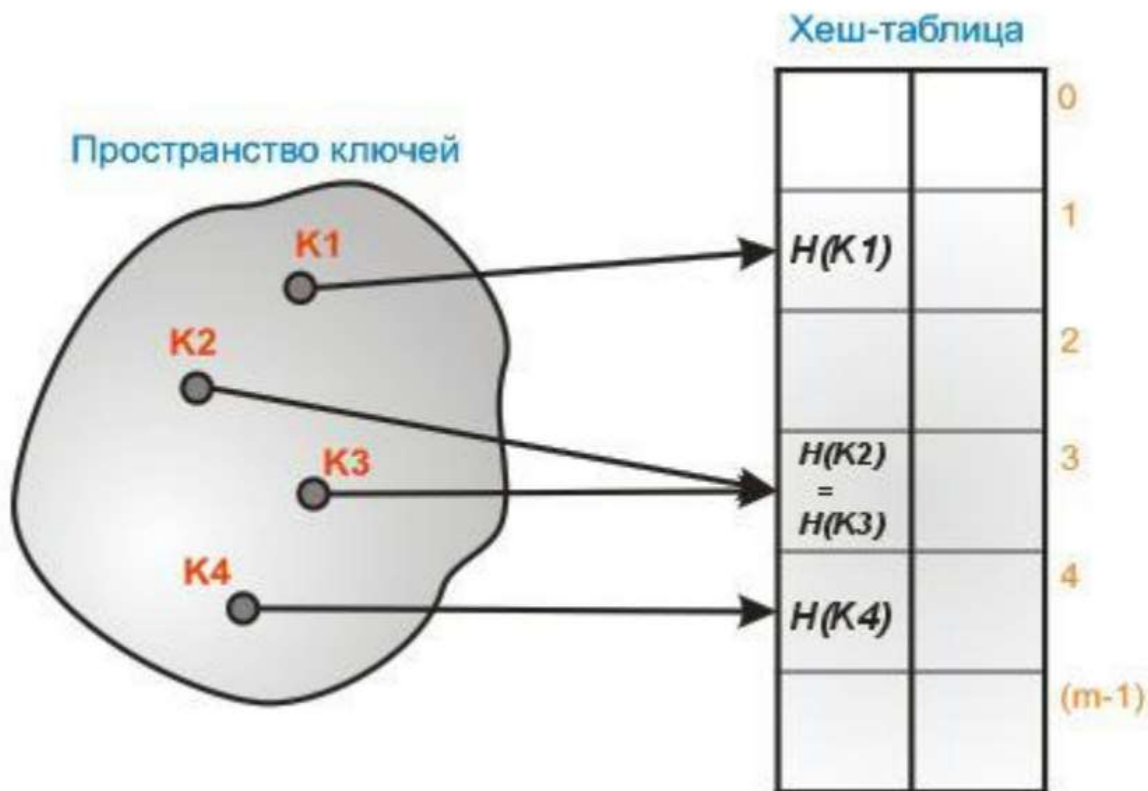


Рис. 1.13. Хеш-таблица

Так же данный рисунок иллюстрирует одну из базовых проблем. При достаточно низком значении m (размер хеш-таблицы) по отношению к n (количество ключей) или при некачественной хеш-функции, может случиться так, что оба ключа будут помещены в одну и ту же ячейку массива H . Такую ситуацию называются коллизией.

При выборе хеш-функции необходимо минимизировать вероятность коллизий, однако, учитывая то, что пространство всех возможных ключей может быть больше размера нашей хеш-таблицы H , всё же избежать их вряд ли удастся. На этот случай имеются несколько технологий для разрешения коллизий. Основные из них представлены далее.

1.5.2. Хеширование с цепочками

В случае открытого хеширования (другое название хеширования цепочками), происходит объединение элементов, хешированных в одну и ту же ячейку, в связный список (рис. 1.14).

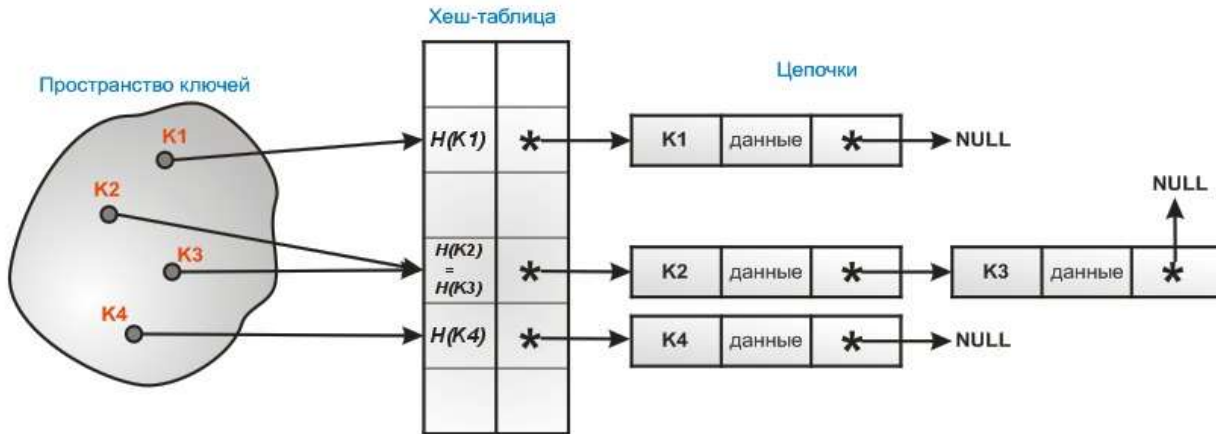


Рис. 1.14. Пример открытого хеширования

Так, идея достаточно проста. Если при добавлении в хеш-таблицу в заданную ячейку встречается ссылка на элемент связного списка, то случается коллизия. Поэтому можно просто вставить новый элемент как узел в список. При поиске программа проходит по цепочкам, сравнивая ключи между собой на эквивалентность, пока не доберётся до нужного. При удалении ситуация такая же.

Процедура вставки выполняется даже в наихудшем случае за $O(1)$, при отсутствии вставляемого элемента в таблице. Время поиска зависит от длины списка, и в худшем случае равно $O(n)$. Эта ситуация, когда все элементы хешируются в единственную ячейку. Если функция распределяет n ключей по m ячейкам таблицы равномерно, то в каждом списке будет содержаться порядка n/m ключей. Это число называется коэффициентом заполнения хеш-таблицы. Математический анализ хеширования с цепочками показывает, что в среднем случае все операции в такой хеш-таблице в среднем выполняются за время $O(1)$.

1.5.3. Хеширование с открытой адресацией

В случае метода открытой адресации (или по-другому: закрытого хеширования) все элементы хранятся непосредственно в хеш-таблице, без использования связанных списков. В отличие от хеширования с цепочками, при использовании метода открытой адресации, может возникнуть ситуация, когда хеш-таблица окажется полностью заполненной, так что будет невозможно добавлять в неё новые элементы. При возникновении такой ситуации решением может быть динамическое увеличение размера хеш-таблицы с одновременной её перестройкой.

Для разрешения коллизий применяются несколько подходов. Самый простой из них – метод линейного исследования. В этом случае при возникновении коллизии следующие за текущей ячейки проверяются одна за другой, пока не найдётся пустая ячейка, куда и помещается элемент. Так, при достижении последнего индекса таблицы, программа возвращается в ее начало, рассматривая таблицу как «циклический» массив. Иллюстрация этого способа представлена на рисунке 1.15:

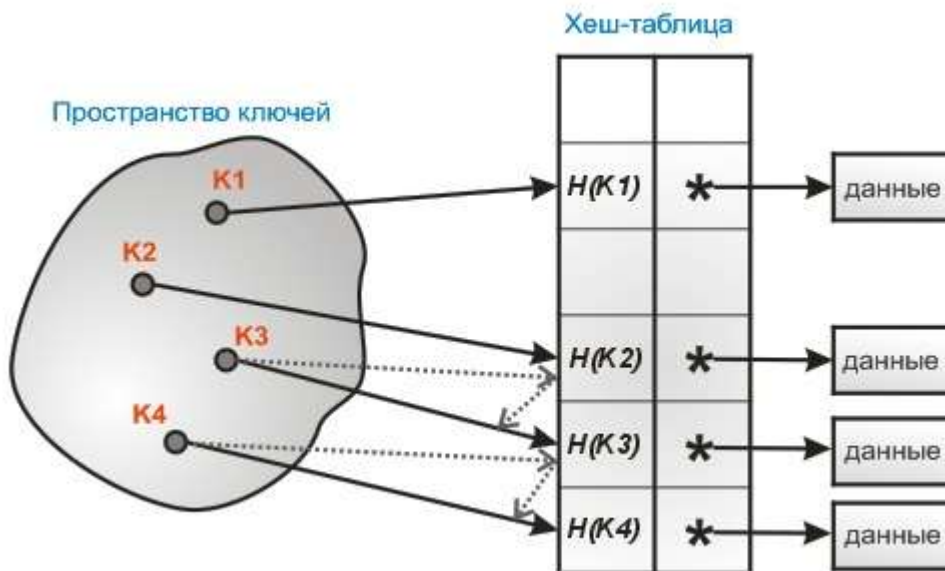


Рис. 1.15. Пример разрешения коллизий

Линейное хеширование достаточно просто реализуется, однако с ним связана существенная проблема – кластеризация. Это явление создания длинных

последовательностей занятых ячеек, которое увеличивает среднее время поиска в таблице. Для снижения эффекта кластеризации используется другая стратегия разрешения коллизий – двойное хеширование. Основная идея заключается в том, что для определения шага смещения исследований при коллизии в ячейке используется другая хеш-функция, вместо линейного смещения на одну позицию.

Одной из сложных вопросов реализации хеширования с открытой адресацией – это операция удаления элемента. При удалении некоего элемента из хеш-таблицы, становится невозможным поиск ключа, в процессе вставки которого текущая ячейка оказалась заполненной. Этого можно избежать, если пометить очищенные ячейки, чтобы впоследствии это учитывать.

2. АНАЛИЗ ВХОДНЫХ ДАННЫХ И ВАРИАНТОВ ОПТИМИЗАЦИИ АЛГОРИТМА А*

2.1. Краткий анализ деятельности предприятия

Исследование проводилось и апробировалось на базе строительных предприятий Иркутской области. Рассмотрим Общество с ограниченной ответственностью «ХХХ», которое осуществляет свою деятельность с 2013 года и управляется командой специалистов, имеющих опыт работы в промышленном строительстве более 15 лет.

Руководство ООО «ХХХ» действует на основании устава. Организация имеет допуски СРО к работе на объектах нефтехимической отрасли, осуществлению электромонтажных работ, устройству магистральных трубопроводов и т.д. Также все сотрудники компании имеют аттестацию НАКС.

Компания осуществляет работы в Иркутской области. В частности, основные объекты расположены:

- г. Братск (Братский завод ферросплавов);
- Ангарск (Ангарский нефтехимический комбинат, Ангарский завод полимеров).

Профилирующие направления деятельности согласно свидетельству СРО о допуске к определенным видам работ:

- Геодезические работы, выполняемые на строительных площадках.
- Подготовительные работы.
- Земляные работы.
- Устройство бетонных и железобетонных монолитных конструкций.
- Монтаж сборных бетонных и железобетонных конструкций.
- Работы по устройству каменных конструкций.
- Монтаж металлических конструкций.
- Защита строительных конструкций, трубопроводов и оборудования (кроме магистральных и промысловых трубопроводов).

- Устройство кровель.
- Фасадные работы.
- Устройство внутренних инженерных систем и оборудования зданий и сооружений.
- Устройство наружных сетей водопровода.
- Устройство наружных сетей канализации.
- Устройство наружных сетей теплоснабжения.
- Устройство наружных и внутренних электрических сетей, сетей автоматизации.
- Монтажные работы.
- Пусконаладочные работы.
- Устройство автомобильных дорог и аэродромов.
- Устройство железнодорожных и трамвайных путей.
- Устройство мостов, эстакад и путепроводов.
- Гидротехнические работы, водолазные работы.
- Работы по организации строительства, реконструкции и капитального ремонта привлекаемым застройщиком или заказчиком на основании договора юридическим лицом или индивидуальным предпринимателем (генеральным подрядчиком).

Компания насчитывает более 100 сотрудников (включая рабочие специальности) и имеет организационную структуру, отображённую на рис.2.1.

Компания располагает соответствующими основными фондами и имеет соответствующий опыт выполнения работ.

