

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
"Пензенский государственный университет
архитектуры и строительства"
(ПГУАС)

К.С. Подшивалова

**ДОСТАВКА
МЕЛКОПАРТИОННЫХ ГРУЗОВ
АВТОМОБИЛЬНЫМ ТРАНСПОРТОМ**

Пенза 2014

УДК 656.135.073

ББК 39.3

П44

Рецензенты: декан автомобильно-дорожного института, доктор технических наук, профессор Ю.В. Родионов (ПГУАС); кандидат технических наук, доцент кафедры «Транспортные машины» А.И. Звижинский (ПГУ).

Подшивалова К.С.

П44 Доставка мелкопартионных грузов автомобильным транспортом: моногр. / К.С. Подшивалова. – Пенза: ПГУАС, 2014. – 124 с. ISBN 978-5-9282-1192-9

Изложены исследования перевозочного процесса доставки мелкопартионных грузов потребителям от нескольких производителей. Рассматривается существующая система организации перевозок мелкопартионных грузов автомобильным транспортом и совершенствование методики организации этих перевозок.

Монография подготовлена на кафедре «Организация и безопасность движения» и предназначена для научных и инженерно-технических работников, студентов вузов, а также аспирантов интересующихся методами маршрутизации транспорта.

ISBN 978-5-9282-1192-9

© Пензенский государственный университет архитектуры и строительства, 2014

© Подшивалова К.С., 2014

ПРЕДИСЛОВИЕ

Дальнейшее развитие автомобилизации страны значительно увеличивает возможности общества, но ставит и новые сложные задачи, связанные с улучшением, как состояния окружающей среды, так и условий жизнедеятельности человека. При переходе экономики страны на рыночные отношения сокращаются жизненные циклы изделий, уменьшаются размеры партий груза, формируется рынок автотранспортных услуг на внутригородских перевозках, возрастает объем мелкопартионных перевозок, наблюдается рост числа торговых точек и предъявляются более жесткие требования к доставке грузов согласно условию «точно в срок». Особенностью таких перевозок является большое количество пунктов реализации на территории населенных пунктов, а также широкий ассортимент, для поддержания которого завоз продукции осуществляется от разных производителей. Вопросами доставки мелкопартионных грузов занимались многие советские и зарубежные ученые, и в настоящее время проводятся исследования, однако до сих пор уровень организации данных перевозок недостаточно эффективен. Поэтому необходим поиск новых научно-практических решений, методов и моделей оптимизации перевозочного процесса. В связи с этим исследования в данной области являются актуальными.

Проведя анализ деятельности автомобильного транспорта, начиная с прошлого столетия, можно сделать вывод, что для практики было достаточно положений, разработанных в тридцатые годы. Работа основывалась в основном на опыте инженеров и экономистов. Однако в современных условиях возникает необходимость совершенствования теории транспортных процессов, и в частности, мелкопартионных перевозок.

ВВЕДЕНИЕ

Автомобильный транспорт занимает одно из ведущих мест в общей транспортной системе страны. При различном государственном устройстве, а также типе экономики, основной задачей транспорта было и остается своевременное и качественное удовлетворение потребностей, как предприятий и организаций, так и населения [76]. Так как в городах возрастает объем перевозок грузов, то для обеспечения наиболее рационального использования подвижного состава, а также сокращения транспортных затрат, большое значение имеет задача маршрутизации [49].

Решение данного вопроса необходимо для эффективного управления перевозочным процессом. Если ранее при доставке различных грузов критерием эффективности было выполнение плана, то в современных условиях целью является доставка груза нужного качества и количества в заданное место, в назначенное время и с минимальными затратами.

Традиционные методы управления уже не оправдывают себя. Для обеспечения вышеуказанных требований необходим комплексный логистический подход к организации перевозочного процесса. Ведь в условиях жесткой конкуренции любое предприятие должно быть нацелено не только на получение прибыли, но и полностью удовлетворять потребителя, т.е. оно должно гибко реагировать на спрос.

Если говорить о системном подходе к организации, то транспорт является ключевой частью единой системы, и от того, насколько он эффективно работает, во многом зависит и экономика предприятия. Практика работы автомобильного транспорта показывает, что значительная часть объема перевозок грузов в различных отраслях осуществляется мелкими отправлениями. Следует заметить, что изменилась и структура торговой сети, и, следовательно, на рынке товаров и услуг наблюдается все возрастающая конкурентная борьба.

1. СОВРЕМЕННОЕ СОСТОЯНИЕ ТЕОРИИ И ПРАКТИКИ ПЕРЕВОЗОК ГРУЗОВ ПО РАЗВОЗОЧНЫМ МАРШРУТАМ

1.1. Обзор научных исследований в области перевозок грузов мелкими партиями

Для эффективного управления перевозочным процессом необходимо решение задач маршрутизации. Однако в настоящее время в новых условиях хозяйствования на автомобильном транспорте нет четко выстроенной теории для решения транспортных задач на уровне маршрутов [5, 76]. Исходя из технологии выполнения, следует различать помашинные и мелкопартионные перевозки.

При выполнении перевозок грузов помашинными отправками различают маятниковые, кольцевые и радиальные схемы маршрутов. При доставке грузов мелкими партиями маршруты бывают развозочные, сборные и развозочно-сборные, которые, в свою очередь, являются разновидностью кольцевого маршрута.

Выбор схемы маршрута движения при перевозке грузов зависит от: размещения пунктов погрузки и разгрузки, от размера партии, требований и условий поставки, грузоподъемности подвижного состава и расположения автотранспортных предприятий [21].

На маятниковом маршруте перевозочный процесс осуществляется между двумя пунктами: пунктом погрузки и пунктом разгрузки. В прямом направлении происходит движение автомобиля с грузом, а в обратном – без груза, т.е. холостой пробег. Наиболее эффективное использование транспортного средства на маятниковом маршруте, когда оно движется с грузом в обоих направлениях, так как холостой пробег отсутствует. Также возможна схема движения на маятниковом маршруте с частично груженым обратным пробегом [20, 21].

Вследствие того, что все больше и больше появляются частные перевозчики для доставки массовых грузов помашинными отправками, маятниковые маршруты получили широкое применение. Ими мало используются экономические и математические методы планирования транспортных схем. Однако распространенное применение маятниковых маршрутов не всегда положительно сказывается на себестоимости перевозок.

Кольцевые маршруты проектируют, когда доставку груза не возможно организовать по маятниковому маршруту с гружеными пробегами в обоих направлениях [21]. Благодаря этому исключаются нерациональные холостые пробеги, так как кольцевой маршрут представляет собой замкнутый контур с одним пунктом начала и конца. Схема кольцевого маршрута может иметь различный вид в зависимости от набора пунктов разгрузки и погрузки и их расположения.

При доставке грузов по радиальным маршрутам погрузка осуществляется в одном центральном пункте и развозится по нескольким периферийным или наоборот. Организация работы автомобилей на таком маршруте гораздо сложнее, чем на кольцевом и тем более на маятниковом, вследствие пересечения грузовых потоков в центральном пункте [76].

Данная работа посвящена маршрутизации перевозок мелкопартионных грузов. На автомобильном транспорте мелкопартионными грузами считаются партии весом от 10 до 2000 кг [14]. Количество мелкопартионных перевозок увеличивается вследствие их широкого применения для доставки социально-значимых грузов, продовольственных товаров, грузов сферы бытового обслуживания, почты и т.д. Под мелкой отправкой подразумевается груз, предъявляемый к единовременной перевозке в один адрес, который не обеспечивает полную загрузку автомобиля, используемого для его доставки [18]. Поэтому перевозка мелкопартионных грузов должна осуществляться по развозочным маршрутам.

В последнее время наблюдается рост появления торговых точек, а это обстоятельство вызывает увеличение объема перевозок мелких партий грузов, т.е. данный вид доставки наиболее характерен для торговли [16]. Исходя из этого, можно утверждать, что с увеличением спроса на мелкопартионные перевозки автомобильный транспорт занимает устойчивое положение на рынке транспортных услуг. Поэтому для доставки торговых грузов разнообразной номенклатуры, которые формируются наибольшими партиями, широкое применение получили развозочные и сборочные маршруты.

Развозочные маршруты, как правило, используются для внутригородского обслуживания. Их типичным примером, как уже было сказано выше, являются перевозки продовольственных товаров в торговую сеть. Для такого рода грузов, как правило, используются автомобили-фургоны, хотя наряду с ними могут использоваться и бортовые автомобили. Обычно применяются автомобили-фургоны с массой отправки до двух тонн, что составляет более 40% от объема перевозок [26]. Однако следует добавить, что такие маршруты получили распространение и в междугороднем сообщении, вследствие их экономичности. Доставка продовольственных товаров мелкими партиями имеет ряд характерных особенностей. Во-первых, эти товары отличаются от других грузов сроками реализации (хлеб, молоко и т.д.). Во-вторых – сложностью задачи планирования, вследствие большого числа получателей, партионностью и высокой себестоимостью перевозок.

Актуальность применения развозочных маршрутов очевидна. Поэтому вопросами их совершенствования, а также экономического обоснования занимались и занимаются многие российские и зарубежные ученые: А.И. Воркут, Б.Л. Геронимус, В.А. Гудков, А.В. Вельможин, Л.Б. Миротин, М. Кристофидес, С. Эйлон, Т. Гаскель, В.А. Житков, К.В. Ким, Дж. Литтл

и др. Основы современной теории транспортного процесса были заложены С.Р. Лейдеманом, Л.Л. Афанасьевым и др. [27, 45, 59, 110 и др.]. Они были использованы в последствии в научных трудах многих авторов и специалистов-практиков для анализа производительности автомобиля [3, 58, 94 и др.]. Следует заметить, что теория, созданная Лейдерманом С.Р. предназначена для описания работы одного автомобиля на маятниковом маршруте с обратным холостым пробегом, но она не учитывала особенностей работы автомобиля на других типах маршрутов. Исходя из этого, возникает необходимость совершенствования теории транспортного процесса доставки грузов мелкими отправками. Перевозка мелких партий груза является дорогостоящим процессом. В работе [41] авторы показали, что на перевозку массовых грузов транспортные затраты сравнительно ниже, чем на доставку мелких партий грузов. Их исследования показали, что мелкопартионные перевозки составляют около 3% от общей транспортной работы. Однако, транспортные затраты на эти перевозки колеблются в пределах 35% от всей суммы. Нерациональный выбор маршрутов движения ведет к перепробегу автомобилей, который может достигать более 30%. При этом транспортные затраты на перевозку мелкопартионных грузов составляют порядка 25% от общей суммы затрат [1, 41, 42, 132, 133].

В своих научных трудах многие ученые рассматривали различные факторы, влияющие на эффективность работы подвижного состава. Так в работах [11, 12, 40, 89, 99] акцент ставится на создание новых технологий перевозок мелкопартионных грузов. Работа [91] посвящена совершенствованию критерия «функции выгоды» при маршрутизации данных перевозках. Несколько другие подходы, а именно, учитывающие производство, накопление и хранение мелких партий груза рассматривались в работах [22, 61].

Профессором В.И. Николиным была создана классификация автотранспортных систем [76]. В данном случае маршруты движения рассматриваются в виде системы. Согласно предложенной классификации следует различать семь их типов, в соответствии с их иерархическим расположением. Развозочные маршруты, с точки зрения системного подхода, имеют более сложную структуру. Здесь наблюдается комбинация нескольких кольцевых, либо кольцевых и маятниковых маршрутов с центральными или несколькими пунктами погрузки. Также в данной работе выявлены закономерности влияния технико-эксплуатационных показателей на эффективность систем. Автор различает помашинные и мелкопартионные перевозки по многим особенностям. Однако выполненное им исследование и полученные результаты характерны только для помашинных перевозок.

Работа [18] посвящена развозочно-сборным автотранспортным системам. Автор является последователем профессора В.И. Николина, и указал некоторые замечания по поводу вышеописанной работы. Речь идет о том,

что созданная классификация недостаточно отражает системы доставки грузов мелкими партиями. Однако при работе в них автомобили осуществляют значительное количество перевозок в городах. Поэтому автор счел необходимым восполнить пробел в теории и разработал классификацию развозочно-сборных автотранспортных систем, а также представил транспортный процесс доставки мелких партий груза, как систему с дискретным состоянием.

В работе [104] А.Ю. Тюрина предлагается использование коэффициента развозочного маршрута $K_{\text{разв}}$, который определяется следующим способом:

$$K_{\text{разв}} = \frac{L_{\text{маят}}}{L_{\text{кольц}}}, \quad (1.1)$$

где $L_{\text{маят}}$ – пробег на маятниковом маршруте, км;

$L_{\text{кольц}}$ – пробег на кольцевом маршруте, км.

В ходе исследований была получена зависимость между коэффициентом кольцевого маршрута и количеством пунктов развоза Π_3 :

$$K_{\text{кольц}} = \frac{\Pi_3}{a\Pi_3 + b}, \quad (1.2)$$

где a, b – коэффициенты пропорциональности.

Однако вызывает сомнение дробно-линейный вид зависимости коэффициента от количества пунктов, а также степень точности расчета при использовании данного коэффициента.

Большое внимание в развитии теории транспортного процесса доставки грузов мелкими партиями уделено исследованию технико-эксплуатационных показателей. Они в разной степени влияют на производительность автомобиля, которая в свою очередь является одним из основных обобщающих показателей эффективности использования подвижного состава [4, 14, 18, 20, 21, 22, 76].

Классическими критериями оценки эффективности маршрутизации являются пробег и транспортная работа. А транспортная работа АТС, выполняемая в единицу времени, представляет собой ни что иное, как производительность. Эти критерии просты и удобны в использовании. Кроме того, все статьи эксплуатационных расходов на транспорте включают пробег автомобиля. На практике принято оценивать выработку автомобиля в тоннах и тонно-километрах. Однако в своих работах [20–22] профессор А.И. Воркут утверждает, что на развозочных маршрутах выработка в тонно-километрах не является критерием эффективности, так как величина грузооборота при таком виде доставки определяется последовательностью объезда пунктов завоза (вывоза) и грузоподъемностью, от которой зависит

количество пунктов в маршруте. Следовательно, нужно использовать часовую производительность. Проведя экспериментальные исследования, автор указал на то, что для развозочных маршрутов коэффициент использования пробега, а также динамический коэффициент использования грузоподъемности не могут быть приняты в качестве показателей для выбора оптимального маршрута.

Проанализируем классическую формулу часовой производительности автомобиля [12, 13, 58, 59]:

$$W_{ar} = \frac{q\gamma_c V_T \beta_e}{L_{er} + t_{пр} V_T \beta_e}, \quad (1.3)$$

где W_{ar} – часовая производительность, т/ч;

q – номинальная грузоподъемность, т;

γ_c – статический коэффициент использования грузоподъемности;

V_T – техническая скорость автомобиля, км/ч;

L_{er} – длина ездки с грузом, км;

$t_{пр}$ – время простоя автомобиля под погрузкой и разгрузкой, ч;

β – коэффициент использования пробега.

Данное выражение можно привести к виду:

$$W_{ar} = q\gamma_c \left(\frac{1}{L_0/V_T + t_{пр}} \right), \quad (1.4)$$

где L_0 – общий пробег автомобиля, км.

Чем больше значение коэффициента использования грузоподъемности, тем выше производительность автомобиля. Коэффициента γ_c напрямую зависит от класса перевозимого груза. Увеличить коэффициент γ_c можно при помощи следующих мероприятий [14]:

2. подвижной состав должен соответствовать условиям перевозок;
3. сортировка и укрупнение мелких партий;
4. груз должен тщательно укладываться в кузове;
5. применение автомобилей с увеличенным объемом кузова;
6. наращивание бортов.

Время, затрачиваемое на погрузочно-разгрузочные работы $t_{пр}$, зависит от степени механизации проведения данной операции.

Поэтому при планировании сборно-развозочных маршрутов оперируют лишь с величинами количества перевезенного груза q , общего пробега L_0 и технической скорости V_T . Справедливо отметить, что величина технической скорости существенно влияет на производительность, так как возрастает выработка и уменьшается требуемое количество подвижного

состава. Однако в условиях реальной транспортной сети имеют место задержки, как следствие нестабильной пропускной способности. Следует добавить, что и превышение допустимой скорости невозможно.

Исходя из этого, по всему маршруту движения принимают величину технической скорости постоянной.

Из сказанного выше следует, что производительность автомобиля является функцией от количества перевезенного груза и общего пробега:

$$W_{ar} = f(qL_0). \quad (1.5)$$

Следует заметить, что эффективность организации перевозок зависит не только от производительности автомобиля, но и его снаряженной массы. Чем она меньше, тем меньше затрат приходится на перевозку одной тонны груза. В настоящее время стоимость топлива резко возросла и затраты на передвижение самого автомобиля составляют значительную долю от общих затрат. Однако в теории автомобильных перевозок это учитывается лишь косвенно при помощи коэффициента β .

Организацию перевозочного процесса следует осуществлять из условия минимизации нерациональных холостых пробегов, которые в свою очередь, оказывают влияние на величину общего пробега автомобиля. Одним из путей повышения эффективности развозочных маршрутов может быть доставка попутных партий груза от нескольких производителей, то есть совместная развозка грузов, совместимых по условиям перевозок. Это позволит снизить величину взаимных холостых пробегов и повысить производительность труда в целом. Однако теория этих маршрутов не до конца разработана. В частности не исследован вопрос о критерии совместимости, когда целесообразно объединять маршруты в один. Кроме того, не разработана методика определения пункта заезда за попутным грузом. Он должен определяться при решении задачи маршрутизации. Следует отметить, что заезд за попутным грузом может создавать на маршруте подциклы. Поэтому для расчета таких маршрутов необходимо использовать методы их учитывающие.

В настоящее время организация развозочных маршрутов в основном производится с целью максимально снизить протяженность перевозки. Но при этом не учитывается количество завозимого (отправляемого) груза, а это, в свою очередь, приводит к повышению транспортной работы.

1.2. Анализ методов решения задач маршрутизации при перевозке мелких партий грузов

Организация перевозок мелкопартионных грузов подразумевает решение задач, которые относятся к конечным оптимизационным задачам. Однако, их решение сложнее, чем при осуществлении помашинных отправок.

В общем виде, задачу маршрутизации можно сформулировать следующим образом [21, 81, 92]: задана дислокация грузоотправителей и грузополучателей; объемы вывоза и завоза грузов; подвижной состав; транспортная сеть, а также условия движения по ней. В итоге необходимо найти такие упорядоченные множества пунктов, связанных между собой, которые и формируют маршруты, а доставка грузов по ним ведет к достижению оптимального значения целевой функции. Следует добавить, что должны выполняться требования к организации транспортного процесса, учитывающие фактор времени. На развозочных (сборных) маршрутах, когда партия груза отгружаемая в i -й пункт (из i -го пункта), меньше фактической загрузки автомобиля, задача маршрутизации представляет собой сложную математическую проблему, которую необходимо решать, применяя методы математического моделирования. Она сводится к определению набора пунктов, входящих в цикл перевозок, а также определению последовательности их объезда. Математическая задача определения кольцевого маршрута известна в двух постановках: как «задача коммивояжера» и «задача развозки».

Задача «странствующего коммивояжера» [49, 51, 62, 72, 82, 92] заключается в том, что имеется n городов. Коммивояжер выезжает из одного из них и объезжает все города с условием побывать в каждом из них только один раз. Расстояние между городами не одинаково, поэтому каждая последовательность городов дает разное суммарное расстояние пробега. Из всех этапов необходимо найти такой, у которого эта сумма была бы минимальна.

В отличие от «задачи коммивояжера», когда для объезда всех пунктов должен быть построен только один маршрут движения, в «задаче развозки» строятся несколько развозочных маршрутов, замкнутых у одного отправителя [40].

Оптимальным считается маршрут, на котором достигается критическое значение целевой функции в зависимости от поставленной цели. Целевая функция, в свою очередь, должна описывать изменение основного критерия оптимизации, в качестве которого могут выступать: пробег автомобиля, время движения, транспортная работа, транспортные расходы и т.д.

Планирование маршрутов должно осуществляться с условием, что грузоподъемность автомобиля не должна превышать, а суммарная длина всех маршрутов была бы минимальной.

Согласно [40, 41], при организации мелкопартионных перевозок необходимо использовать различные приемы, учитывая выбранный критерий оптимальности.

Получение оптимального значения целевой функции возможно посредством использования точных методов [28, 40, 41, 50, 62, 106], к которым относятся: метод «ветвей и границ», целочисленного линейного и ди-

намического программирования. Следует заметить, что точные методы предполагают большой перебор вариантов. Однако память и быстродействие персональных компьютеров (ПК) быстро растут. Поэтому альтернативы точным методам в будущем нет.

В работах [38–41, 44, 121] описаны способы приближенных решений, которые можно подразделить на методы: локальной оптимизации, случайного поиска и эвристические, которые базируются на материалах опыта прошлых решений.

Использование приближенных методов дало возможность решать задачи в допустимые сроки с различными ограничениями. Из всех допустимых решений они дают достаточно хорошие, но не обязательно лучшие из них.

Однако следует заметить, что в современных условиях жесткой конкуренции актуальным является использование точных методов, дающих оптимальное решение, так как у приближенных методов погрешность составляет более 10% , которую очень сложно преодолеть.

При решении задачи маршрутизации, с целью получения точного решения, впервые была сделана попытка использования динамического метода [6, 7, 109].

Динамическое программирование представляет собой математический метод оптимизации, который позволяет планировать многоэтапные процессы, зависящие от времени [49, 66, 127]. Определение кратчайших расстояний на заданной транспортной сети относится к наиболее типичным задачам данного метода.

Суть динамического программирования заключается в следующем. Вся цепочка вычислений разбивается на $n + 1$ этапов (n – количество пунктов разгрузки). На каждом k -том этапе рассматривается пункт, причем его номер соответствует номеру этапа. Затем подсчитывается оценка (функция состояния) f_{k-1} для каждой дуги, которая выходит из этого пункта. Выбирается минимальное значение оценки из всех. Комбинация вершин, согласно выбранной дуге, должна удовлетворять условиям:

- только одна дуга выходит из вершины;
- только одна дуга выходит в вершину;
- отсутствие подциклов на рассматриваемом участке маршрута.

При нарушении вышеуказанных условий производится возврат на один этап назад и принятая на этом этапе дуга $(k - 1) - j$ игнорируется. Следующая дуга выбирается из условия, что $f_{(k-1)-i} \geq f_{(k-1)-j}$ ($i \neq j$). Переход на один этап вперед происходит, когда хотя бы для одной дуги выполняются вышеуказанные условия. Когда процесс достигает $n+1$ этап, то расчет

заканчивается. В общем виде функциональное уравнение для данной задачи будет следующим [49]:

$$f_i = \min(\ell_{ij} + f_j), \quad (1.6)$$

где ℓ_{ij} – расстояние между пунктами i и j ;

f_j – функция состояния (min расстояние) от пункта j к конечному пункту.

Согласно алгоритму метода динамического программирования анализ вариантов начинается с конца процесса.

Достоинством данного метода является многоэтапность решения задачи. Тем самым сложная многовариантная задача оптимизации сводится к совокупности более простых задач [49].

Недостатком метода динамического программирования является отсутствие общего алгоритма решения, пригодного для всех задач. Он дает лишь общее направление решения. Также наблюдается значительная трудоемкость при решении задач большой размерности. [40,42]. Алгоритм позволяет находить оптимальное значение лишь для задач с ограниченной размерностью.

В работах [1, 48, 49, 115, 130, 131] рассматривается еще один метод достижения точного решения «задачи коммивояжера». Здесь задача маршрутизации решается путем целочисленного программирования. Целочисленность в математическом программировании означает получение решения только в целых числах [49]. Суть данного метода заключается в том, что система линейных уравнений дополняется ограничениями. Это обеспечивает отсутствие подциклов и целочисленность переменных.

В работе [124] рассматривается несколько интерпретированный, модифицированный метод. В этом случае в исходную систему уравнений вводятся дополнительные ограничения, которые не позволяют маршрутам пересекаться. Это ведет к повышению сходимости, а также к уменьшению числа вариантов. Применение такой модификации метода при решении практических задач не всегда может быть целесообразным. Данный метод используется для набора пунктов в маршрут, а порядок их объезда целесообразнее определять с помощью других методов.

Метод «ветвей и границ» был предложен Дж. Литлом, К. Мурти, Д. Суини и К. Кэрлом [49, 62] и получил широкое распространение для решения «задачи коммивояжера» [15, 107, 113, 114, 116]. Он является одним из точных методов для решения задач маршрутизации и сводится к нахождению целевой функции в виде:

$$L = \sum_{i=1}^n \ell_i, \quad (1.7)$$

где n – количество ветвей;

ℓ_i – длина ветви между вершинами.

Расчет по данному методу довольно прост и заключается в нижеследующем. На первом этапе для всего множества предполагаемых решений находится нижняя граница значения функционала, больше которой величина целевой функции быть не может.

По минимальному расстоянию выполняется приведение исходной матрицы. Оно заключается в следующем. В матрице определяются минимальные элементы, так называемые константы приведения, которые вначале вычитаются из каждого элемента данной строки, а затем из каждого элемента данного столбца. Нижней границей функционала и будет являться сумма этих констант приведения.

В приведенной матрице в каждой строке и каждом столбце имеется хотя бы один нулевой элемент. Сначала рассматриваются эти элементы, поскольку клетка с нулевым расстоянием означает минимальный маршрут по протяженности. Затем для всех нулевых элементов определяется оценка A_{i-j} , которая показывает возможные потери, если этот элемент не будет включен в матрицу. Поэтому в первую очередь в матрицу включают нулевой элемент с наибольшей оценкой. Далее на каждом шаге множество разбивается на два не пересекающихся подмножества, для каждого из которых также определяется нижняя граница решения и выбирается из них наименьшее.

При этом одно подмножество образует маршруты, которые включают звено графа $K - S$, тогда:

$$L_{K-S} = L(U), \quad (1.8)$$

где $L(U)$ – нижняя граница.

А другое – маршруты, которые это звено исключают:

$$L_{\overline{K-S}} = L(U) + A_{K-S}. \quad (1.9)$$

В результате вычислений образуется так называемое «дерево решений», которое имеет в каждом узле графа два разветвления.

При прохождении до конца ветви получается матрица размером 2×2 , и недостающая пара дуг определяется однозначно.

Путь является оптимальным, если его длина не превышает нижнюю границу любого из неразбитых подмножеств.

Если у какого-либо неразбитого подмножества нижняя граница меньше, чем длина найденного маршрута, то решение далее выполняется с того подмножества, которое имеет минимальную нижнюю границу. После прохождения по всем ветвям «дерева решений» процесс вычислений заканчивается. В одной из последних работ [90] метод «ветвей и границ» был положен в основу оптимизации последовательности объезда пунктов всех возможных комбинаций маршрутов движения грузового автомобиля. В данной работе автор для решения задачи распределения в сочетании с ме-

тодом «ветвей и границ» использует процедуру рекурсии. Следует заметить, что использование точных методов гарантирует получение оптимального решения.

Одним из недостатков метода «ветвей и границ», как и у некоторых других, является не учет подциклов – звеньев, замкнутых на себя. Это не всегда соответствует практике развозки ввиду специфичности планировки городов, когда подразумевается неоднократный проезд по одним и тем же улицам – ветвям графа.

Другим слабым местом метода является его трудоемкость при решении задач большой размерности для полностью связанного графа. Однако, улицы современных городов и транспортная сеть их отражающая, представляет разреженный, а не полностью связанный граф. Более того, чем больше размерность графа, тем он может быть менее связан. Поэтому затраты времени на решение будут гораздо меньше, чем у связанного графа. Более того, быстродействие и память ПК быстро растут из года в год.

В работе [91] указано, что в 1985 году для решения «задачи коммивояжера» из шести пунктов затраты времени составили 3 минуты 16 секунд, а из 25 пунктов – 16 часов 17 минут. В настоящее время расчет маршрута транспортной сети одного из районов г. Пензы из 50 пунктов занимает 2 секунды (рис.1.1).

Приближенные методы дают из всех допустимых решений достаточно хорошие, но не обязательно лучшие решения. Как уже упоминалось раньше, они могут быть случайного поиска, локальной оптимизации и эвристические.

Методы случайного поиска характерны тем, что при сравнительно низких затратах времени на формирование отдельного решения и сравнительно большом времени процедуры вычисления возможно получить, если не лучшие, то достаточно хорошие решения. Появление методов случайного поиска вызвано нестабильностью спроса, а также, как следствие, нарушением периодичности завоза груза клиентам. Для реализации данных методов необходима статистическая обработка всех поступающих заявок клиентуры и выявление закономерностей. При этом заявки клиентов рассматриваются как случайные величины. Далее производится либо микрорайонирование клиентов [10, 23, 98, 117, 136], либо применяется ситуационное планирование, т.е. выделение типичных ситуаций [123].

Такой подход, как микрорайонирование клиентов подразумевает объединение нескольких пунктов в микрорайон. Объединение в районы может осуществляться различными способами. Во-первых, это может быть сектор с вершиной в грузообразующем пункте [23, 117]. Во-вторых, расположение объединенных в микрорайон пунктов может быть ограничено квадратом, диагональ которого составляет 5–6 км [98]. И, в-третьих, когда эта территория ограничена произвольной фигурой, но с условием, что объем перевозок в микрорайонах примерно одинаков, т.е. образуются так называемые «территории равного спроса» [10]. Далее, каждый микрорайон рассматривается отдельно, и решение осуществляется с использованием того или иного метода.

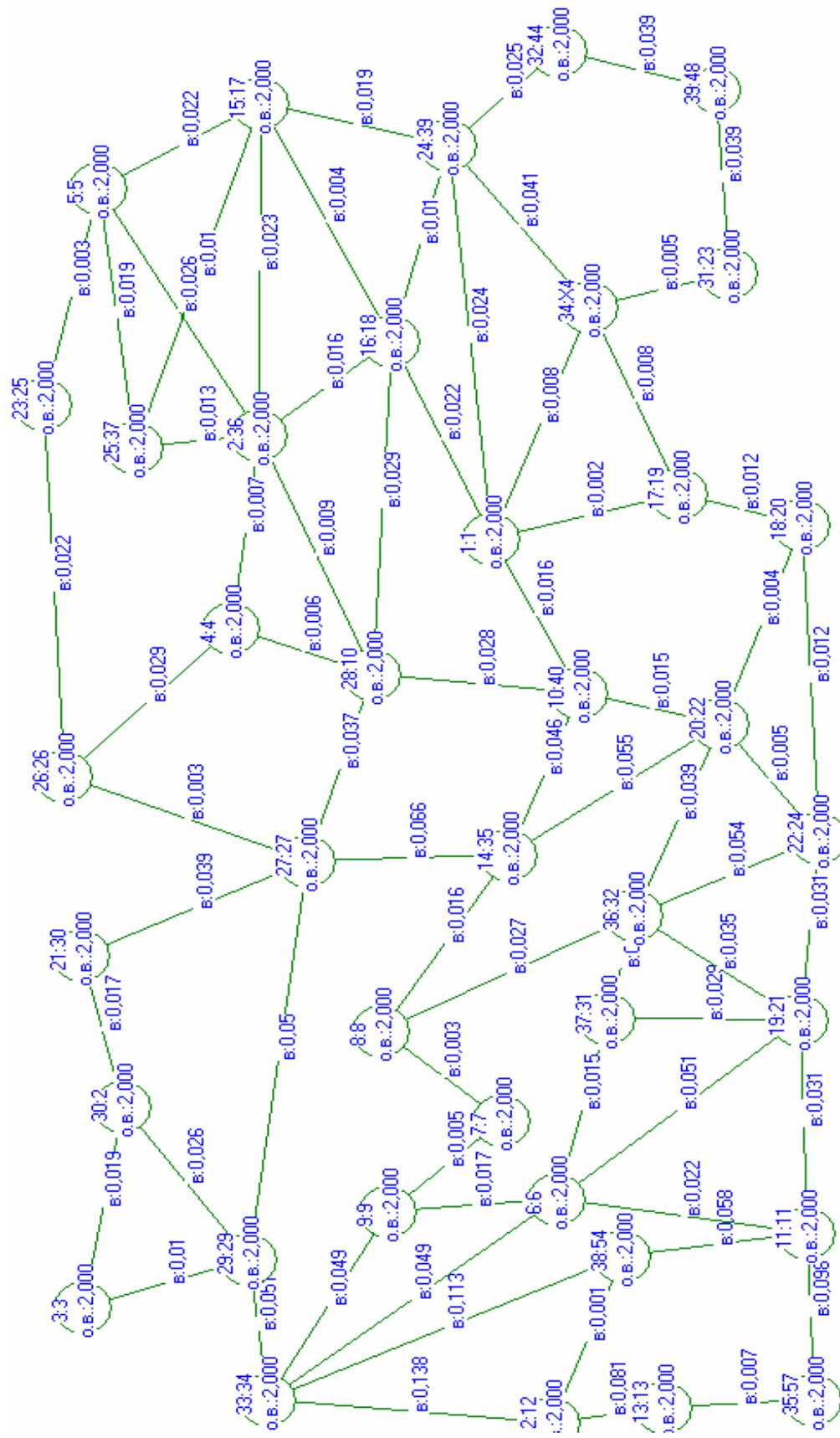


Рис. 1.1. Схема транспортной сети одного из районов г. Пензы

Задачи такого рода, когда транспортная сеть разбивается на районы, т.е. происходит зонирование, решаются с применением различных подходов [88]. Так в работе [91] рассматривается, декомпозиция транспортной сети. В этом случае сеть разбивается на несколько подсетей (районов). В случае, когда какой-либо пункт принадлежит к нескольким районам, то он считается граничной точкой.