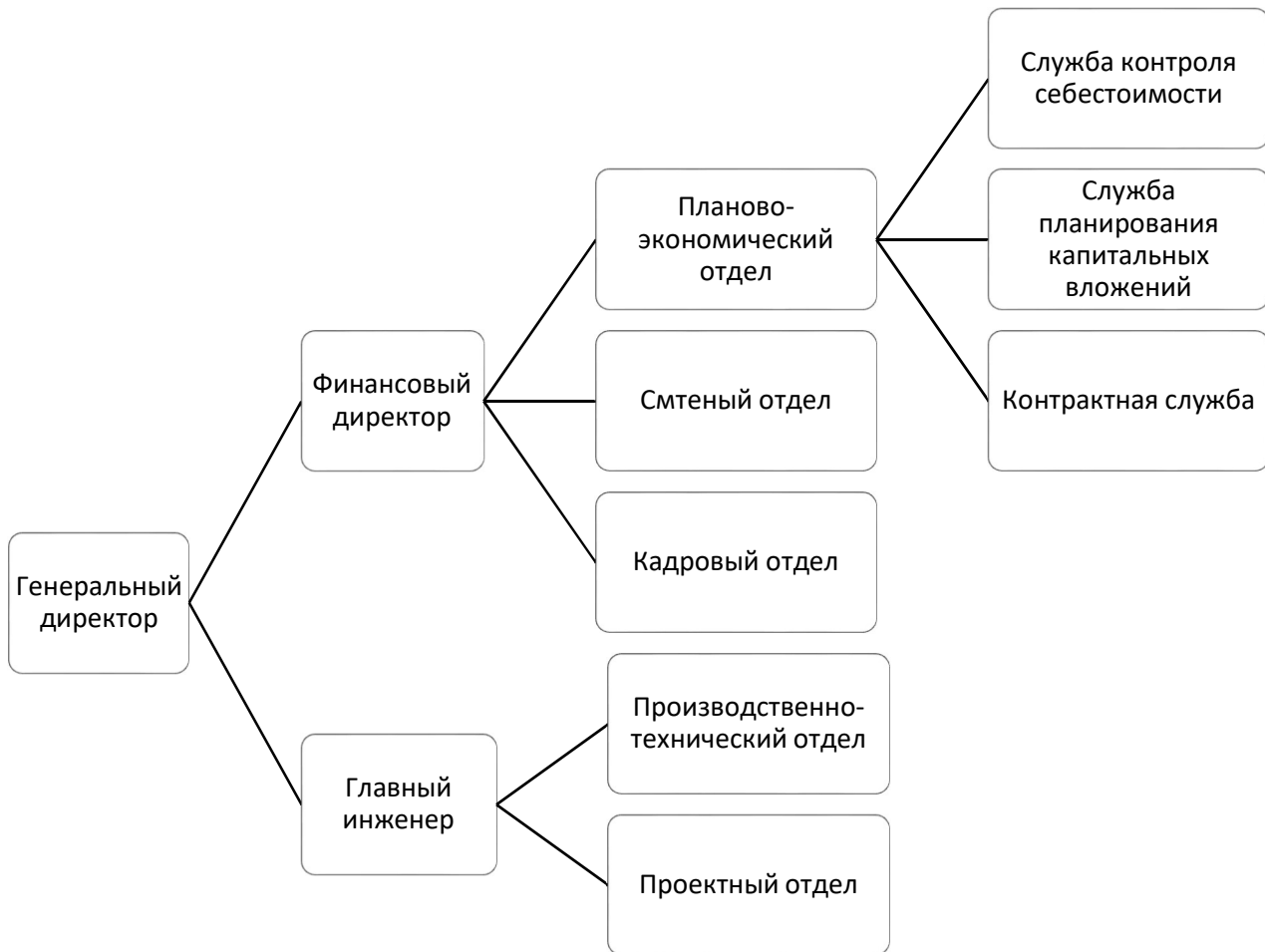


Рис.2.1. Организационная структура компании

2.2. Входные данные и их анализ

Пространственные данные подразделяются на 2 основных типа растровые и векторные.

Векторные данные

К векторным данным относятся: точки, полилинии, многоугольники.

1. Точки

Их обычно используют для обозначения географических объектов, для которых важны не форма и размеры, а местоположение. Возможность обозначения объектов точками напрямую зависит от масштаба карты. В то время как на карте страны или мира города принято обозначать точечными объектами, то на карте города или района сам город следует представлять в виде массива объектов.

2. Полилинии

Полилинии служат для отображения линейных объектов. Полилиния — это ломаная линия, составленная из прямых отрезков. Полилиниями обычно отображают реки, улицы, дороги, железнодорожные пути и т.д. Допустимость отображения объектов полилиниями так же зависит от масштаба карты. Например, крупная река в масштабах континента вполне может изображаться линейным объектом, тогда как уже в масштабах города требуется её изображение площадным объектом. Основной характеристикой линейного объекта является длина.

3. Многоугольники (полигоны)

Полигоны служат для обозначения площадных объектов с четко заданными границами. Например, страны, континенты, озера, парки, здания. Характеризуются длиной и площадью периметра.

Растровые данные

К растровым данным относятся так называемые гриды - изображения поверхностей или данные по пространственным свойствам. В то время как векторные объекты используют геометрические переменные (точки, полилинии и полигоны) для изображения объектов реального мира, растровые данные основаны на ином подходе. Растры являются матрицами пикселей (также называемых ячейками), каждый из которых хранит определенное значение для области, соответствующую этому пикселу (рис.2.2).

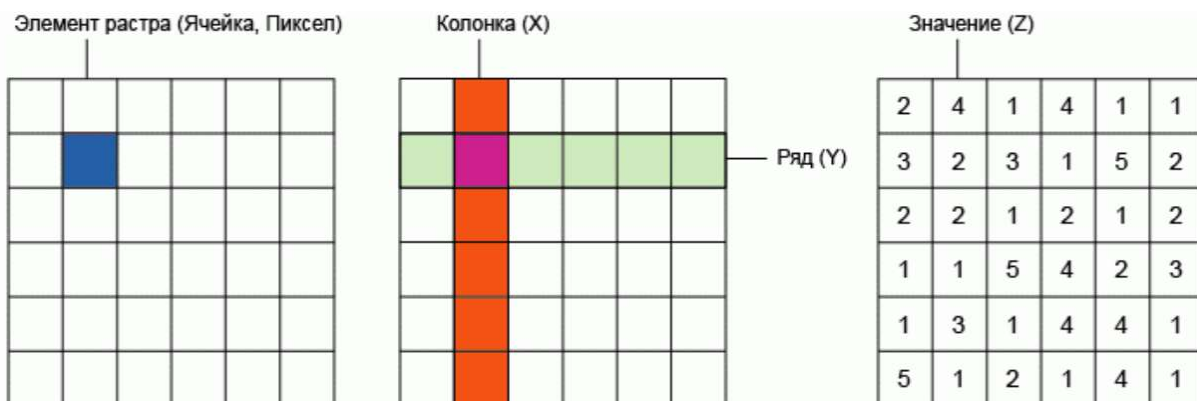


Рис. 2.2. Растровые данные

Иногда растровые данные создаются на основании векторных, так как держатели этих данных хотят опубликовать их в доступном формате. Если пользователю необходимо иметь возможность просматривать и анализировать атрибутивную таблицу, растровые форматы не оптимальны для такой задачи, так как большинство из них не может иметь связанной атрибутивной таблицы.

Векторная и растровая модель данных ведет себя по-разному при масштабировании (рис. 2.3), следовательно, для более точной работы используется масштаб 1:50000, как утвержденный.

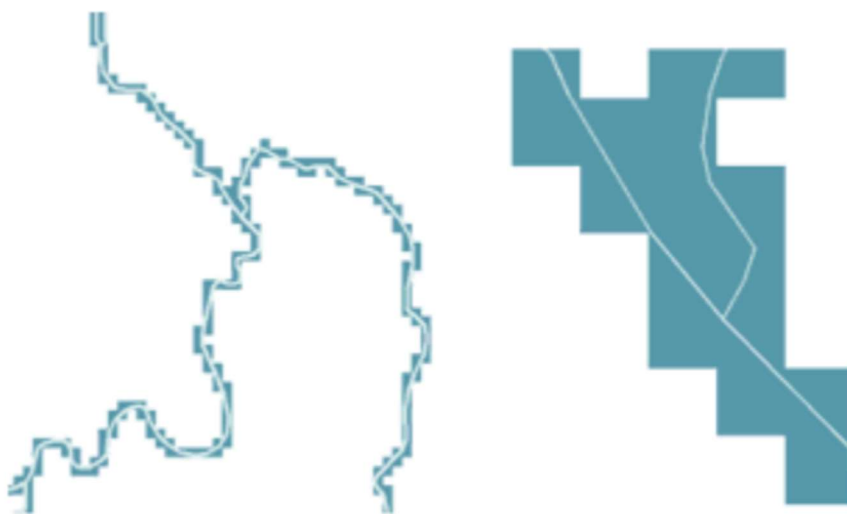


Рис. 2.3. Масштабирование

Сравнение векторной и растровой модели данных продемонстрировано в таблице 9.

Таблица 9. Сравнение растровой и векторной модели данных

Свойство/модель данных	Растровая	Векторная
Масштабируемость	-	+
Избыточность (данных)	-	+
Передача непрерывных свойств	+	-
Передача дискретных объектов	-	+
Легкость создания	+	-

Из таблицы видно, разные модели данных подходят для разных целей.

Проекты по обустройству месторождений нефти и газа обычно не имеют достаточную обеспеченность актуальной пространственной информацией, но

для этапа технико-экономического обоснования этого достаточно. В качестве картографической основы используют топографические карты масштаба от 1:25 000.

Данные, которые следует учитывать при проектировании маршрута:

- населенные пункты;
- промышленные предприятия;
- сельскохозяйственные предприятия;
- особо охраняемые зоны, военные части;
- заповедники;
- уклон местности;
- болота и обводненные участки;
- вечномёрзлые грунты;
- дороги;
- реки;
- озера;
- территории с просадочными грунтами;
- растительность.

Карты местности постоянно задействованы в проектных работах, при этом, отображаемые на них пространственные данные следует учитывать совместно. Так, при прокладке внутрипромысловых трубопроводов учитываются такие параметры, как наличие на пути трасы водоёмов, болот, древесной растительности, уклоны территории и т.п. Суммирование тематических растровых карт позволяет получить интегральную карту, на которой качественным фоном отображаются благоприятные и неблагоприятные условия для строительства.

В результате суммирования карт образуется общая картографическая основа для проектирования. На рис.2.4 оттенками зелёного показаны участки, наиболее благоприятные для прокладки линейного объекта, оттенками красного – менее благоприятные. В данном примере использовались три параметра оценки. Но в каждой конкретной ситуации возможно введение дополнительных

параметров и поправочных коэффициентов, в зависимости от имеющихся данных.

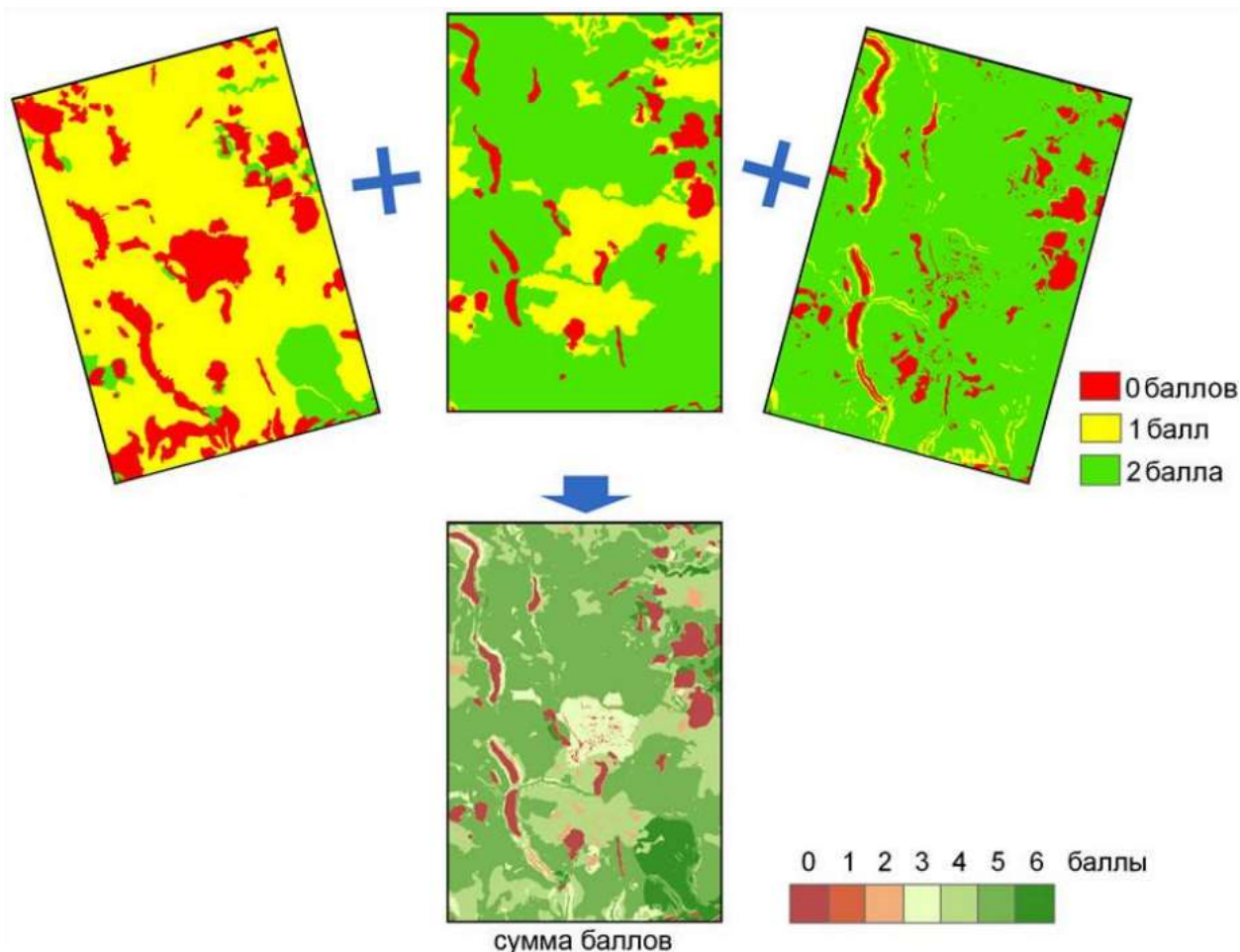


Рис. 2.4. Суммирование растровых тематических карт

2.3. Анализ алгоритма A*

По результатам исследования, представленным выше, для поиска оптимального маршрута был выбран алгоритм A*. Далее проведём его подробный анализ.

Для начала обозначим основную терминологию, используемую в анализе:

Узел - позиция на карте.

Открытый список - список узлов в которые можно переместиться и которые являются смежными с закрытыми узлами.

Закрытый список - список узлов, в которые можно переместиться, но которые уже были пройдены.

На рис.2.5 проиллюстрированы узлы из которых строится карта. Фактически, узлом будет являться каждый квадрат карты (ячейка). На карте отмечены начальная позиция А и конечная позиция В. Поскольку начальная позиция всегда будет являться частью искомого маршрута, то она автоматически включается в закрытый список пройденных узлов.