В работе [46] в качестве базовой методики предлагается использовать алгоритм кластерного анализа [103] (кластеризации), представляющий эвристическую методику разделения пространства объектов на ряд подмножеств (кластеров). Предлагаемый алгоритм позволяет объединить вершины графа, находящиеся наиболее близко друг к другу, в кластер. Это объединение необходимо продолжать до тех пор, пока в пределах графа не останется вершин, которые могут быть объединены между собой. Следует заметить, что хотя разделение на зоны или объединение нескольких пунктов в группы и ведет к упрощению транспортной сети, однако необходимо учитывать следующие обстоятельства. При появлении дополнительных пробегов внутри района нарушаются реальные расстояния на транспортной сети, вследствие чего, происходят отклонения плановых показателей от фактических, что в свою очередь сказывается на точности результатов. Так как поток заявок, а также клиентурная база с течением времени может меняться, то вероятно искажение дислокации пунктов погрузки и разгрузки. Следовательно, погрешности расчета расстояний могут достигать критических значений. Таким образом, возникает постоянная необходимость в корректировке графа.

Наиболее примечательным примером метода локально оптимизации является метод инверсии [126, 134]. Сущность данного метода заключается в следующем. Изначально формулируется допустимый план перевозок и его характеристики, т.е. задается определенный маршрут. Затем, исходя из выбранного критерия оптимальности, добиваются совершенствования имеющегося плана перевозок путем манипуляции со звеньями маршрута. Таким образом, происходит разбиение маршрута на две части, а затем, заменяя звенья, снова их объединяют, но с условием, что одна часть маршрута будет иметь противоположное направление, т.е. она будет инвертирована. Когда инверсия двух любых звеньев позволяет сформировать маршрут, который уже не представляется возможным улучшить, то его называют два-оптимальным. Манипуляцию можно проводить также по трем, четырем и т.д. звеньям. При достижении три-оптимального маршрута улучшения очевидны, однако возрастает количество вариантов, которые необходимо проанализировать. Так, если для сравнения воспользоваться эвристическим методом Кларка-Райта, который будет рассмотрен ниже, то два-оптимальный маршрут по методу инверсии не дает преимуществ, а трудоемкость расчетов выше. Следует добавить, что она возрастает в 4 раза, ес-

ли маршрут улучшить до три-оптимального [40], однако при этом качество возрастает до 5%.

Наиболее ярким примером эвристических алгоритмов является метод Кларка-Райта [21,41,121]. Данный метод является лучшим по универсальности, точности решения и быстродействию из приближенных методов. За рубежом метод Г. Кларка и Дж. Райта известен как «экономизирующий», так как в качестве целевой функции выступает экономия пробега. Сущность данного метода заключается в следующем. Из одного узла (грузообразующего пункта) исходят несколько маятниковых маршрутов, которые объединяются в кольцевой таким образом, чтобы на каждой итерации два маршрута объединялись в один так, чтобы его суммарная длина была минимальна. Следовательно, на каждом этапе получается экономия от подключения пары вершин (i, j) или так называемая «функция выгоды» [91]:

$$\Delta_{i,j} = l_{in} + l_{nj} - l_{ij}, \quad (1.10)$$

где Δ_{ij} – «функция выгоды» от сокращения пробега при объединении пунктов i и j в маршрут;

l_{in}, l_{nj} – кратчайшие расстояния от пункта i до n и от n до j ;

l_{ij} – кратчайшее расстояние между пунктами i и j .

Наглядно процесс расчета «функция выгоды» Δ_{ij} можно представить следующим образом (рис. 1.2).

В итоге для всех объединений получают величины «функций выгоды», которые формируют матрицу экономий. Рассматриваемый метод может быть использован для парка автомобилей различной грузоподъемности, а также для несимметричной матрицы исходных данных. В качестве недостатка метода Кларка-Райта можно отметить отсутствие просмотра вперед. Связь, включенную в маршрут, на последних итерациях удалить невозможно, даже если она ухудшает решение.

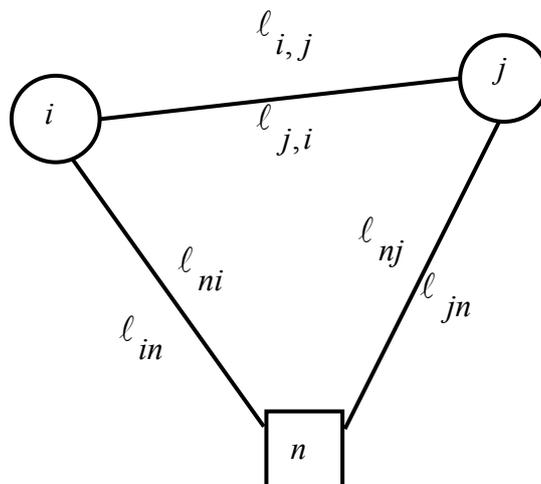


Рис. 1.2. Схема определения экономии для двух вершин

Также, следует заметить, что при использовании данного метода не учитывается реальное местоположение АТП, т.к. принимается, что оно находится в непосредственной близости с грузообразующим (ГОП) или грузопоглощающим (ГПП) предприятием. При развозке груза АТП закрепляется за первым пунктом погрузки для минимизации первого нулевого пробега, а второй нулевой пробег игнорируется. То же самое можно наблюдать и при сборе груза, т.е. минимизация второго нулевого пробега и не учет первого. В качестве важного замечания можно отметить то, что критерием оптимальности служит только минимальный пробег, который не учитывает величину транспортной работы. Это справедливо, когда она незначительна. Однако на практике встречаются задачи, когда учет транспортной работы в тонно-километрах необходим, поскольку это позволяет снизить эксплуатационные затраты. Метод Кларка-Райта не гарантирует получение оптимального результата, поэтому необходимо проверить целесообразность перестановки пунктов маршрута, исходя из минимальной транспортной работы автомобиля.

Одним из наиболее простых приближенных методов является суммирование по столбцам [21, 25, 49]. Данный алгоритм применяется в случае симметричной матрицы расстояний. Сущность данного метода заключается в том, что изначально выбираются три пункта с максимальным суммарным расстоянием, которые образуют базовый маршрут. Затем все остальные пункты поочередно включаются в него, путем последовательного перебора вариантов включения. При своей простоте использования этот метод имеет и недостатки. Если рассматривать большое количество пунктов, то трудоемкость процесса существенно возрастет. Также следует, что алгоритм по методу суммирования по столбцам не всегда удастся автоматизировать, вследствие того, что необходимо учитывать направление движения, т.к. возможно возникновение ситуации, когда в базовый маршрут могут быть включены наиболее отдаленные пункты, но с равным направлением движения.

Другим распространенным методом является выбор по кратчайшей связывающей сети [21, 29, 78]. Целью является соединить все пункты в кратчайшую связывающую сеть, т.е. на каждой итерации добавляется звено с наименьшими расстояниями. Данный метод, как и предыдущий, используется для симметричной матрицы.

Также известным является метод «метлы» [125]. Сущность данного метода заключается в определении кратчайших расстояний от пункта, принятого за начало сети, до всех остальных. Решение задачи ведется путем последовательного составления однотипных таблиц. Для данной задачи алгоритм решения достаточно прост и позволяет рационально использовать оперативную память ЭВМ [49]. Даже для транспортной сети с большим количеством вершин решение выполняется достаточно быстро. Подчеркнем, что в целом приближенные методы, дают результат, который в той или иной степени близок к оптимальному. Однако без знания точного решения эта величина отклонения будет неизвестна.

2. СОВЕРШЕНСТВОВАНИЕ МЕТОДА «ВЕТВЕЙ И ГРАНИЦ»

2.1. Разработка алгоритма решения задачи

Как уже было отмечено в первой главе, эффективность перевозочного процесса можно повысить с помощью планирования совмещенных маршрутов. Однако процесс совместной доставки мелкопартионных грузов в торговую сеть характеризуется некоторыми особенностями. Такой способ доставки подразумевает неоднократный заезд в пункт маршрута, что не удовлетворяет условию классической задачи маршрутизации, когда каждый пункт должен посещаться только один раз. Кроме того, такая идеальная теоретическая постановка вопроса не редко противоречит реальному маршруту движения, установленному опытным путем, когда выгоднее проехать по одной и той же улице транспортной сети несколько раз. Также, следует заметить, что граф транспортной сети, в силу специфичности планировки улиц городов, может иметь подциклы, то есть звенья, замкнутые на себе.

В современных условиях требования, предъявляемые к точности расчета оптимального маршрута движения, возрастают, и поэтому альтернативы точным методам решения нет. В настоящей работе в основу оптимизации развозочных маршрутов был положен метод «ветвей и границ». Недостатком данного метода является то, что он не может удовлетворять особенностям, характерным совмещенным маршрутам, то есть он не учитывает подциклы. Это обстоятельство ограничивает применение метода «ветвей и границ» и заставляет исследователей искать приближенные пути решения задачи.

В данной работе предложена методика устранения указанного недостатка посредством усовершенствования классического метода. Идея методики заключается во введении дополнительных фиктивных узлов и связей между ними и пунктами исследуемой транспортной сети, когда это необходимо.

Фиктивным узлом (пунктом) называется дополнительно введенный узел, которого не было в исходной транспортной сети. Аналогично, фиктивной связью называется дополнительно введенная дуга между ним и действительным узлом или между фиктивными узлами. Вместе они образуют фиктивную ветвь или звено.

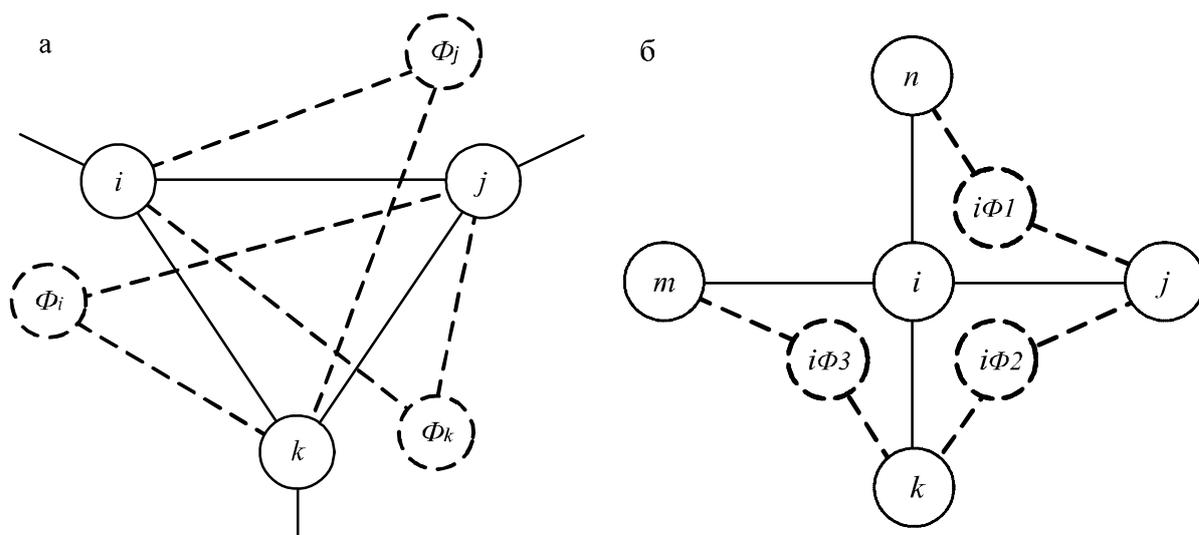


Рис. 2.1.Элементы графа «фиктивных ветвей»:
 а – треугольный элемент; б – узловый элемент

Проблема заключается в разработке методики и способа их введения, учитывающей особенности используемого метода.

Перейдем к общей постановке задачи маршрутизации, которую можно сформулировать следующим образом. Пусть имеется транспортная сеть, на которой необходимо осуществить перевозку груза. Грузы имеют одну характеристику, что позволяет использование одного типа подвижного состава. Имеется одно автотранспортное предприятие, которое обеспечивает доставку продукции от одного производителя в несколько пунктов потребления (торговые точки). Задано количество груза, предъявляемого к перевозке, для каждого из грузополучателей. За один маршрут осуществляется обслуживание нескольких пунктов. Необходимо так организовать перевозочный процесс, чтобы удовлетворить всех клиентов, но при этом затраты на транспортировку были бы минимальны. В качестве целевой функции может выступать минимальная длина маршрута или время, соответственно:

$$f(L) = \sum \ell_i \rightarrow \min; f(T) = \sum t_i \rightarrow \min. \quad (2.1)$$

При наличии нескольких маршрутов одинаковой длины, выбирается тот, у которого минимальная транспортная работа, затрачиваемая при перевозке груза. Заметим, что под транспортной работой понимается работа, затрачиваемая только на перевозку груза [77].

Перейдем к формализации поставленной задачи. Граф транспортной сети включает в себя N вершин. Количество пунктов разгрузки i находится в пределах $i = 1, 2, 3 \dots n$. Все вершины соединены связями, вес (длина или время) которых известен. Исходя из этого, составляется исходная матрица весов между пунктами [43, 79, 108]:

$$L = \begin{pmatrix} \infty & l_{1,2} & l_{1,3} & \dots & l_{1,n} & \dots & l_{1,N} \\ l_{2,1} & \infty & l_{2,3} & \dots & l_{2,n} & \dots & l_{2,N} \\ l_{3,1} & l_{3,2} & \infty & \dots & l_{3,n} & \dots & l_{3,N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ l_{n,1} & l_{n,2} & l_{n,3} & \dots & \infty & \dots & l_{n,N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ l_{N,1} & l_{N,2} & l_{N,3} & \dots & l_{N,n} & \dots & \infty \end{pmatrix}$$

Заметим, что ∞ – представляет собой очень большое число и означает блокировку движения между этими пунктами. Возможно, что $l_{ij} \neq l_{ji}$, тогда матрица получается несимметричная.

Для решения поставленной задачи будем придерживаться нижеследующего алгоритма расчета:

1. Составляем исходную матрицу весов L между пунктами l_{ij} исследуемой транспортной сети. Следует заметить, что в ее клетках могут быть значения не только расстояний, но и другие показатели.

2. Переходим к приведенной матрице – $L^{пр}$. Для этого в каждой строке находим минимальный элемент h_i и вычитаем его из всех остальных элементов l_{ij} , расположенных в рассматриваемой строке:

$$l'_{ij} = l_{ij} - h_i, \quad i = 1, 2, 3 \dots n. \quad (2.2)$$

Затем, в полученной матрице находим минимальный элемент в каждом столбце h_j и вычитаем его из всех остальных элементов l'_{ij} , расположенных в рассматриваемом столбце:

$$l''_{ij} = l'_{ij} - h_j, \quad j = 1, 2, 3 \dots m. \quad (2.3)$$

3. Определяем для каждого элемента с $l''_{ij} = 0$ оценку по формуле:

$$A = \min l''_{ik} + \min l''_{sj}, \quad (2.4)$$

где l''_{ik} – наименьший элемент в строке i ;

l''_{sj} – наименьший элемент в столбце j ;

$$k \neq j, \quad s \neq i, \quad k, s = 1, 2, 3 \dots n.$$

4. Находим пару $k - s$ с максимальной оценкой:

$$A_{ks} = \max A_{ij}, \quad i, j = 1, 2, 3 \dots n. \quad (2.5)$$

5. Переходим к новой матрице L_1 . Для этого вычеркиваем из L^{np} строку k и столбец s с наибольшей оценкой. Блокируем ячейку на пересечении строки s и столбца k , а также ветвь, ведущую к заикливанью цепи со звеном $k - s$.

6. Создаем новые фиктивные матрицы Φ_k и Φ_s , вводя в L_1 фиктивные узлы k и s . Выполняем над полученными матрицами L_1 , Φ_k и Φ_s операции приведения и оценки, описанные в п. 2–4, до тех пор, пока последняя вычеркиваемая ветвь не станет очевидной.

7. Устанавливаем, что количество фиктивных узлов равно числу входящих в узел ветвей минус единица.

8. Оптимальный маршрут устанавливается путем сравнения вариантов.

На основании предложенного алгоритма была разработана компьютерная программа, блок-схема которой приведена на рисунке 2.2. Программа позволяет осуществить перебор всех возможных вариантов маршрутов, в соответствии с разработанным алгоритмом. Если в процессе расчета получается несколько маршрутов, имеющих одинаковое минимальное значение целевой функции, то она выбирает оптимальный маршрут с наименьшей транспортной работой.

В ней предусмотрен контроль над вводом количества фиктивных узлов по усмотрению оператора. Это позволяет сократить количество вариантов и время решения задачи. В качестве целевой функции в программе можно принять расстояние или время. Программа функционирует под управлением операционной системы Windows 95, 98, XP. Для вычисления используется методика рекурсивного левостороннего алгоритма. Все модули программы реализованы на языке C++ [2,100] в среде программирования Borland C++ Builder 6.0.

Описание разработанной программы определения наименьшего значения целевой функции, приведенное в прил. 1, и включает в себя следующее:

- описание интерфейса программы, которое содержит описание ввода исходных данных и описание рабочего режима программы;
- техническое описание программы, включающее в себя описание основных элементов главной формы программы;
- описание классов и интерфейсных функций.

Кроме того, для поиска кратчайшего расстояния или времени между двумя любыми вершинами графа, была разработана программа, основанная на методе «метлы». Она использует ту же исходную матрицу весов.

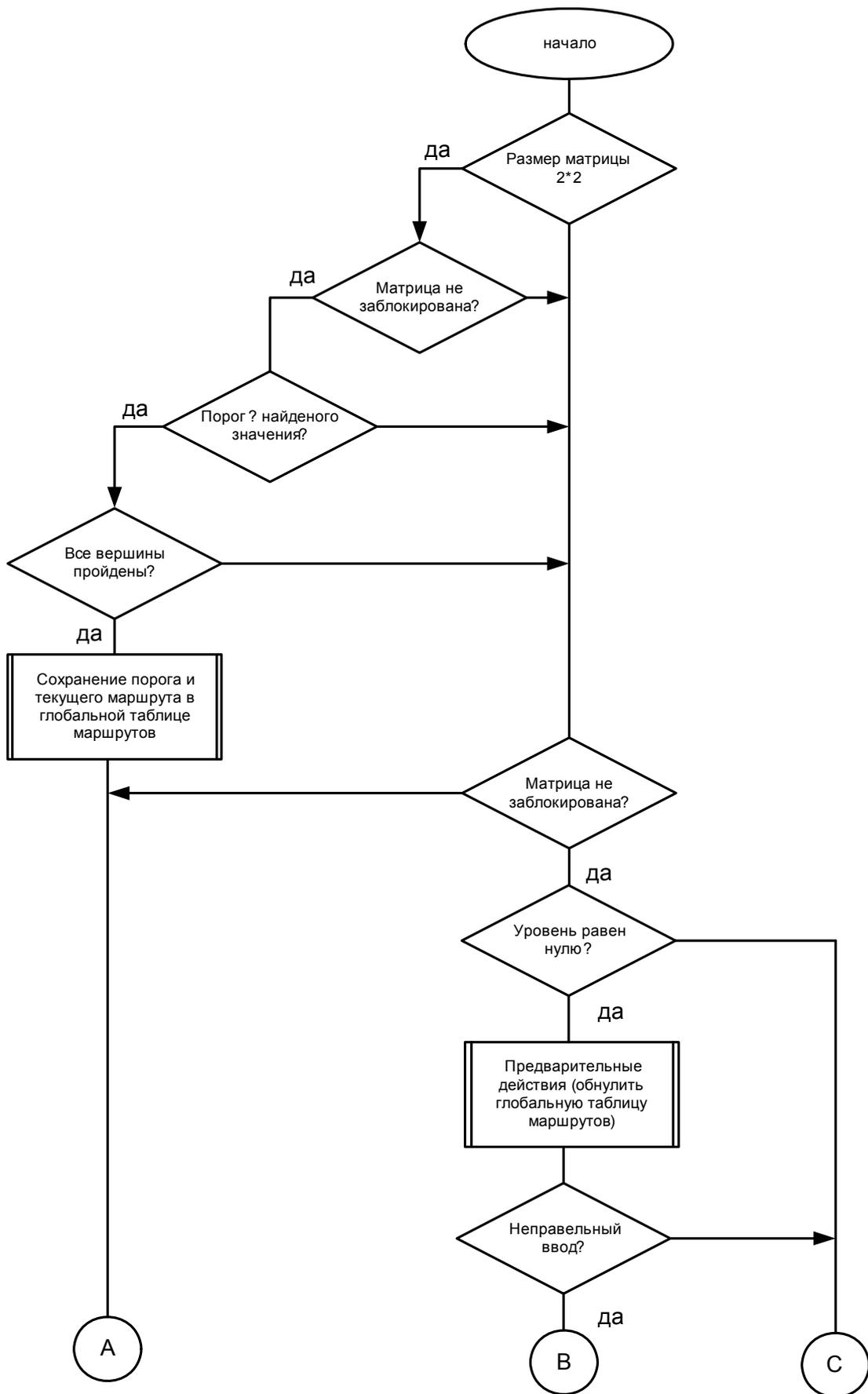


Рис. 2.2. Блок-схема метода «фиктивных ветвей» (начало)

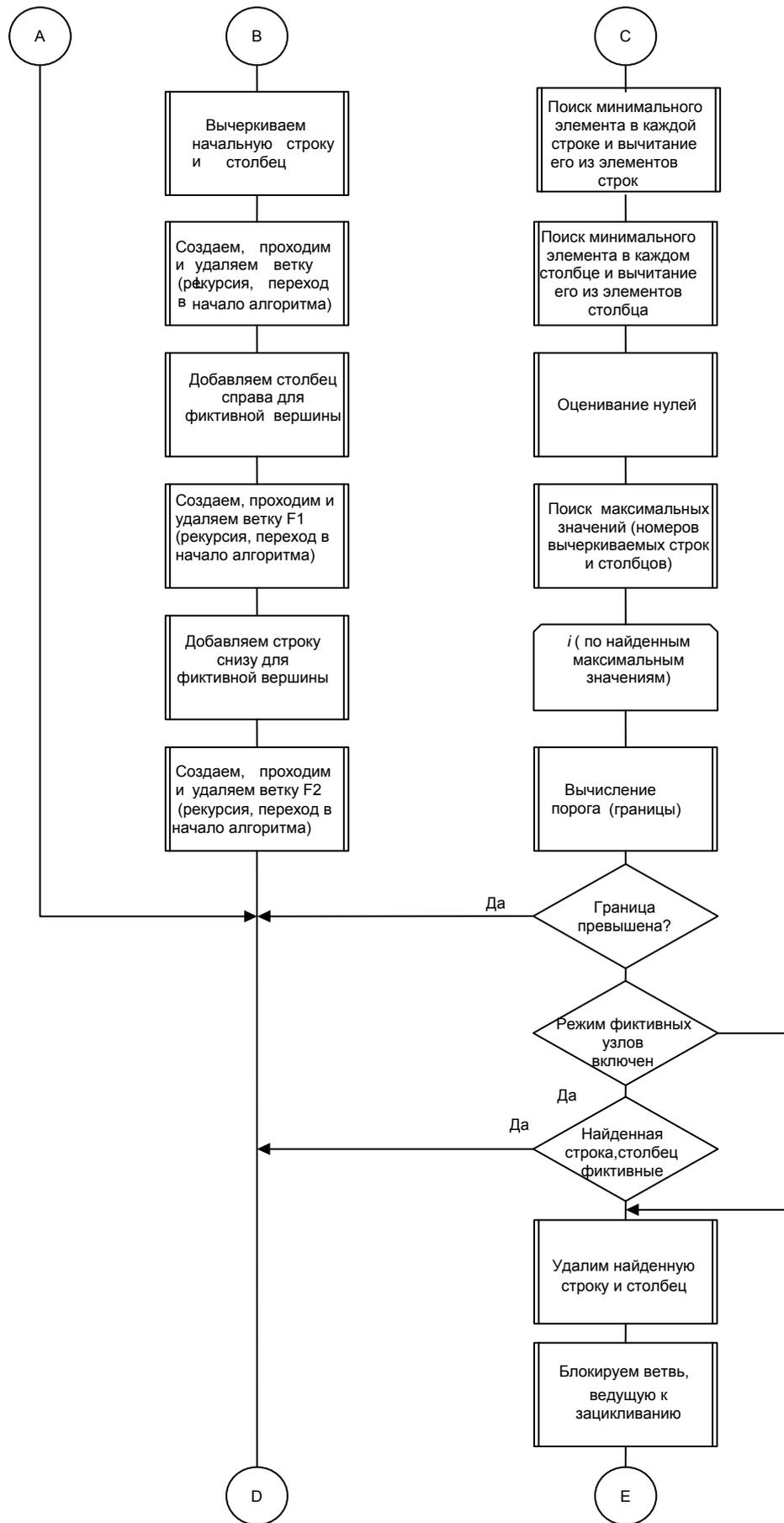


Рис. 2.2. Блок-схема метода «фиктивных ветвей» (продолжение)

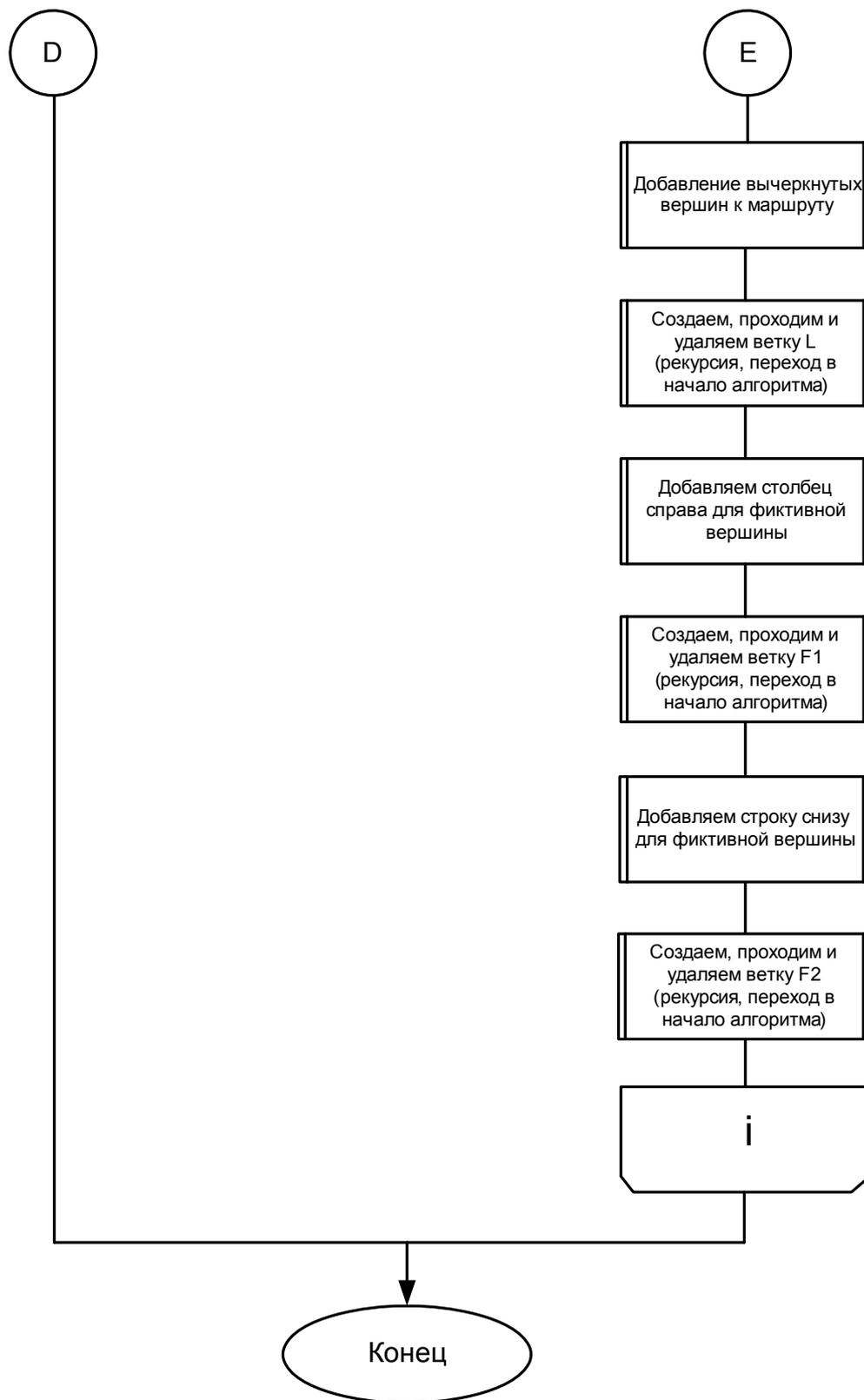


Рис. 2.2. Блок-схема метода «фиктивных ветвей» (окончание)

2.2. Примеры решения задач по методике «фиктивных ветвей»

Покажем применение предложенного алгоритма на примере транспортной сети из пяти пунктов, представленной на рис. 2.3. Расчет по программе показал, что решение данной задачи имеет 848 вариантов. Объем работы не позволяет привести решение всех вариантов подробно. Поэтому для раскрытия сущности предлагаемой методики рассмотрим наиболее значимые из них. На первом этапе составляется исходная матрица расстояний L (табл. 2.1). Результаты вычислений, согласно второму этапу, приведены в табл. 2.2 и 2.3.

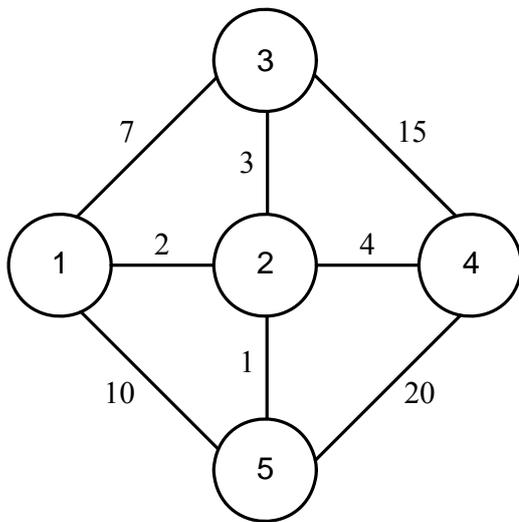


Рис. 2.3. Схема транспортной сети

Таблица 2.1

	1	2	3	4	5
1		2	7		10
2	2		3	4	1
3	7	3		15	
4		4	15		20
5	10	1		20	

L

Таблица 2.2

	1	2	3	4	5
1		0	5		8
2	1		2	3	0
3	4	0		12	
4		0	11		16
5	9	0		19	

Таблица 2.3

	1	2	3	4	5
1		0	3		8
2	0		0	0	0
3	3	0		9	
4		0	9		16
5	8	0		16	

На третьем этапе решения задачи определяются оценки для каждого нулевого элемента матрицы, которые указываются в правом верхнем углу

ячеек. На четвертом этапе находится ветвь с максимальной оценкой. Как видно из таблицы 2.4, наибольшую оценку имеют ветви 2 – 4 и 4 – 2, равную 9.

Таблица 2.4

	1	2	3	4	5
1		0^3	3		8
2	0^3		0^3	0^9	0^8
3	3	0^3		9	
4		0^9	9		16
5	8	0^8		16	

L^{np}

Таблица 2.5

	1	2	3	5
1		0	3	8
3	3	0		
4		∞	9	16
5	8	0		

L_1

Согласно пятому этапу, можно вычеркнуть любую из указанных ветвей. Таким образом, дерево решений раздваивается и в дальнейшем каждая ветвь проверяется отдельно. Рассмотрим пример решения задачи при вычеркивании ветви 2–4. Полученная новая матрица L_1 представлена в табл. 2.5, где во избежание заикливания ветвь 4–2 блокируется. Составляем фиктивные матрицы Φ_k и Φ_s , представленные в табл. 2.6 и 2.7, путем поочередного введения в L_1 фиктивных узлов вычеркнутой ветви из матрицы L^{np} .