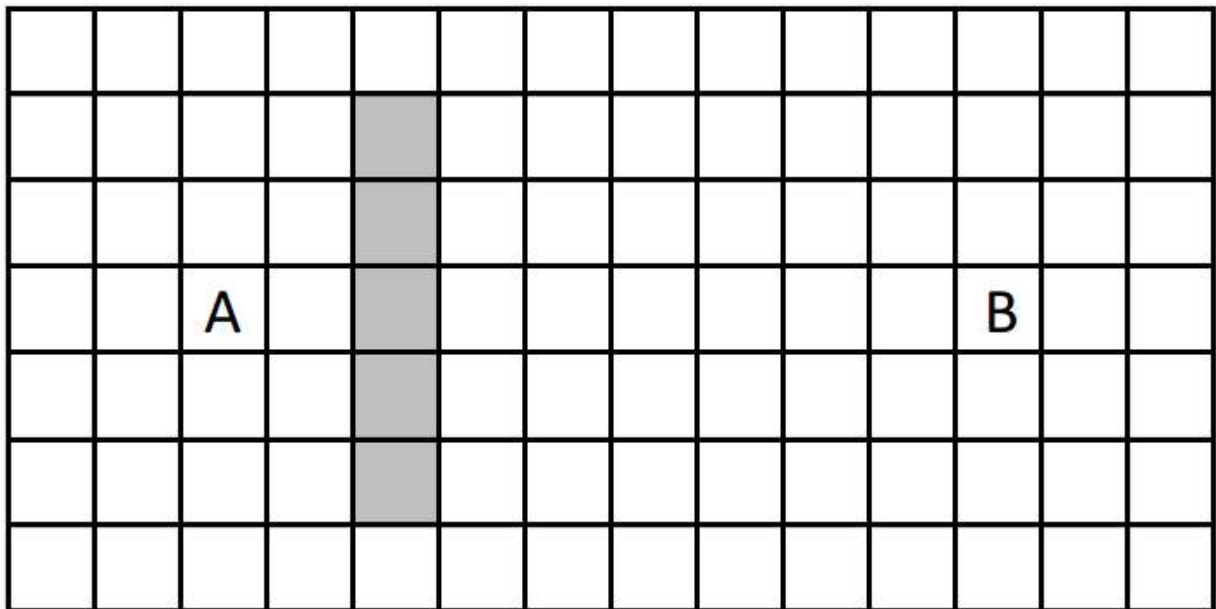


Рис. 2.5. Терминология в алгоритме A*

Узлы, соседствующие с последним узлом из закрытого списка пройденных узлов, помещаются в открытый список. Это продемонстрировано на рис. 2.6.

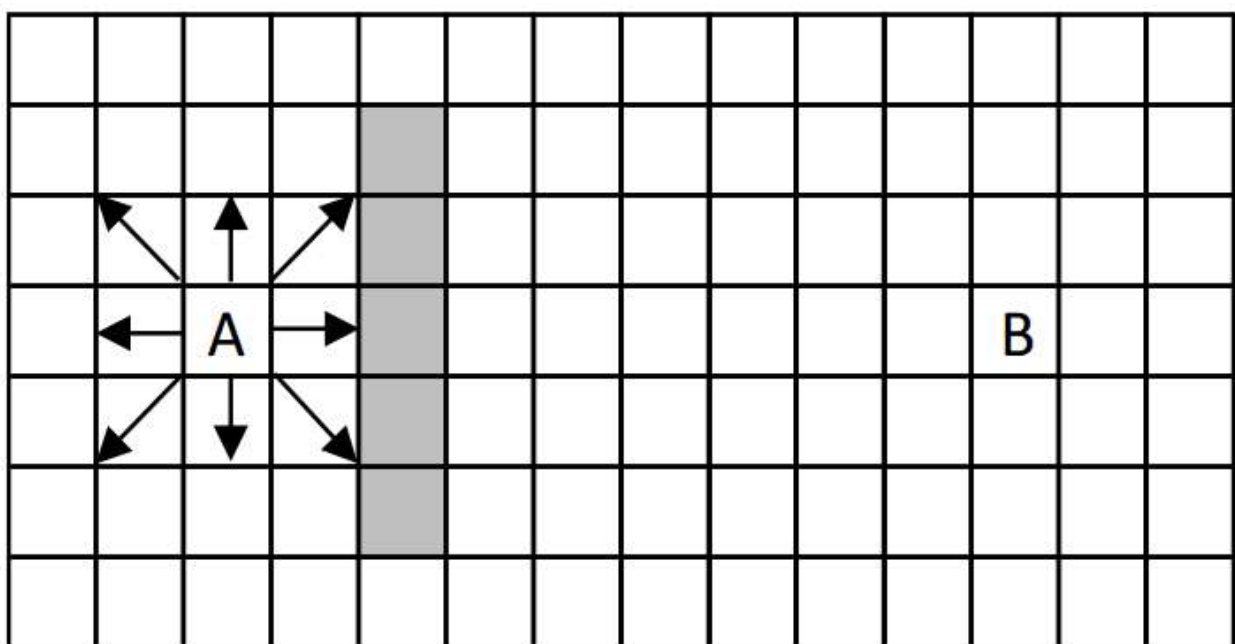


Рис. 2.6. Добавление узлов в открытый список

На рис.2.6. изображены восемь узлов, входящих в открытый список и один узел, входящий в закрытый список. Стрелки указывают направление перемещения из того закрытого узла, к которому относятся эти открытые узлы. Закрытый узел в этом случае принято называть родительским узлом каждого из открытых узлов.

Первое, что делает алгоритм A^* — это добавление начального узла в закрытый список. Это делается потому, что начальный узел всегда будет первым узлом полученного пути. Сделав это необходимо найти все узлы, которые являются смежными с начальным и в которые можно перейти. Если смежный узел доступен, он добавляется в открытый список. Так как в самом начале нет никаких открытых узлов, перед началом работы алгоритма открытый список пуст.

Этапы поиска:

- 1) Поместить начальный узел в закрытый список.
- 2) Поместить доступные смежные узлы в открытый список.

Следующим этапом является вычисление стоимости узлов.

У каждого узла в закрытом и открытом списках есть его вес, поэтому некоторые из них лучше подходят для создания пути, чем другие. Чтобы выяснить, какой узел является самым лучшим, необходимо каждому узлу в закрытом и открытом списках назначить его итоговую «стоимость». Как только всем узлам назначена стоимость, достаточно простой сортировки, чтобы выяснить какой узел является самым дешевым. Для вычисления стоимости узлов в алгоритме A^* необходимы следующие значения:

- Базовая стоимость узла.
- Стоимость возврата к начальному узлу.
- Стоимость достижения цели.

Базовая стоимость узла. Базовой стоимостью узла называется стоимость передвижения через данный узел. В простейшем случае базовая стоимость всех доступных узлов одна и та же. Однако, в случае с поставленной задачей у

различных узлов различная стоимость, в зависимости от типа ландшафта и других факторов.

Стоимость относительно цели. Последний компонент стоимости — это стоимость достижения цели из данного узла. Она вычисляется путем сложения количества строк и столбцов на которые текущий узел отстоит от цели.

Общая стоимость. Как только все три упомянутых выше стоимости вычислены, нужно их просуммировать и получить общую стоимость узла.

Зная общую стоимость каждого узла, очень просто найти наилучший узел, для добавления его в закрытый список. Необходимо отсортировать узлы по значению общей стоимости и выбрать тот из них, у которого она меньше всего.

После того, как узел с наименьшей общей стоимостью найден, нужно добавить его в закрытый список в качестве кандидата на участие в итоговом пути. Так же необходимо удалить этот узел из открытого списка, чтобы он не был обработан снова.

Продолжение поиска. Если в открытом списке отсутствует конечный пункт маршрута, следует продолжать поиск пути добавив в открытый список те узлы, которые находятся вокруг узла только что добавленного в закрытый список. После этого нужно найти открытый узел с наименьшей стоимостью и добавить его в закрытый список. Эти действия будут повторяться до тех пор, пока конечный пункт маршрута не окажется в открытом списке.

Как только конечный пункт маршрута окажется в открытом списке, необходимо будет составить путь обратно к исходной точке. Для этого берется родитель открытого узла, в котором расположен конечный пункт. Затем родитель родителя и так далее до тех пор, пока не произойдет возвращение к исходной позиции. В результате будет найден путь от конечного пункта до начального. Теперь необходимо инвертировать полученный путь, чтобы получить итоговый маршрут от исходной точки до цели.

2.4. Оптимизация A*

Можно выделить, ряд способов модифицировать алгоритм поиска, для того чтобы получить оптимальные результаты при использовании ограниченных ресурсов:

Лучевой поиск. Одним из основных способов борьбы с ограничением по памяти принято считать применение ограничений на количество узлов в открытом списке. Когда список переполнен и необходимо прибавить новый узел, просто исключается узел с наихудшим показателем. Список закрытых узлов также может быть исключен, если каждая его ячейка хранит в себе протяженность наилучшего пути и обратный указатель. Данный алгоритм не гарантирует оптимальности пути, по причине того, что узел, ведущий к нему, может быть исключен, но все равно может позволить отыскать разумный путь.

Алгоритм последовательных приближений. Для алгоритма A* возможно использование алгоритма последовательных приближений, включение его полностью избавляет от необходимости хранить списки открытые (Open) и закрытые (Closed). Производится обычный рекурсивный поиск, собирается накопленная стоимость маршрута $g(x,y)$, и поиск останавливается при достижении значением $f(x,y) = g(x,y) + h(x,y)$ заданных пределов. Начинать нужно с пределом остановки, равным $h(x_1, y_1)$, и в каждой последовательной итерации, устанавливать новый предел остановки, равным минимальному значению $f(x,y)$, которое превысило прежнюю границу.

Метрика $h(n)$. Как обсуждалось выше, если приближение оставшейся стоимости пути слишком мало, то A* может быть очень неэффективным. Но если приближение слишком высоко, то для найденного пути не гарантируется оптимальность. На картах диапазон изменения стоимостей ландшафта широк - от болот до населенных пунктов, поэтому можно поэкспериментировать с

различными промежуточными значениями, чтобы найти точный баланс между эффективностью поиска и качеством найденного пути.

Поведение алгоритма напрямую зависит от того, какая метрика используется, выбор метрики зависит от постановки задачи:

- если мы можем перемещаться в четырех направлениях, в качестве метрики стоит выбрать манхэттенское расстояние. [49]

$$h(x, y) = |x_1 - x_2| + |y_1 - y_2| \quad (5)$$

- расстояние Чебышева [39] применяется, когда к четырем направлениям добавляются диагонали:

$$h(vx, y) = \max(|x_1 - x_2|, |y_1 - y_2|) \quad (6)$$

- если передвижение не ограничено сеткой, то можно использовать евклидово расстояние:

$$h(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (7)$$

A^* работает на любом графе, где можно выделить функцию расстояния (если ее нет, он становится неэффективен), поэтому можно использовать разные варианты сетки, например, вариант не 4-угольной сетки, а 6-угольной.

Использование «запрещенных» областей. В задаче выбора оптимального пути для прокладки трубопровода на карте есть ряд ограничений по выбору маршрута. Например, трубопровод не может быть проложен на территориях заповедников, военных частей и вблизи населенных пунктов. Эти ограничения необходимы с точки зрения нормативных актов и действующего законодательства, но в то же время полезны, так как позволяют заранее отсеять области, в которых строительство не может быть произведено.

Для того, чтобы сократить время выполнения алгоритма, можно уменьшить количество входных данных, предварительно обработав данные и присвоив выбранным «запрещенным областям» значение ∞ .

В процессе работы алгоритма A^* для вершин рассчитывается функция $f(x, y) = g(x, y) + h(x, y)$, и если есть область, значения которой «бесконечно большие», то алгоритм не будет их рассматривать.

3. РАЗРАБОТКА СЦЕНАРНОГО ЯЗЫКА ПРИМЕНЕНИЯ АЛГОРИТМА А* И ЕГО АПРОБАЦИЯ НА ПРИМЕРЕ НЕФТЕПРОВОДА «А МЕСТОРОЖДЕНИЕ – Б»

Сценарный язык для построения оптимального маршрута трубопровода был реализован на языках python 2.7 и java Script, библиотеке ArcGis для java Script (ArcGIS Web API for JavaScript API 4.0). В приложении 1 описывается видение продукта.

3.1. Проектирование сценарного языка

Проектирование:

а) Функциональные требования: определение цели разработки и исходных требований.

Цель создания сценарного языка – позволить пользователям прокладывать оптимальный маршрут линейных объектов, в частности нефте- и газопроводов, посредством программного обеспечения ArcGIS.

Исходные требования:

- Определение стоимости прохода по конкретному типу участка;
- Создание оптимального маршрута прокладки трубопровода по двум точка.

В таблице 10 проведем анализ пользователей.

Таблица 10.

Анализ пользователей

Название	Представитель	Роль
Конечный пользователь	Специалист, занимающийся расчетом стоимости обустройства месторождений	Используется функционал системы для автоматизации работы

б) Концептуальное проектирование: моделирование процесса, для которого разрабатывается приложение.

В таблице 11 изображена концепция проектируемого продукта.

Таблица 11.

Концептуальное проектирование

Сейчас	Будет
<p>- Специалист получает задание построить оптимальный маршрут на конкретном месторождении с конкретными точками начала и конца, а также карту в масштабе 1:50 000</p> <p>- Специалист вручную строит маршрут, полагаясь исключительно на свой опыт</p> <p>- Специалист определяет координаты точек перегиба</p> <p>Специалист предоставляет карту с маршрутом и координатами точек перегиба</p>	<p>Специалист получает задание построить оптимальный маршрут на конкретном месторождении с координатами точек начала и конца Специалист получает карту в масштабе 1:50 000</p> <p>- Программа определяет типы местности</p> <p>- Специалист определяет стоимость прокладки трубопровода по типам</p> <p>- Программа строит маршрут, основываясь на оценках специалиста - Специалист вносит корректировки (при необходимости)</p> <p>- Специалист экспортирует карту с маршрутом и координатами точек перегиба</p> <p>Специалист не получает карту</p> <p>- Специалист выбирает участок и слои из БД</p> <p>- Программа строит маршрут, основываясь на оценках, предусмотренных в БД</p> <p>- Специалист вносит корректировки (при необходимости)</p> <p>- Специалист экспортирует карту с маршрутом и координатами точек перегиба</p> <p>Специалист предоставляет карту с маршрутом и координатами точек перегиба</p>

в) Физическое проектирование: выбор платформы, на которой будет реализовываться проект.