Таблица 2.6

	1	2	3	5	4ф1
1		0	3	8	
3	3	0			9
4			9	16	
5	8	0			16
4ф1		0	9	16	

Φ_k

Таблица 2.7

	1	2	3	5	2ф1
1		0	3	8	0
3	3	0			0
4			9	16	0
5	8	0			0
2ф1	0		0	0	

Φ_s

Выполняем над полученными фиктивными матрицами операции приведения и оценки, аналогично ранее рассмотренным, до тех пор, пока в результате не получится матрица размером 2×2 . В начале рассмотрим решение матрицы Φ_s . В табл. 2.8 приводим оценки всех нулевых элементов и

по максимальной из них вычеркиваем ветвь 4 – 2ф1, получая новую матрицу (табл. 2.9).

Т а б л и ц а 2.8

	1	2	3	5	2ф1
1		0^0	3	8	0^0
3	3	0^0			0^0
4			9	16	0^9
5	8	0^0			0^0
2ф1	0^3		0^3	0^8	

Т а б л и ц а 2.9

	1	2	3	5
1		0	3	8
3	3	0		
5	8	0		
2ф1	0		0	0

Проводим над ней аналогичную операцию приведения и следующей вычеркиваем ветвь 5 – 2, табл. 2.10. В полученную матрицу, представленную в табл. 2.11, вводим поочередно фиктивные узлы 5ф1 и 2ф2.

Т а б л и ц а 2.10

	1	2	3	5
1		0^3	3	8
3	3	0^3		
5	8	0^8		
2ф1	0^3		0^3	0^8

Т а б л и ц а 2.11

	1	3	5
1		3	8
3	3		
2ф1	0	0	0

Вначале в новую матрицу вводим фиктивный узел 5ф1 и вычеркиваем ветвь 3 – 1 (табл. 2.12). Чтобы исключить заикливание, блокируем ветвь 1 – 3 (табл. 2.13).

Из таблицы 2.13 видно, что с узлом 5ф1 связь отсутствует, поэтому его ввод не целесообразен. Возвращаемся в табл. 2.12 и рассматриваем вариант вычеркивания ветви 5ф1 – 1 (табл. 2.14).

Таблица 2.12

	1	3	5	5ф1
1		0^5	5	5
3	0^∞			
2ф1	0^0	0^0	0^5	0^5
5ф1	0^∞			

Таблица 2.13

	3	5	5ф1
1	∞	5	5
2ф1	0	0	0
5ф1			

Таблица 2.14

	1	3	5	5ф1
1		0^5	5	5
3	0^∞			
2ф1	0^0	0^0	0^5	0^5
5ф1	0^∞			

Таблица 2.15

	3	5	5ф1
1	0	5	∞
3			
2ф1	0	0	0

Из табл. 2.15 видно, что с узлом 3ф1 связь тоже отсутствует. Таким образом, прохождение по данной ветви дерева решений не дает.

Возвращаемся в табл. 2.10 и теперь рассмотрим вариант введения второго фиктивного узла 2ф2, табл. 2.16. В табл. 2.17 приводим оценки и вначале вычеркиваем ветвь 1 – 2ф2.

Таблица 2.16

	1	3	5	2ф2
1		3	8	0
3	3			0
2ф1	0	0	0	
2ф2	0	0	0	

Таблица 2.17

	1	3	5	2ф2
1		3	8	0^3
3	3			0^3
2ф1	0^0	0^0	0^0	
2ф2	0^0	0^0	0^0	

На следующем этапе, после приведения матрицы, вычеркивается ветвь 3 – 1 (табл. 2.18). Далее решение доводится до получения матрицы разме-

ром 2×2 . Ветвь $2\phi 2 - 3$ блокируется, и поочередно вычеркиваются ветви $2\phi 1 - 3$ и $2\phi 2 - 5$ (табл. 2.19).

Т а б л и ц а 2.18

	1	3	5
3	0^∞		
2φ1	0^0	0^0	0^0
2φ2	0^0	0^0	0^0

Т а б л и ц а 2.19

	3	5
2φ1	0^∞	0^0
2φ2	∞	0^∞

В результате прохождения по данной ветви дерева решений получен маршрут $2 - 4 - 2\phi 1 - 3 - 1 - 2\phi 2 - 5 - 2$ длиной 22 км.

Возвращаемся в таблицу 2.17 и проверяем вариант вычеркивания ветви $3 - 2\phi 2$ (табл. 2.20). Получаем новую матрицу, где блокируем ветвь $2\phi 2 - 3$ (табл. 2.21).

Т а б л и ц а 2.20

	1	3	5	2φ2
1		3	8	0^3
3	3			0^3
2φ1	0^0	0^0	0^0	
2φ2	0^0	0^0	0^0	

Т а б л и ц а 2.21

	1	3	5
1		3	8
2φ1	0	0	0
2φ2	0	∞	0

После приведения матрицы и оценки нулевых элементов вычеркиваем ветвь $1 - 3$ (табл. 2.22). Далее решение доводится до получения матрицы размером 2×2 . Ветвь $2\phi 2 - 1$ блокируется, и поочередно вычеркиваются ветви $2\phi 1 - 1$ и $2\phi 2 - 5$ (табл. 2.23).

Т а б л и ц а 2.22

	1	3	5
1		0^5	5
2φ1	0^0	0^0	0^0
2φ2	0^0		0^0

Т а б л и ц а 2.23

	1	5
2φ1	0^∞	0^0
2φ2	∞	0^∞

При прохождении по рассматриваемой ветви дерева решений получен маршрут $2 - 4 - 2\phi 1 - 1 - 3 - 2\phi 2 - 5 - 2$ длиной 22 км.

Возвращаемся в табл. 2.21 и теперь рассмотрим вариант введения фиктивного узла $3\phi 1$ (табл. 2.24). Ветвь $2\phi 2 - 3$ блокируем, чтобы избежать заикливание.

В табл. 2.25 выполняем приведение матрицы и оценку нулевых элементов, в результате чего вычеркиваем ветвь $3\phi 1 - 1$. Получаем новую матрицу, где блокируем соответственно ветвь $1 - 3\phi 1$ (табл. 2.26).

Т а б л и ц а 2.24

	1	3	5	3φ1
1		3	8	3
2φ1	0	0	0	0
2φ2	0	∞	0	0
3φ1	3			

Т а б л и ц а 2.25

	1	3	5	3φ1
1		0^0	5	0^0
2φ1	0^0	0^0	0^0	0^0
2φ2	0^0		0^0	0^0
3φ1	0^∞			

Вычеркиваем ветвь $1 - 3$ (табл. 2.27), а ветвь $1 - 3\phi 1$ блокируется. Далее решение доводится до получения матрицы размером 2×2 . При этом в табл. 2.28 блокируется ветвь $2\phi 1 - 5$ и поочередно вычеркиваются ветви $2\phi 1 - 3\phi 1$ и $2\phi 2 - 5$.

Т а б л и ц а 2.26

	3	5	3φ1
1	0	5	∞
2φ1	0	0	0
2φ2		0	0

Т а б л и ц а 2.27

	3	5	3φ1
1	0^5	5	∞
2φ1	0^0	0^0	0^0
2φ2		0^0	0^0

Т а б л и ц а 2.28

	5	3φ1
2φ1	∞	0^∞
2φ2	0^∞	0^0

При прохождении по рассматриваемой ветви дерева решений получен маршрут $2 - 4 - 2\phi 1 - 3\phi 1 - 1 - 3 - 2\phi 2 - 5 - 2$ длиной 30 км.

Возвращаемся в табл. 2.26 и теперь рассмотрим вариант введения фиктивного узла $3\phi 2$ (табл. 2.29). Ветвь $1 - 3\phi 1$ блокируем, чтобы избежать заикливание. Узел $3\phi 2$ не имеет связей, а, следовательно, путь не замкнут. Поэтому при его вводе задача решения не имеет.

Таблица 2.29

	3	5	3ф1	3ф2
1	0	5		0
2ф1	0	0	0	0
2ф2		0	0	0
3ф2				

Таблица 2.30

	3	5	3ф1	1ф1
1	0^0	5		0
2ф1	0^0	0^0	0^0	0^0
2ф2		0^0	0^0	0^0
1ф1	0^0	5	0^0	

Возвращаемся в табл. 2.26 и проверяем вариант решения при вводе фиктивного узла 1ф1, в результате чего вычеркивается ветвь 1 – 3 (табл. 2.30). Ввод фиктивных узлов 1 и 3 в табл. 2.30 невозможен, так как отсутствуют соответствующие столбцы и строки.

Возвращаемся в табл. 2.21 и проверяем вариант ввода фиктивного узла 2ф3. После приведения матрицы и оценки нулевых элементов вычеркиваем ветвь 1 – 2ф3 (табл. 2.31). Получаем новую матрицу, где блокируем ветвь 2ф3 – 1 (табл. 2.32).

Таблица 2.31

	1	3	5	2ф3
1		3	8	0^3
2ф1	0^0	0^0	0^0	0^0
2ф2	0^0		0^0	
2ф3	0^0	0^0	0^0	

Таблица 2.32

	1	3	5
2ф1	0^0	0^0	0^0
2ф2	0^0		0^0
2ф3	∞	0^0	0^0

Согласно алгоритму решения задачи (пункт 7), количество фиктивных узлов не должно превышать число связей минус единица. Следовательно, очередной фиктивный узел 2ф4 вводить нельзя. Далее рассмотрим решение матрицы без введения фиктивных узлов. Из табл. 2.32 видно, что все элементы имеют одинаковые оценки. Поэтому необходимо поочередно выполнить вычеркивание ветвей, а оптимальный маршрут найти путем сравнения. В ходе расчета установлено, что все семь вариантов дают одинаковый результат решения – маршрут длиной 20 км.

Возвращаемся в табл. 2.32 и проверяем вариант ввода фиктивного узла 1ф1, табл. 2.33. Вычеркиваем ветвь 1ф1 – 3 и получаем новую матрицу, где блокируем ветвь 2ф2 – 1ф1 (табл. 2.34).

Таблица 2.33

	1	3	5	1ф1
2ф1	0^0	0^0	0^0	0^0
2ф2	0^0		0^0	0^0
2ф3		0^0	0^0	0^0
1ф1		0^5	5	

Таблица 2.34

	1	5	1ф1
2ф1	0^0	0^0	0^0
2ф2	0^0	0^0	∞
2ф3		0^0	0^0

Во всех ячейках с нулевыми элементами одинаковые оценки, поэтому поочередно вычеркивается каждая ветвь. В результате решения установлено, что длина маршрута составила 26 км.

Из сравнения всех вариантов определяем оптимальный маршрут движения длиной 20 км, показанный на рис. 2.4 а. Заметим, можно также двигаться и в обратном направлении. Если выполнить расчет маршрута классическим методом «ветвей и границ», то последовательность получается следующей: 2-4-3-1-5-2, длиной 37 км (рис. 2.4 б).

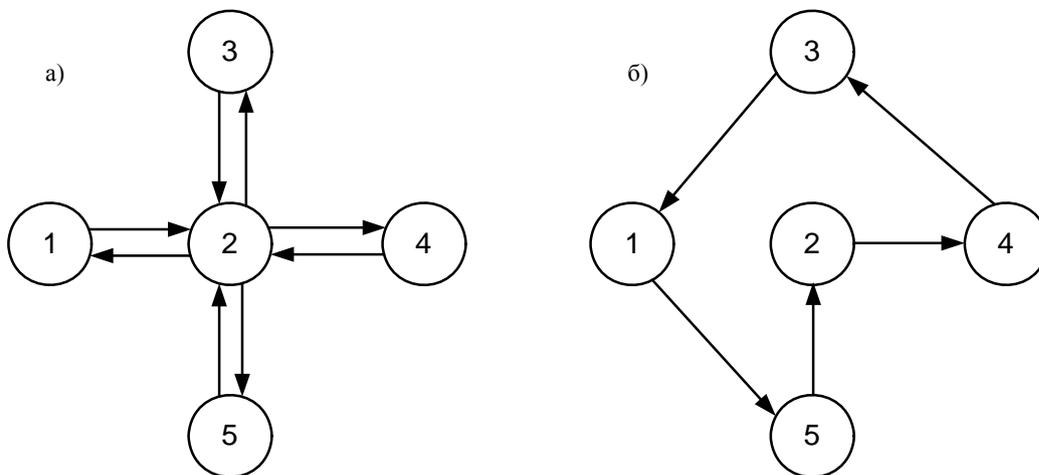


Рис. 2.4. Граф движения:
а – по предложенной методике; б – по методу «ветвей и границ»

Покажем применение предложенного алгоритма на примере транспортной сети из тринадцати пунктов (рис. 2.5). В этом случае классический метод «ветвей и границ» не дает решения задачи без декомпозиции, так как заранее ясно, что узел 10 требуется посетить два раза.

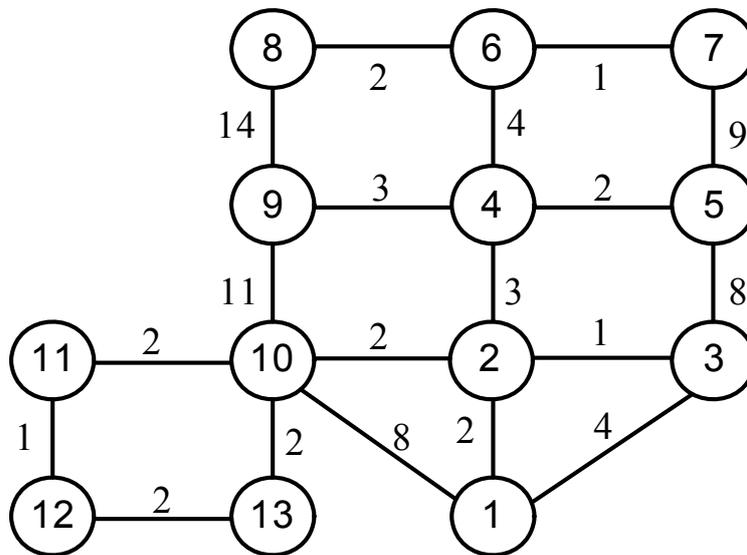


Рис. 2.5. Схема транспортной сети

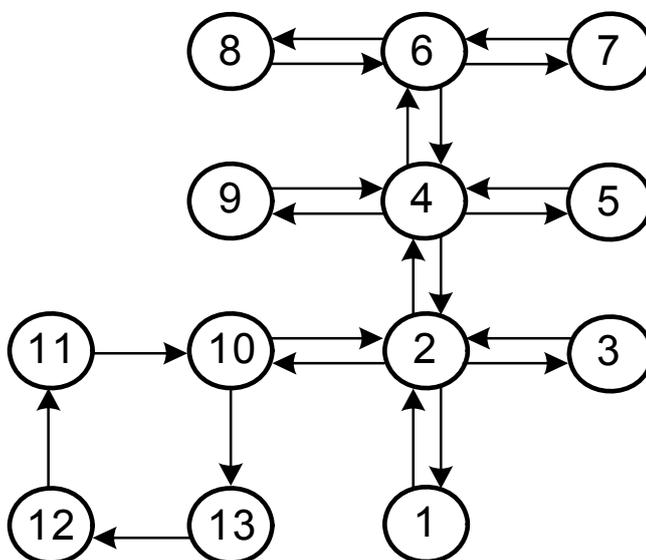


Рис. 2.6. Граф оптимального маршрута движения

Исходная матрица расстояний L приведена в табл. 2.35.

Таблица 2.35

	1	2	3	4	5	6	7	8	9	10	11	12	13
1		2	4							8			
2	2		1	3						2			
3	4	1			8								
4		3			2	4			3				
5			8	2			9						
6				4			1	2					
7					9	1							
8						2			14				
9				3				14		11			
10	8	2							11		2		2
11										2		1	
12											1		2
13										2		2	

В правом верхнем углу табл. 2.36 дана оценка пар, рассчитывая по выше приведенному алгоритму для приведенной матрицы L^{np} . В пустых клетках таблицы находятся большие числа. Из табл. 2.36 вычеркиваем пару 8 – 6 с наибольшей оценкой 11 и блокируем ветвь 6 – 8. Полученная из нее матрица L_1 , представлена в табл. 2.37. В ней наибольшую оценку ∞ имеют две ветви 7 – 5 и 9 – 8. Как показало дальнейшее исследование, их вычеркивание не дает оптимального маршрута.

Таблица 2.36

	1	2	3	4	5	6	7	8	9	10	11	12	13
1		0^2	2							6			
2	0^2		0^2	2						1			
3	2	0^2			7								
4		1			0^7	2			0^8				
5			6	0^6			7						
6				3			0^7	0^{10}					
7					8	0^8							
8						0^{11}			11				
9				0^8				10		8			
10	5	0^0							8		0^0		0^1
11										1		0^1	
12											0^1		1
13										0^1		0^0	

Фиктивная матрица получается из табл. 2.36 после вычеркивания из нее ветви 8 – 6 и введения фиктивного узла бф1. Из нее получаем приведенную матрицу Φ_s , представленную в табл. 2.38. В ней имеются две ветви, имеющие наибольшую оценку 10. Вычеркиваем последовательно каждую из них: бф1 – 8 и 7 – бф1. Далее алгоритм повторяется.

Т а б л и ц а 2.37

	1	2	3	4	5	7	8	9	10	11	12	13
1		0^2	2						6			
2	0^2		0^2	2					1			
3	2	0^2			7							
4		1			0^0			0^8				
5			6	0^6		7						
6				3		0^7	0^{10}					
7					0^∞							
9				0^0			0^∞		8			
10	5	0^0						8		0^0		0^1
11									1		0^1	
12										0^1		1
13									0^1		0^0	

Т а б л и ц а 2.38

	1	2	3	4	5	7	8	9	10	11	12	13	бф1
1		0^2	2						6				
2	0^2		0^2	2					1				
3	2	0^2			7								
4		1			0^7			0^8					2
5			6	0^6		7							
6				3		0^3							
7					8								0^{10}
9				0^8			10	8	8				
10	5	0^0						8		0^0		0^1	
11									1		0^1		
12										0^1		1	
13									0^1		0^0		
бф1				3		0^0	0^{10}						

Процесс приведения, вычеркивания и ввода фиктивных узлов продолжается до тех пор, пока не останется матрица размером 2×2 , и, следовательно, удаляемая ветвь станет очевидной. При этом, согласно приведенному алгоритму, общее количество фиктивных узлов не должно превышать количество ветвей входящих в узел минус один. Полученный в результате решения на ПК оптимальный маршрут длиной 47 км показан на рисунке 2.6: $8 - 6 - 4\text{ф}2 - 2\text{ф}3 - 1 - 2 - 1\text{ф}1 - 13 - 12 - 11 - 10 - 2\text{ф}2 - 3 - 2\text{ф}1 - 4\text{ф}3 - 5 - 4\text{ф}1 - 9 - 4 - 6\text{ф}2 - 7 - 6\text{ф}1 - 8$.

В табл. 2.39 представлена приведенная матрица Φ_k , полученная из табл. 2.37 путем введения фиктивного узла $8\text{ф}1$ в вершину 8. Она имеет четыре ветви с оценкой ∞ : $7 - 5$, $9 - 8$, $9 - 8\text{ф}1$ и $8\text{ф}1 - 9$. Проведенный анализ показал, что последовательное их вычеркивание не дает оптимального маршрута.

Т а б л и ц а 2.39

	1	2	3	4	5	7	8	9	10	11	12	13	8ф1
1		0^2	2						6				
2	0^2		0^2	2					1				
3	2	0^2			7								
4		1			0^0			0^0					
5			6	0^6		7							
6				3		0^{10}							
7					0^∞								
9				0^0			0^∞		8				0^∞
10	5	0^0						8		0^0		0^1	
11									1		0^1		
12										0^1		1	
13									0^1		0^0		
8ф1								0^∞					

3. РАЗРАБОТКА МОДЕЛИ СОВМЕЩЕНИЯ МАРШРУТОВ ПРИ ДОСТАВКЕ МЕЛКОПАРТИОННЫХ ГРУЗОВ

3.1. Совершенствование теории совместной развозки грузов от двух производителей

Повышение эффективности перевозочного процесса, а также производительности подвижного состава можно достичь путем увеличения объема перевозимого груза и сокращения холостых пробегов. Идеальным вариантом является маятниковый маршрут с обратным груженым пробегом. Чтобы повысить эффективность развозочного маршрута при перевозке мелкопартионных грузов необходимо поступать аналогично – заезжать за дополнительным (попутным) грузом в другой пункт погрузки. Однако в этом случае, в отличие от маятникового маршрута, он может загружаться в транспортное средство не только после полной разгрузки основного груза, но и по ходу его движения. В первом случае возникает развозочный маршрут с обратным груженым пробегом, когда в кузове автомобиля всегда перевозится только один груз: основной или дополнительный. Во втором случае – после заезда во второй пункт погрузки возникает этап совместной развозки сразу двух грузов. Заметим, что при разгрузке двух грузов в пунктах потребления сокращаются затраты времени на заезд и оформление документов. Однако эта особенность в современной теории не учитывается. Конечно, следует заметить, что речь идет о грузах, имеющих одну транспортную характеристику.

В современной теории вопрос об объединении маршрутов недостаточно разработан, однако решение задачи с несколькими производителями представляет значительную сложность ввиду многообразия накладываемых ограничений. Поэтому в качестве примера рассмотрим более простой вариант, когда доставка грузов осуществляется от двух производителей.

При организации маршрута перевозки с двумя пунктами погрузки необходимо решить два вопроса:

1. Каким должно быть максимально допустимое время заезда во второй пункт погрузки, чтобы совмещение было целесообразным?
2. С какого пункта транспортной сети необходимо осуществлять этот заезд?

Чтобы ответить на первый вопрос необходимо сравнить технико-экономические показатели при раздельной развозке основного и дополнительного груза двумя автомобилями с объединенным их вариантом, когда на маршруте работает один автомобиль. Для определения пункта заезда за вторым грузом необходимо решить задачу маршрутизации совместно с определением наличия свободного места в кузове автомобиля.

Вначале рассмотрим решение первой задачи. В качестве критерия целесообразности примем время, которое является наиболее значимым показателем перевозочного процесса в сложившихся рыночных условиях. Это подтверждается тарифами автоперевозчиков как по всей России, так и в городе Пензе, которые выражаются в руб./час.

Известно, что время на маршруте складывается из времени выполнения трех основных технологических операций:

$$t = t^{3p} + t^{np} + t^{\ell}, \quad (3.1)$$

где t^{3p} – организационный простой – время на заезд и оформление документов, ч;

t^{np} – технологический простой – время на погрузо-разгрузочные работы, ч;

t^{ℓ} – время движения, ч.

При совмещении основного и дополнительного маршрутов можно сократить затраты времени за счет уменьшения t^{3p} . Тогда, в качестве критерия целесообразности совмещения можно принять условие:

$$\Delta t = \Delta t^{3p} - \Delta t^{\ell} \geq 0. \quad (3.2)$$

Таким образом, разница времени сэкономленном на заездах в пункты разгрузки нескольких грузов Δt^{3p} и времени затраченном на изменение длины маршрута Δt^{ℓ} должно быть больше или равно нулю. В выражении (3.2):

$$\Delta t^{3p} = \sum_{i=1}^{\Pi} t_i^{3p} (\Gamma_i - 1), \quad (3.3)$$

где Γ_i – количество грузов одновременно разгружаемых в i -м пункте, т;

t_i^{3p} – время на заезд в пункт разгрузки, ч;

Π – количество пунктов совместной разгрузки, шт.

Изменение времени движения, есть:

$$\Delta t^{\ell} = t^{\text{сов}} - (t^{\text{осн}} + t^{\text{доп}}), \quad (3.4)$$

где $t^{\text{сов}}$ – время движения на совмещенном маршруте, ч;

$t^{\text{осн}}$ – время движения на основном маршруте, ч;

$t^{\text{доп}}$ – время движения на дополнительном маршруте, ч.

Нижняя граница критерия целесообразности заезда получается из условия (3.2), если приравнять его к нулю. Исходя из этого, получаем уравнение целесообразности совмещения в виде:

$$\sum_{i=1}^{\Pi} t_i^{3p} (\Gamma_i - 1) - t^{\text{сов}} + (t^{\text{осн}} + t^{\text{доп}}) = 0. \quad (3.5)$$

Таким образом, решение задачи формирования одного маршрута из двух сводится к определению сокращения затрат времени при их совмещении из уравнения (3.5). Оно будет зависеть от схем маршрутов движения основного и дополнительного грузов.

Рассмотрим различные схемы маршрутов движения и установим взаимосвязь между ними. На рис. 3.1 представлена достаточно обобщенная схема совместной доставки грузов, когда пункты разгрузки основного и дополнительного грузов не совпадают и заезд за последним осуществляется с некоторого промежуточного пункта m . Здесь: в вершине A расположено АТП; в вершине B – пункт погрузки основного груза; C – первый пункт разгрузки основного груза на первом этапе и последний пункт разгрузки дополнительного груза на третьем этапе; D – вершина в которой загружается дополнительный груз; E – первый пункт разгрузки одновременно двух грузов – первый пункт совместной развозки; K – конечный пункт совместной развозки; n – конечный пункт разгрузки основного груза; F – первый пункт разгрузки на третьем этапе, когда развозится только дополнительный груз.

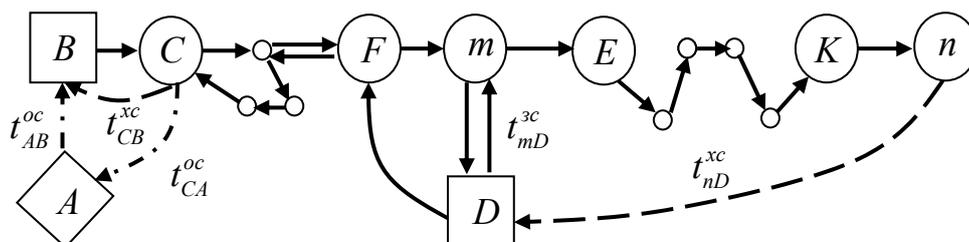


Рис. 3.1. Граф совместной развозки груза

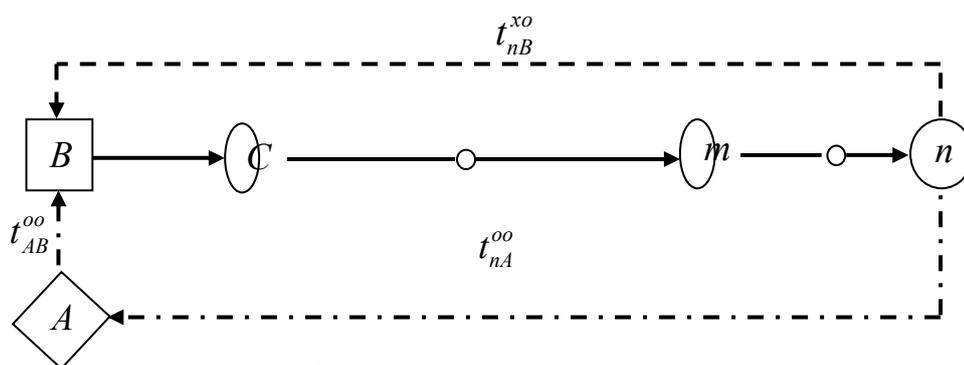


Рис. 3.2. Граф основного маршрута

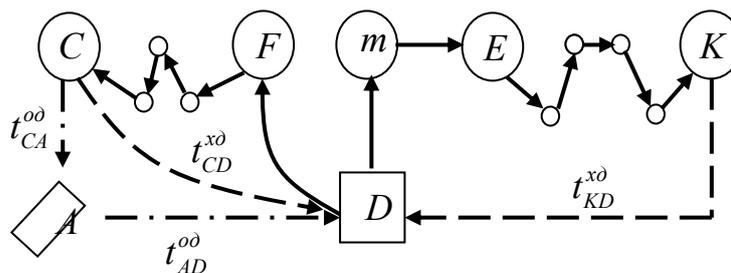


Рис. 3.3. Граф дополнительного маршрута

Следует подчеркнуть, что эффект от объединения уже существующих маршрутов отдельных маршрутов доставки основного и дополнительного грузов будет отличаться от такового, если организовать новый заезд за вторым грузом.

В отличие от совмещенного (рис. 3.1), на основном и дополнительном маршрутах (рис. 3.2 и 3.3), в верхнем индексе буква «с» заменяется на буквы «о» или «д», соответственно.

В общем случае совместный маршрут доставки мелкопартионных грузов можно разделить на три характерных этапа: первый – когда развозят только основной груз, второй – когда происходит совместная перевозка и разгрузка двух грузов и третий – когда развозится только дополнительный груз.

Время, затраченное на основном маршруте, будет равно:

$$t^{\text{осн}} = t_{AB}^{\text{оо}} + (t_{Bm}^{\Gamma\text{o}} + t_{mn}^{\Gamma\text{o}} + t_{nB}^{\text{xо}})Ц - t_{nB}^{\text{xо}} + t_{nA}^{\text{оо}}. \quad (3.6)$$

Аналогично на дополнительном маршруте:

$$t^{\text{доп}} = t_{AD}^{\text{од}} + (t_{DK}^{\Gamma\text{д}} + t_{KD}^{\text{xд}} + t_{DF}^{\Gamma\text{д}} + t_{FC}^{\Gamma\text{д}} + t_{CD}^{\text{xд}})Ц - t_{CD}^{\text{xд}} + t_{CA}^{\text{од}}. \quad (3.7)$$

На совмещенном маршруте затраченное время есть:

$$t^{\text{сов}} = t_{AB}^{\text{оо}} + (t_{Bm}^{\Gamma\text{с}} + t_{mD}^{\text{зс}} + t_{Dn}^{\Gamma\text{с}} + t_{nD}^{\text{xс}} + t_{DF}^{\Gamma\text{с}} + t_{Fc}^{\Gamma\text{с}} + t_{CB}^{\text{xс}})Ц - t_{CB}^{\text{xс}} + t_{CA}^{\text{оо}}. \quad (3.8)$$

В формулах (3.6) – (3.8) «Ц» – количество циклов заездов. Буква «Г» – в верхнем индексе обозначает пробег с грузом. Как следует из рис. 3.1 и формулы (3.8), время, затрачиваемое на совместный маршрут, складывается из времени: нулевого пробега из АТП в первый пункт погрузки $t_{AB}^{\text{оо}}$; грузового пробега до пункта заезда за дополнительным грузом $t_{Bm}^{\Gamma\text{с}}$; заезда во второй пункт погрузки $t_{mD}^{\text{зс}}$; грузового пробега от вершины D до пункта n совместной развозки $t_{Dn}^{\Gamma\text{с}}$; холостого пробега от пункта n до вершины D $t_{nD}^{\text{xс}}$, когда требуется дозагрузка дополнительный грузом для его развозки на третьем этапе; грузового пробега от вершины D до пункта C , когда развозится только дополнительный груз ($t_{DF}^{\Gamma\text{с}} + t_{Fc}^{\Gamma\text{с}}$); холостого пробега $t_{CB}^{\text{xс}}$, если необходимо совершить не один цикл движения по данному маршруту; нулевого пробега $t_{CA}^{\text{оо}}$, если $Ц = 1$ и автомобиль возвращается в исходный пункт АТП.

Решение задачи формирования одного маршрута из двух сводится к определению сокращения затрат времени при их совмещении и нахождении максимально допустимого времени на заезд за дополнительным грузом.

Изменение времени движения по формуле (3.4) составит:

$$\Delta t^\ell = t_{mD}^{3c} \cdot \Pi - t_{nA}^{00} - t_{AD}^{0d} + (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd}) + \\ + (t_{Dn}^{\Gamma c} - t_{DK}^{\Gamma d} - t_{mn}^{\Gamma o}) \cdot \Pi + (t_{nD}^{xc} - t_{KD}^{xd}) \cdot \Pi. \quad (3.9)$$

Экономия времени, согласно (3.2), будет равна:

$$\Delta t = \Pi \cdot \sum_{i=1}^{\Pi} t_i^{3p} - t_{mD}^{3c} \cdot \Pi + t_{nA}^{00} + t_{AD}^{0d} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd}) - \\ - (t_{Dn}^{\Gamma c} - t_{DK}^{\Gamma d} - t_{mn}^{\Gamma o}) \cdot \Pi - (t_{nD}^{xc} - t_{KD}^{xd}) \cdot \Pi. \quad (3.10)$$

Тогда критерий целесообразности (3.5), в общем виде выражающий максимально допустимое время заезда за дополнительным грузом, запишется в виде:

$$\left[t_{mD}^{3c} \right] = \frac{t_{nA}^{00} + t_{AD}^{0d} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd})}{\Pi} + \sum_{i=1}^{\Pi} t_i^{3p} - t_{Dn}^{\Gamma c} + t_{DK}^{\Gamma d} + t_{mn}^{\Gamma o} - t_{nD}^{xc} + t_{KD}^{xd}. \quad (3.11)$$

Из анализа формулы (3.11) следует, что максимально допустимое время заезда зависит от времени, затрачиваемого на грузовые, холостые и нулевые пробеги, а также от времени заезда в пункты разгрузки и количества циклов движения.

В частном случае при одном цикле заезда, $\Pi=1$, получим:

$$\left[t_{mD}^{3c} \right] = \sum_{i=1}^{\Pi} t_i^{3p} + t_{nA}^{00} + t_{AD}^{0d} - t_{Dn}^{\Gamma c} + t_{DK}^{\Gamma d} + t_{mn}^{\Gamma o} - t_{nD}^{xc} + t_{KD}^{xd}. \quad (3.12)$$

Как следует из анализа выражений (3.11) и (3.12), эффективность объединения маршрутов и максимальное допустимое время заезда возрастает с увеличением:

- затрат времени на заезд в пункт разгрузки и их количество $\sum_{i=1}^{\Pi} t_i^{3p}$;
- дальности расположения конечного пункта развозки t_{nA}^{00} ;
- дальности расположения второго пункта погрузки t_{AD}^{0d} ;
- экономии времени груженого пробега на участке совместной развозки двух грузов:

$$\Delta t^\Gamma = t_{mn}^{\Gamma o} + t_{DK}^{\Gamma d} - t_{Dn}^{\Gamma c};$$

- экономия времени на холостых пробегах между дополнительным и совмещенным маршрутами до пункта погрузки второго груза:

$$\Delta t^x = t_{KD}^{xd} - t_{nD}^{xc}.$$

Таким образом, объединение маршрутов целесообразнее производить для более удаленных от АТП пунктов разгрузки и пункта погрузки второго

груза, во-первых. Во-вторых, для этого выгоднее использовать автомобиль большей грузоподъемности, позволяющий объединять большее количество пунктов разгрузки.

Перейдем к рассмотрению вопроса об экономии времени при организации второго вида совместной развозки, когда обслуживание второго производителя данным АТП ранее не выполнялось. Отметим, что в этом случае расчетная схема получается аналогичная, приведенной на рис. 3.1. Изменение времени будет равно:

$$\Delta t = \Pi \cdot \sum_{i=1}^{\Pi} t_i^{3p} - t_{mD}^{3c} \cdot \Pi + t_{nA}^{oo} - t_{CA}^{oc} - \\ - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo}) - \Pi(t_{Dn}^{\Gamma c} + t_{DF}^{\Gamma c} + t_{FC}^{\Gamma c} - t_{mn}^{\Gamma o}) - t_{nD}^{xc} \cdot \Pi. \quad (3.13)$$

Отсюда найдем максимально допустимое время заезда за дополнительным грузом:

$$\left[t_{mD}^{3c} \right] = \frac{t_{nA}^{oo} - t_{CA}^{oc} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo})}{\Pi} + \sum_{i=1}^{\Pi} t_i^{3p} - t_{Dn}^{\Gamma c} - t_{DF}^{\Gamma c} - t_{FC}^{\Gamma c} + t_{mn}^{\Gamma o} - t_{nD}^{xc}. \quad (3.14)$$

По структуре формулы (3.14) аналогично выражению (3.11) и, следовательно, в данном случае эффективность организации тоже будет зависеть от холостых, нулевых и груженых пробегов, а так же времени заезда в пункты разгрузки и количество циклов движения.

При $\Pi=1$ найдем:

$$\left[t_{mD}^{3c} \right] = \sum_{i=1}^{\Pi} t_i^{3p} + t_{nA}^{oo} - t_{CA}^{oc} - t_{Dn}^{\Gamma c} - t_{DF}^{\Gamma c} - t_{FC}^{\Gamma c} + t_{mn}^{\Gamma o} - t_{nD}^{xc}. \quad (3.15)$$

В результате исследования формул (3.14) и (3.15) установлено, что максимально допустимое время заезда и эффективность организации попутного маршрута будут возрастать:

- при увеличении затрат времени на заезд в пункт погрузки и их количества $\sum_{i=1}^{\Pi} t_i^{3p}$;
 - с увеличением дальности расположения последнего пункта разгрузки основного груза t_{nA}^{oo} ;
 - при уменьшении расстояния до расположения последнего пункта совместной развозки t_{CA}^{oc} ;
 - с увеличением разности времени груженого пробега с одним и двумя грузами после второго пункта погрузки:
- $$\Delta t^{\Gamma} = t_{mn}^{\Gamma o} - (t_{Dn}^{\Gamma c} + t_{DF}^{\Gamma c} + t_{FC}^{\Gamma c});$$
- с уменьшением времени холостого пробега t_{nD}^{xc} .

Таким образом, для второго вида совместной развозки так же можно утверждать, что целесообразнее использование автомобиля большей грузоподъемности. Сравнения формул (3.12) и (3.15) показало, что эффективность организации совместной развозки первого вида выше, так как слагаемые t_{CA}^{oc} , $t_{DF}^{\Gamma c}$, $t_{FC}^{\Gamma c}$ и t_{nD}^{xc} входят в формулу (3.15) со знаком минус.

Рассмотрим некоторые частные случаи совместной развозки двух видов.

На рис. 3.4 представлена схема движения совместного маршрута, когда не требуется повторный заезд в вершину D . Следовательно, из рис. 3.1 исключаются холостые пробеги t_{nD}^{xc} и t_{KD}^{xd} , а так же появляется время на грузовой пробег $t_{nF}^{\Gamma c}$ из пункта n в пункт F .

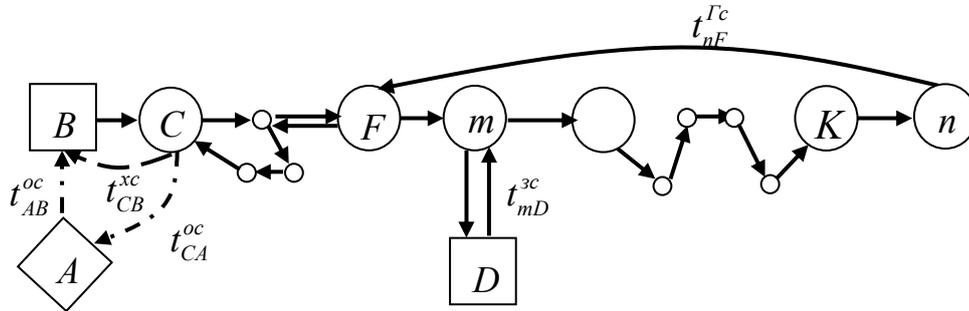


Рис. 3.4. Граф совместной развозки при $t_{nD}^{xc}=0$

Решение находится по формулам (3.16) – (3.18) из выражений (3.10) – (3.12), аналогично рассмотренному выше, при $t_{nD}^{xc} = t_{KD}^{xd} = 0$. Если цикл движения один, то максимальное допустимое время заезда $[t_{mD}^{zc}]$ за дополнительным грузом по выражению (3.18) не зависит от холостых пробегов, в отличие от формулы (3.12).

$$\Delta t = \Pi \sum_{i=1}^{\Pi} t_i^{zp} - t_{mD}^{zc} \Pi + t_{nA}^{oo} + t_{AD}^{од} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd}) - (t_{DnF}^{\Gamma c} - t_{DKF}^{\Gamma d} - t_{mn}^{\Gamma o}) \Pi. \quad (3.16)$$

$$[t_{mD}^{zc}] = \frac{t_{nA}^{oo} + t_{AD}^{од} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd})}{\Pi} + \sum_{i=1}^{\Pi} t_i^{zp} - t_{DnF}^{\Gamma c} + t_{DKF}^{\Gamma d} + t_{mn}^{\Gamma o}. \quad (3.17)$$

При $\Pi=1$ получим:

$$[t_{mD}^{zc}] = \sum_{i=1}^{\Pi} t_i^{zp} + t_{nA}^{oo} + t_{AD}^{од} - t_{DnF}^{\Gamma c} + t_{DKF}^{\Gamma d} + t_{mn}^{\Gamma o}. \quad (3.18)$$

Следует заметить, что структура полученных формул (3.16) – (3.18) аналогично выражениям (3.10) – (3.12).

Для попутного маршрута из формул (3.13) – (3.15) найдем:

$$\Delta t = \Pi \cdot \sum_{i=1}^{\Pi} t_i^{3p} - t_{mD}^{3c} \cdot \Pi + t_{nA}^{oo} - t_{CA}^{oc} -$$

$$-(\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo}) - \Pi(t_{Dn}^{\Gamma c} + t_{nF}^{\Gamma c} + t_{FC}^{\Gamma c} - t_{mn}^{\Gamma o}). \quad (3.19)$$

$$[t_{mD}^{3c}] = \frac{t_{nA}^{oo} - t_{CA}^{oc} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo})}{\Pi} + \sum_{i=1}^{\Pi} t_i^{3p} - t_{Dn}^{\Gamma c} - t_{nF}^{\Gamma c} - t_{FC}^{\Gamma c} + t_{mn}^{\Gamma o}. \quad (3.20)$$

При $\Pi=1$ получим:

$$[t_{mD}^{3c}] = \sum_{i=1}^{\Pi} t_i^{3p} + t_{nA}^{oo} - t_{CA}^{oc} - t_{Dn}^{\Gamma c} - t_{nF}^{\Gamma c} - t_{FC}^{\Gamma c} + t_{mn}^{\Gamma o}. \quad (3.21)$$

Выведенные формулы (19) – (21) по структуре аналогичны выражениям (3.13) – (3.15).

На рис. 3.5 и 3.6 представлены схемы движения аналогичные рис. 3.1 и 3.4, соответственно, но когда пункты разгрузки основного и дополнительного грузов совпадают. Тогда в формулах (3.11) – (3.12) исключаются не только холостые пробеги $t_{nD}^{xc} = t_{KD}^{xd} = 0$, но и грузовые пробеги, так как $t_{Dn}^{\Gamma c} - t_{DK}^{\Gamma d} = 0$. Из них найдем

$$\Delta t = \Pi \sum_{i=1}^{\Pi} t_i^{3p} - t_{mD}^{3c} \Pi + t_{nA}^{oo} + t_{AD}^{од} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd}) + t_{mn}^{\Gamma o} \Pi, \quad (3.22)$$

$$[t_{mD}^{3c}] = \frac{t_{nA}^{oo} + t_{AD}^{од} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd})}{\Pi} + \sum_{i=1}^{\Pi} t_i^{3p} + t_{mn}^{\Gamma o}. \quad (3.23)$$

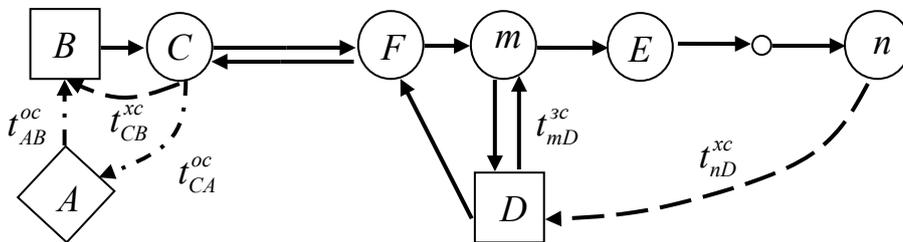


Рис. 3.5. Граф совместной развозки при совпадении пунктов основного и дополнительного маршрутов

При $\Pi=1$ получим:

$$\left[t_{mD}^{3c} \right] = \sum_{i=1}^{\Pi} t_i^{3p} + t_{nA}^{oo} + t_{AD}^{od} + t_{mn}^{\Gamma o}. \quad (3.24)$$

В результате анализа выражений (3.11) и (3.23) установлено, что наибольшее допустимое время заезда для первого вида совместной развозки получается при совмещении наибольшего количества пунктов разгрузки основного и дополнительного груза, согласно схемам движения на рис. 3.5 и 3.6. Для дополнительного маршрута по рис. 3.5 формула определения экономия времени совпадает с выражениями (3.14) и (3.15). При движении по рис. 3.6 можно использовать выражения (3.19) – (3.21).

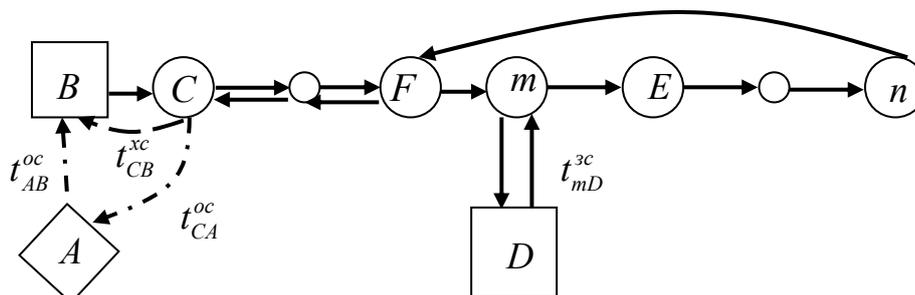


Рис. 3.6. Граф совместной развозки при $t_{nD}^{xc}=0$ и совпадении маршрутов

Рассмотрим схему маршрута, показанную на рис. 3.7, когда заезд за дополнительным грузом производится из конечного пункта n в вершину D и его развозка осуществляется в обратном направлении из n в C . Вследствие отсутствия второго этапа в выражениях (3.22) – (3.24), экономии времени на заезды $\sum_{i=1}^{\Pi} t_i^{3p}$ не будет и $t_{mn}^{\Gamma o} = 0$. Однако в полученных формулах (3.26) – (3.28) присутствуют количество циклов и время, затрачиваемое на холостые пробеги.

$$\Delta t = t_{nA}^{oo} + t_{AD}^{od} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd}) - t_{nD}^{3c} \Pi. \quad (3.26)$$

$$\left[t_{mD}^{3c} \right] = \frac{t_{nA}^{oo} + t_{AD}^{od} - (\Pi - 1)(t_{CB}^{xc} - t_{nB}^{xo} - t_{CD}^{xd})}{\Pi}. \quad (3.27)$$

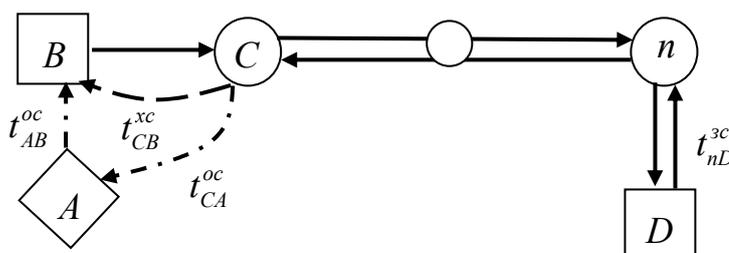


Рис. 3.7. Граф совместной развозки из конечного пункта разгрузки основного груза при совпадении маршрутов

При $\Psi=1$ найдем:

$$\left[t_{nD}^{3c} \right] = t_{nA}^{oo} + t_{AD}^{од}. \quad (3.28)$$

Как установлено из сравнения выражений (3.26) – (3.28) с формулами (3.22) – (3.24), организация совместной развозки первого вида при заезде за дополнительным грузом с последнего пункта разгрузки основного груза менее выгодна.

Если рассматривать второй вид совместной развозки, то из формул (3.19) – (3.21) получим:

$$\Delta t = t_{nD}^{3c} \cdot \Psi + t_{nA}^{oo} - t_{CA}^{oc} - (\Psi - 1)(t_{CB}^{xc} - t_{nB}^{xo}) - t_{Dn}^{\Gamma c} \Psi - t_{nC}^{\Gamma c} \Psi. \quad (3.29)$$

Так как $t_{nD}^{3c} = t_{Dn}^{\Gamma c}$, то следует:

$$2 \left[t_{nD}^{3c} \right] = \frac{t_{nA}^{oo} - t_{CA}^{oc} - (\Psi - 1)(t_{CB}^{xc} - t_{nB}^{xo})}{\Psi} + t_{nC}^{\Gamma c}. \quad (3.30)$$

При $\Psi=1$ найдем:

$$2 \left[t_{nD}^{3c} \right] = t_{nA}^{oo} - t_{CA}^{oc} - t_{nC}^{\Gamma c}. \quad (3.31)$$

Сравнение полученных выражений (3.29) – (3.31) с формулами (3.19) – (3.21) показала, что если время груженого пробега на участке $n - C$ $t_{nC}^{\Gamma c}$ по рис. 3.4 меньше, чем по рис. 3.7: $t_{nC}^{\Gamma c} \leq t_{nF}^{\Gamma c} + t_{FC}^{\Gamma c}$, то организация совместной развозки второго вида по рисунку 3.4 является более выгодной.

Приведём результаты расчета схемы маршрута, представленного на рис. 3.8, когда пункты разгрузки основного и дополнительного грузов не совпадают:

$$t^o = t_{AB}^{oo} + (t_{Bm}^{\Gamma o} + t_{mB}^{xo}) \Psi - t_{mB}^{xo} + t_{mA}^{oo};$$

$$t^{\Delta} = t_{AD}^{од} + (t_{Dn}^{\Gamma \Delta} + t_{nD}^{x\Delta}) \Psi - t_{nD}^{x\Delta} + t_{nA}^{од}; \quad (3.32)$$

$$t^c = t_{AB}^{oc} + (t_{Bm}^{\Gamma c} + t_{mD}^{3c} + t_{Dn}^{\Gamma c} + t_{nB}^{xc}) \Psi - t_{nB}^{xc} + t_{nA}^{oc}; \quad (3.33)$$

$$\Delta t = -(\Psi - 1)(t_{nB}^{xc} - t_{mB}^{xo} - t_{nD}^{x\Delta}) + t_{mA}^{oo} + t_{AD}^{од} - \Psi t_{mD}^{3c}. \quad (3.34)$$

При $\Delta t = 0$:

$$\left[t_{mD}^{3c} \right] = \frac{t_{mA}^{oo}}{\Psi} + \frac{t_{AD}^{од}}{\Psi} - \frac{(\Psi - 1)(t_{nB}^{xc} - t_{mB}^{xo} - t_{nD}^{x\Delta})}{\Psi}. \quad (3.35)$$

При $\Pi = 1$:

$$\left[t_{mD}^{3c} \right] = t_{mA}^{oo} + t_{AD}^{od} \quad (3.36)$$

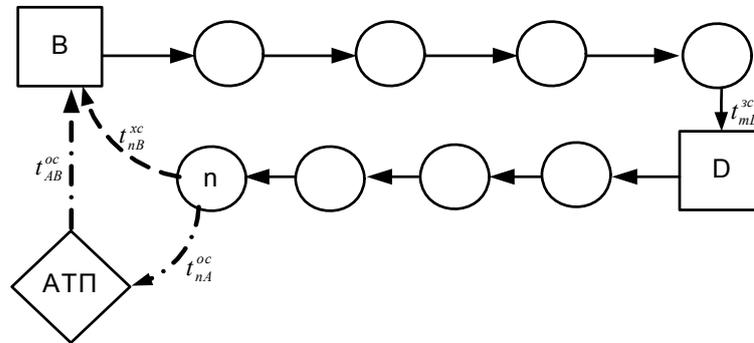


Рис. 3.8. Граф совместной развозки из конечного пункта разгрузки основного груза при несовпадении маршрутов

Таким образом, на основании проведенного анализа различных схем маршрутов совместной доставки грузов с помощью предложенного критерия целесообразности (3.2) установлено:

- для первого вида совмещения целесообразнее объединять пункты наиболее удаленные от АТП;
- наибольший эффект достигается, когда пункты разгрузки для основного и дополнительного грузов совпадают;
- эффект объединения маршрутов возрастает с увеличением количества пунктов и времени заезда в них;
- второй вид совмещения, как правило, менее выгоден по сравнению с первым;
- для организации совместной развозки необходимо использовать автомобиль большей грузоподъемности.

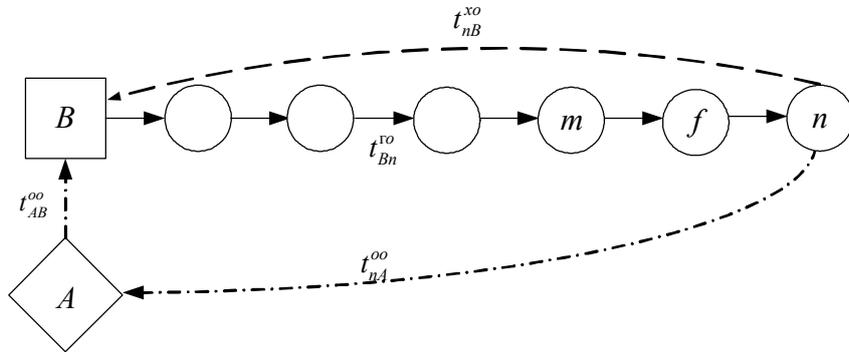
Рассмотрим более сложный вариант организации совмещенного маршрута, когда оба предприятия имеют и основной и дополнительный маршруты (рис. 3.9). При этом нижний индекс означает номер производителя груза.

На рис. 3.10 представлен граф совмещенного маршрута, который получился в результате объединения маршрутов: $t_1^o, t_2^o, t_1^d, t_2^d$. Время движения на каждом из них соответственно равно:

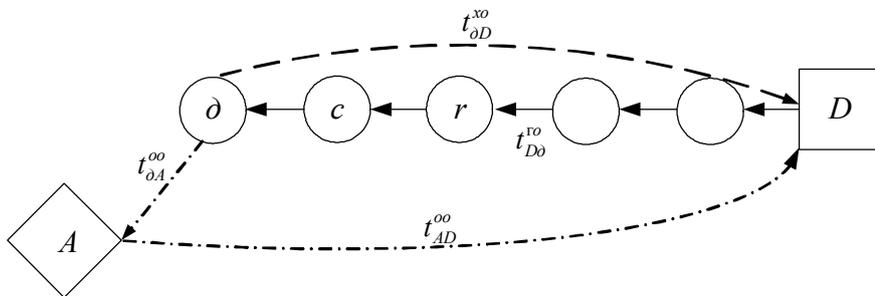
$$\begin{aligned} t_1^o &= t_{AB}^{oo} + (t_{Bn}^{go} + t_{nB}^{xo})\Pi - t_{nB}^{xo} + t_{nA}^{oo}; \\ t_2^o &= t_{AD}^{oo} + (t_{Dd}^{go} + t_{dD}^{xo})\Pi - t_{dD}^{xo} + t_{dA}^{oo}; \\ t_1^d &= t_{AD}^{od} + (t_{Dv}^{gd} + t_{vD}^{xd})\Pi - t_{vD}^{xd} + t_{vA}^{od}; \end{aligned} \quad (3.37)$$

$$t_2^d = t_{AB}^{од} + (t_{Ba}^{гд} + t_{aB}^{хд}) \Pi - t_{aB}^{хд} + t_{aA}^{од};$$

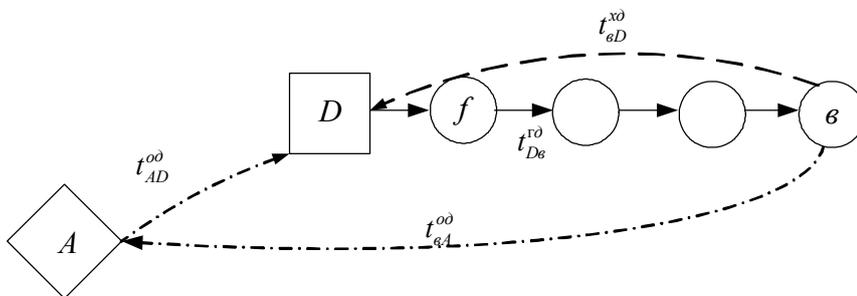
$$t^c = t_{AB}^{оc} + (t_{Bm}^{гc} + t_{mD}^{3c} + t_{De}^{гc} + t_{eD}^{3c} + t_{Dr}^{гc} + t_{rB}^{3c} + t_{By}^{гc} + t_{yB}^{xc}) \Pi - t_{yB}^{xc} + t_{yA}^{оc}.$$



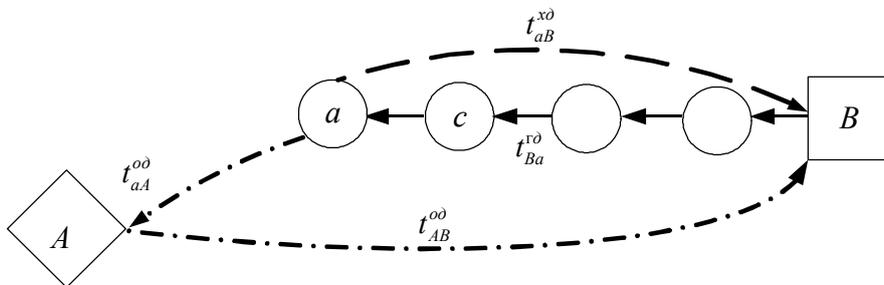
Основной маршрут t_1^o



Основной маршрут t_2^o



Дополнительный маршрут t_1^o



Дополнительный маршрут t_2^o

Рис. 3.9. Графы основных и дополнительных маршрутов до совмещения

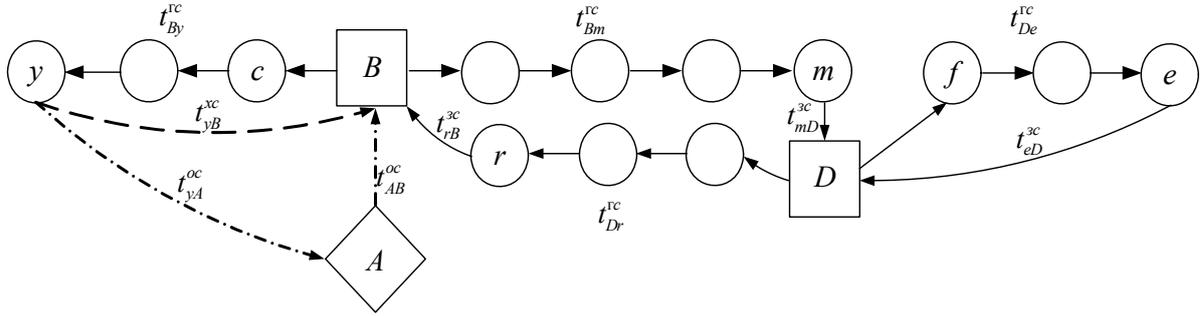


Рис. 3.10. Граф совмещенного маршрута t^c

Экономия времени от совмещения будет равна:

$$\Delta t^\ell = t^c - (t_1^o + t_2^o + t_1^d + t_2^d). \quad (3.38)$$

Если при совмещении маршрутов порядок объезда пунктов доставки не совпадает, то (3.38) примет вид:

$$\Delta t^\ell = \Pi \Sigma t^{\text{rc}} - \Pi \Sigma t^{\text{ro}} - \Pi \Sigma t^{\text{rd}} + \Pi \Sigma t^{\text{zc}} - (1 - \Pi)(t^{\text{xc}} - \Sigma t^{\text{xo}} - \Sigma t^{\text{xd}}) - \Sigma t_1^o + \Sigma t_2^o, \quad (3.39)$$

где $\Sigma t^{\text{rc}} = t_{Bm}^{\text{rc}} + t_{Dc}^{\text{rc}} + t_{Dr}^{\text{rc}} + t_{By}^{\text{rc}}$;

$$\Sigma t^{\text{ro}} = t_{Bn}^{\text{ro}} + t_{Dd}^{\text{ro}};$$

$$\Sigma t^{\text{rd}} = t_{De}^{\text{rd}} + t_{Ba}^{\text{rd}};$$

$$\Sigma t^{\text{zc}} = t_{mD}^{\text{zc}} + t_{eD}^{\text{zc}} + t_{rB}^{\text{zc}}; \quad (3.40)$$

$$\Sigma t^{\text{xo}} = t_{nB}^{\text{xo}} + t_{dD}^{\text{xo}};$$

$$\Sigma t^{\text{rd}} = t_{BD}^{\text{rd}} + t_{ab}^{\text{rd}};$$

$$\Sigma t_1^o = t_{AD}^{\text{oo}} + t_{AD}^{\text{od}} + t_{AB}^{\text{od}};$$

$$\Sigma t_2^o = t_{yA}^{\text{oc}} - t_{nA}^{\text{oo}} - t_{dA}^{\text{oo}} - t_{vA}^{\text{od}} - t_{aA}^{\text{od}}.$$

Если выражение (3.39) подставить в выражение (3.2), при условии что $\Delta t = 0$, то получим максимально допустимое время заезда:

$$\left[\Sigma t^{\text{zc}} \right] = \Sigma t^{\text{zp}} - \Sigma t^{\text{rc}} + \Sigma t^{\text{ro}} + \Sigma t^{\text{rd}} + (1 - \Pi) \frac{t^{\text{xc}} - \Sigma t^{\text{xo}} - \Sigma t^{\text{xd}}}{\Pi} - \frac{\Sigma t_1^o}{\Pi} + \frac{\Sigma t_2^o}{\Pi}. \quad (3.41)$$

При $\Pi = 1$ из выражения (3.41) имеем:

$$\left[\Sigma t^{\text{zc}} \right] = \Sigma t^{\text{zp}} - \Sigma t^{\text{rc}} + \Sigma t^{\text{ro}} + \Sigma t^{\text{rd}} - \Sigma t_1^o + \Sigma t_2^o. \quad (3.42)$$

Если при совмещении маршрутов порядок объезда пунктов доставки совпадает после заезда во второй пункт погрузки, то, согласно выражению(3.38):

$$\Delta t^\ell = \Pi \Sigma t^{\Gamma c} - \Pi \Sigma t^{\Gamma o} + \Pi \Sigma t^{3c} + (1 - \Pi) \Pi \Sigma t^{x0} + t^{xд} - \Sigma t_1^o - \Sigma t_2^o, \quad (3.43)$$

где $\Sigma t^{\Gamma c} = t_{Bm}^{\Gamma c} + t_{Dr}^{\Gamma c}$;

$$\Sigma t^{\Gamma o} = t_{Bn}^{\Gamma o} + t_{Dд}^{\Gamma o} ;$$

$$\Sigma t^{3c} = t_{mD}^{3c} + t_{rB}^{3c} ; \quad (3.44)$$

$$\Sigma t^{x0} = t_{nB}^{x0} + t_{дD}^{x0} ;$$

$$\Sigma t_1^o = t_{AD}^{oo} + t_{AD}^{од} + t_{AB}^{од} ;$$

$$\Sigma t_2^o = t_{nA}^{oo} + t_{дA}^{oo} + t_{BA}^{од} .$$

Подставив (3.43) в выражение (3.2) при $\Delta t = 0$ найдем:

$$\left[\Sigma t^{3c} \right] = \Sigma t^{3p} - \Sigma t^{\Gamma c} + \Sigma t^{\Gamma o} - (1 - \Pi) \frac{\Sigma t^{x0}}{\Pi} - \frac{t^{xд}}{\Pi} + \Sigma t_1^o + \Sigma t_2^o . \quad (3.45)$$

При $\Pi = 1$ из формулы (3.45) найдем:

$$\left[\Sigma t^{3c} \right] = \Sigma t^{3p} - \Sigma t^{\Gamma c} + \Sigma t^{\Gamma o} - t^{xд} + \Sigma t_1^o + \Sigma t_2^o . \quad (3.46)$$

Если при совмещении порядок объезда пунктов доставки совпадает на всем маршруте, то согласно формуле (3.38):

$$\Delta t^\ell = \Pi \Sigma t^{\Gamma o} + \Pi \Sigma t^{3c} + (1 - \Pi) \Pi \Sigma t^{x0} + t^{xд} - \Sigma t_1^o - \Sigma t_2^o, \quad (3.47)$$

где $\Sigma t^{\Gamma o} = t_{mf}^{\Gamma o} + t_{fn}^{\Gamma o} + t_{rc}^{\Gamma o} + t_{cd}^{\Gamma o}$;

$$\Sigma t^{3c} = t_{mD}^{3c} + t_{rB}^{3c} ;$$

$$\Sigma t^{x0} = t_{nB}^{x0} + t_{дD}^{x0} ; \quad (3.48)$$

$$\Sigma t_1^o = t_{AD}^{oo} + t_{AD}^{од} + t_{AB}^{од} ;$$

$$\Sigma t_2^o = t_{nA}^{oo} + t_{дA}^{oo} + t_{BA}^{од} .$$

Подставив (3.47) в выражение (3.2) получим, при условии что $\Delta t = 0$:

$$\left[\Sigma t^{3c} \right] = \Sigma t^{3p} - \Sigma t^{\Gamma o} - (1 - \Pi) \frac{\Sigma t^{x0}}{\Pi} - \frac{t^{xд}}{\Pi} + \Sigma t_1^o + \Sigma t_2^o . \quad (3.49)$$

При $\Pi = 1$ из выражения (3.49) имеем:

$$\left[\Sigma t^{3c} \right] = \Sigma t^{3p} - \Sigma t^{\Gamma o} - t^{xд} + \Sigma t_1^o + \Sigma t_2^o . \quad (3.50)$$

Из анализа полученных формул (3.41) - (3.50) следует, что допустимое время заезда зависит от комбинации грузового, холостого и нулевого пробогов на основном, дополнительном и совмещенном маршрутах, экономии времени заезда в пунктах разгрузки, их количества, а также количества совершаемых циклов перевозки.