Проект реализован на python 2.7, java Script, библиотеке ArcGis для java Script (ArcGIS Web API for JavaScript API 4.0).

Последовательность действий проиллюстрирована на диаграмме деятельности (рис. 3.1)

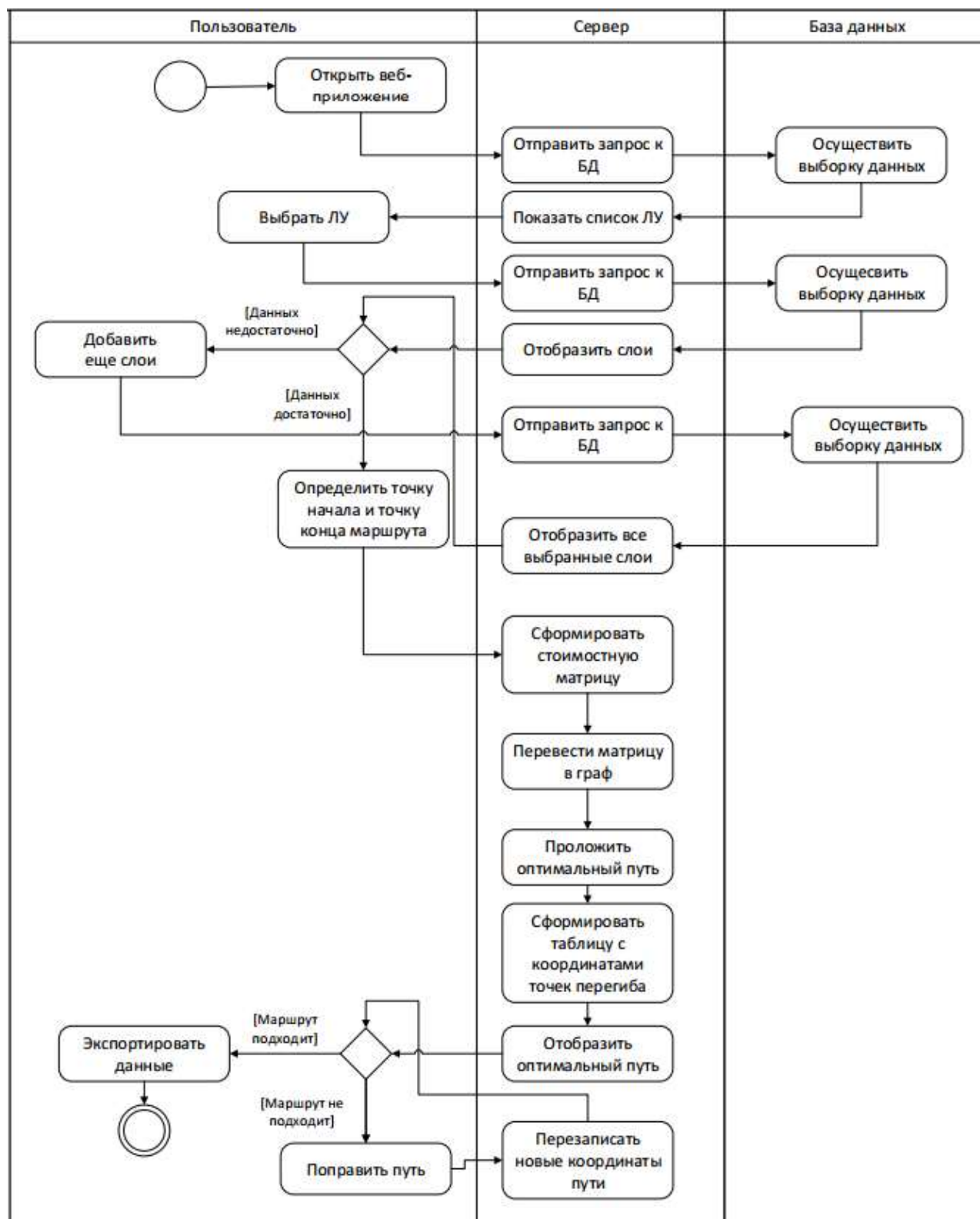


Рис. 3.1. Диаграмма деятельности

Для создания общей карты необходимо преобразовать все объекты в растр, а затем просуммировать стоимости прохода по каждой ячейке в зависимости от объектов, к которым она относится.

3.2. Общие сведения о программной среде ArcGIS

ArcGIS основана на общей модульной библиотеке отдельно взятых программных ГИС - компонентов. В состав библиотеки входит обширный набор программных модулей, позволяющих описать различные по сложности объекты. В комплексе эти модули предоставляют пользователям богатую функциональность современной ГИС.

В ArcGIS Desktop входят следующие приложения:

- ArcGlobe
- ArcScene;
- ArcCatalog;
- ArcToolbox;
- ModelBuilder.
- ArcMap;

С помощью данных приложений и их интерфейсов возможно выполнять разные задачи ГИС, включая редактирование, анализ данных, их визуализацию и обработку.

Также предусмотрена возможность разработки своих собственных расширений к ArcGIS, с помощью ArcObjects – API библиотеки программных компонентов ArcGIS.

ArcToolbox (рис. 3.2) содержит широкий набор функций геообработки, включая следующие инструменты для:

- обработки покрытий;
- статистического анализа;
- векторного анализа;
- управления данными;

- конвертации данных;
- геокодирования.



Рис. 3.2. Окно ArcToolbox

Arcgis позволяет компоновать последовательности инструментов геообработки, передавая выход одного инструмента в другой инструмент как вход.

ModelBuilder — это специальное приложение, для создания последовательности инструментов геообработки из встроенных функций и скриптов на языке Python 2.7.

3.3. Преобразование векторных объектов в растровые с помощью API ArcGIS

Для того, чтобы можно было создать общую карту для построения оптимального маршрута прокладки трубопровода, необходимо было преобразовать все данные в растровые. Это было осуществлено с помощью API ArcGIS, в частности инструментов для конвертации точек, полилиний и полигонов в растр.

В таблице 12 рассмотрены входные параметры для использования API.

Таблица 12.

Входные параметры, необходимые для использования API

Параметр	Объяснение	Тип данных
in_features	Входной набор линейных объектов, который будет конвертирован в растр.	Feature Layer
value_field	Поле, используемое для присвоения значений выходному растру	Field
out_rasterdataset	Создаваемый выходной набор растровых данных.	Raster Dataset

В качестве входных объектов берутся все слои, добавленные пользователем, для присвоения стоимости по умолчанию используется поле «Cost».

Алгоритм преобразования представлен на рисунке 3.3.

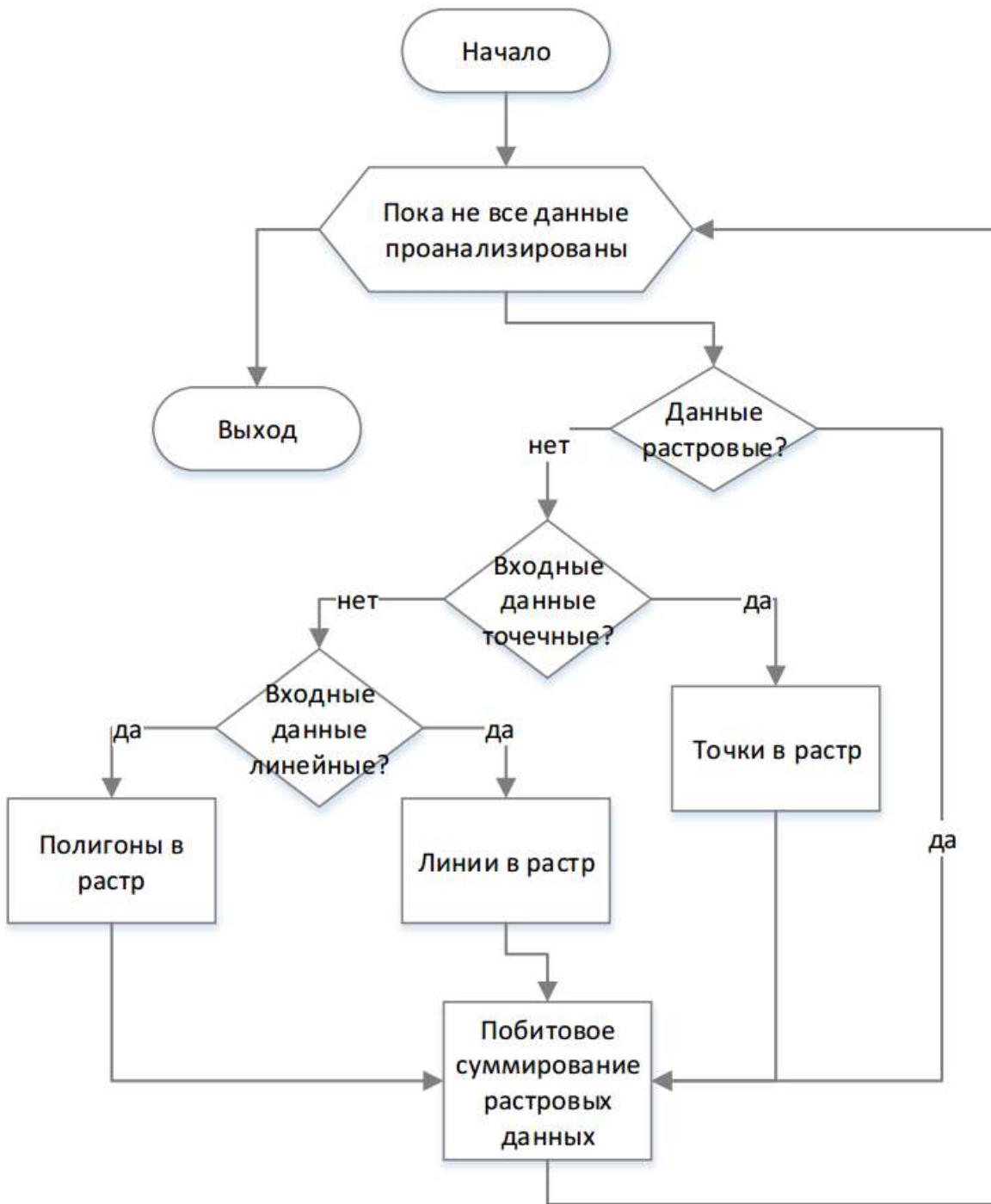


Рис. 3.3. Алгоритм перевода векторных объектов в растровые

Преобразование матрицы в граф связности происходит по шаблону ферзя. Алгоритм представлен в блок-схеме на рисунке 3.4.

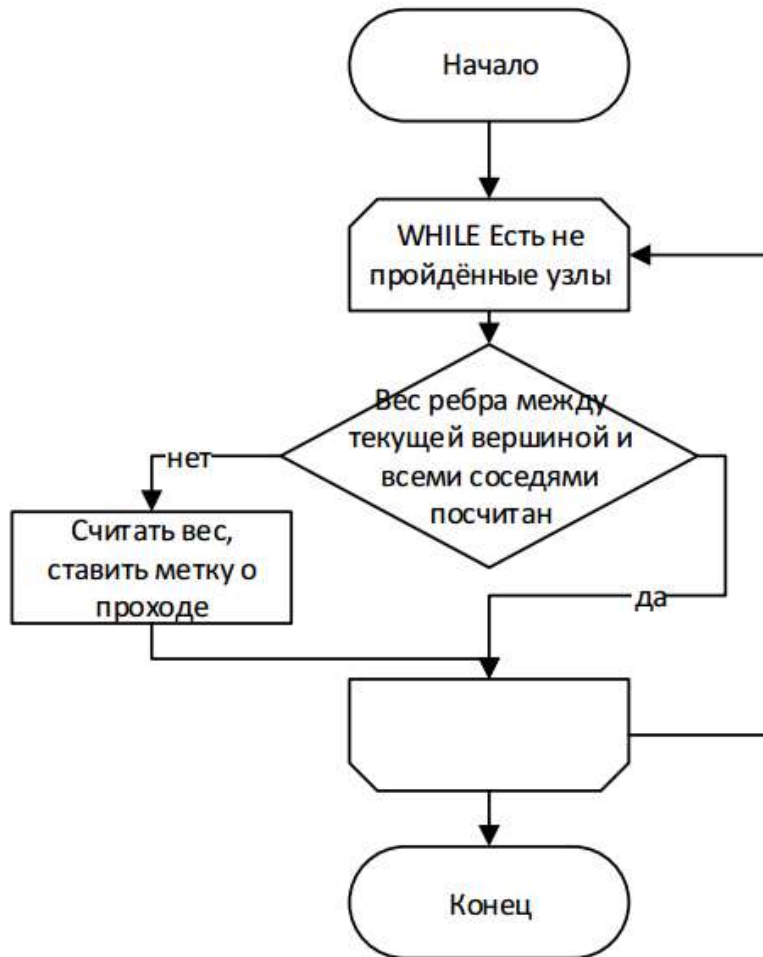


Рис. 3.4. Алгоритм построения графа связности

Расчет веса ребра происходит по простым математическим формулам, представленным выше в пункте про построение графа связности по данным растровой поверхности.

В классической версии алгоритм A* выполняется в 2 этапа: сначала ищется оптимальный маршрут, а потом выполняется обратная трассировка для нахождения пути. При этом сам алгоритм выглядит так рисунок 3.5.