Следует отметить, что в общем случае проанализировать влияние указанных факторов на время заезда невозможно, ввиду сложности задачи. Поэтому в качестве примера исследуем схему передвижения, показанную на рис. 3.6, когда пункты разгрузки первого и второго груза совпадают и Ц=1. Выражение (3.24) представим в виде:

$$[t_{mD}^{3c}] = \sum_{i=1}^{\Pi} t_i^{3p} + t^o + \frac{\ell_{mn}^{\Gamma}}{V}, \quad (3.51)$$

где $t^o = t_{nA}^{oo} + t_{AD}^{od}$ – время, затрачиваемое на нулевые пробеги при перевозке основного и дополнительного груза, ч.;

ℓ_{mn}^{Γ} – длина второго груженого участка при совместной развозке двух грузов, км.;

V – техническая скорость, км/ч.

Для анализа полученной зависимости (3.24) используем характеристический график (рис.3.11).

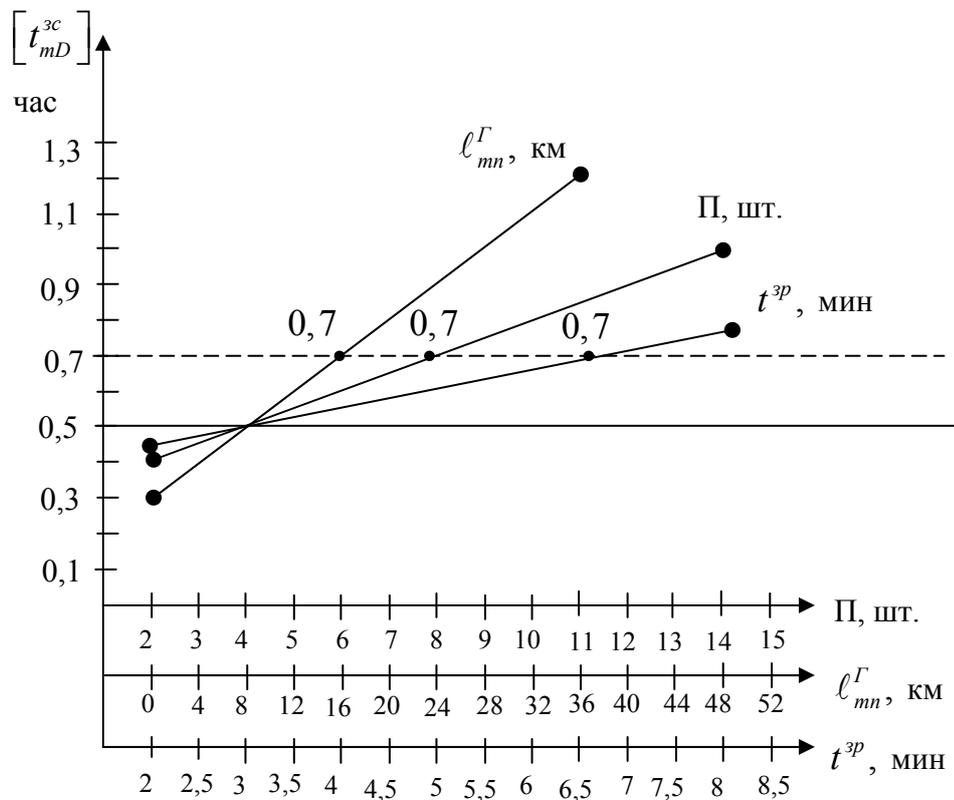


Рис. 3.11.Характеристический график

Примем следующие исходные данные: $V = 40$ км/ч, $t^0 = 0,1$ ч. В качестве переменных используется: ℓ_{mn}^Γ – длина груженого пробега; Π – количество пунктов разгрузки; t^{3p} – время заезда под разгрузку. При $\Pi = 4$, $\ell_{mn}^\Gamma = 8$ км и $t^{3p} = 3$ мин максимально допустимое время заезда за дополнительным грузом не должно превышать 0,5 ч. Чтобы увеличить $[t_{mD}^{3c}]$ до 0,7 часа в 1,4 раза, необходимо изменить: расстояние ℓ_{mn}^Γ до 16 км – в 2 раза или количество пунктов до 8 шт. – в 2 раза, или время заезда до 6 минут – в 2 раза.

Полученные данные подтверждают сделанные ранее выводы.

3.2. Определение пункта заезда и экономическая оценка при совмещении маршрутов

Как уже было сказано ранее, вторая задача совместной развозки заключается в определении оптимального пункта заезда за вторым грузом. В качестве критерия оптимальности маршрута можно принять наименьшее значение любого из параметров: длина маршрута или время движения на нём. Оптимальный маршрут движения определяется при использовании методики «фиктивных ветвей».

В качестве примера за критерий оптимальности выберем наименьшую длину маршрута. Рассмотрим задачу совместной развозки, когда у основного и дополнительного маршрутов имеется одно начало – АТП. Для перевозки используется автомобиль ГАЗ – 3302 «Газель» массой 1,65 тонны и грузоподъемностью 1,85 тонны. Масса первого груза составляет 1,8 тонны, а второго – 0,9 тонны. Время t^{np} составляет 0,5 часа за 1 тонну. Время заезда в пункт $t^3 = 0,05$ часа. Техническая скорость $V = 30$ км/ч. Решение задачи предлагается производить в семь этапов, блок-схема которых показана на рис. 3.12.

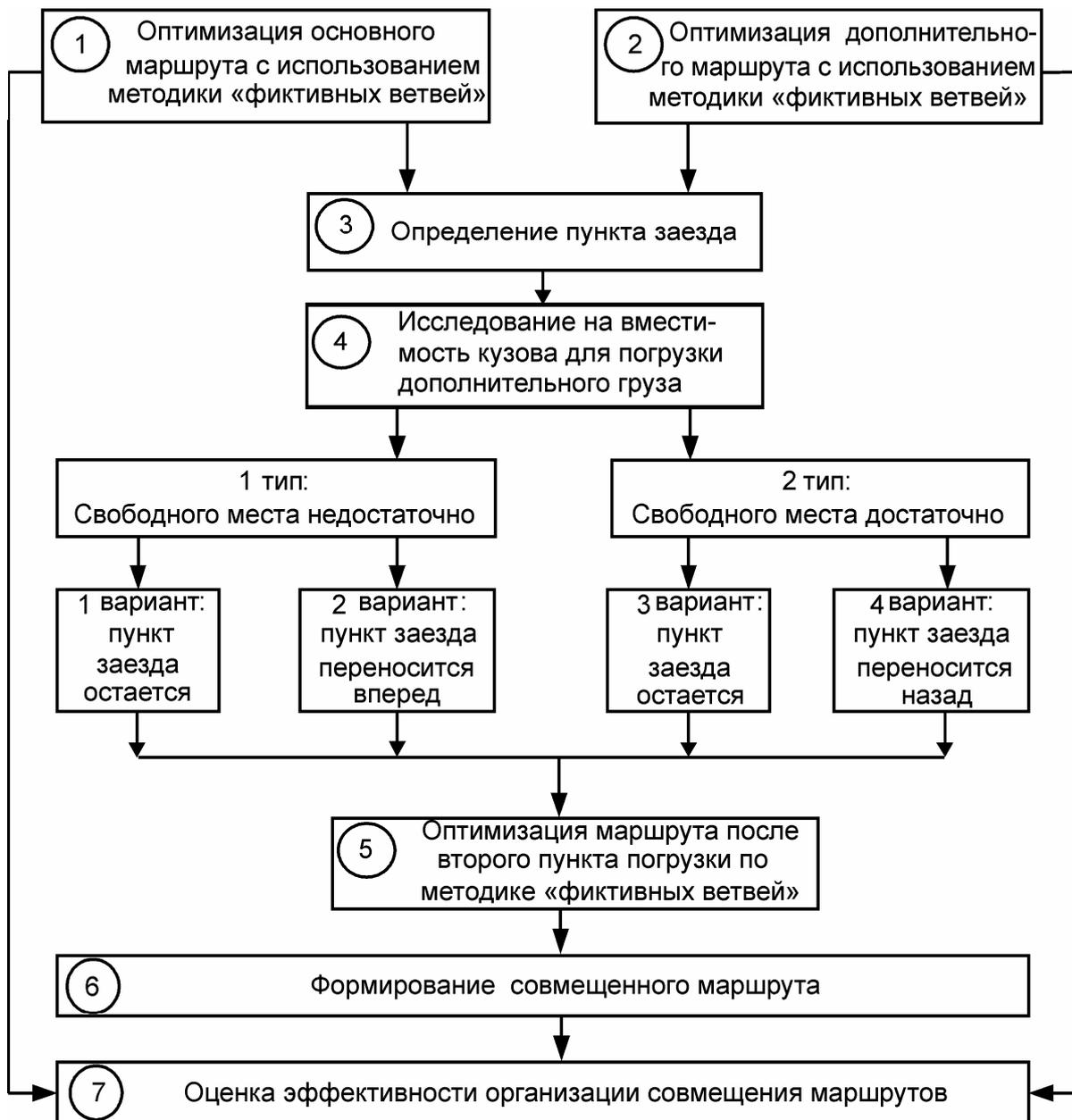


Рис. 3.12. Блок-схема

Оптимальный маршрут развозки основного груза длиной 43 км показан на рис. 3.13: 6-11-7-8-9-10-1-2-4-5-4-6. Здесь в пункте 6 расположено АТП, а в вершине 11 имеется база погрузки основного груза. Дополнительный груз развозится по оптимальному маршруту длиной 40 км согласно рис. 3.14: 6-3-9-10-1-2-4-5-4-6. Для определения пункта заезда за дополнительным грузом на третьем этапе, в транспортную схему по рис. 3.13 вводится пункт 3, где расположена база второго груза. В результате решения установлено, что заезд в вершину 3 будет осуществляться с пункта 8 (рис. 3.15). Заметим, что в общем случае полученный маршрут может отличаться от приведенного на рис. 3.15. На четвертом этапе установлено, что место в транспортном средстве для погрузки второго груза после пункта 8 освободилось достаточно. Если его будет не хватать, то необходимо

перейти к заезду из следующего пункта по ходу движения, сделав ветвь 9–3 ориентированную. На пятом этапе в транспортную схему дополнительного маршрута по рис. 3.14 добавляются пункты разгрузки первого груза, расположенные после заезда за вторым грузом из вершины 3 (рис.3.16). Совместный маршрут длиной 44 км получается путем сложения пути передвижения основного груза до пункта 3 по рис. 3.15 и пути следования после него по рис. 3.16: 6-11-7-8-3-9-10-1-2-4-5-4-6. В общем случае на нем можно выделить четыре характерных участка: первый нулевой пробег от АТП до пункта погрузки первого груза – 11; участок перевозки основного груза от вершины 11 до пункта 3; участок совместной развозки двух грузов между узлами 3 и 5; второй нулевой пробег от пункта 5 до АТП. На седьмом этапе производится экономическое обоснование полученного совместного маршрута в микросистеме [80].

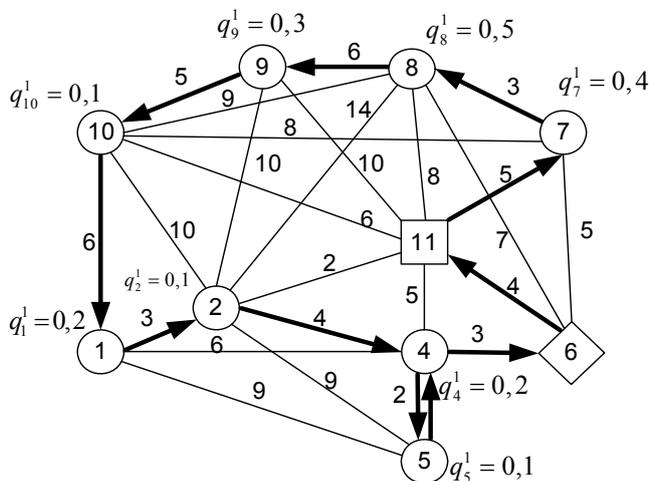


Рис. 3.13

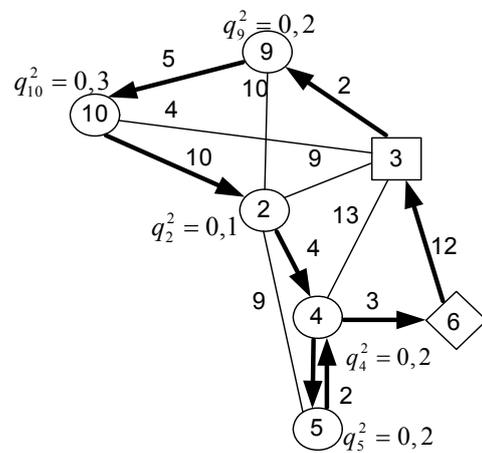


Рис. 3.14

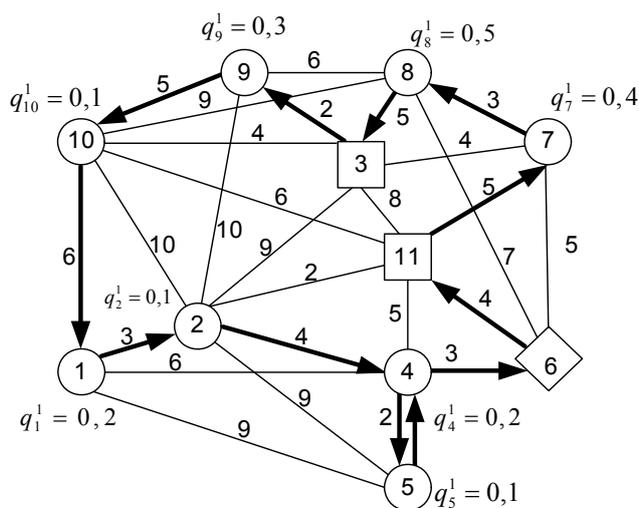


Рис. 3.15

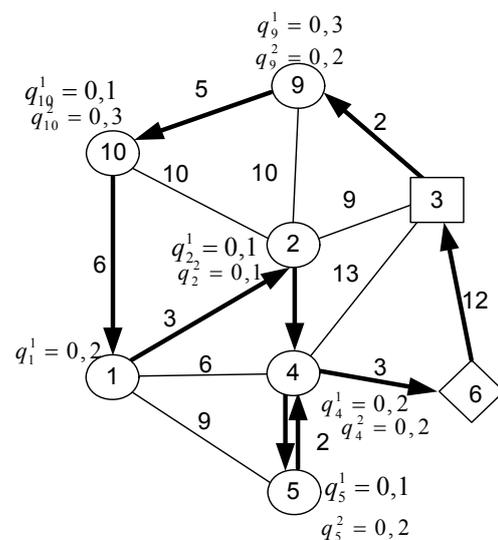


Рис. 3.16

Известно, что процесс перевозки состоит из двух базовых технологических звеньев – движения и простоя под погрузкой и разгрузкой. В свою очередь, движение бывает двух типов – с грузом и без него. Простой также предлагается разделить на технологический и организационный. Технологический простой – время, затрачиваемое непосредственно на разгрузку (погрузку) груза из кузова автомобиля и зависящее от выбранной технологии работы. Организационный простой – время, предшествующее непосредственно началу разгрузки (погрузки). Оно зависит от степени совершенствования организации заезда под погрузку или разгрузку. Следовательно, экономический эффект при совмещении основного и дополнительного маршрутов может быть достигнут за счет сокращения затрат на пробег с грузом и без него, это во-первых. Во-вторых, он может быть получен за счет уменьшения времени на организационный простой во время совместной разгрузки нескольких грузов [30–32]. Характеристики этих двух этапов перевозки различны. Для первого характерна длина пробега, а для второго – время.

Таким образом, для оценки затрат на пробег в общем случае необходимо использовать покилометровую оценку, а для простоя – почасовую. Следовательно, затраты на маршруте должны состоять из трех частей:

$$z_i = z_i^{дв} + z_i^{пр} + z_i^0, \quad (3.52)$$

где $z_i^{дв}$ – затраты на передвижение, руб.;

$z_i^{пр}$ – затраты на простой автомобиля, руб.;

z_i^0 – организационный простой на заезд в пункт, руб..

Отметим, что при движении энергия сжигаемого топлива затрачивается на работу по перемещению масс автомобиля и груза. Работу на перемещении груза считаем независимой от таковой для автомобиля. Поэтому затраты на маршруте при движении найдем по формуле:

$$z_i^{дв} = z_i^A + z_i^Г, \quad (3.53)$$

где z_i^A – затраты на перемещение автомобиля без груза по всему маршруту, руб.;

$z_i^Г$ – затраты на перемещение груза.

В формуле (3.53):

$$z_i^A = c_i^{\ell} L, \quad (3.54)$$

где c_i^{ℓ} – стоимость километра пробега рассматриваемой марки автомобиля, руб.;

L – длина ездки, км.

В формуле (3.53):

$$z_i^\Gamma = c_i^q P_i^{\text{TP}}, \quad (3.55)$$

где c_i^q – стоимость единицы работы в т·км. при перевозке груза, руб.;

P_i^{TP} – величина работы по перемещению груза, т·км.

Работу по перемещению всего груза на i -м маршруте представим в виде суммы:

$$P_i^{\text{TP}} = \sum_{j=1}^{\Pi} q_{ij} \ell_{ij}^\Gamma, \quad (3.56)$$

где q_{ij} – масса разгружаемого груза в пункте, т.;

ℓ_{ij}^Γ – грузовое плечо от пункта погрузки груза до пункта его разгрузки;

Π – количество пунктов разгрузки, шт.

Стоимость затрат времени на погрузочно-разгрузочные работы рассчитываем используя почасовую оплату:

$$z_i^{\text{TP}} = t_i^{\text{TP}} c_i^{\text{ч}}, \quad (3.57)$$

где $c_i^{\text{ч}}$ – стоимость одного часа использования автомобиля, руб.;

t_i^{TP} – время на погрузочно-разгрузочные операции, ч.

Стоимость затрат времени на организационный заезд в пункт разгрузки:

$$z_i^{\text{O}} = t_i^{\text{O}} c_i^{\text{ч}}, \quad (3.58)$$

где t_i^{O} – время заезда в пункт разгрузки, ч.

Подставляя (3.53)-(3.58) в (3.52), получим общую стоимость затрат на i -м маршруте:

$$z_i = c_i^\ell \ell_i + c_i^q \sum_{j=1}^{\Pi} q_{ij} \ell_{ij}^\Gamma + c_i^{\text{ч}} t_i^{\text{TP}} + t_i^{\text{O}} c_i^{\text{ч}}. \quad (3.59)$$

К этим затратам необходимо добавить расходы связанные с зарплатой водителя:

$$z_i^{\text{B}} = t c_i^{\text{B}}, \quad (3.60)$$

где t – время ездки, ч;

c_i^{B} – стоимость одного часа работы водителя, руб.

При необходимости к выше перечисленным затратам можно добавить накладные расходы.

Стоимостные показатели c_i^{ℓ} и c_i^g находим в зависимости от марки автомобиля по формулам [77]:

$$c_i^{\ell} = 0.01N_s \text{Ц}; \quad (3.61)$$

$$c_i^g = 0.01N_w \text{Ц}, \quad (3.62)$$

где N_s – норма расхода топлива на пробег автомобиля, л/100 км;

N_w – норма расхода топлива на перевозку полезного груза, л/100 т км;

Ц – стоимость одного литра топлива, руб.

Стоимость одного часа эксплуатации автомобиля и зарплата водителя может отличаться в различных регионах и фирмах.

В качестве примера для рассмотренного выше маршрута принимаем: $c_i^{\text{н}} = 100$ руб./час и $c_i^{\text{в}} = 100$ руб./час. Стоимость одного километра пробега автомобиля «Газель» при цене бензина 20 руб./литр составит $c_i^{\ell} = 3,2$ руб./км, а величина $c_i^g = 0,4$ руб./т·км. В результате расчета по формулам (3.52) – (3.62) для основного маршрута получим: $P^{\text{тп}} = 29,7$ т·км, $P^{\text{авт}} = 70,95$ т·км; $t_1^{\text{тп}} = 1,8$ ч; $t_1^{\ell} = 1,43$ ч; $t_1^3 = 0,45$ ч; $t_1 = 3,72$ ч; $Z_1 = 746,48$ руб. При развозке дополнительного груза найдем аналогично: $P^{\text{тп}} = 13$ т·км; $P^{\text{авт}} = 66$ т·км; $t_2^{\text{тп}} = 0,9$ ч; $t_2^{\ell} = 1,33$ ч; $t_2^3 = 0,3$ ч; $t_2 = 2,53$ ч; $Z_2 = 506,2$ руб. Маршрут совместной развозки будет иметь следующие характеристики: $P^{\text{тп}} = 42,2$ т·км; $P^{\text{авт}} = 72,6$ т·км; $t_c^{\text{тп}} = 2,7$ ч; $t_c^{\ell} = 1,47$ ч; $t_c^3 = 0,5$ ч; $t_c = 4,67$ ч; $Z_c = 944,68$ руб.

Уменьшение времени при совмещении маршрутов будет:

$$\Delta t = t_1 + t_2 - t_c = 3,72 + 2,53 - 4,67 = 1,58 \text{ ч},$$

что составит 25,3 %.

Сокращение расходов на перевозку будет:

$$\Delta Z = Z_1 + Z_2 - Z_c = 746,48 + 506,2 - 944,68 = 308 \text{ руб.},$$

что составит 24,6 %.

Полная работа на совмещенном маршруте уменьшилась по сравнению с раздельной развозкой в:

$$\frac{P_1 + P_2}{P_c} = \frac{179.65}{114.8} = 1.56 \text{ раза}.$$

Отметим, что при организации совместной развозки двух маршрутов, не имеющих общее начало в АТП задача решается аналогично рассмотренной.

Таким образом, совместная развозка грузов позволяет повысить эффективность мелкопартионных перевозок на кольцевом маршруте.

3.3. Оценка эффективности организации перевозочного процесса

Одними из основных показателей работы автотранспортных средств (АТС) является выработка в тоннах и тонно-километрах. Однако следует заметить, что с увеличением расстояния перевозок первый показатель уменьшается, а второй возрастает. Это является основным противоречием классической методики и, следовательно, эти показатели не могут объективно отражать эффективность организации перевозок на кольцевом маршруте. Поэтому необходимо разработать новый критерий, основанный на оценке выполненной работы при перевозке грузов.

Полная работа автомобиля складывается:

$$P = P^{\text{авт}} + P^{\text{тр}}, \quad (3.63)$$

где $P^{\text{авт}}$ – работа, затрачиваемая на передвижение автомобиля, т·км;

$P^{\text{тр}}$ – транспортная работа, т.е. работа, затрачиваемая на перемещение груза, т·км.

Приведём выражение (1) к виду:

$$\frac{P}{P^{\text{авт}}} = 1 + \frac{P^{\text{тр}}}{P^{\text{авт}}}. \quad (3.64)$$

Таким образом, чем больше транспортной работы приходится на единицу работы автомобиля, тем процесс перевозки будет происходить эффективнее. Для характеристики величины этого отношения в формуле (3.64) предлагается использовать коэффициент относительной транспортной работы:

$$K^{\text{тр}} = \frac{P^{\text{тр}}}{P^{\text{авт}}} = \frac{\sum_{j=1}^n q_{ij} \ell_{ij}}{G_{\text{сн}} L}, \quad (3.65)$$

где q_{ij} – масса груза в j -м пункте разгрузки, .;

L – длина маршрута, км;

$G_{\text{сн}}$ – снаряженная масса автомобиля, т;

ℓ_{ij} – расстояние от i -го пункта погрузки до j -го пункта разгрузки, км.

Формулу (3.65) преобразуем к виду:

$$K^{\text{тр}} = K^Q \gamma_d \beta, \quad (3.66)$$

где β – коэффициент использования пробега;

γ_d – динамический коэффициент использования грузоподъемности;

K^Q – коэффициент удельной грузоподъемности, равный:

$$K^Q = Q_a / G_{сн}, \quad (3.67)$$

где Q_a – грузоподъемность автомобиля, т.

Предлагаемый коэффициент $K^{тр}$ является удобной безразмерной величиной, так как не зависит от скорости, времени и длины маршрута, а также учитывает снаряженную массу автомобиля. Отметим, что наибольшее его значения достигается на маятниковом маршруте с обратным груженым пробегом при $\gamma_d \beta = 1$:

$$K^{тр} = K^Q \quad (3.68)$$

Это выражение подтверждает правильность выбранного подхода. Более того, с увеличением коэффициента удельной грузоподъемности эффективность перевозки возрастает. Это обстоятельство подтверждается практикой и логикой рынка – что всегда надо стремиться перевозить наибольшее количество груза на автомобиле меньшей массы. Таким образом, при оценке организации перевозочного процесса необходимо учитывать снаряженную массу АТС.

Для подтверждения этого факта рассмотрим расход топлива на единицу массы груза для некоторых марок автомобилей, приведенные в табл. 3.1 и на рис. 3.17 и 3.18 [77]. В табл. 3.1: H_S – базовая норма расхода топлива на пробег автомобиля в снаряженном состоянии в зависимости от его марки, л/100 км; H_W – норма расхода топлива на транспортную работу по перемещению груза, л/100 т·км.

Т а б л и ц а 3 . 1

№ п/п	Марка автомобиля	Q_a , т	$G_{сн}$, т	K^Q	H_S , л	H_S/Q_a , л/т
1.	Иж – 27156	0,37	1,095	0,34	11	29,7
2.	Иж – 2715-01	0,45	1,015	0,44	11	24,4
3.	Иж – 27151-01	0,5	0,965	0,52	11	22
4.	АЗЛК – 2335	0,5	0,99	0,51	10	20
5.	УАЗ – 3741	0,8	1,7	0,47	16	20
6.	ЕрАЗ – 762В	1,15	1,475	0,78	14	12,2
7.	ГАЗ – 3302	1,5	1,85	0,81	11	7,3
8.	ГЗСА – 3726	3,4	4,35	0,78	27	8
9.	ГЗСА – 37121	4,0	3,75	1,07	27	6,7
10.	ГЗСА – 3768-20	4,36	3,335	1,31	27	6,2
11.	ГАЗ – 5312	4,5	3,2	1,41	25	5,5
12.	ЗИЛ – 4314	6,0	4,175	1,44	31	5,2
13.	ЗИЛ – 4314 с приц. ГКБ – 8328	11,5	6,875	1,67	36,4	3,2

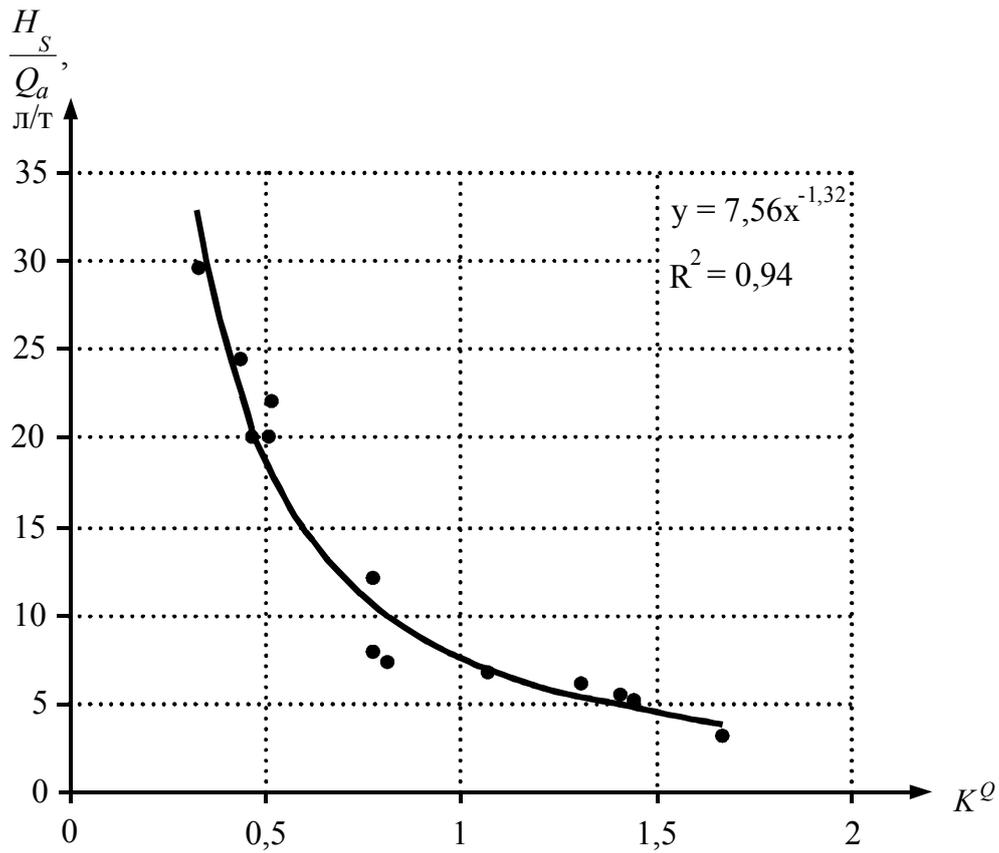


Рис. 3.17. Зависимость расхода топлива на тонну груза от коэффициента удельной грузоподъемности

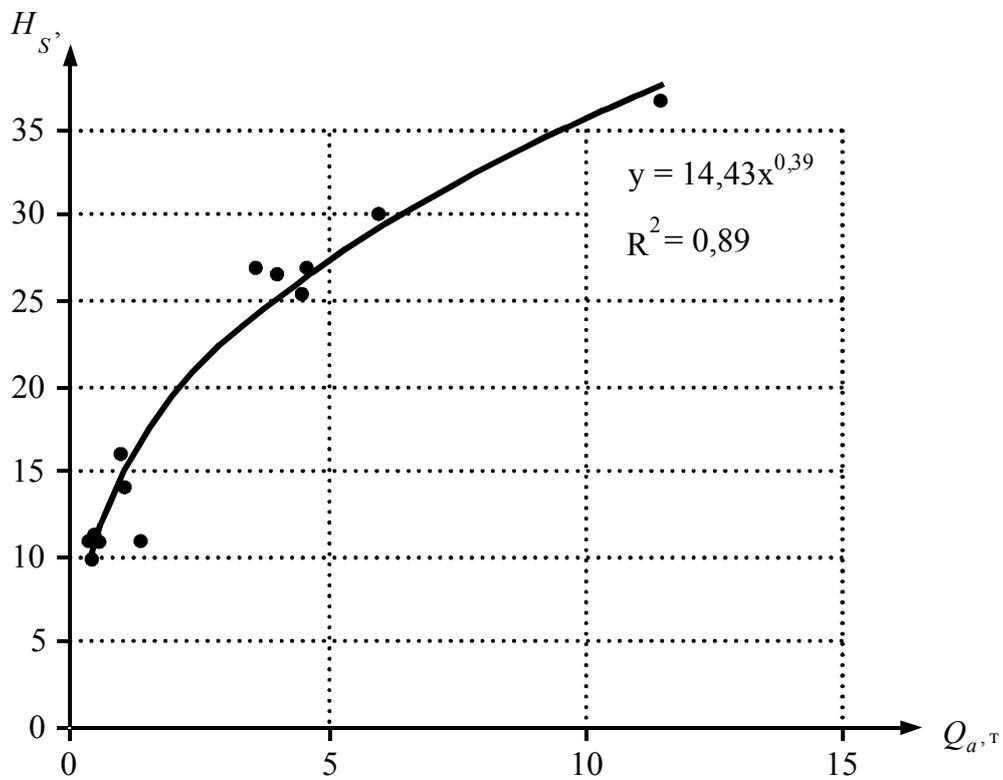


Рис. 3.18. Зависимость расхода топлива от грузоподъемности автомобиля

Из данных табл. 3.1 и рис. 3.17 и 3.18 видно, что с ростом грузоподъемности, а также при использовании прицепов расход топлива на единицу массы груза уменьшается. Следовательно, это ещё раз подтверждает ранее полученный из формулы (3.66) вывод, что для оценки эффективности организации кольцевого маршрута целесообразно использовать показатель, который учитывает снаряженную массу АТС.