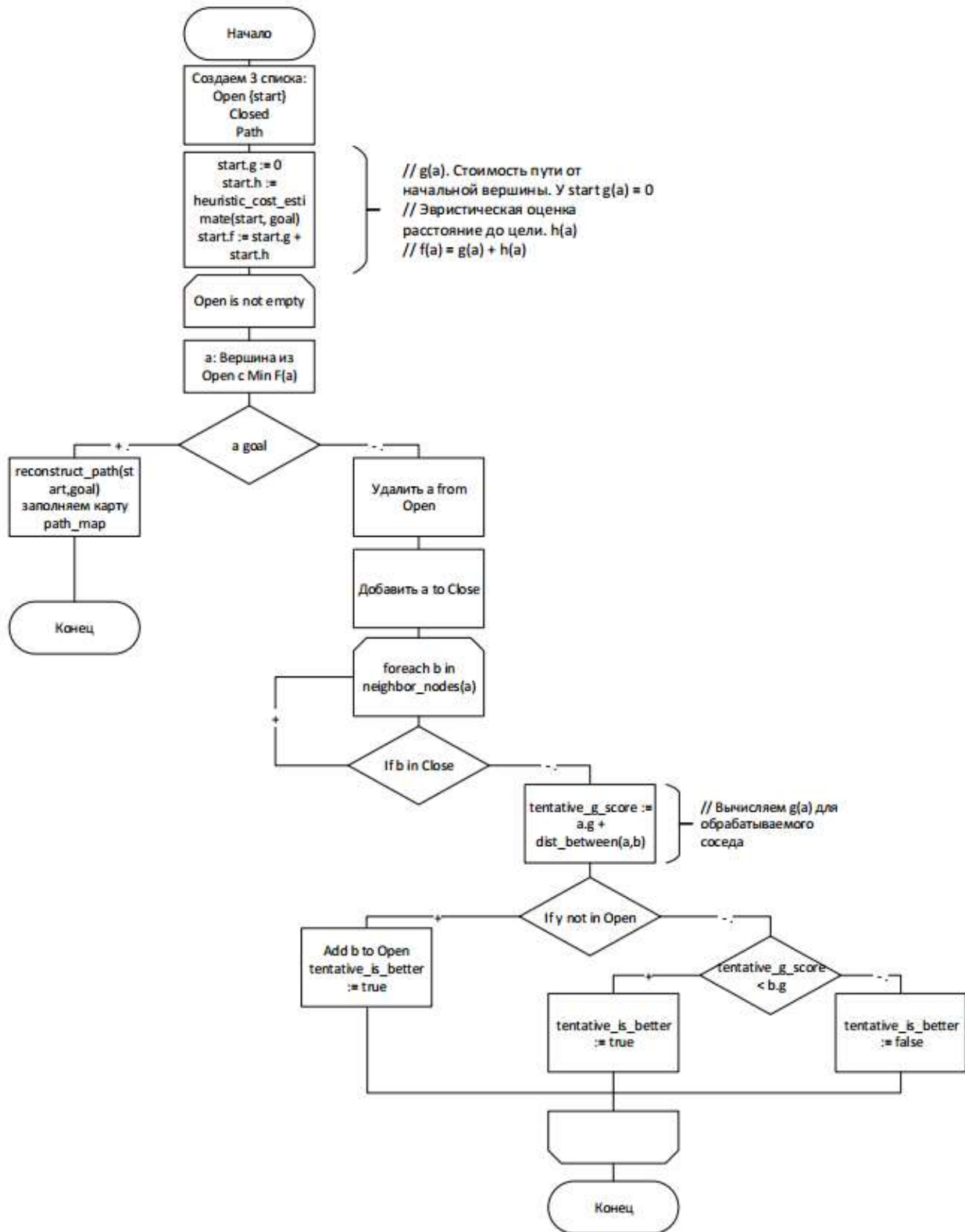


Рис. 3.5. Блок-схема алгоритма A*

Start и goal – начальная и конечная ячейки соответственно. A и b обозначают ячейки, просматриваемые алгоритмом, при этом у них есть данные по координатам.

На рисунке 3.6 представлена функция обратной трассировки.

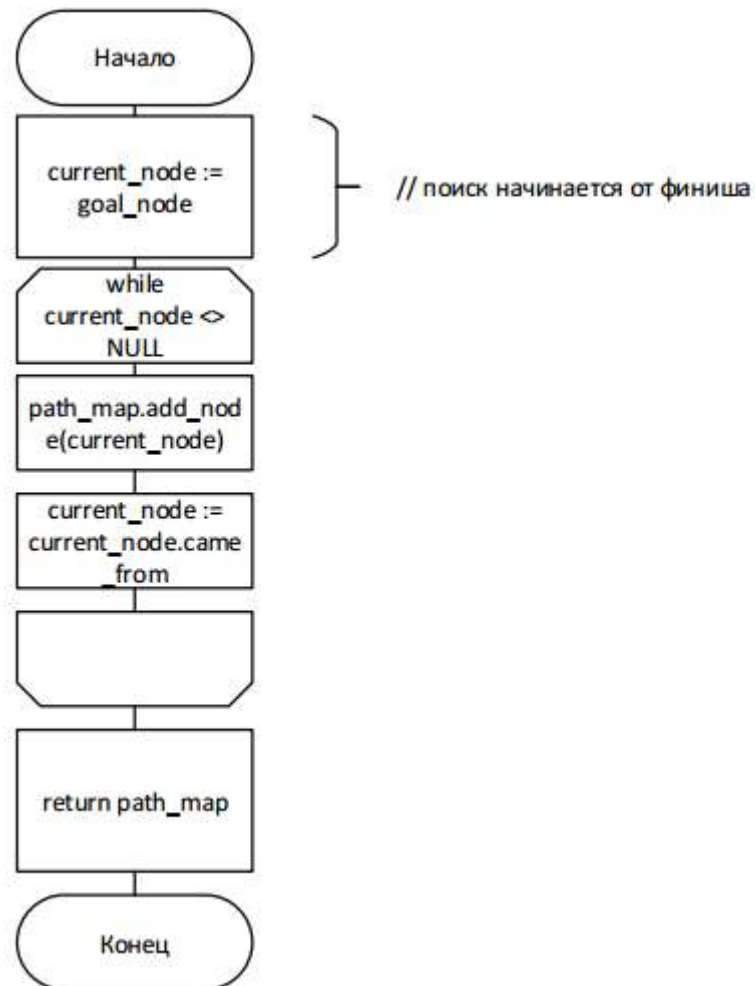


Рис. 3.6 - Блок-схема алгоритма A*. Функция обратной трассировки

В данной работе алгоритм был модифицирован с помощью последовательных приближений и «запрещенных областей», для хранения и возможности быстрого доступа к данным использовались хэш-таблицы. Лучевой поиск не был реализован, потому что в этом случае алгоритм не гарантирует оптимальности пути, хотя и может позволить найти разумный путь

Как был реализован алгоритм в этой работе представлено на рисунках 3.7 и 3.8.

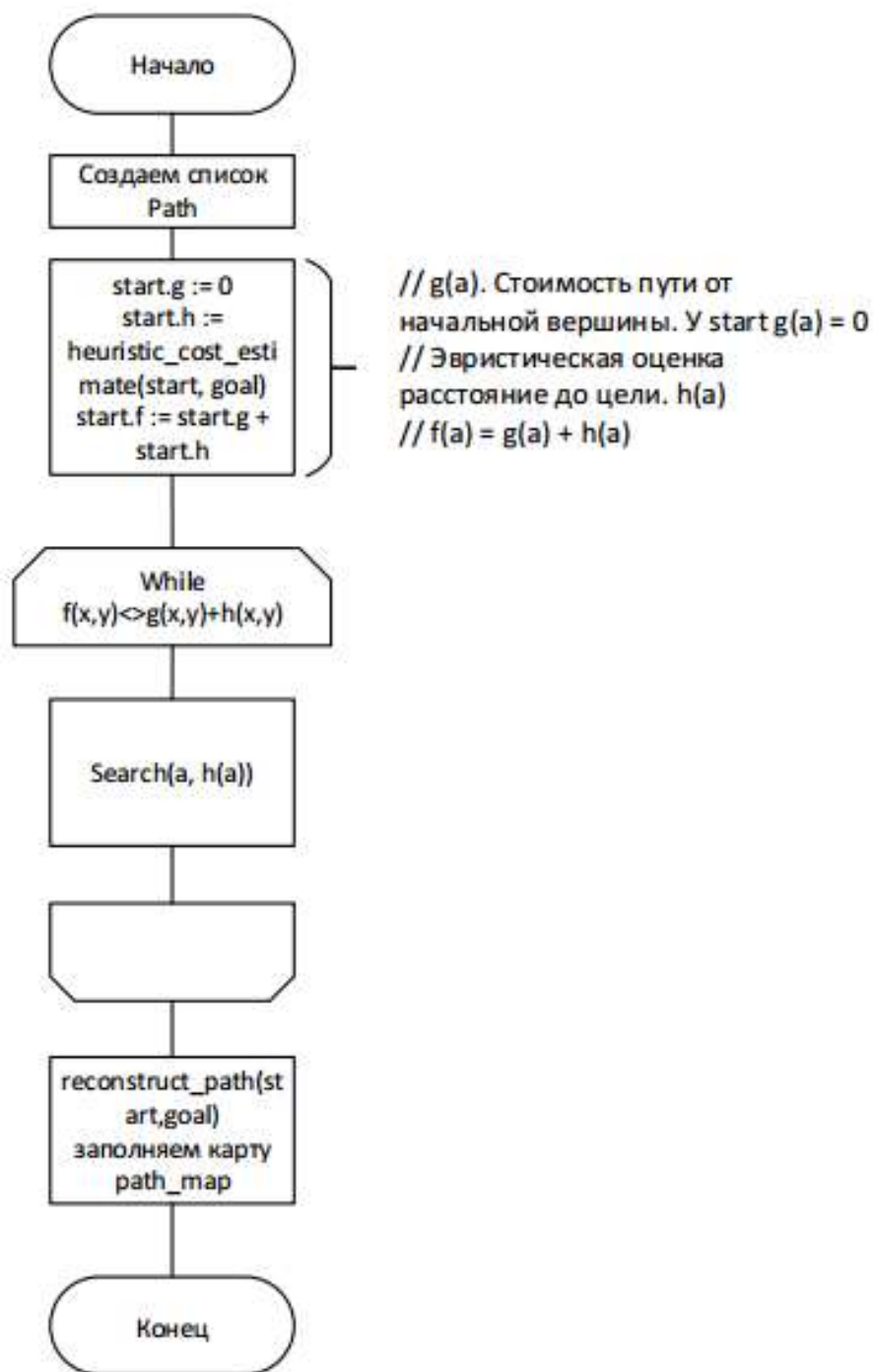


Рис. 3.7. Блок-схема алгоритма A* после модификации.

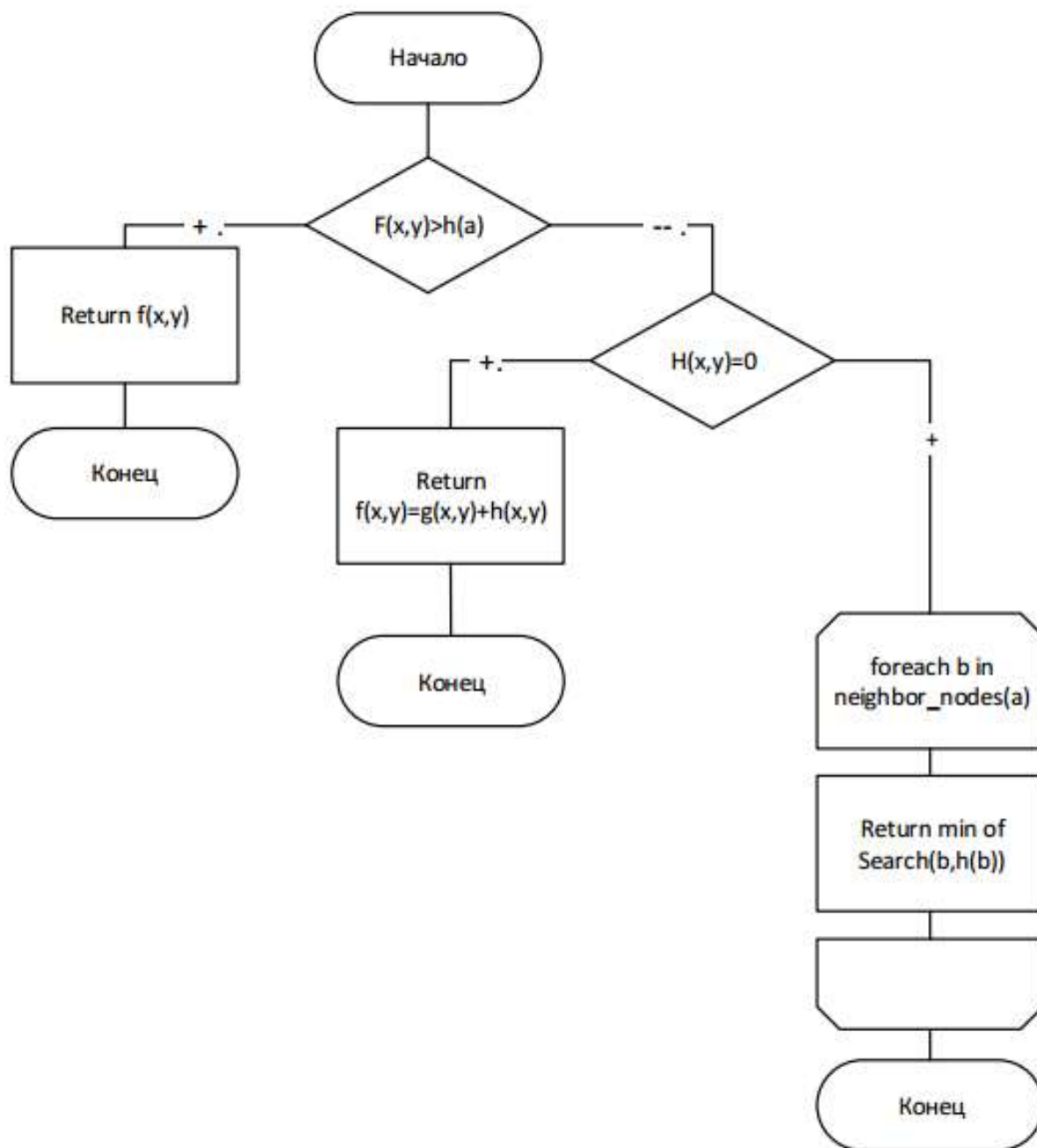


Рис. 3.8. Функция для алгоритма последовательных приближений.