Однако одинаковую по величине работу можно выполнить за разное время и автомобилями разной мощности. Очевидно, что его производительность и расход топлива будут тоже разными. Существующая методика ее расчета не учитывает этого обстоятельства. Одна из причин заключается в том, что в 30-е годы, когда разрабатывалась теория автомобильных перевозок, не было такого разнообразия марок автомобилей различной мощности, как сейчас и поэтому его массой пренебрегали. Также в эти годы отсутствовала конкуренция между перевозчиками. Кроме того, в настоящее время стоимость топлива резко возросла и продолжает расти. Следовательно, затраты на передвижение автомобиля также резко увеличились. Они становятся сопоставимы со стоимостью груза. Это обстоятельство должно менять подход к оценке организации перевозочного процесса в целом. Таким образом, методика расчета производительности нуждается в совершенствовании.

Из классического курса механики известно, что мощность (в нашем случае производительность) определяется по выражению

$$W = \frac{P^{TP}}{t}. \quad (3.69)$$

Используя (3.65), найдем производительность АТС в тонно-километрах:

$$W = \frac{K^{TP} P^{авт}}{t} = \frac{K^Q \gamma_d \beta}{\frac{L}{V_T} + t^{пp}} P^{авт}, \quad (3.70)$$

где V_T – техническая скорость, км/ч;

$t^{пp}$ – время погрузочно-разгрузочных работ, ч;

t – время ездки, ч.

Разделив производительность АТС в тонно-километрах на работу, затрачиваемую на передвижение АТС, получим коэффициент относительной производительности (час^{-1}):

$$K^W = \frac{K^{TP}}{t} = \frac{K^Q \gamma_d \beta}{\frac{L}{V_T} + t^{пp}} = \frac{P^{TP}}{P^{авт} t}. \quad (3.71)$$

Из анализа выражения (3.71) следует, что чем больше K^W , тем выше относительная производительность автомобиля. Таким образом, он учитывает снаряженную массу АТС и его работу.

Известно, что самая высокая производительность наблюдается на маятниковом маршруте с грузеным обратным пробегом при $\gamma_d\beta = \gamma_c\beta = 1$. Тогда из выражения (3.71) найдем:

$$K^W = \frac{K^Q}{\frac{L}{V_T} + t_{\text{пр}}}. \quad (3.72)$$

Как следует из выражения (3.72), чем больше удельная грузоподъемность, тем выше относительная производительность. Это подтверждается опытом эксплуатации большегрузных автомобилей и прицепов с более высоким параметром K^Q (см. табл.3.1). Кроме того, повысить производительность можно при уменьшении длины маршрута и времени на погрузочно-разгрузочные работы, а также при увеличении скорости движения. Следует отметить, что формула (3.71) является универсальной и может быть использована для любого типа маршрута. Как следует из нее, для увеличения производительности необходимо стремиться к увеличению коэффициента $K^{\text{ТР}}$ и уменьшению времени ездки.

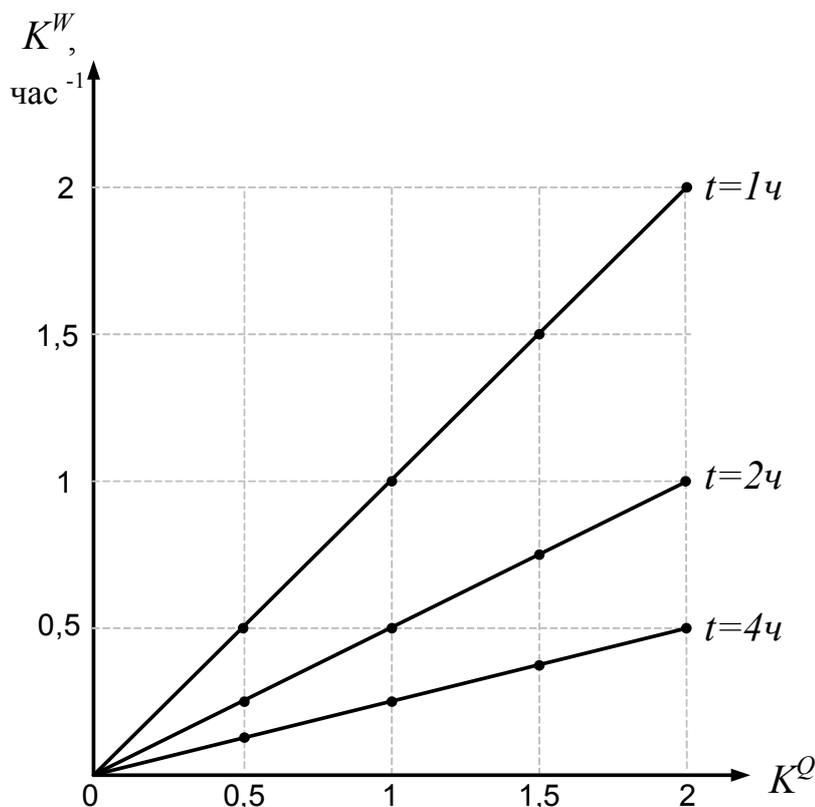


Рис. 3.19. Зависимость коэффициента относительной производительности от коэффициента удельной грузоподъемности

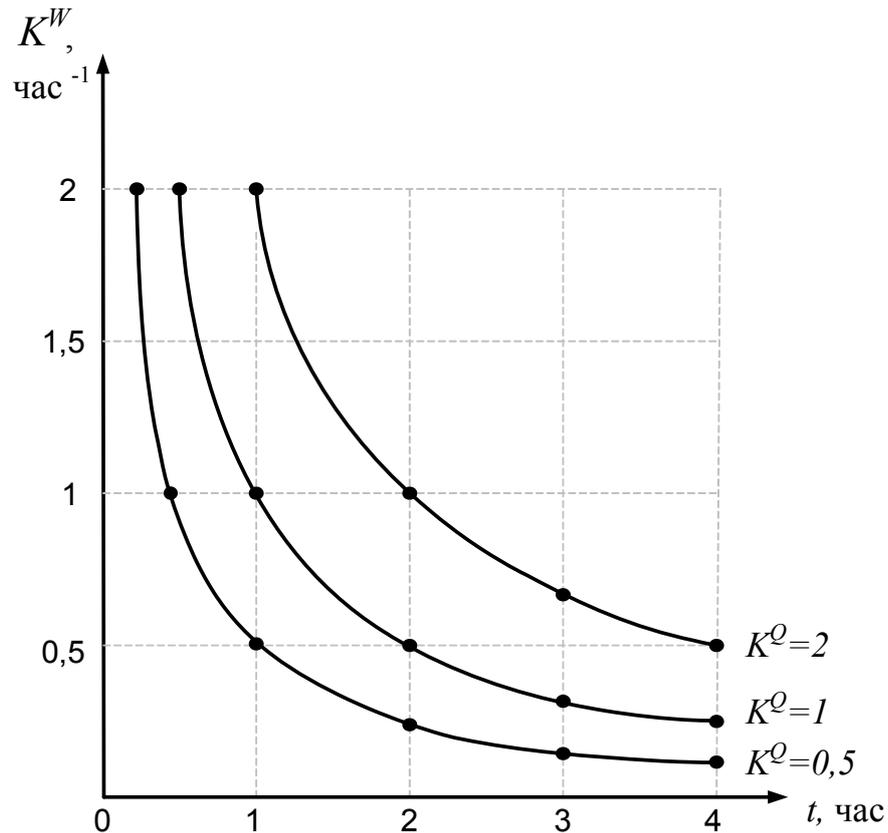


Рис. 3.20. Зависимость коэффициента относительной производительности от времени

Отметим, что критерий K^W исключает ранее отмеченную двойственную противоречивую оценку производительности в т/час и ткм/час.

Графическая интерпретация предложенного критерия относительной производительности в зависимости от коэффициента K^Q и времени показана на рис. 3.19 и 3.20, соответственно.

Как видно из рис. 3.19 и формулы (3.72), зависимость относительной производительности от коэффициента K^Q линейная. С уменьшением времени ездки параметр K^W возрастает сильнее. Например, увеличение K^Q в два раза повышает K^W с 0,25 до 2 час⁻¹, т.е. в 8 раз, при уменьшении времени маршрута в 4 раза.

Таким образом, чтобы повысить K^W необходимо уменьшать расходы на перемещение автомобиля, то есть повышать K^Q . Другим резервом повышения K^W является уменьшение затрат времени на маршруте. Зависимость критерия K^W от времени имеет гиперболический характер. Время движения может изменяться в зависимости от трех переменных: длины, скорости и времени погрузочно-разгрузочных работ, которые в свою очередь могут быть связаны с величиной грузопотока. Чтобы проанализировать влияние каждого фактора на относительную грузоподъемность по-

строим характеристический график (рис. 3.21). В качестве примера прием следующие исходные данные:

$$L = 5 \text{ км}, V_T = 25 \text{ км/ч}, K^Q = \gamma_c = \beta = 0,5 \text{ и } t^{np} = 0,3 \text{ час.}$$

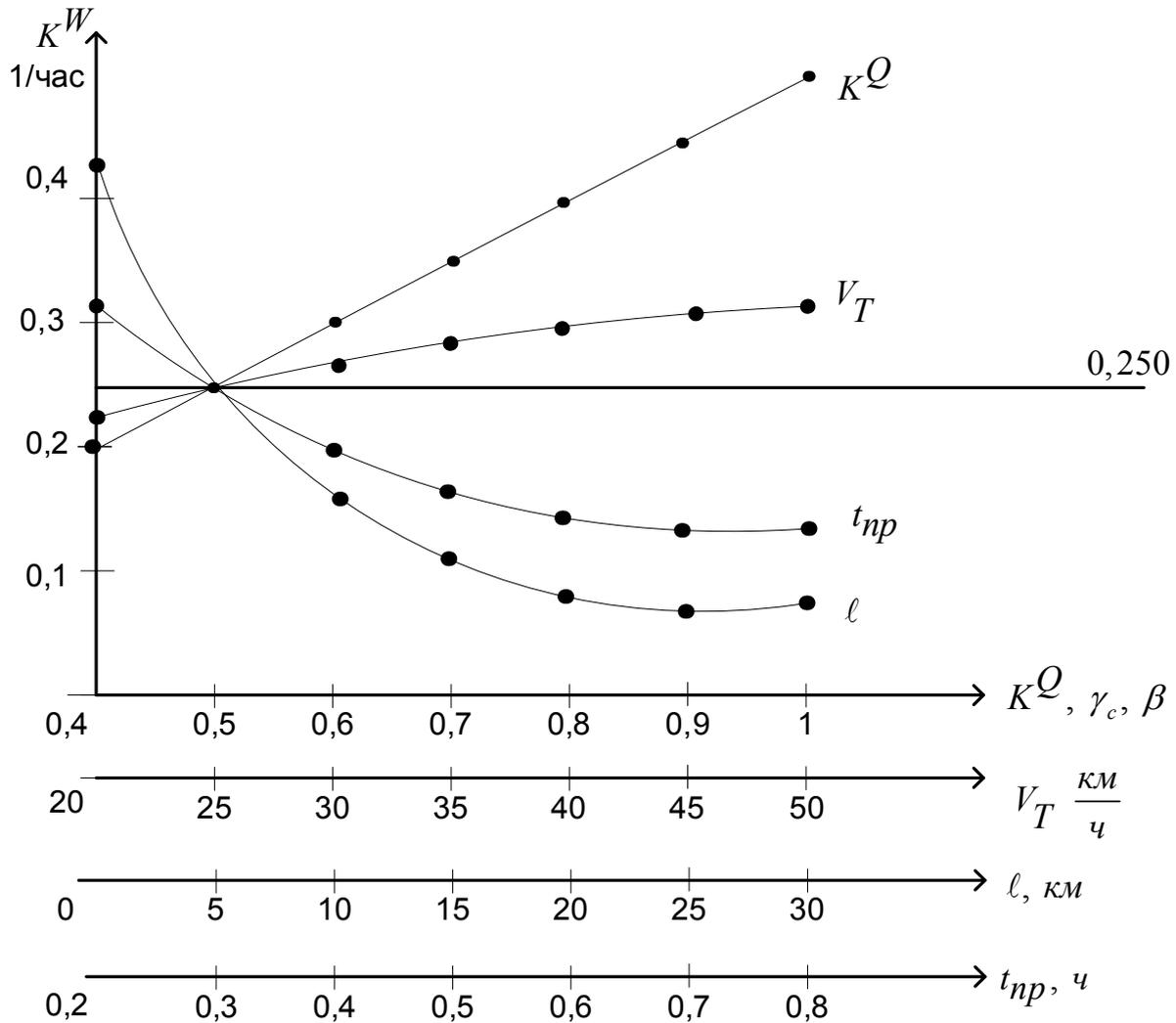


Рис. 3.21. Характеристический график

Горизонтальная линия на рисунке 3.21 соответствует относительной производительности $K^W = 0,250 \text{ час}^{-1}$. Увеличим коэффициент K^W в 1,2 раза и получим его значение $K^W = 0,3 \text{ час}^{-1}$. Такой рост относительной производительности может быть достигнут при увеличении: скорости до 45 км/ч (в 1,8 раза), коэффициентов удельной грузоподъемности, использования пробега и динамического коэффициента использования грузоподъемности до 0,6 (в 1,2 раза); а также уменьшения: длины ездки до 2,9 км (в 1,72 раза) и времени погрузочно-разгрузочных работ до 0,22 часа (в 1,36 раза). Наибольшее влияние на параметр K^W оказывают коэффициенты: K^Q , γ_d и β , входящие в величину коэффициента относительной транспортной работы $K^{тр}$.

4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ РАЗРАБОТАННОЙ МОДЕЛИ

4.1. Цель и задачи эксперимента

Целью экспериментальных исследований является: выполнить проверку на адекватность разработанной методики оптимизации перевозок на развозочных маршрутах при доставке мелкопартионных грузов в торговую сеть с использованием усовершенствованного математического метода «ветвей и границ».

Для достижения указанной цели были поставлены следующие задачи, которые решались в процессе проведения эксперимента:

➤ собрать исходную информацию на предприятии города Пензы, содержащую данные о клиентурной базе и дислокации торговых точек. На основании этой информации составить транспортную схему и существующие маршруты движения.

➤ экспериментальным путём получить данные, которые характеризуют транспортный процесс доставки готовой продукции от производителя до потребителя.

➤ провести статистическую обработку данных, собранных в ходе натуральных наблюдений.

➤ разработать модель совмещенного маршрута при доставке грузов от двух производителей.

➤ провести сопоставление технико-экономических показателей предложенного объединённого и существующих маршрутов.

➤ произвести сравнение расчётов по классическому методу «ветвей и границ» с предложенной методикой.

4.2. Выбор и описание объекта исследования

Использование логистического подхода в хозяйственной практике предприятий подразумевает системное решение проблемы минимизации общих расходов в цепи поставщик – производитель – потребитель [32, 37, 63, 64, 70, 71]. Основная задача любого предприятия – это получение прибыли. Однако необходимо так организовать устойчивый производственный процесс, чтобы конечной целью являлось не только достижение экономического эффекта, но и полное удовлетворение потребительского спроса, как в количественном, так и в качественном отношении. В наибольшей степени это относится к перевозкам социально-значимых грузов, таких, например, как хлеб и хлебобулочные изделия. Поэтому для реализации разработанной модели экспериментальные исследования проводились на крупном предприятии хлебопекарной промышленности города Пензы.

В качестве объекта исследований было выбрано объединение «Пензахлебопродукт», в состав которого входят: агрофирмы, элеваторы, мукомольная фабрика, хлебозаводы и фирменные магазины, то есть целый комплекс предприятий, выполняющих весь процесс, от выращивания хлебных культур до сбыта готовой продукции.

Ранее в состав данного объединения входило четыре хлебозавода. Однако в условиях современного рынка все чаще наблюдается процесс укрупнения предприятий. Поэтому в настоящее время в структуру объединения «Пензахлебопродукт» входят только два хлебопекарных предприятия: ОАО «Хлебозавод № 2» и ОАО «Хлебозавод № 4». На этих заводах и были собраны исходные данные для выявления реальной ситуации, складывающейся при доставке хлеба и хлебобулочных изделий в торговую сеть.

ОАО «Хлебозавод № 4» находится в Первомайском районе г. Пензы, а ОАО «Хлебозавод № 2» – в Октябрьском районе и является более крупным предприятием. Объем выпуска хлебобулочной продукции на данных предприятиях составляет около 40 т/сут и 65 т/сут, соответственно. Производство на рассматриваемых хлебозаводах направленно на массовое обслуживание населения и имеет широкую номенклатуру продукции, которая включает в себя не только хлебобулочные, но и кондитерские, пряничные, бараночные и макаронные изделия.

Схема движения материальных потоков представлена на рис. 4.1. Для производства хлебобулочной продукции в качестве сырья требуется не только мука, но и ряд других исходных продуктов, таких как: соль, сахар, масло, дрожжи, молоко и т.д. Поэтому хлебозаводы имеют сеть поставщиков, которые, благодаря информационным потокам, обеспечивают стабильное снабжение необходимым сырьем для производства.

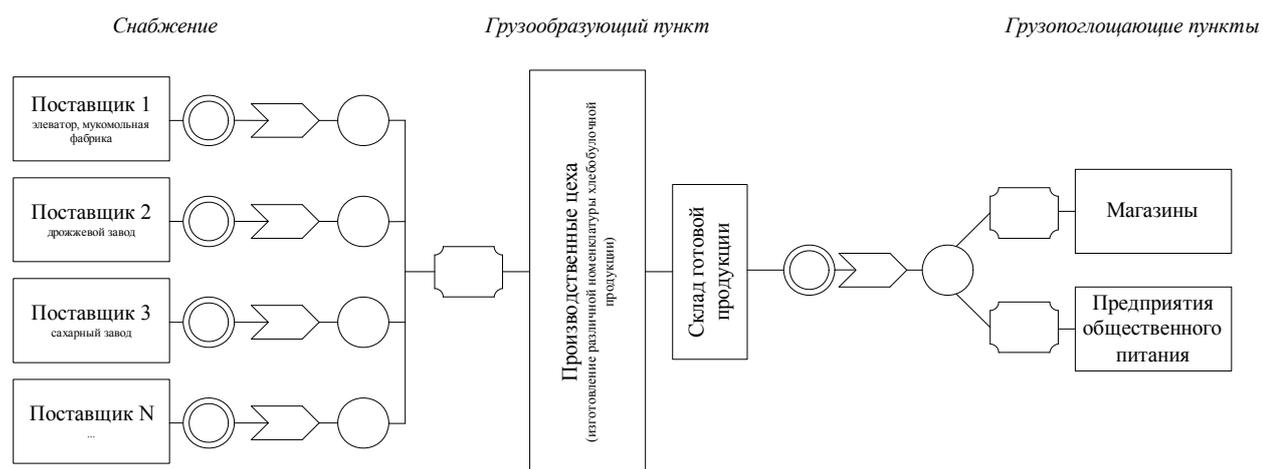


Рис. 4.1. Схема движения материальных потоков:

⊙ – погрузка; ➤ – транспортирование; ○ – разгрузка; ▭ – складирование

Из производственных цехов хлебобулочные изделия попадают на склад готовой продукции, где они должны отлежаться не менее 6 часов после выхода из печи (требования СанПиН). Затем готовая продукция поступает в

систему сбыта (не позднее 16 часов после выхода из печи) для реализации, как в различные торговые точки, так и в предприятия общественного питания.

Для перевозки хлебобулочной продукции заводы используют собственный подвижной состав. Применяются автомобили–фургоны следующих марок: ГАЗ-27073 (96 лотков), ГАЗ-33021 (96,108,120лотков), ГАЗ-3307 (128,160,200,250 лотков), ЗИЛ-130 (325 лотков), ЗИЛ-5301 (360 лотков). Возраст подвижного состава находится в пределах от 1 года до 10 лет. Для обслуживания районов области применяют автомобили большей грузоподъемности ЗИЛ-130 и ЗИЛ-5301. Около 25 % груза перевозится частным транспортом.

Технологическая схема доставки продукции представлена на рис. 4.2.



Рис. 4.2. Технологическая схема доставки хлебобулочной продукции

В г. Пензе существуют и другие хлебопекарные предприятия меньшей производственной мощности, а также частные пекарни. Для устойчивой конкурентоспособности рассматриваемых хлебозаводов сотрудниками отделов сбыта и маркетинга регулярно проводится изучение ассортимента и выявление новой продукции, пользующейся спросом у потребителя [65].

4.3. Методика проведения экспериментальных исследований

Экспериментальные исследования на указанных ранее предприятиях хлебопекарной промышленности проводились в течение одного календарного года с целью определения исходных данных для разработанной модели. Натурные исследования осуществлялись для получения предварительной информации, характеризующей: работу отделов сбыта и диспетчерских служб,

исходную транспортную сеть, а также продолжительности различных этапов процесса доставки. Следующим этапом была статистическая обработка результатов эксперимента с целью определения статистических характеристик [24] и проверки принадлежности генеральной совокупности данных определённому закону распределения. Для проверки правдоподобия гипотезы, в качестве оценочного критерия был применён критерий χ^2 Пирсона.

В результате анализа работы диспетчерской службы и потока заявок от заказчиков было выявлено, что клиентурная база рассматриваемых производителей включает около 1500 пунктов доставки (не считая обслуживания районов области). Согласно этой информации и дислокации торговых точек, на карте г. Пензы был составлен граф исходной транспортной сети, который охватывал 4 района города. В результате, для проведения замеров моделируемых характеристик, было отобрано 69 реальных маршрутов движения, в пунктах доставки которых реализовывалась продукция обоих производителей. Их выбор производился в соответствии с поставленной задачей совмещения. Поэтому рассматривались маршруты движения, проходящие мимо хлебозаводов, а также между ними.

Проведение натурных исследований этапа погрузки хлебобулочной продукции осуществлялось для автомобилей ГАЗ – 33021, вместимость которого 108 лотков (0,5т) и ГАЗ – 3307, вместимость которого 200 лотков (1т). На исследуемых предприятиях преобладает немеханизированный способ погрузки, когда грузчики (а порой и сами водители) выкатывают со склада готовой продукции контейнеры на погрузочную платформу и вручную производят погрузку в транспортное средство. В результате исследований было установлено, что время погрузки $t^п$ практически не зависит от типа и грузоподъёмности автомобиля и поэтому все замеры производились на одну тонну груза. Полученный график распределения представлен на рис. 4.3.

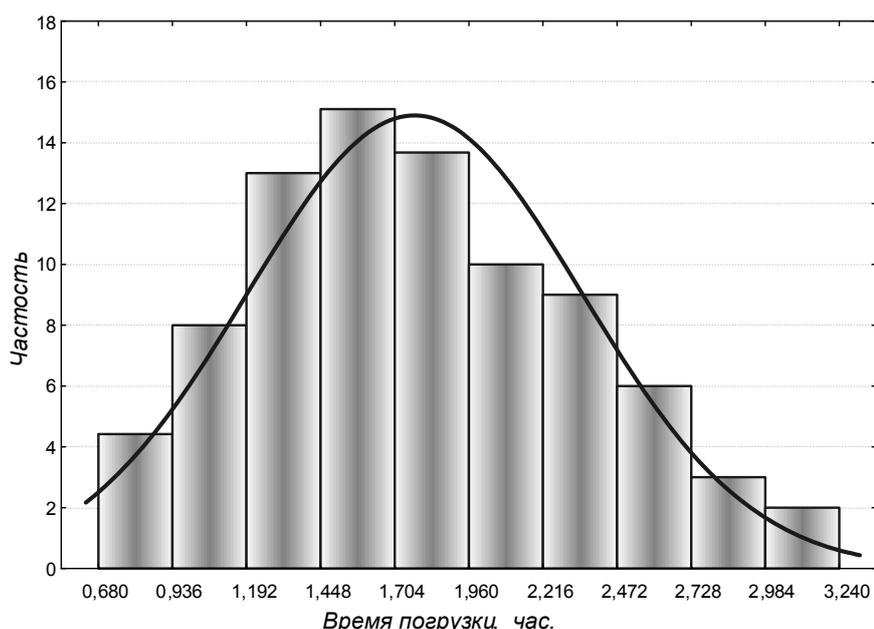


Рис. 4.3. График распределения величины $t^п$

Для каждого из выбранных 69 маршрутов проводились экспериментальные наблюдения. Были выполнены замеры времени работы автомобиля на маршруте, а также его длины при помощи спидометра. Как известно, степень полезного использования общего пробега характеризует коэффициент использования пробега β . Полученное его распределение приведено на рис. 4.4.

На основе продолжительности этапа транспортирования были рассчитаны значения технической скорости V_T для различных марок автомобилей. На рис. 4.5 представлен график распределения технической скорости для автомобиля ГАЗ-3307. Аналогичными будут графики распределения данной величины и для других марок автомобилей.

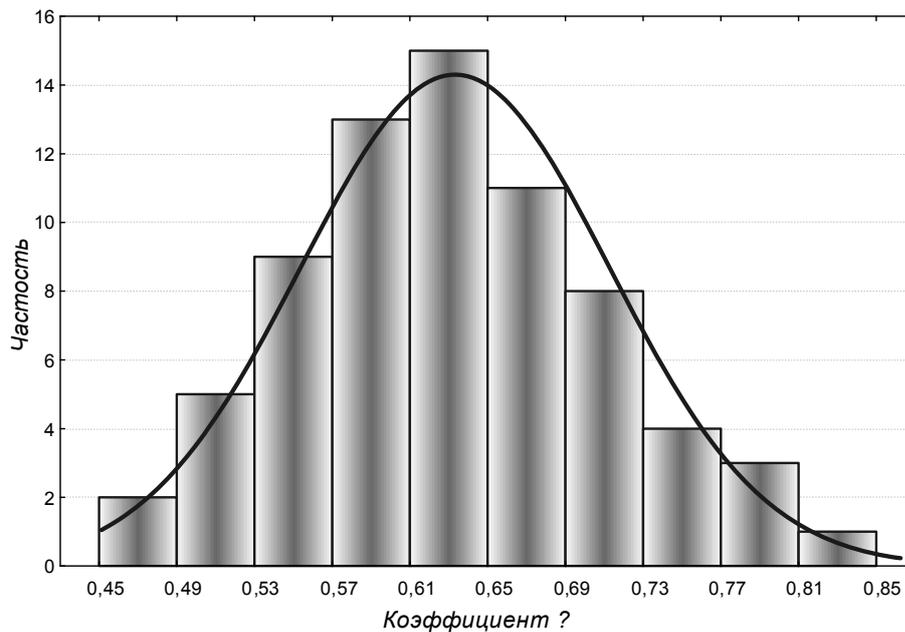


Рис. 4.4. График распределения величины коэффициента β

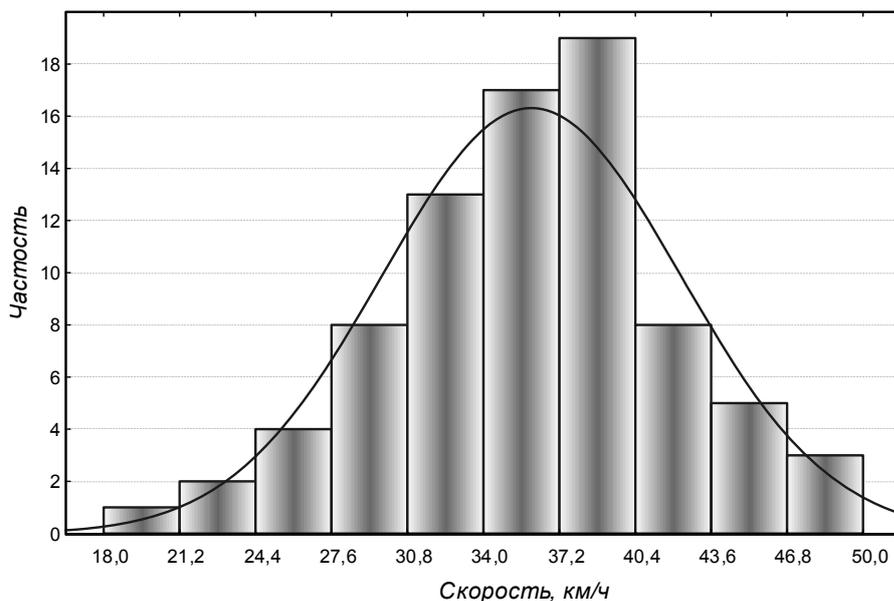


Рис. 4.5. График распределения величины V_T для автомобиля ГАЗ-3307

Под продолжительностью заезда в пункт разгрузки t^{3P} понимается время, затрачиваемое на этапы маневрирования, ожидание разгрузки и оформление документов. Следовательно, разброс величины t^{3P} наблюдается вследствие неравномерности показателей, характеризующих её. В ходе наблюдений было установлено, что, как правило, этап оформления документов осуществляется параллельно с этапом разгрузки. Распределение времени заезда в пункт разгрузки представлено на рис. 4.6.

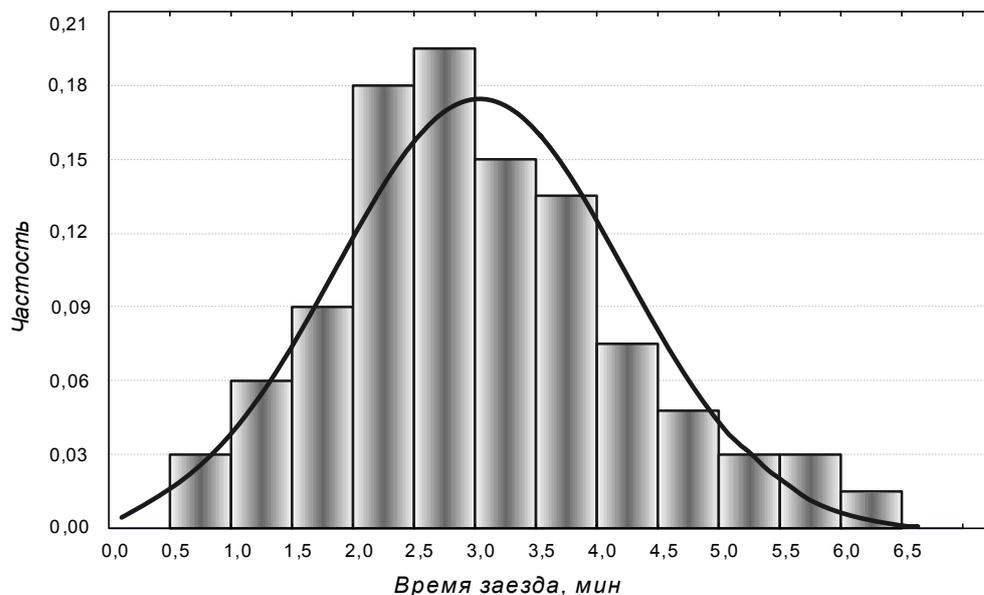


Рис. 4.6. График распределения величины t^{3P}

Продолжительность этапа разгрузки t^P , как и погрузки, не зависит от типа подвижного состава, поэтому замеры производились из расчета на одну тонну груза. Варьирование величины t^P обусловлено степенью механизации процесса разгрузки, наличием или отсутствием грузчиков в пункте доставки, так как зачастую их функцию выполняет сам водитель, а также расстоянием от кузова автомобиля до места хранения груза, либо непосредственно до торгового зала. На рис. 4.7 представлено распределение времени разгрузки хлебобулочной продукции.

Проведенные исследования показали, что на рассматриваемых предприятиях наблюдается нерациональность выбора кольцевых маршрутов движения и набора пунктов доставки, включаемых в них. Как показал анализ работы диспетчерских служб, вопросам организации и планирования перевозок не уделяется должного внимания. Практически нет постоянных маршрутов движения по времени доставки, набору пунктов и количеству перевозимого груза, за исключением некоторых маятниковых маршрутов. Как правило, диспетчер, располагая информационным потоком заявок, выдаёт водителю лишь товарно-транспортную накладную с указанием адресов клиентов и объёма партий, а порядок объезда пунктов доставки осуществляется по усмотрению водителя.

нию водителя. При анализе накладных документов зачастую наблюдались неоправданные перепробеги автомобилей. Водитель, несомненно, несёт ответственность за сохранность груза, но отчитывается он лишь за соответствие расхода топлива величине пробега. Возникающие при этом увеличение пробегов вызваны нежеланием работников диспетчерской службы маршрутизировать работу автомобилей.