3.4. Анализ на возможность оптимизации нефтепровода

«А месторождение – Б»

3.4.1. Показатели действующего трубопровода «А месторождение – Б»

Для апробации разработанного метода был выбран нефтепровод, пролегающий между УПН (установка подготовки нефти) А месторождения и ПСП (пункт приема-сдачи сырья) Б. Данный трубопровод принадлежит нефтяной компании и является связующим между месторождениями и магистральным трубопроводом «Восточная Сибирь – Тихий океан». Нефтепровод Б является связующим между месторождениями Западной и Восточной Сибири и рынками сбыта в Азии и США.

Длина линейной части трубопровода составляет 61 км, строительство завершено в 2011 году. В конечной точке нефтепровода расположен пункт сдачи-приёмки сырья (ПСП), первая очередь которого была завершена в январе 2011 года. Мощность ПСП составляет 3,4 млн тонн в год.

Характеристики нефтепровода «А месторождение – Б»:

Протяженность: 61 км.

Мощность: 4,50 млн. тонн/год

Состояние: в эксплуатации

Дата ввода в эксплуатацию: 2011 г.

Суммарные затраты: 2,2 млрд. руб.

Затраты из расчёта на 1м нефтепровода: 36,066 млн. руб.

Тип в зависимости от транспортируемой среды: Нефтепровод

Тип в зависимости от вида прокладки: Подземный

Протяженность участка в сложных условиях: 14 км.

Координаты точки А

Координаты точки Б

Используются координаты точек перегибов.

Таблица 13.

Координаты точек поворотов действующего нефтепровода
«А месторождение – Б».

	Широта	Долгота
А	57.98	106.79
1	57.98	106.81
2	57.95	106.80
---	---	---
20	57.59	106.85
21	57.53	107.02
Б	57.53	107.03

Графически данный нефтепровод изображен на рис. 3.9.



Рис. 3.9. Маршрут пролегания нефтепровода Яракта – Б.

3.4.2. Применение алгоритма оптимизации и анализ результатов

Для проведения оптимизации будем использовать следующие факторы:

1) Топографическая карта перепада высот. Данный растр оценивается с точки зрения влияния перепада высот ландшафта на перегибы трубопровода в вертикальной проекции и увеличения объема земляных работ.

2) Карта доступности инфраструктуры. Данный растр учитывает доступность места пролегания трубопровода с позиции подъезда спецтехники, доставки людей, подвоза материалов.

3) Карта охраняемых и заповедных земель. Данный растр учитывает области особо охраняемых и заповедных земель, прокладка трубопровода по которым невозможна или сопровождается существенными затратами.

4) Карта экологического влияния. Данный растр учитывает затраты на вырубку деревьев и прочие факторы экологического влияния.

5) Карта рек и водоемов. Данный растр оценивает кол-во пересекаемых рек и водоемов, так как каждое пересечение рек и водоемов сопровождается большим количеством затрат и сложностей в проектировании.

6) Карта тектонической подвижности.

В приложениях 1, 2, 3, 4 проиллюстрированы данные растры, сведённые в четыре карты с оценкой каждой точки прохождения.

Далее проведём наложение растров друг на друга с последующим суммированием их в один растр с общими показателями оптимальности трассировки трубопровода. Следует учитывать, что разные факторы могут обладать разной итоговой ценностью.

В полученной картографической основе, представленной в приложении 5, для проектирования оттенками зелёного показаны участки, наиболее благоприятные для прокладки дороги, оттенками красного – менее благоприятные.

Средствами программного обеспечения ArcGIS и описанного ранее алгоритма оптимизации на языке программирования Python проведём оптимизацию маршрута из точки А в точку Б

Полученный оптимальный маршрут представлен на рис. 3.10, и более подробно в приложении 6.

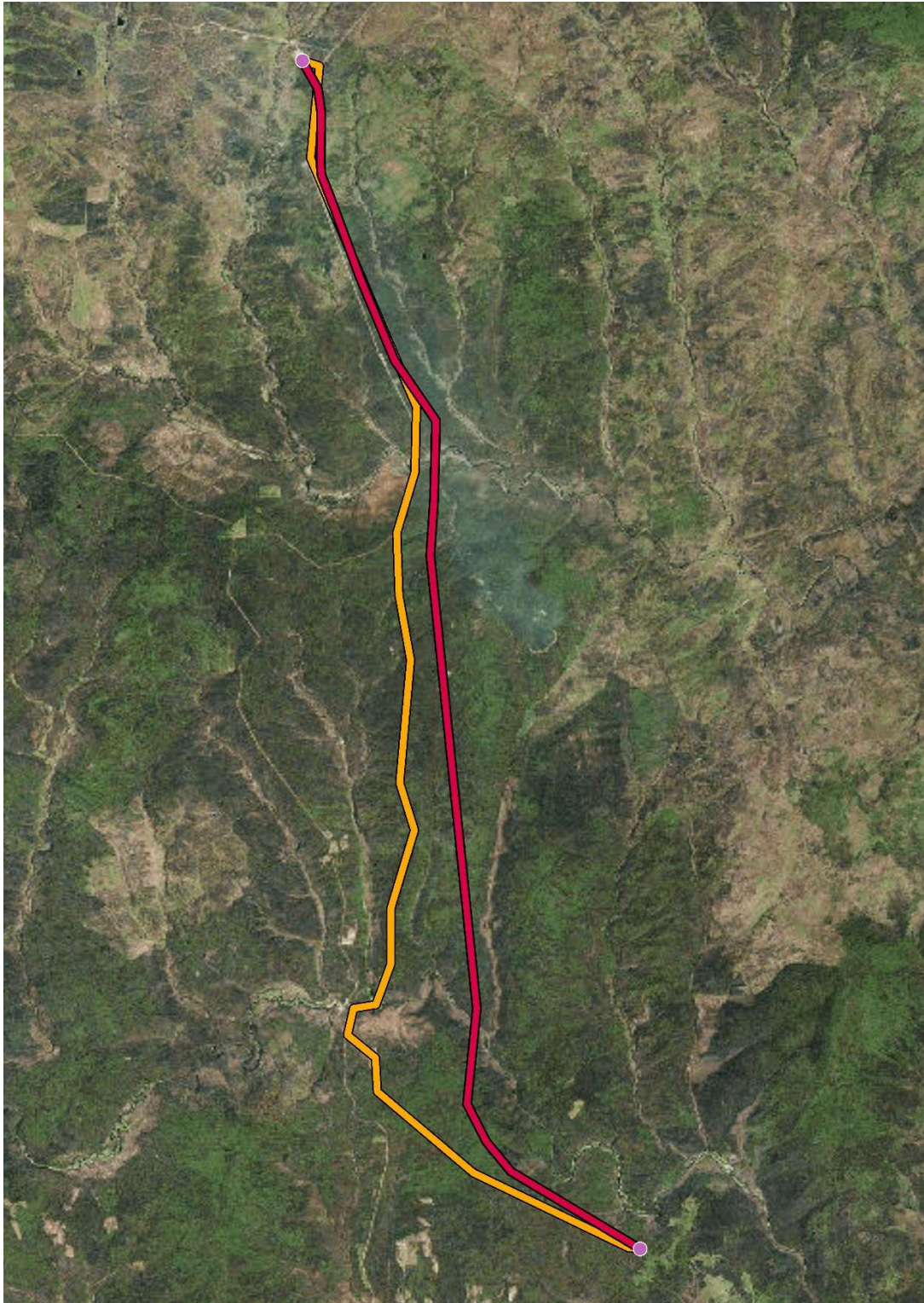


Рис. 3.10. Результат оптимизации пути нефтепровода.

Из приложения можно экспортировать точки перегибов трубопровода, построенного автоматически, в Excel. В таблице 14 приведены точки перегибов в табличной форме.

Таблица 14

Координаты точек поворота оптимизированного варианта.

	Широта	Долгота
А	106,80	57,98
1	57,96	106,81
2	57,94	106,81
---	---	--
11	57,56	106,95
Б	57,53	107,03

Для оценки результатов оптимизации планирования необходимо провести сравнение по различным критериям получившейся трассы с трассой, существующей.

В таблице 15 приведены показатели для оценки эффективности получившихся вариантов трассировки нефтепровода.

Таблица 15. Сравнение результатов

Факторы оценки эффективности	Единица измерения	Существующий вариант	Оптимизированный вариант
Протяженность	м	61 300	56 240
Высота средняя	м	498	478
Высота максимальная	м	607	569
Высота минимальная	м	353	343
Разница максимальной и минимальной высот	м	254	135

Количество переходов через крупные реки	Шт.	2	2
Количество переходов через малые реки	Шт.	1	1
Пересечение крупных водных преград	Шт.	-	-
Протяженность участка в сложных условиях, км.	м	14 000	11 500
Число углов поворота	Шт.	19	11
Средний угол поворота	°	31,75	25,6
Максимальный угол наклона	°	60	45
Средняя удалённость от автодорог	км	5	6,5
Максимальная удаленность от автодорог	км	8	8,4
Участки, проходящие по лесам	км	36	33

Для более точного проведения оценки наилучшего варианта, необходимо ввести поправочные коэффициенты, корректирующие влияние каждого фактора на принятие решения. Следует отметить, что не все факторы будут рассматриваться с точки зрения влияния на стоимостные показатели. Так же учтено влияние на экологию, что имеет особую актуальность, так как 2017 год объявлен годом экологии в России. Помимо прочего, большую значимость имеет технологическая целесообразность выполнения проекта, так как слишком большие показатели перегибов или перепада высот вызовут необходимость пересчёта мощностей насосных станций.

Для дальнейшего анализа была создана экспертная комиссия, на базе организации, каждый член экспертной комиссии оценивал влияние каждого фактора оценки эффективности на общую эффективность проекта.

Состав комиссии:

- 1) Главный инженер
- 2) Начальник сметного отдела
- 3) Начальник проектного отдела
- 4) Финансовый директор

Так как при заключении экспертных оценок влияния факторов используются лингвистические переменные, затрудняющие их анализ в рамках традиционной булевой логики, то принято решение использовать методы нечеткой логики для конечного анализа и суммирования факторов оценки эффективности построения маршрута для нефтепровода «А месторождение – Б». Данный метод авторами был рассмотрен в рамках доклада «Применение методов нечеткой логики в оценке рисков инвестиционно-строительных проектов» на IV ON-LINE Международной научно-практической конференции «Проблемы экономики и управления строительством в условиях экологически ориентированного развития». По каждому фактору дается оценка принимающая значение от 0 до 1 где:

- от 0 до 0,2 – негативное влияние на проект;
- от 0,2 до 0,45 – оцениваемый вариант менее эффективен чем базовый;
- от 0,45 до 0,55 – оцениваемый вариант имеет незначительную эффективность, по причине схожести с базовым;
- от 0,55 до 0,75 – эффективное решение;
- от 0,75 до 1 – высоко эффективное решение.

Результаты присвоения весов факторам оценки сведём в таблицу 15. Данные показатели являются усреднёнными оценками экспертов

Матрица факторов оценки эффективности

Фактор	Комплексная оценка влияния на стоимость, экологию, технологи.
Протяженность	0,88
Высота средняя	0,62
Высота максимальная	0,68
Высота минимальная	0,53
Разница максимальной и минимальной высот	0,94
Количество переходов через крупные реки	0,5
Количество переходов через малые реки	0,5
Протяженность участка в сложных условиях, км.	0,68
Число углов поворота	0,58
Средний угол поворота	0,73
Максимальный угол наклона	0,9
Средняя удалённость от автодорог	0,4
Максимальная удаленность от автодорог	0,45
Участки, проходящие по лесам	0,61
Суммарное влияние	0,64

Полученные данные сведём в таблицу 16, где проанализируем полученные данные в зависимости от их влияния на стоимостные, экологические и технологические показатели.

Анализ эффективности по блокам.

	Стоимостные показатели	Экологическое воздействие	Технологические показатели
Протяженность	0,88		
Высота средняя			0,62
Высота максимальная			0,68
Высота минимальная			0,53
Разница максимальной и минимальной высот	0,94		0,94
Количество переходов через крупные реки	0,5	0,5	0,5
Количество переходов через малые реки	0,5	0,5	0,5
Протяженность участка в сложных условиях, км.	0,68		0,68
Число углов поворота			0,58
Средний угол поворота			0,73
Максимальный угол наклона	0,9		0,9
Средняя удалённость от автодорог	0,4		

Максимальная удаленность от автодорог	0,45		
Участки, проходящие по лесам		0,61	
Суммарное влияние по показателям	0,65625	0,536667	0,666
Заключение	Эффективно	Незначительная эффективность	Эффективно

Так, средняя общая оценка разработанного алгоритма составила 0,64, что характеризует его как эффективный. При разбивке показателей по блокам влияния на стоимостной, экологический и технологический, выявлен наибольший эффект на стоимостные и технологические параметры, при этом экологическая составляющая отличается не значительно от базового варианта.

В результате проведённого исследования можно заключить, что полученный новый маршрут пролегания нефтепровода от А месторождения до магистрального нефтепровода Восточная Сибирь – Тихий океан является более эффективным в стоимостных и технологических параметрах. Трассировка линейных объектов разрабатывается на этапе ТЭО, следовательно, наибольшее значение будут иметь факторы, влияющие на конечную стоимость проекта.

ЗАКЛЮЧЕНИЕ

В рамках данного исследования рассмотрена возможность применения разных алгоритмов оптимизации пролегания пути, выявлен наиболее подходящий алгоритм для проектирования маршрута пролегания трубопровода.

В результате построения оптимального маршрута пролегания нефтепровода «А месторождение – Б» был опробован способ применения разработанного сценарного языка алгоритма А*.