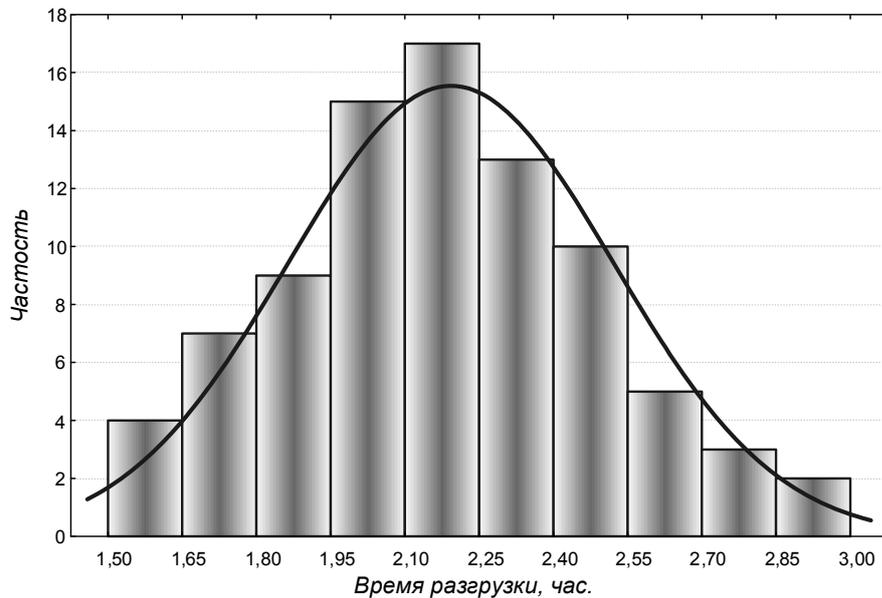


Рис. 4.7. График распределения величины t^p

Проведённые экспериментальные исследования показали, что все величины, полученные в ходе натурных наблюдений, носят случайный характер [82]. Статистические характеристики приведены в табл. 4.1. Приведённые гистограммы распределения рассматриваемых характеристик описываются нормальным законом, что также подтверждает случайность исследуемых величин.

Таблица 4.1

Статистические характеристики распределения величин.

Исследуемая величина	Математическое ожидание	Среднее квадратическое отклонение	Критерий Пирсона
Время погрузки $t^п$, час	$M(t^п)=1,765$	$\sigma(t^п)=0,576$	$\chi^2(t^п)=0,986$
Время разгрузки $t^р$, час	$M(t^р)=2,189$	$\sigma(t^р)=0,327$	$\chi^2(t^р)=0,985$
Время заезда пункт разгрузки $t^{3р}$, мин	$M(t^{3р})=3,079$	$\sigma(t^{3р})=1,174$	$\chi^2(t^{3р})=0,972$
Техническая скорость V_T , км/час	$M(V_T)=35,925$	$\sigma(V_T)=6,259$	$\chi^2(V_T)=0,984$
Коэффициент использования пробега β	$M(\beta)=0,623$	$\sigma(\beta)=0,792$	$\chi^2(\beta)=0,992$

Проблемы, возникающие при организации перевозочного процесса, как на рассматриваемых хлебозаводах, так и на большинстве других предприяти-

ях, связаны с отсутствием мероприятий по использованию экономико-математических методов планирования.

4.4. Моделирование совмещённого маршрута и оценка его эффективности

Рассмотрим процесс моделирования совмещённого маршрута доставки продукции от двух производителей. Проанализированные ранее 69 маршрутов, характеристики которых приведены в табл. 4.2–4.7, разбиты на шесть блоков, в зависимости от направления движения:

Блок 1 – маршруты движения в направлении от хлебозавода №2, в которых конечные пункты доставки находятся в районе хлебозавода №4.

Блок 2 – маршруты движения, в которых пункты доставки находятся между хлебозаводами в направлении от хлебозавода №2.

Блок 3 – маршруты движения в направлении от хлебозавода №4, в которых конечные пункты доставки находятся в районе хлебозавода №2.

Блок 4 – маршруты движения, в которых пункты доставки находятся между хлебозаводами в направлении от хлебозавода №4.

Блок 5 – маршруты движения в направлении от хлебозавода №4 в пункты доставки, находящиеся в этом же районе.

Блок 6 – маршруты движения в направлении от хлебозавода №2 в пункты доставки, находящиеся в этом же районе.

Рассмотрено два типа совмещения: первый, когда заезд во второй пункт погрузки осуществляется с промежуточного пункта основного маршрута (рис. 3.10) и второй, когда заезд производится с его конечного пункта (рис. 3.8).

Следует отметить, что в отличие от предложенной модели, представленной на рисунке 3.10, рассматриваемые хлебозаводы имеют собственный парк подвижного состава. При совмещении маршрутов предлагается обслуживание двух хлебозаводов транспортными средствами, входящими в состав АТП, принадлежащего хлебозаводу №2. Это обусловлено тем, что данное АТП более крупное, а также тем, что оба хлебозавода входят в состав одного объединения. Исходя из этого, формула (3.42) преобразуется и выражение примет следующий вид:

$$\left[t_{mD}^{3c} \right] = \Sigma t^{3p} - \Sigma t^{\Gamma o} + \Sigma t^{\Gamma c} + \Sigma t^o, \quad (4.1)$$

где $\Sigma t^o = t_{nB}^{oo} + t_{дD}^{oo} + t_{вD}^{од}$.

Для анализа формулы (4.1) проводилось объединение маршрутов доставки основного груза, входящих в блоки 1 и 3 с маршрутами доставки дополнительного груза, входящих в блоки 5 и 6. Совмещение осуществлялось в три этапа. На первом этапе объединялись маршруты из 1 и 5 блоков, и получался первый промежуточный совмещенный маршрут.

Таблица 4.2

Технико-экономические показатели 1 блока маршрутов

№ маршр.	Р _{гр} , ткм	Р _а , ткм	Р _{общ} , ткм	L ^г , км	L ^х , км	L ^{общ} , км	П, шт.	t ^{общ} , ч	Q, т	W, тч	W _p , ткмч	β	З, руб.	K ^ш
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	9,935	112,688	122,623	17,556	14,187	31,743	9	4,805	0,878	0,183	2,068	0,55	1000,9	0,088
2	13,778	126,227	140,005	20,473	15,084	35,557	10	5,232	0,948	0,181	2,633	0,58	1092,5	0,109
3	10,850	107,448	118,298	19,569	10,698	30,267	10	5,058	0,941	0,186	2,092	0,65	1050,45	0,101
4	12,860	173,698	186,558	34,257	14,672	48,929	12	5,633	0,930	0,165	2,283	0,7	1187,6	0,074
5	9,712	104,512	114,224	16,081	13,359	29,44	12	4,973	0,900	0,181	1,953	0,55	1032,09	0,093
6	14,169	139,21	153,38	23,551	15,664	39,125	14	5,479	0,934	0,170	2,586	0,6	1146,18	0,102
7	11,031	109,457	120,488	18,398	12,435	30,833	14	5,246	0,934	0,178	2,103	0,6	108879	0,101
8	11,059	121,769	132,828	19,316	14,985	34,301	16	5,209	0,875	0,168	2,123	0,56	1085,41	0,091

Таблица 4.3

Технико-экономические показатели 2 блока маршрутов

№ маршру.	P_{TP} , ткм	P_A , ткм	$P_{OБЦ}$, ткм	L^I , км	L^X , км	$L^{OБЦ}$, км	П, шт.	$t^{OБЦ}$, ч	Q , т	W , тч	W_P , ткмч	β	З, руб.	K^W
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2,687	32,688	35,375	9,556	10,255	19,811	4	2,587	0,465	0,18	1,039	0,48	526,851	0,082
2	4,735	46,652	51,387	18,962	9,312	28,274	5	2,931	0,480	0,164	1,615	0,67	600,11	0,101
3	2,547	21,757	24,304	8,556	4,627	13,186	6	2,527	0,471	0,186	1,01	0,65	511,932	0,117
4	3,069	33,743	36,812	11,719	8,731	20,45	4	2,514	0,442	0,176	1,221	0,57	512,653	0,091
5	2,535	36,968	39,503	13,446	8,959	22,405	5	2,571	0,430	0,167	0,986	0,6	524,646	0,069
6	4,594	35,325	39,919	14,329	7,080	21,409	6	3,071	0,551	0,3179	1,496	0,67	625,106	0,13
7	2,938	36,577	39,515	12,526	9,642	22,168	4	2,653	0,465	0,175	1,107	0,56	541,041	0,08
8	6,527	46,685	53,212	19,579	8,715	28,294	5	2,979	0,492	0,165	2,191	0,7	610,426	0,14
9	3,516	36,78	40,296	12,553	9,738	22,291	6	2,677	0,445	0,166	1,313	0,56	546,207	0,096
10	3,998	27,463	31,461	10,979	5,665	16,644	4	2,566	0,482	0,188	1,558	0,66	521,873	0,146
11	5,681	48,682	54,363	17,833	11,671	29,504	5	2,985	0,485	0,162	1,903	0,6	611,791	0,117
12	3,308	27,068	30,376	10,01	6,396	16,405	5	2,57	0,472	0,184	1,287	0,61	522,269	0,122
13	2,935	28,215	31,15	9,109	7,991	17,1	4	2,433	0,445	0,183	1,206	0,53	494,944	0,104
14	3,164	29,476	32,64	10,596	7,268	17,864	5	2,445	0,430	0,176	1,294	0,59	497,753	0,107
15	3,502	37,325	40,827	13,813	8,808	22,621	6	2,824	0,480	0,17	1,24	0,61	575,824	0,094

Таблица 4.4

Технико-экономические показатели 3 блока маршрутов

№ маршру.	P_{TP} , ТКМ	P_A , ТКМ	$P_{OБЦ}$, ТКМ	L^I , км	L^X , км	$L^{OБЦ}$, км	П, шт.	$t^{OБЦ}$, ч	Q , т	W , тч	W_P , ТКМЧ	β	З, руб.	K^W
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	9,845	84,6	94,445	11,997	11,854	23,831	6	4,715	950	0,201	2,088	0,5	974,1	0,116
2	11,668	97,771	109,439	14,424	13,117	27,541	7	5,12	1,014	0,198	2,279	0,52	1060,19	0,119
3	9,807	98,253	108,06	15,482	12,195	27,677	8	4,807	0,921	0,192	2,04	0,56	996,88	0,1
4	9,999	113,494	123,49	17,71	14,260	31,97	9	4,775	0,870	0,182	2,094	0,55	995,428	0,088
5	10,958	109,163	120,12	16,333	14,417	30,75	10	5,067	0,940	0,186	2,163	0,53	1052,94	0,1
6	11,591	114,026	125,617	20,064	12,056	32,120	11	5,076	0,920	0,181	2,283	0,62	1056,57	0,102
7	11,148	110,895	122,043	18,6	12,637	31,238	12	5,386	0,992	0,184	2,131	0,6	1117,36	0,101
8	10,563	134,073	144,635	24,489	13,278	37,767	14	5,336	0,908	0,170	1,98	0,65	1114,49	0,079
9	11,074	106,244	117,318	18,097	11,831	29,928	14	5,323	0,960	0,18	2,08	0,6	1103,28	0,104
10	15,794	107,114	122,908	18,283	11,890	30,173	9	5,337	1,025	0,192	2,96	0,61	1108,16	0,147

Таблица 4.5

Технико-экономические показатели 4 блока маршрутов

№ маршр.	Р _{тр} , ткм	Р _а , ткм	Р _{общ} , ткм	L ^г , км	L ^х , км	L ^{общ} , км	П, шт.	t ^{общ} , ч	Q, т	W, тч	W _p , ткмч	β	З, руб.	K ^ш
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4,055	30,475	34,53	10,915	7,555	18,47	5	2,659	0,480	0,181	1,525	0,59	541,232	0,133
2	3,875	35,363	39,238	12,189	9,243	21,432	6	2,594	0,430	0,166	1,494	0,57	529,366	0,11
3	2,084	28,291	30,375	8,717	8,429	17,146	4	2,45	0,449	0,183	0,851	0,51	498,04	0,074
4	3,594	29,433	33,027	9,94	7,898	17,838	6	2,644	0,468	0,177	1,36	0,56	537,789	0,122
5	2,994	26,708	29,703	8,413	7,774	16,187	5	2,615	0,485	0,185	1,145	0,52	531,11	0,112
6	3,759	31,507	35,266	10,667	8,428	19,095	6	2,754	0,487	0,177	1,365	0,56	560,379	0,119
7	2,223	21,13	23,353	6,404	6,402	12,806	4	2,294	0,440	0,192	0,969	0,5	465,041	0,105
8	3,683	27,242	30,925	9,156	7,354	16,510	5	2,652	0,492	0,186	1,389	0,55	538,846	0,135
9	4,028	33,492	37,520	14,778	5,520	20,298	6	2,701	0,465	0,172	1,491	0,73	550,298	0,12
10	5,054	34,85	39,904	16,166	4,955	21,121	6	2,818	0,489	0,174	1,793	0,76	574,588	0,145
11	2,759	28,84	31,599	8,514	8,965	17,479	6	2,492	0,432	0,173	1,107	0,49	506,869	0,096
12	3,937	34,48	38,417	14,167	6,73	20,897	5	2,482	0,418	0,168	1,586	0,68	506,712	0,114
13	3,143	32,614	35,757	10,9	8,866	19,766	4	2,602	0,469	0,18	1,208	0,55	529,924	0,096
14	2,355	26,77	29,125	8,72	7,504	16,224	5	2,605	0,482	0,185	0,904	0,54	528,706	0,088
15	4,043	31,639	35,673	10,165	9,010	19,175	6	2,669	0,465	0,174	1,511	0,53	543,587	0,128
16	3,163	34,33	37,493	10,764	10,042	20,806	4	2,615	0,465	0,178	1,21	0,52	532,989	0,092

Таблица 4.6

Технико-экономические показатели 5 блока маршрутов

№ маршру.	$P_{\text{тр}}$, ткм	$P_{\text{а}}$, ткм	$P_{\text{общ}}$, ткм	L^{Γ} , км	L^{χ} , км	$L^{\text{общ}}$, км	II, шт.	$t^{\text{общ}}$, ч	Q , т	W , тч	W_p , ткмч	β	З, руб.	$K^{\#}$
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1,937	17,898	19,835	5,955	4,892	10,847	6	2,41	0,458	0,190	0,804	0,55	487,436	0,108
2	3,579	24,684	28,263	9,985	4,975	14,960	7	2,717	0,494	0,182	1,317	0,67	551,119	0,145
3	2,563	20,039	22,602	7,852	4,293	12,145	5	2,396	0,458	0,191	1,070	0,65	485,445	0,128
4	1,665	15,164	16,828	5,841	3,349	9,190	6	2,487	0,489	0,197	0,670	0,64	501,912	0,110
5	1,918	15,956	17,873	7,984	1,686	9,670	5	2,379	0,471	0,198	0,806	0,83	480,662	0,120
6	2,160	23,417	25,577	8,528	5,664	14,192	8	2,528	0,439	0,174	0,854	0,60	512,511	0,092
7	2,470	22,631	25,101	8,642	5,074	13,716	5	2,48	0,468	0,189	0,996	0,63	502,699	0,109
8	1,949	16,917	18,866	8,564	1,689	10,253	5	2,399	0,472	0,197	0,812	0,83	484,95	0,115
9	1,417	21,661	23,078	7,464	5,664	13,128	4	2,283	0,435	0,191	0,621	0,57	462,69	0,065
10	2,266	19,325	21,591	6,553	5,159	11,712	6	2,462	0,465	0,189	0,920	0,56	498,268	0,117

Таблица 4.7

Технико-экономические показатели 6 блока маршрутов

№ маршру.	P_{TP} , ТКМ	P_A , ТКМ	$P_{OБЦ}$, ТКМ	$L^Г$, км	L^X , км	$L^{OБЦ}$, км	П, шт.	$t^{OБЦ}$, ч	Q , т	W , тч	W_P , ТКМЧ	β	Z , руб.	K^W
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2,101	15,297	17,398	6,681	2,590	9,271	9	2,446	0,440	0,180	0,859	0,72	493,86	0,137
2	1,791	18,147	19,938	5,811	5,187	10,998	4	2,232	0,437	0,196	0,802	0,53	451,69	0,099
3	1,637	19,868	21,505	6,792	5,249	12,041	5	2,386	0,456	0,191	0,686	0,56	482,873	0,082
4	2,211	18,381	20,592	8,591	2,549	11,140	6	2,442	0,464	0,190	0,907	0,77	494,037	0,120
5	1,566	13,294	14,860	6,290	1,767	8,057	5	2,243	0,448	0,200	0,698	0,78	452,709	0,118
6	0,662	7,308	7,970	2,219	2,210	4,429	4	2,053	0,438	0,213	0,322	0,50	412,76	0,091
7	1,453	11,345	12,798	5,823	1,053	6,876	4	2,078	0,427	0,205	0,699	0,85	419,014	0,128
8	2,068	13,487	15,555	6,284	1,890	8,174	5	2,259	0,451	0,200	0,915	0,77	455,98	0,153
9	1,028	10,410	11,438	4,080	2,229	6,309	5	2,163	0,440	0,203	0,475	0,65	435,725	0,099
10	1,050	14,560	15,610	3,938	4,886	8,824	5	2,202	0,432	0,196	0,477	0,45	444,448	0,065

Таблица 4.8

Средние значения показателей по блокам маршрутов

№ блока	P_{TP} , ТКМ	P_A , ТКМ	$P_{OБЦ}$, ТКМ	$L^Г$, км	L^X , км	$L^{OБЦ}$, км	П, шт.	$t^{OБЦ}$, ч	Q , т	W , тч	W_P , ТКМЧ	β	Z , руб.	K^W
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	11,674	124,376	136,05	21,150	13,885	35,050	12	5,2	0,918	0,176	2,23	0,6	1085,5	0,095
2	3,716	34,863	38,743	12,905	8,324	21,228	5	2,69	0,469	0,175	1,364	0,6	548,23	0,106
3	11,245	107,563	118,808	17,546	12,754	30,300	10	5,09	0,950	0,187	2,210	0,57	1057,9	0,106
4	3,421	30,448	33,87	10,661	7,792	18,453	5	2,6	0,464	0,178	1,307	0,57	529,7	0,112
5	2,192	19,769	21,961	7,737	4,245	11,982	6	2,45	0,465	0,190	0,890	0,65	469,77	0,111
6	1,557	14,210	15,767	5,651	2,961	8,612	5	2,25	0,443	0,197	0,684	0,66	454,31	0,109

Затем на втором – соединялись маршруты из 3 и 6 блоков, и получался второй промежуточный совмещенный маршрут. На последнем этапе из полученных формировался общий совмещенный маршрут. В качестве примера были рассмотрены маршруты: 3 из блока 1, 9 из блока 3, 6 из блока 5 и 1 из блока 6, совмещение которых представлено на рисунке 4.8.

В результате расчета по формуле (4.1) получено, что допустимое время заезда 1,95 часа превышает фактически затраченное время заезда во второй пункт погрузки, которое составляет 0,18 часа. Следует отметить, что 0,75 часа в допустимом времени заезда экономится за счет совмещения пунктов разгрузки. В среднем по всем маршрутам допустимое время заезда во второй пункт погрузки составляет 1,93 часа, а фактическое – 0,15 часа.

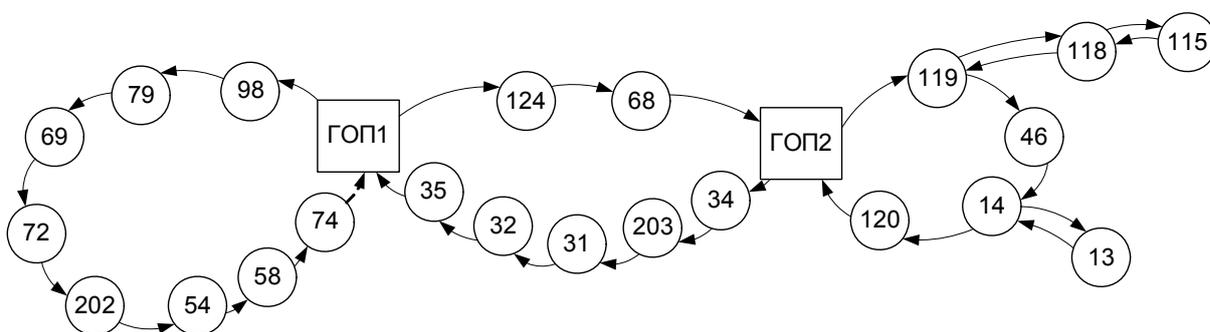


Рис. 4.8. Маршрут, полученный в результате совмещения

При анализе модели по второму типу совмещения проводилось объединение маршрутов доставки основного груза из блока 4 с маршрутами доставки дополнительного груза из блока 2.

Так как предлагается обслуживание одним АТП, то аналогичное изменение нулевых пробегов следует внести и в формулу (3.36). Следовательно, оно примет вид:

$$\left[t_{mD}^{3c} \right] = \sum t^o, \quad (4.2)$$

где $\sum t^o = t_{mB}^{oo} + t_{nD}^{од} + t_{nB}^{oc}$.

В качестве примера были взяты маршруты: 2 из блока 2 и 2 из блока 4, совмещение которых представлено на рис. 4.9.

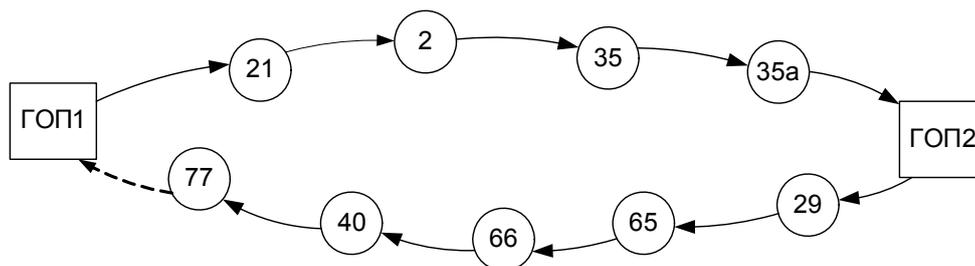


Рис. 4.9. Маршрут, полученный в результате совмещения

В результате расчета по формуле (4.2) получено, что для маршрута, представленного на рис. 4.9, допустимое время заезда 0,47 часа. Оно превышает фактически затраченное время заезда во второй пункт погрузки, которое составляет 0,01 часа. В среднем по всем маршрутам допустимое время заезда во второй пункт погрузки составляет 0,35 часа, а фактическое – 0,09 часа, что подтверждает целесообразность этого заезда.

Следует отметить, что в общем случае при совмещении маршрутов из рассматриваемой выборки можно получить различные комбинации их объединения. Однако маршруты имеют разные характеристики: количество перевозимого груза, число пунктов доставки, расстояние, тип транспортного средства и его грузоподъемность. Поэтому для определения эффективности совмещения в целом по всем маршрутам, в каждом блоке вычислялись средние технико-экономические показатели, которые приведены в табл. 4.8. Затем по этим показателям и оценивалась эффективность совмещения. В табл. 4.9 приведены данные, характеризующие целесообразность совмещения маршрутов по первому типу, то есть когда заезд во второй пункт погрузки осуществляется с промежуточного пункта основного маршрута. Графическая иллюстрация такого вида совмещения представлена на рис. 4.10.

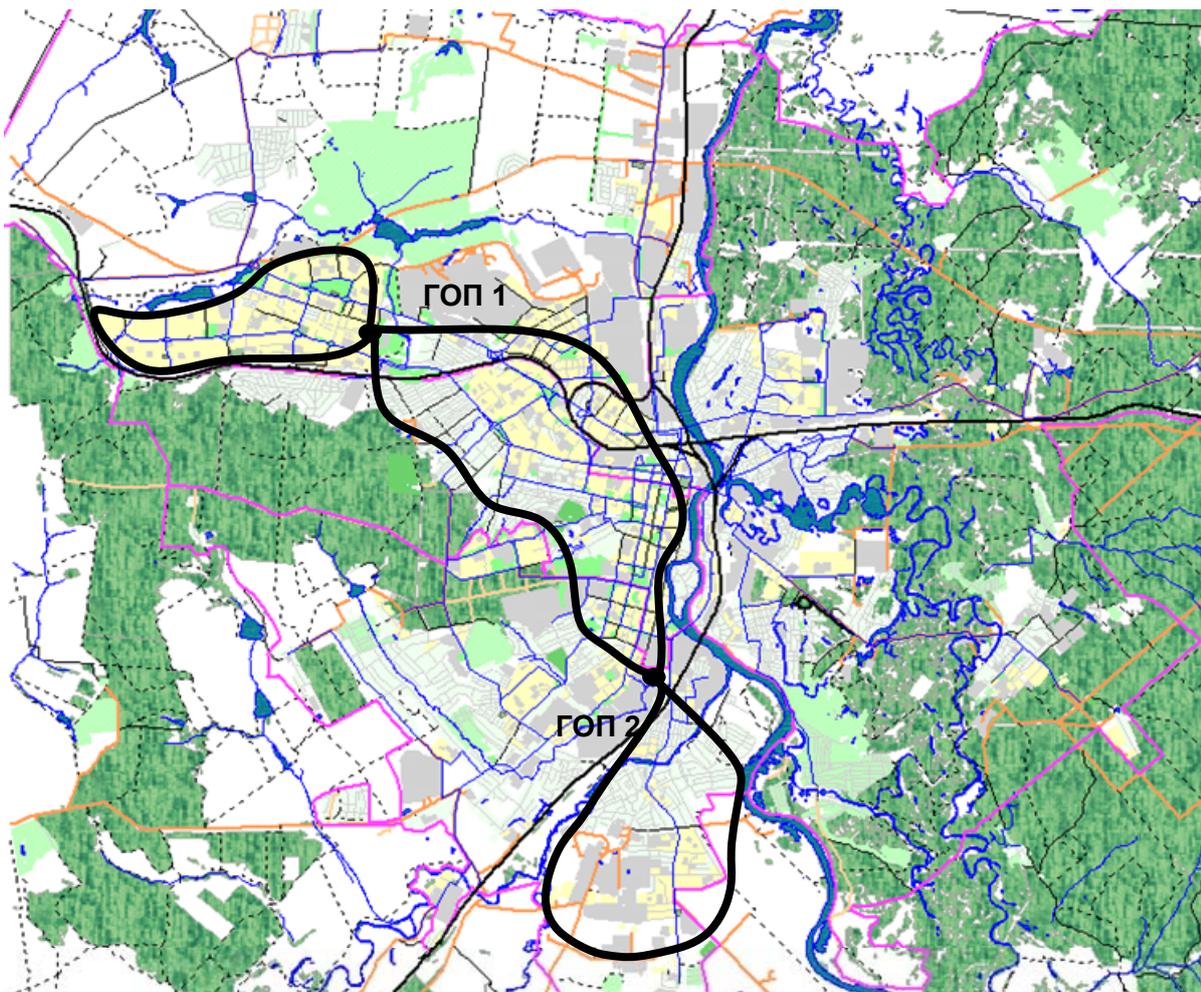


Рис. 4.10. Моделируемая транспортная схема первого типа

Таблица 4.9

Маршруты	$t^{\text{общ}}$	W	W_p	L^x	β	$З$	K^w
До совмещения	15,00	0,158	1,778	33,845	0,61	3067,5	0,007
После совмещения	13,35	0,206	1,843	7,823	0,83	2733,8	0,011

Из анализа данных, приведенных в табл. 4.9, следует:

1. Общая продолжительность маршрута сократилась на 11 %;
2. Производительность W в т/ч повысилась на 30 %;
3. Производительность W_p в ткм/ч повысилась на 4 %;
4. Длина холостых пробегов сократилась на 76 %;
5. Коэффициент использования пробега возрастает на 36 %;
6. Затраты сократились на 11 %.
7. Коэффициент относительной производительности увеличился на 57 %;

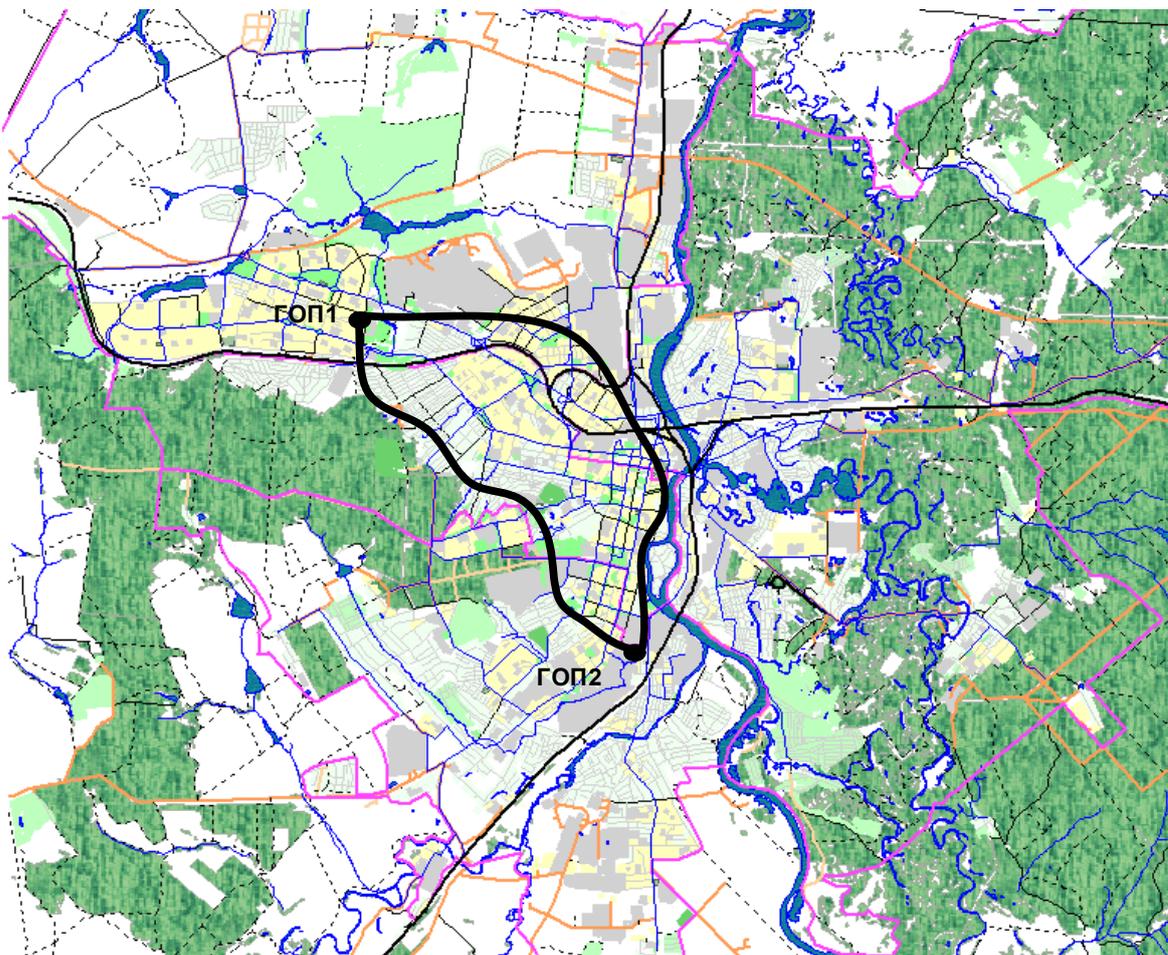


Рис. 4.11. Моделируемая транспортная схема второго типа

Т а б л и ц а 4.10

Маршруты	$t^{\text{общ}}$	W	W_p	L^x	β	$З$	K^w
До совмещения	5,3	0,176	1,347	16,12	0,59	1078	0,021
После совмещения	5,03	0,185	1,418	6,965	0,77	1023	0,028

В табл. 4.10 приведены данные, характеризующие эффективность совмещения маршрутов по второму типу, когда заезд во второй пункт погрузки производится с последнего пункта основного маршрута. Графическая иллюстрация такого вида совмещения представлена на рис. 4.11.

Из анализа данных, приведенных в табл. 4.10, следует:

1. Общая продолжительность маршрута сократилась на 5 %;
2. Производительность W в т/ч повысилась на 5 %;
3. Производительность W_p в ткм/ч повысилась на 5 %;
4. Длина холостых пробегов сократилась на 57 %;
5. Коэффициент использования пробега возрастает на 30 %;
6. Затраты сократились на 5 %.
7. Коэффициент относительной производительности увеличился на 33 %;

Данные, приведенные в табл. 4.9 и 4.10, подтверждают целесообразность совмещения маршрутов по рассмотренным моделям. Следует отметить, что применение автомобилей большей грузоподъемности приводит к сокращению их количества на маршруте, а, следовательно, экономии фонда заработной платы водителей.