Итоговые данные продемонстрировали актуальность применения данного алгоритма, полученные результаты необходимо проанализировать более подробно с привлечением большего количества экспертов, так как качество экспертных оценок имеет значительное влияние на конечный результат при использовании методов нечетной логики.

Данная работа имеет высокую научную значимость, так как идентифицирует алгоритм А* как наиболее эффективный при работе с линейными объектами, в то время как на сегодняшний день наиболее распространённым является алгоритм Дейкстры.

На основе полученного метода возможно создание отдельного специализированного программного обеспечения, что позволит ускорить процесс обработки информации при работе с максимально полным объемом данных. Разработка отдельного приложения так же позволит облегчить внедрение программного продукта в проектные организации, у которых достаточно ресурса для накопления больших объемов данных, что увеличивает точно работы алгоритма.

Список использованной литературы

- 1) Федеральный закон от 27 июля 2010 г. N 210-ФЗ "Об организации предоставления государственных и муниципальных услуг"
- 2) Постановление Правительства РФ от 16.02.2008 №87 (ред. От 12.11.2016) «О составе разделов проектной документации и требования к их содержанию»
- 3) Министерство регионального развития Российской Федерации письмо от 20 мая 2011 г. № 13137-ИП/08
- 4) ГОСТ Р 21.1001-2009. Основные требования к проектной и рабочей документации. – М.: Национальные стандарты, 2009. – 51 с.
- 5) ГОСТ 28397-89. Языки программирования. Термины и определения. – М.: Издательство стандартов, 1989. – 8 с.
- 6) СНиП 2.05.06-85. Магистральные трубопроводы. – М.: ФГУП ЦПП, 2000. – 7 с.
- 7) СНиП Ш-42-80. Магистральные трубопроводы. – М.: ВНИИСТ, 1997. – 33 с.
- 8) СП 103-34-96. Свод правил сооружения магистральных трубопроводов. Подготовка строительной полосы. - М.: ИРЦ Газпром, 1996. – 13 с.
- 9) СП 104-34-96. Свод правил сооружения магистральных трубопроводов. Производство земляных работ. – М.: ИРЦ Газпром, 1996. – 33 с.
- 10) ОАО «Газпром». Стандарт организации. Магистральные газопроводы. СТО Газпром 2-2.1-249-2008: принят и введен в действие 26.08.08г. // СПС «Консультант плюс».
- 11) Алиев, Р.А. Трубопроводный транспорт нефти и газа / Р.А. Алиев, В.Д. Белоусов, А.Г. Немудров, В.А. Юфин, Е.И. Яковлев. – М.: Недра, 1988. – 235 с
- 12) Адигеев, М.Г. Анализ сложности параллельных алгоритмов. Методические указания для студентов факультета математики, механики и

компьютерных наук [Электронный ресурс] / М.Г. Адигеев. – Ростов-на-Дону: ЮФУ, 2013. – Режим доступа: <https://open-edu.sfedu.ru/pub/2416>. Дата обращения: 08.03.2017 г.

13) Баевский О.А. Правовые аспекты планировки территории размещения линейных объектов: московский опыт // Имущественные отношения в Российской Федерации. 2013. №6 (141). С. 83-95.

14) Барсукова Галина Николаевна, Радчевский Николай Михайлович Современные Проблемы управления земельными ресурсами // Научный журнал КубГАУ - Scientific Journal of KubSAU. 2017. №125. С.408-428

15) Басов, Е.Д. Технология и организация строительства переходов магистральных трубопроводов через малые преграды / Е.Д. Басов. - М.: МИНГП им. И.М. Губкина, 2004. – 109 с.

16) Бородавкин, П.П. Выбор оптимальных трасс трубопроводов большой протяженности / П.П. Бородавкин, Е.М. Сощенко, Б.И. Ким. – М.: ВНИИОЭНГ, 1977. – 57 с.

17) Бородавкин, П.П. Строительство магистральных трубопроводов в сложных условиях / П.П. Бородавкин, А.Х. Сунарчин. - М.: Недра, 1965. – 207 с.

18) Бочаров М.В., Королев Д.В. Оформление земельных участков под объектами недропользования и линейными объектами: актуальные проблемы и перспективы законодательного регулирования // Имущественные отношения в Российской Федерации. 2010. №11. С. 76-86.

19) Варгина Т.В. Разработка схем территориального планирования субъектов РФ (методический аспект) // Промышленное и гражданское строительство. 2008. №1. С. 30-32.

20) Жаркова О.А. Новый порядок размещения объектов трубопроводного транспорта и иных линейных объектов // Петербургский юрист. 2014. №3. С. 48-55.

21) Иванина, Д.С. Совершенствование проектирования линейной части магистральных трубопроводов с использованием математических методов и

средств вычислительной техники / Д.С. Иванина. - К.: ИК им. В.М. Глушкова, 1984. – 234 с.

22) Корсей, С.Г. Применение ГИС в трубопроводном транспорте [Электронный ресурс] / С.Г. Корсей, Р.Г. Прохожаев, И.А. Рыльский. – М.: ЗАО «АРКОН», 2008. – Режим доступа: <http://neftegaz.ru/science/view/265>. Дата обращения: 12.12.2016 г.

23) Корсей, С.Г. ГИС-технологии в трубопроводном транспорте [Электронный ресурс] / С.Г. Корсей, Н.Б. Дьякова. – М.: ЗАО «АРКОН», 2007. - Режим доступа: <http://neftegaz.ru/science/view/208>. Дата обращения: 17.02.2017 г.

24) Кузнецов, И.С. Поиск маршрута прокладки инженерных сетей с наименьшей стоимостью / И.С. Кузнецов, Р.Н. Кузнецов, А.А. Горских // Строительство и архитектура. – 2009. - № 4. – С. 12-16.

25) Курносов, М.Г. Амортизационный анализ (amortized analysis) [Электронный ресурс] / М.Г. Курносов. – Режим доступа: <http://www.mkurnosov.net/teaching/index.php/DSA/Fall2015>. Дата обращения: 28.01.2017 г.

26) Митягин С.Д. В помощь застройщику // Зодчий 21 век. 2014. №1(50). С. 25-27.

27) Митягин С.Д. Как упростить систему проектно-правового обеспечения инвестиционно-градостроительной деятельности // Зодчий 21 век. 2014. №3(52). С. 28-29.

28) Митягин С.Д., Клещельская М.В. Актуальные задачи разработки документации по планировке территории // Вестник гражданских инженеров. 2012. №5 (34). С. 28-33.

29) Молочко А. В., Жучков П. С. Возможности применения геоинформационных технологий в решении задачи оптимизации трассы магистрального газопровода (на примере Петровского района саратовской области) // Изв. Сарат. ун-та Нов. сер. Сер. Науки о Земле. 2013. №1. URL: <http://cyberleninka.ru/article/n/vozmozhnosti-primeneniya-geoinformatsionnyh->

tehnologiy-v-reshenii-zadachi-optimizatsii-trassy-magistralnogo-gazoprovoda-na-primere (дата обращения: 15.05.2017).

30) Рыльский, И.А. Оптимизация трасс трубопроводов с использованием ГИС-технологий / И.А. Рыльский // Вестник Московского Университета, серия География. – 2004. - № 4. - С. 34-41.

31) Рыльский, И.А. Геоинформационное обеспечение автоматизированного трассирования трубопроводов [Электронный ресурс] / И.А. Рыльский. – М.: МГУ им. М.В. Ломоносова, 2006. - Режим доступа: <http://www.dissercat.com/content/geoinformatsionnoe-obespechenie-avtomatizirovannogo-trassirovaniya-truboprovodov-0>. Дата обращения: 03.02.2016 г.

32) Сарычев Д.С., Скворцов А.В. Применение графовых моделей для анализа инженерных сетей // Вестник Томского гос. ун-та, 2002, Т. 273, апрель, с. 70-74.

33) Семенченко Е.А. О проектах планировки и межевания территорий, предусматривающих размещение линейных объектов // Юрист. 2012. №15. С. 6-15.

34) Серапинас, Б. Б. Математическая картография: учебник для вузов / Б. Б. Серапинас. - М.: Академия, 2005. - 339 с.

35) Скворцов А.В. Комплексное исследование и разработка эффективных вычислительно устойчивых алгоритмов вычислительной геометрии и их реализация в геоинформационной системе. Дисс. докт. техн. наук. — Томск, 2002.-307 с.

36) Тесленок Сергей Адамович, Тесленок Кирилл Сергеевич, Горелов Александр Владимирович Анализ динамики лесных ландшафтов административного района средствами геоинформационных технологий // Вестник ВолГУ. Серия 11. Естественные науки. 2015. №4. С.68-80

37) Алгоритм A* для новичков [Электронный ресурс]. – Режим доступа: http://www.policyalmanac.org/games/aStarTutorial_rus.htm. Дата обращения: 25.12.2016 г.

38) Кратчайшие пути в графах [Электронный ресурс]. - Режим доступа: http://neerc.ifmo.ru/wiki/index.php?title=%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D1%8F:%D0%9A%D1%80%D0%B0%D1%82%D1%87%D0%B0%D0%B9%D1%88%D0%B8%D0%B5_%D0%BF%D1%83%D1%82%D0%B8_%D0%B2_%D0%B3%D1%80%D0%B0%D1%84%D0%B0%D1%85. Дата обращения: 05.04.2015 г.

39) Расстояние Чебышёва [Электронный ресурс]. - Режим доступа: https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5_%D0%A7%D0%B5%D0%B1%D1%8B%D1%88%D1%91%D0%B2%D0%B0. Дата обращения: 14.04.2017 г.

40) Хеш-таблица [Электронный ресурс]. – Режим доступа: <http://kvodo.ru/hash-table.html>. Дата обращения: 19.11.2015 г.

41) ArcGIS Resources [Электронный ресурс]. – Режим доступа: <http://resources.arcgis.com/ru/help/main/10.1/index.html###/004700000053000000>. Дата обращения: 29.03.2017 г.

42) Chowdhury, Rezaul A. Analysis of Algorithms [Электронный ресурс] / Rezaul A. Chowdhury. – NY.: Stony Brook University, 2015. - Режим доступа: <http://www3.cs.stonybrook.edu/~rezaul/CSE548-S15.html>. Дата обращения: 09.09.2015 г.

43) Cormen, Thomas H. Introduction to Algorithms Third Edition Thomas / Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. – Ldn. : The MIT Press, 2009. – 1313 с.

44) Harabor, D. Clearance-based Pathfinding [Электронный ресурс] / D. Harabor. – Cbr.: The Australian National University, 2009. – Режим доступа: <https://harablog.wordpress.com/2009/01/29/clearance-based-pathfinding/>. Дата обращения: 13.10.2016 г.

45) Harabor, D. Fast Pathfinding via Symmetry Breaking [Электронный ресурс] / D. Harabor. – Cbr.: The Australian National University, 2011. – Режим доступа: <https://harablog.wordpress.com/2011/08/26/fast-pathfinding-viasymmetry-breaking/>. Дата обращения: 02.10.2016 г.

46) Harabor, D. Jump Point Search [Электронный ресурс] / D. Harabor. – Cbr.: The Australian National University, 2011. – Режим доступа: <https://harablog.wordpress.com/2011/09/07/jump-point-search/>. Дата обращения: 02.10.2016 г.

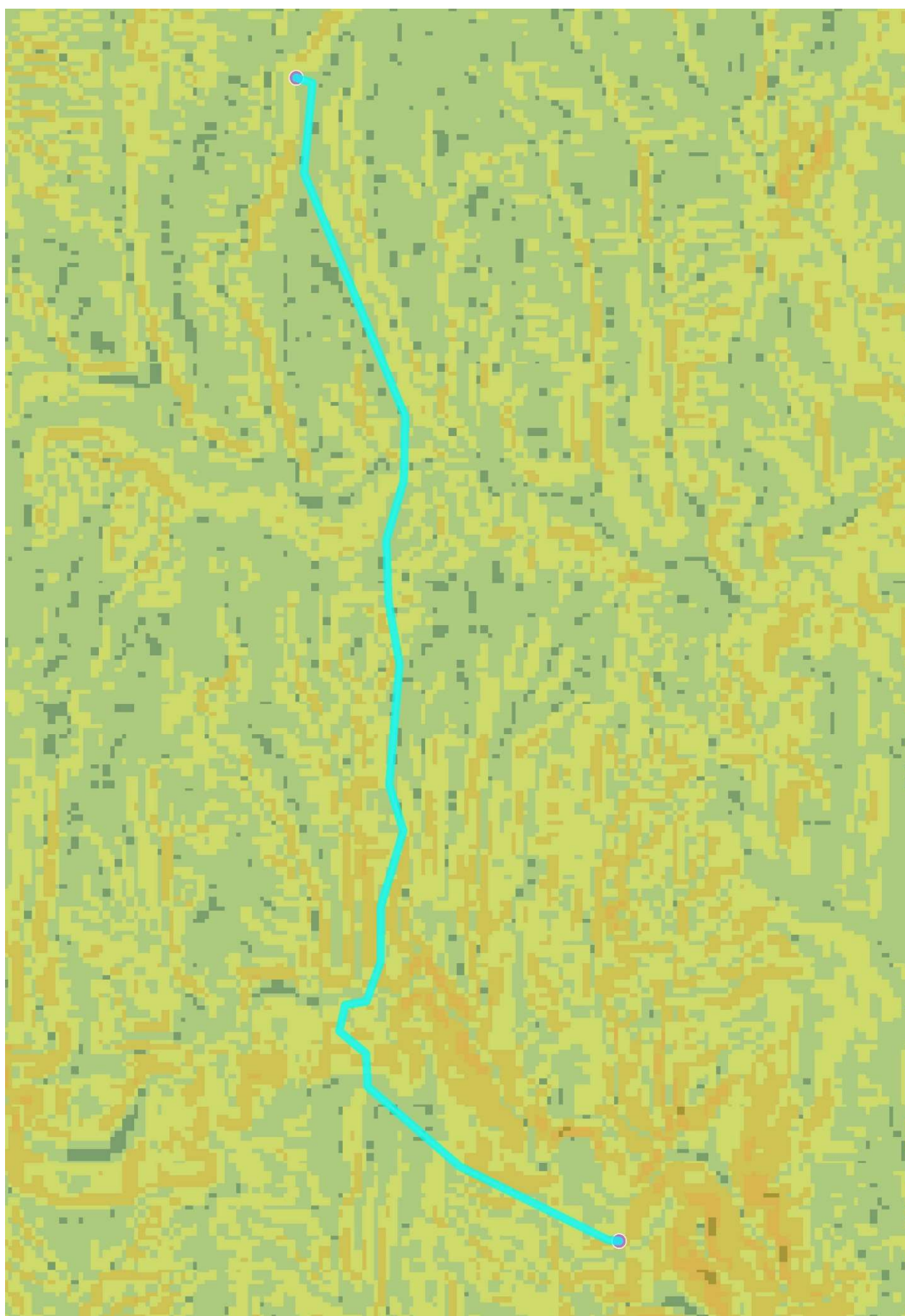
47) Harabor, D. Rectangular Symmetry Reduction [Электронный ресурс] / D. Harabor. – Cbr.: The Australian National University, 2011. – Режим доступа: <https://harablog.wordpress.com/2011/09/01/rectangular-symmetry-reduction/>. Дата обращения: 02.10.2016 г.

48) Heuristics and A* Pathfinding [Электронный ресурс]. – Режим доступа: <http://www.policyalmanac.org/games/heuristics.htm>. Дата обращения: 16.10.2015 г.
71 Incremental heuristic search [Электронный ресурс]. - Режим доступа: http://en.wikipedia.org/wiki/Incremental_heuristic_search. Дата обращения: 12.05.2017 г.

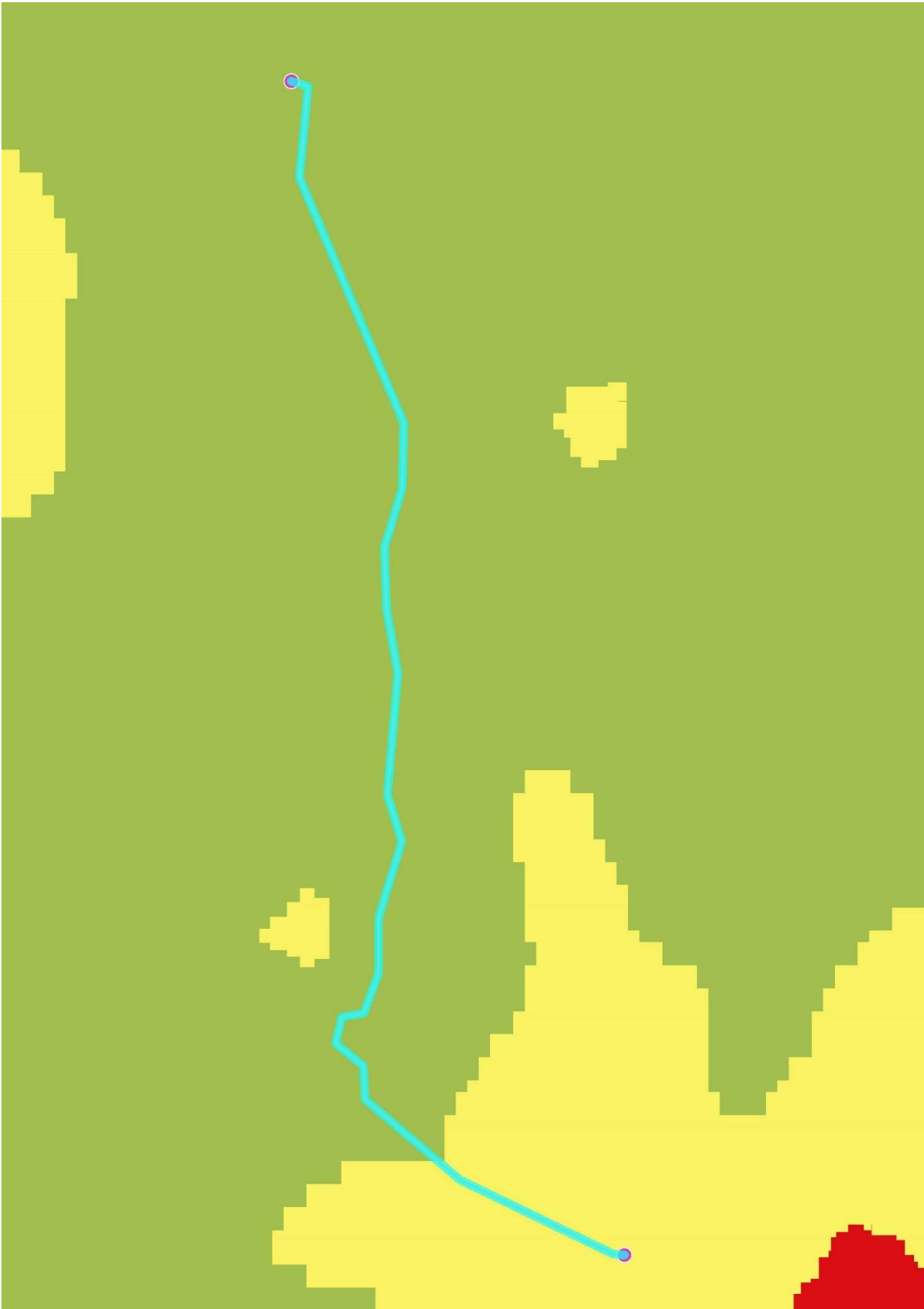
49) Taxicab geometry [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Taxicab_geometry. Дата обращения: 24.04.2017 г.

50) Sun, X. Generalized Adaptive A*S / X. Sun, S. Koenig, W. Yeoh // In Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS). – 2008. – С. 469-476.

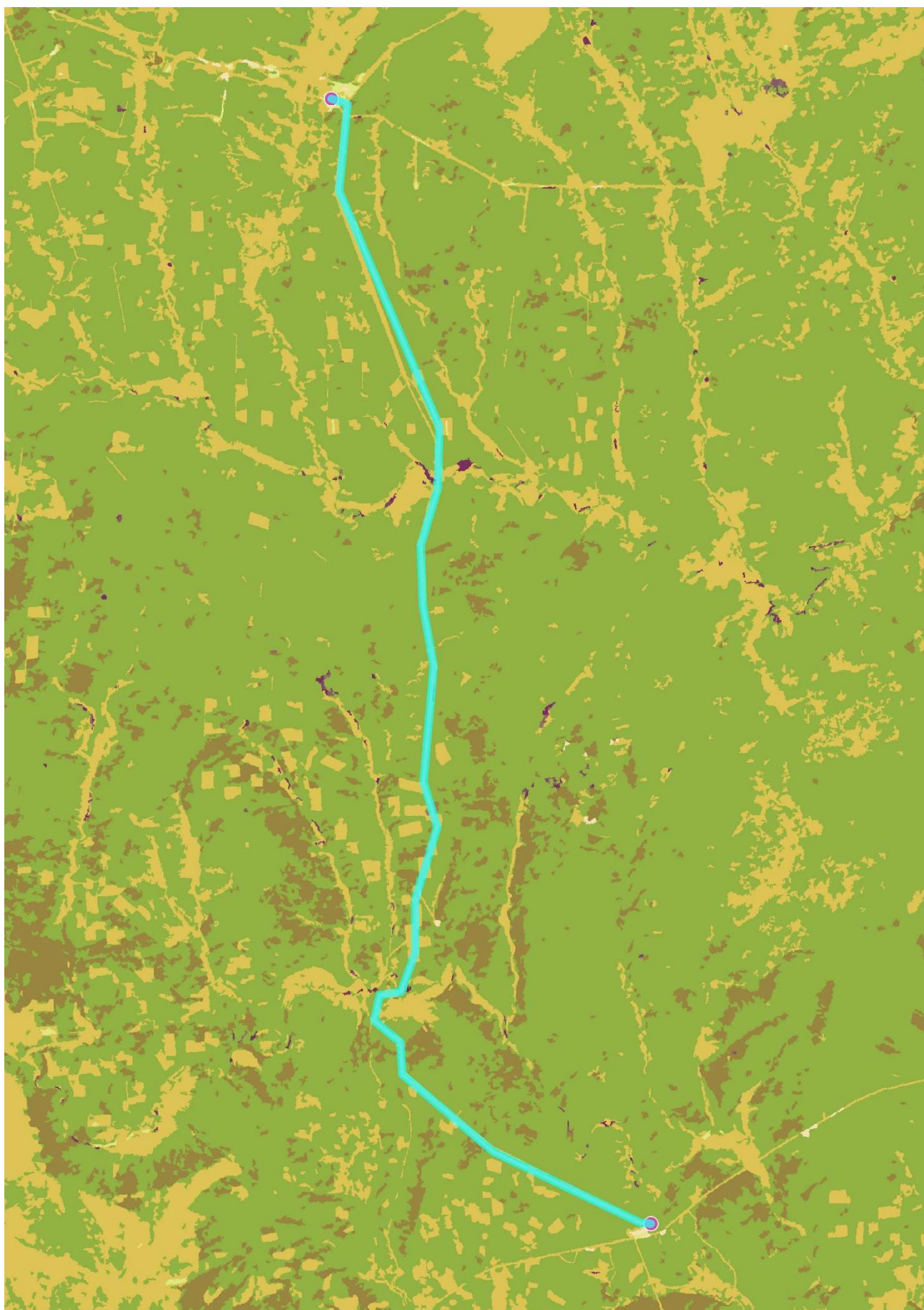
Растр факторов рельефа



Растр доступности земель



Растр инфраструктурных и технологических факторов



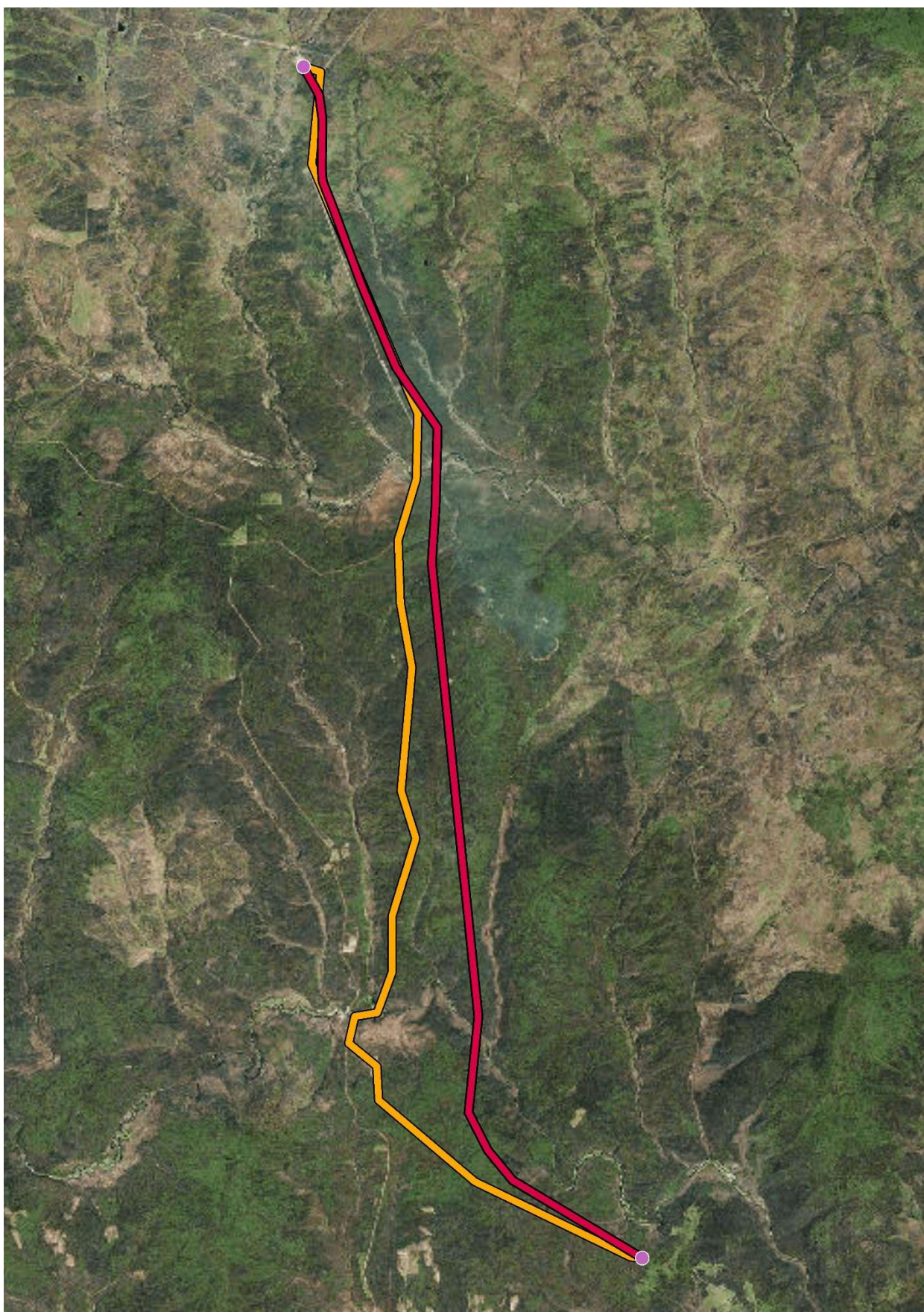
Растр экологических факторов



Картографическая основа проектирования



Нефтепровод существующий и построенный по алгоритму A*



Научное издание

Э. В. Батоева

Д.Ю. Мельников

**ПРОЕКТИРОВАНИЕ ОПТИМАЛЬНОГО МАРШРУТА
ПРОЛЕГАНИЯ ЛИНЕЙНЫХ ОБЪЕКТОВ**

Монография

*Под редакцией
доктора экономических наук
профессора Плотникова А.Н.*

Подписано в печать: 15.09.2018г.

Формат 60 x 84

1/16

Бумага офсетная

Усл. печ. л. 6,8

Тираж 500 экз.

Изготовлено в полиграфической
компании «Эль-Принт»