4.5. Оценка применения методики «фиктивных ветвей»

Известно, что классический метод «ветвей и границ» подразумевает однократное прохождение по ветви графа. Однако на практике часто встречаются ситуации, требующие повторного посещения какого-либо его звена. Поэтому для нахождения оптимального маршрута в таких случаях была предложена методика «фиктивных ветвей», разработке которой была посвящена вторая глава.

Для сравнения предложенной методики с классическим методом были проанализированы 80 вариантов маршрутов, которые могли бы получиться при различных комбинациях совмещения маршрутов из блоков 1 и 5 рассмотренной транспортной сети. Как показали расчеты, 27 из них имеют

Т а б л и ц а 4.11

Расчет по методу	Длина маршрута L , км	Время $t^{\text{общ}}$, час
«Ветвей и границ»	37,009	7,029
«Фиктивных ветвей»	30,512	6,848

Как видно из табл. 4.11, сокращение длины маршрута от расчета по предложенной методике составило 21,3 %. В среднем по всем маршрутам оно составило 8 %. Таким образом, предложенная методика позволяет сократить длину и время пробега.

Следует отметить, что затраты времени на расчет по предложенной методике с помощью разработанной компьютерной программы для данного примера составили 1 сек.

ЗАКЛЮЧЕНИЕ

В настоящее время в связи с уменьшением жизненного цикла изделий и, как следствие этого, роста количества мелкопартионных перевозок грузов, возникает необходимость в повышении использования подвижного состава. Этого можно достичь за счет организации совместной развозки на кольцевых маршрутах, т.е. доставкой попутного груза.

В результате проведенных исследований разработан инструментарий, позволяющий формировать наиболее рациональные маршруты доставки мелкопартионных грузов и за счет этого повышать эффективность перевозок.

Модернизирован метод «ветвей и границ», используемый для решения задач маршрутизации грузовых перевозок, суть которого заключается в совершенствовании классического алгоритма расчета путем введения фиктивных узлов и связей в граф транспортной сети.

Разработана и реализована на ПЭВМ математическая модель планирования совмещения кольцевых маршрутов, содержащая ряд новых элементов. Это позволило провести анализ показателя целесообразности совмещения. Установлено, что эффективность совмещения кольцевых маршрутов тем выше, чем больше величина встречных холостых и нулевых пробегов. В результате совмещения маршрутов в среднем повышаются: коэффициент использования пробега – на 30–36%, производительность в т/ч на – 5–30%, производительность в ткм/ч – на 4–5%, коэффициент относительной производительности – на 33–57%. При этом сокращаются: величина холостых пробегов автомобилей – на 57–76%, время работы автомобиля на маршруте и затраты в рублях – на 5–11%.

Таким образом, создание модификаций точных методов является наиболее перспективным направлением для решения задач маршрутизации доставки мелкопартионных грузов и позволяет усовершенствовать оперативное планирование и тем самым повысить эффективность автомобильных перевозок.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Азизов, Ф.Х. Применение математических методов при планировании перевозок товаров [Текст] / Ф.Х. Азизов, А.И. Ахмедов, В.П. Кочулов. – М.: Госторгиздат, 1963. – 104 с.
2. Алгоритмы. Построение и анализ [Текст] / Томас Кормен [и др.]. – 2-е изд.: пер. с англ. – М.–СПб.–Киев: Изд. дом «Вильямс», 2007. – 1296 с.
3. Афанасьев, Л.Л. Производительность работы автомобильного парка при грузовых перевозках [Текст]: дис... канд. техн. наук / Л.Л. Афанасьев – М., 1940. – 146 с.
4. Афанасьев, Л.Л. Единая транспортная система и автомобильные перевозки [Текст]: учеб. для студентов вузов / Л.Л. Афанасьев, Н.Б. Островский, С.М. Цукерберг. – 2-е изд., перераб. и доп. – М.: Транспорт, 1984. – 333 с.
5. Ахметзянов, А. Моделирование грузовых и пассажирских потоков в городе и регионе [Текст] / А.Ахметзянов // Логинфо. – 2002. – №1.
6. Беллман, Р. Применение динамического программирования к задаче о коммивояжере [Текст] / Р. Беллман // Кибернетический сб. – 1964. – Вып.9. – С. 219 – 222.
7. Беллман, Р. Прикладные задачи динамического программирования [Текст] / Р. Беллман, С. Дрейфус. – М.: Наука, 1965. – 458 с.
8. Бенсон, Д. Транспорт и доставка грузов [Текст] / Д.Бенсон, Дж.Уайтхед: пер. с англ. – М.: Транспорт, 1990. – 279 с.
9. Брегман, В.И. Графы в задачах управления производством [Текст] / В.И. Брегман. – М.: Статистика, 1974. – 144 с.
10. Вадас, П. Новые методы по разработке маршрутов с помощью ЭВМ [Текст]: перевод №Б-832/2 / П. Вадас – М.: ВЦП, 1982. – 32 с.
11. Великанов, Д.П. Эффективность автомобиля [Текст] / Д.П. Великанов. – М.: Транспорт, 1989.
12. Вельможин, А.В. Теория организации и управления автомобильными перевозками: логистический аспект формирования перевозочных процессов [Текст]: монография / А.В. Вельможин, В.А. Гудков, Л.Б. Миротин. – Волгоград: Волгоград. гос. тех. ун-т, 2001. – 178 с.
13. Вельможин, А.В. Технология, организация и управление грузовыми автомобильными перевозками [Текст]: учеб. для вузов / А.В. Вельможин, В.А. Гудков, Л.Б. Миротин. – 2-е изд., доп. – Волгоград: Волгоград. гос. тех. ун-т., 2000. – 304 с.
14. Вельможин, А.В. Грузовые автомобильные перевозки [Текст]: учеб. для вузов / А.В. Вельможин [и др.]. – М.: Горячая линия – Телеком, 2006 – 560 с.
15. Венгерова, И.В. К вопросу об эффективности метода ветвей и границ [Текст] / И.В. Венгерова, Ю.Ю. Финкельштейн // Экономика и мат. методы. – 1975. – № 1.

16. Вензик, Н.Г. Торговля и транспорт [Текст] / Н.Г. Вензик // БТИ. – 1999. – №4. – С. 13 – 15.
17. Титюхин, Н.Ф. Взаимодействие участников логистической цепи. Место транспортно-экспедиторской компании в логистической цепи [Текст] / Н.Ф. Титюхин // Логинфо. – 2003. – №12.
18. Витвицкий, Е.Е. Развозочно-сборные автотранспортные системы перевозки грузов [Текст]: монография / Е.Е. Витвицкий – Омск: Изд-во «Вариант-Сибирь», 2003. – С. 18.
19. Воркут, А.И. Автомобильные перевозки партионных грузов [Текст] / А.И. Воркут. – Киев: Вища школа, 1974. – 184 с.
20. Воркут, А.И. Грузовые автомобильные перевозки (Основы теории транспортного процесса) [Текст]: учеб. пособ. для вузов / А.И. Воркут. – Киев: Вища школа, 1979. – 392 с.
21. Воркут, А.И. Грузовые автомобильные перевозки [Текст] / А.И. Воркут. – 2-е изд., перераб. и доп. – М.: Транспорт, 1986. – 447 с.
22. Воркут, А.И. Разработка теоретических основ и методов рациональной организации транспортного процесса при автомобильных перевозках партионных грузов [Текст]: дис... докт. техн. наук / А.И. Воркут. – Киев, 1986. – 442 с.
23. Воркут, А.И. Транспортное обслуживание торгово-оптовых баз [Текст] / А.И. Воркут. – Киев: Техніка, 1985. – 112 с.
24. Герасимович, А.И. Математическая статистика [Текст] / А.И. Герасимович. – Минск: Выш. школа. 1983. – 279 с.
25. Геронимус, Б.Л. Математические методы оперативного планирования грузовых автомобильных перевозок [Текст] / Б.Л. Геронимус. – М.: Транспорт, 1972. – 102 с.
26. Голубев, С. Доставка молока в торговую сеть [Текст] / С. Голубев, А. Бречко // Автомобильный транспорт. – 1973. – №1. – С. 23 – 24.
27. Гольденберг, Я.М. Эксплуатация автогужевого транспорта [Текст] / Я.М. Гольденберг. – М.: Союзтранс, 1932. – Вып.1 – 77 с.
28. Гольштейн, Е.Г. Задачи линейного программирования транспортного типа [Текст] / Е.Г. Гольштейн, Д.Б. Юдин. – М.: Наука, 1969. – 384 с.
29. Ванчукевич, В.Ф. Грузовые автомобильные перевозки [Текст]: учеб. пособ. / В.Ф. Ванчукевич, В.Н. Седюкевич, В.С. Холупов. – Минск: Выш. шк., 1978. – 272 с.
30. Гудков, В.А. Влияние работы автотранспортного обслуживающего комплекса на качество хлебобулочных изделий и экономическую эффективность системы доставки [Текст] / В.А. Гудков, С.А. Кашеев, С.А. Ширяев // Наземные транспортные системы: межвуз. сб. науч. тр. – Волгоград: РПК «Политехник», 2000. – С. 111–114.
31. Гудков, В.А. Особенности функционирования производственно-транспортной системы доставки продуктов питания [Текст] / В.А. Гудков,

С.А. Ширяев, С.А. Кащеев // Проблемы качества и эксплуатации автотранспортных средств: сб. матер. I междунар. науч.-техн. конф. (4–6 апреля 2000 г.). – Пенза, 2000. – Ч. 2. – С. 125 – 130.

32. Гудков, В. А. Формирование логистической цепи доставки продуктов питания (на примере хлебобулочных изделий) [Текст] / В.А. Гудков, С. А. Ширяев, С.А. Кащеев // Бизнес и логистика–2001: сб. матер. Московского международного логистического форума (Москва, 1–4 февраля 2001 г.) / Координационный совет по логистике. – М., 2001. – С. 180 – 182.

33. Гудков, В. А. Логистический подход к организации процесса перевозки хлебобулочной продукции [Текст] / В.А. Гудков, С.А. Ширяев, К.С. Подшивалова // Прогресс транспортных средств и систем: сб. матер. междунар. науч.-практ. конф. (20–23 сентября 2005 г.) – Волгоград, 2005. – Ч. 1. – С. 523 – 524.

34. Гудков, В. А. К теории совместной развозки партионных грузов [Текст] / В. А. Гудков, С. А. Ширяев, К. С. Подшивалова, С.Ф. Подшивалов // Проблемы качества и эксплуатации автотранспортных средств: сб. матер. междунар. науч.-практ. конф. (10 – 12 мая 2006 г.). – Пенза, 2006. Ч. 1. – С. 298 – 302.

35. Гудков, В. А. Новый взгляд на показатели работы автомобилей [Текст] / В. А. Гудков, С. А. Ширяев, К. С. Подшивалова, С.Ф. Подшивалов // Проблемы качества и эксплуатации автотранспортных средств: сб. матер. междунар. науч.-практ. конф. (10 – 12 мая 2006 г.) – Пенза, 2006. – Ч. 1. – С. 303 – 307.

36. Гудков, В. А. Совместная развозка партионных грузов на кольцевом [Текст] / В. А. Гудков, С. А. Ширяев, К. С. Подшивалова, С.Ф. Подшивалов // Проблемы качества и эксплуатации автотранспортных средств: сб. матер. междунар. науч.-практ. конф. (10 – 12 мая 2006 г.). – Пенза, 2006. – Ч. 1. – С. 307 – 313.

37. Джонсон, Дж. Современная логистика [Текст] / Дж. Джонсон [и др.] – 7-е изд. – М.: Вильямс, 2002. – С. 8–27.

38. Житков, В.А. Алгоритм приближенного решения задачи развозки [Текст] / В.А. Житков // Методы оптимизации перевозочного процесса на автотранспорте. – М., 1976. – С. 84 – 90.

39. Житков, В.А. Математические методы организации и планирования автомобильных перевозок мелких партий грузов [Текст]: дис... канд. техн. наук / В.А. Житков. – М.: МАДИ, 1968. – 134 с.

40. Житков, В.А. Планирование автомобильных перевозок грузов мелкими партиями [Текст] / В.А. Житков. – М.: Транспорт, 1976. – 112 с.

41. Житков, В.А. Методы оперативного планирования грузовых автомобильных перевозок [Текст] / В.А. Житков, К.В. Ким. – М.: Транспорт, 1984. – 218 с.

42. Зарецкий, Л.С. Решение задачи коммивояжера и задач развозки методом коррекции функции состояния [Текст] / Л.С. Зарецкий // Методы оптимизации перевозочного процесса на автотранспорте. – М., 1976. – С. 70 – 83.
43. Зыков, А.А. Основы теории графов [Текст] / А.А. Зыков. – М.: Наука, 1987.
44. Исследование операций. Методологические основы и математические методы [Текст]: пер. с англ. / Под ред. И.М. Макарова, И.М. Бескровного. – М.: Мир, 1981. – Т. 1. – 712 с.
45. Каниовский, П.В. Эксплуатация автомобильного транспорта [Текст] / П.В. Каниовский. – М.: Изд-во МКХ РСФСР, 1947.
46. Кашеев, С.А. Повышение эффективности функционирования системы доставки грузов в торговую сеть автомобильным транспортом [Текст]: дис... канд. техн. наук / С.А. Кашеев. – Волгоград, 2004. – 195 с.
47. Клепик, Н.К. Статистическая обработка эксперимента в задачах автомобильного транспорта [Текст]: учеб. пособие / Н.К. Клепик. – Волгоград: ВГТУ, 1995. – 96 с.
48. Ковалев, М.М. Дискретная оптимизация (целочисленное программирование) [Текст] / М.М. Ковалев. – 2-е изд., стер. – М.: Едиториал УРСС, 2003. – 192 с.
49. Кожин, А.П., Математические методы в планировании и управлении грузовыми автомобильными перевозками [Текст]: учеб. для вузов / А.П. Кожин, В.Н. Мезенцев. – М.: Транспорт, 1994 – С. 124.
50. Корбут, А.А. Дискретное программирование [Текст] / А.А. Корбут, Ю.Ю. Финкельштейн. – М.: Наука, 1969. – 368 с.
51. Коробков, В.К. Некоторые алгоритмы для решения задачи коммивояжера [Текст] / В.К. Коробков, Р.И. Кричевский // Математические модели и методы оптимального управления. – Новосибирск: Наука, 1966. – С. 106–108.
52. Корягин, М.Е. Планирование автомобильных перевозок в торговой сети [Текст] / М.Е. Корягин, А.Ю. Тюрин // Вести КузГТУ. – 2000. – №1. – С. 43 – 45.
53. Косован, А.П. Новые подходы к калькуляции себестоимости и формированию ассортимента продукции на хлебозаводах [Текст] / А.П. Косован [и др.] // Хлебопечение России. – 2000. – №1. – С. 10 – 12.
54. Кофман, А. Сетевые методы планирования и их применение [Текст] / А. Кофман, Г. Дебазей. – М.: Прогресс, 1968. – 181 с.
55. Крампе, Х. Логистика, как фактор развития производства в условиях рыночной экономики [Текст] / Х. Крампе // Подъёмно-транспортная техника и склады. – 1991. – № 6. – С. 43 – 45.
56. Крестофидес, Н. Теория графов. Алгоритмический подход [Текст] / Н. Крестофидес. – М.: Мир, 1978. – 396 с.

57. Левачев, Н.А. Механизация погрузочно-разгрузочных, транспортных и складских работ в пищевой промышленности [Текст] / Н.А. Левачев. – М.: Легкая и пищевая промышленность, 1984. – 184 с.
58. Лейдерман, С.Р. Анализ влияния эксплуатационно-технических измерителей на производительность и себестоимость работы автомобилей [Текст] / С.Р. Лейдерман // сб. тр. ЦНИИАТ. – М.: Коммухоз, 1949. – Вып.4.
59. Лейдерман, С.Р. Основные измерители и коэффициенты автомобилей [Текст] / С.Р. Лейдерман // Мотор. – 1932. – №6.
60. Лесин, В.В. Основы методов оптимизации [Текст] / В.В. Лесин, Ю.П. Лисовец. – М.: Изд-во МАИ, 1995. – 344 с.
61. Лесов, Ю.И. Совершенствование внутригородских автомобильных перевозок продовольственных и промышленных товаров [Текст]: докл. по совокупности опубликованных работ на соискание ученой степени канд. техн. наук / Ю.И. Лесов. – М.: 1968. – 38 с.
62. Литтл, Дж. Алгоритм решения задачи коммивояжера [Текст] / Дж. Литтл [и др.] // Экономика и математические методы. – 1965. – №1. – С. 94 – 107.
63. Логистика: Управление в грузовых транспортно-логистических системах [Текст]: учеб. пособие / Под ред. Л.Б. Миротина. – М.: Юристъ, 2002.
64. Лукинский, В.С. Теоретические проблемы современной логистики [Текст] / В.С. Лукинский, И.А. Цвиринько, А.Э. Троянов // Логистика в современном бизнесе: сб. тез. и докл. междунар. семинара в рамках выставки «ТРАНСРОССИЯ 2002». – М.: Изд-во ГУ-ВШЭ, 2002. – С. 16–20.
65. Магомедов, М.Д. Снижение транспортных затрат на перевозку хлеба – фактор повышения конкурентоспособности хлебозавода [Текст] / М.Д. Магомедов, К.Л. Быстров // Хлебопечение России. – 1999. – №2. – С. 9–11.
66. Малышев, В.В. Обобщенное динамическое программирование. Общие положения [Текст] / В.В. Малышев, Д.Э. Чернов // Автоматика и телемеханика. – 1993. – №12. – С.101 – 110.
67. Малышев, В.В. Обобщенное динамическое программирование. Некоторые приложения [Текст] / В.В. Малышев, Д.Э. Чернов // Автоматика и телемеханика. – 1994. – №1. – С.117 – 127.
68. Меламед, И.И. Методы оптимизации в транспортном процессе [Текст] / И.И. Меламед // ИНТ ВИНТИ. Сер. Организация управления транспортом. – 1991. – № 10. – С. 1–164.
69. Механизация работ в хлебохранилищах и экспедициях хлебозаводов [Текст] / Э.А. Бронштейн [и др.]. – М.: Пищевая промышленность, 1973. – 304 с.

70. Миротин, Л.Б. Логистика: обслуживание потребителей [Текст] / Л.Б. Миротин, Ы.Э. Ташибаев, А.Г. Касенов. – М.: ИНФРА-М, 2002. – 190 с.
71. Модели и методы теории логистики [Текст]: учеб. пособие / Под ред. В.С. Лукинского – СПб.: Питер, 2003. – 176 с.
72. Мудров, В.И. Задача о коммивояжере [Текст] / В.И. Мудров. – М.: Знание, 1970.
73. Нестеров, Е.П. Транспортные задачи линейного программирования [Текст] / Е.П. Нестеров. – М.: Транспорт, 1977. – 216 с.
74. Николин, В.И. Автотранспортный процесс и оптимизация его элементов [Текст] / В.И. Николин. – М.: Транспорт, 1990. – 191 с.
75. Николин, В.И. Научные основы совершенствования теории грузовых автомобильных перевозок [Текст]: автореф. дис... докт. техн. наук / В.И. Николин. – М., 2000. – 37 с.
76. Николин, В.И. Грузовые автомобильные перевозки [Текст]: монография / В.И. Николин, Е.Е. Витвицкий, С.М. Мочалин. – Омск: Изд-во «Вариант-Сибирь», 2004. – С. 8–27.
77. Новые нормы расхода топлива и ГСМ [Текст]. – М.: ИНФРА-М., 2002. – 64 с.
78. Оптимизация планирования и управления транспортными системами [Текст] / под ред. В.Н. Лифшица – М.: Транспорт. – 1987.
79. Оре, О. Теория графов [Текст] / О. Оре. – М.: Наука, 1968. – 352 с.
80. Организация, планирование и управление автотранспортными предприятиями [Текст]: учеб. для студентов вузов, обучающихся по специальности «Экономика и организация автомобильного транспорта» / Н.Ф. Билибина [и др.]; под ред. Л.А. Бронштейна, К.А. Савченко-Бельского. – 2-е изд., перераб. и доп. – М.: Высш. шк., 1986. – 360 с.
81. Панов, С.А. Модели маршрутизации на автомобильном транспорте [Текст] / С.А. Панов. – М.: Транспорт, 1974. – 152 с.
82. Петрова, Е.В. Статистика на автомобильном транспорте [Текст] / Е.В. Петрова, О.И. Ганченко. – М.: Транспорт. – 1997.
83. Подшивалов, С. Ф. Оценка эффективности перевозок [Текст] / С. Ф. Подшивалов, К.С. Подшивалова // Прогресс транспортных средств и систем: сб. матер. междунар. науч.-практ. конф. (20–23 сентября 2005 г.). – Волгоград, 2005. – Ч. 1. – С. 498 – 499.
84. Подшивалов, С.Ф., Подшивалова К.С. К построению общего алгоритма решения задачи странствующего коммивояжера [Текст] / С. Ф. Подшивалов, К.С. Подшивалова // Прогресс транспортных средств и систем: сб. матер. междунар. науч.-практ. конф. (20–23 сентября 2005 г.). – Волгоград, 2005. – С. 571 – 572.
85. Подшивалова, К. С. Решение задачи маршрутизации для кольцевого маршрута методом фиктивных матриц [Текст] / К.С. Подшивалова //

Проблемы качества и эксплуатации автотранспортных средств: сб. матер. междунар. науч.-практ. конф. (10 – 12 мая 2006 г.). – Пенза, 2006. Ч. 1. – С. 356 – 370.

86. Подшивалова, К.С. Логистика перевозок хлебобулочной продукции [Текст] / К.С. Подшивалова, В.А. Гудков, С.А. Ширяев // Грузовое и пассажирское автохозяйство. – 2005. – № 11. – С. 59 – 60.

87. Попченко, Я.А. Пути повышения эффективности грузовых автомобильных перевозок [Текст] / Я.А. Попченко, Г.Д. Луцкер. – М.: Транспорт, 1986. – 94 с.

88. Портнов, А.С. Эффективное распределение продукции по торговым филиалам с применением системы планирования, прогнозирования и оптимизации [Текст] / А.С. Портнов // ММЛФ-2001: сб. тезисов выступлений на ММЛФ-2001. – М., 2001. – С. 68 – 71.

89. Поставка товаров народного потребления [Текст]: сб. нормативных актов. – М.: Юридическая литература, 1986. – 128 с.

90. Прокофьева, О.С. Разработка методики оптимизации развозочных маршрутов [Текст]: дис... канд. техн. наук / О.С. Прокофьева. – Иркутск, 2004. – 169 с.

91. Просов, С.Н. Повышение эффективности планирования перевозок по сборно-развозочным маршрутам [Текст]: дис... канд. техн. наук / С.Н. Просов. – М.: МАДИ-ТУ, 1985. – 125 с.

92. Просов, С.Н. Маршрутизация грузовых автомобильных перевозок [Текст]: дис... канд. техн. наук / С.Н. Просов // Материалы ММЛФ-2002. – М.: МАДИ-ТУ, 2002. – С. 77 – 92.

93. Резер, С.М. Математические методы оптимального планирования в транспортных системах [Текст] / С.М. Резер, С.Е. Ловецкий, И.И. Меламед // ИНТ ВИНТИ. Сер. Организация управления транспортом. – 1990. – С. 1–172.

94. Ройтман, Я.М. Исследование влияния технико-эксплуатационных показателей на эффективность использования автомобилей [Текст]: дис... канд. техн. наук / Я.М. Ройтман. – М., 1968.

95. Романовская, А.М. Задачи оптимизации на графах [Текст]: учеб. пособие / А.М. Романовская. – Омск: РГТЭУ, 2007. – 89 с.

96. Романовский, И.В. Односторонний обход дерева вариантов в методе Ленд и Дойг [Текст] / И.В. Романовский, М.П. Сорокина // Вычислительная математика и математическая физика. – 1973. – № 1.

97. Ромашко, М.В. Совершенствование процесса доставки мелкопартионных грузов при сокращении срока доставки [Текст]: автореф. дис... канд. техн. наук / М.В. Ромашко. – М., 2000.

98. Семенов, Е.В. Составление на ЭВМ графика подачи автомашин под погрузку и доставку готовой продукции в торговую сеть с одновременным

подбором оптимальных маршрутов [Текст] / Е.В. Семенов [и др.] // Труды ВНИИМП. – 1970. – Вып. 27. – С. 212 – 215.

99. Семухина, Н.В. Некоторые вопросы организации перевозок мелко-партионных грузов автомобильным транспортом на коротком плече [Текст]: дис... канд. экон. наук / Н.В. Семухина. – М., 1971. – 232 с.

100. Серджвик, Роберт. Фундаментальные алгоритмы на С. Алгоритмы на графах [Текст]: пер. с англ. / Роберт Серджвик. – СПб.: ООО «Диа Софт ЮП», 2003. – 480 с.

101. Смехов, А.А. Математические модели процессов грузовой работы [Текст] / А.А. Смехов. – М.: Транспорт, 1982. – 255 с.

102. Триус, Е.Б. Задачи математического планирования транспортного типа [Текст] / Е.Б. Триус. – М.: Сов. Радио, 1967. – 208 с.

103. Ту, Дж. Принципы распознавания образов [Текст] / Дж. Ту, Р. Гонсалес. – М.: Мир, 1978. – 284 с.

104. Тюрин, А.Ю. Логистическая система управления транспортом в хлебопекарной промышленности [Текст]: автореф. дис... канд. экон. наук / А.Ю. Тюрин. – Ростов н/Д., 2002.

105. Управление транспортным парком [Текст] / К. Кирсанов, С.Луцкий // Логинфо. – 2003. – №5.

106. Фомин, Г.П. Математические методы и модели в коммерческой деятельности [Текст] / Г.П. Фомин. – М.: Финансы и статистика, 2005.

107. Фридман, А.А. Дискретные задачи и метод ветвей и границ [Текст] / А.А. Фридман, А.А. Вотяков // Экономика и мат. методы. – 1974. – № 3.

108. Харари, Фрэнк. Теория графов [Текст] / пер. с англ. и предисл. В.П. Козырева; под ред. Г.П. Гаврилова / Фрэнк Харари. – 2-е изд. – М.: Едиториал УРСС, 2003. – 296 с.

109. Хелд, М. Применение динамического программирования к задачам упорядочения [Текст] / М. Хелд, Р. Карп // Кибернетический сб. – 1964. – Вып.9. – С. 208–218.

110. Чудаков, Е.А. Избранные труды [Текст] / Е.А. Чудаков. – М.: Изд-во Академ. наук СССР, 1961. – Т. 1,2. – С. 462, 361.

111. Юдин, Д.Б. Линейное программирование [Текст] / Д.Б. Юдин, Е.Г. Гольштейн. – М.: Наука, 1969. – 424 с.

112. Юревич, Н.В. Анализ деятельности хлебокомбината с использованием методики «Издержки – оборот – прибыль» [Текст] / Н.В. Юревич, Н.А. Евланова // Хлебопечение России. – 1999. – №3. – С. 9 – 10.

113. Atkinson A., Epstein M. Measure for measure: Realizing the power of the balanced scorecard // CMA Management. – September 2000.

114. Balas E. A note on the branch-and-bound principle. – Oper. Res., 16, N 2, 1968.

115. Balinski M., Quandt R. On an integer program for a delivery problem. – Operational Research, 1964. V. 12. №2. p. 300 – 304.

116. Ballou R.H. Business Logistics Management. Third Edition. – Prentice – Hall International, Inc., 1993
117. Beltrami E., Bodin L. Networks and vehicle routing for municipal waste collection. – Networks, 1974. V. 4. № 1. p. 65 – 94.
118. Binbergen A., Visser J. New urban goods distribution systems. Conference on Urban Transport Systems. Lund, Sweden. 1999, 11 p.
119. Bowersox D.J., Closs D.J., Helderich O. K. Logistical Management. The Integrated Supply Chain Process. – The McGRAW –HILL Companies, ins. New York, 1996.
120. Carpaneto G., Toth P. Some new branching and bounding criteria for the asymmetric traveling salesman problem // Management Science. – 1980. – 26, №7. – p. 736 – 743.
121. Clark G., Wright J. Scheduling of vehicles central depot to an umber of delivery points. – Operational Research Quarterly, 1964. V. 12. № 4. p. 568 – 581.
122. Dantzig G., Fulkerson D., Jonson S. Solution of a largescale traveling salesman problem // Operational Research, 1954. V. 2. № 4. – p. 393 – 410.
123. Evans S.R., Norback J.P. The impact of a decision-support system for vehicle routing in foodservice supply situation. – Operational Research Quarterly, 1985. V. 36. № 4. p. 467 – 472.
124. Foster B.A., Ryan D.M. An integer programming approach to the vehicle scheduling problem. – Operational Research Quarterly, 1976. V. 27. №2. p. 367 – 384.
125. Gillett B., Miller L. A heuristic algorithm for the vehicle dispatch problem. – Operational Research, 1974. V. 22. № 3. p. 340 – 349.
126. Groes G.A. Method for Solving Traveling Salesman Problems. Operational Research, 1958, vol. 6, p. 791–814.
127. Held M. and Karp R.M. A Dynamic Programming Approach to Sequencing Problems. J. Soc. Indust. And Appl. Math., 1962, vol. 10, p. 196 – 210.
128. Lawler E.L., Lenstra I.K., Rinnooy Kan A. U. G., Shomoys D.B. The traveling salesman problem. // J. Wiley Sons. New York, 1985.
129. Lin S. Computer solution of the traveling salesman problem. // Bell System Technical Journal – 1965. – №4. – p. 677 – 682.
130. Miller C.E., Tucker A.W., Zemlin R.A. Integer programming formulation of traveling salesman problems. – Computer Machinery, 1960. V. 27. №2. p. 503 – 511.
131. Mitten L. Branch-and-bound methods: general formulation and properties. – Oper. Res., 18, N 1, 1970.
132. Robertson B. Route optimization is the better way to bigger profits: Walls. – Motor truck. 1984. V. 53. №5. p. 27 – 28.

133. Rusehrt J. Chr.Erfolgs-Erlebnis. – Last auto Omnibus, 1984. №8. p.26 – 27.
134. Russel R.A. An effective heuristic for m-tour traveling salesman problem with some side conditions. – Operational Research, 1977. V. 25. № 3.
135. Sweeney D.W. The Exploration of a New Algorithm for Solving the Traveling Salesman. M.S. Thesis, M., I. T., 1963.
136. Wong K.F., Beasley J.E. Vehicle routing using fixed delivery areas. – Omega. 1984. V.12. № 6. p. 591 – 600.
137. Wren A., Holliday A. Computer scheduling of vehicles from one or more depots to a number of delivery Points // Operational Research Quarterly, 1972. V. 23. №3. p. 333 – 344.

ПРИЛОЖЕНИЕ

МЕТОДИКА ОПТИМИЗАЦИИ СОВМЕЩЕННЫХ МАРШРУТОВ

1. Назначение методики

Для расчета рациональных маршрутов доставки мелкопартионных грузов, в соответствии с поставленной задачей маршрутизации, описанной во второй главе, была разработана компьютерная программа. Она позволяет находить оптимальный маршрут движения на заданной транспортной сети.

В качестве целевой функции можно использовать длину или время пути. Программа позволяет осуществить перебор всех возможных вариантов маршрутов, в соответствии с разработанным алгоритмом. Если в процессе расчета получается несколько маршрутов, имеющих одинаковое минимальное значение целевой функции, то она выбирает оптимальный маршрут с наименьшей транспортной работой.

Алгоритм разработан на основе совершенствования классического метода «ветвей и границ» путем введения, в процессе расчета матрицы весов, фиктивных ветвей.

Для вычисления используется методика рекурсивного левостороннего алгоритма. Все модули программы реализованы на языке C++ в среде программирования Borland C++ Builder 6.0.

Данная программа предусматривает два варианта введения фиктивных ветвей: автоматический и контролируемый. Использование второго варианта позволяет сократить количество фиктивных узлов, а соответственно и время, затрачиваемое на расчет поиска оптимального пути.

Кроме того, для поиска кратчайшего расстояния или времени между двумя любыми вершинами графа, была разработана программа, основанная на методе «метлы». Она использует ту же исходную матрицу весов.

2. Описание методики

Описание разработанной программы определения кратчайшего пути включает в себя следующее:

- описание интерфейса программы, которое включает в себя описание ввода исходных данных и описание рабочего режима программы;

- техническое описание программы, включающее в себя описание основных элементов главной формы программы;

- описание классов и интерфейсных функций.

Приведем описание программы согласно отмеченным пунктам.

2.1. Описание интерфейса программы

Разработанная программа определения оптимального маршрута имеет интуитивно понятный интерфейс (рис. 2.1.1), который соответствует стандартам эргономики.

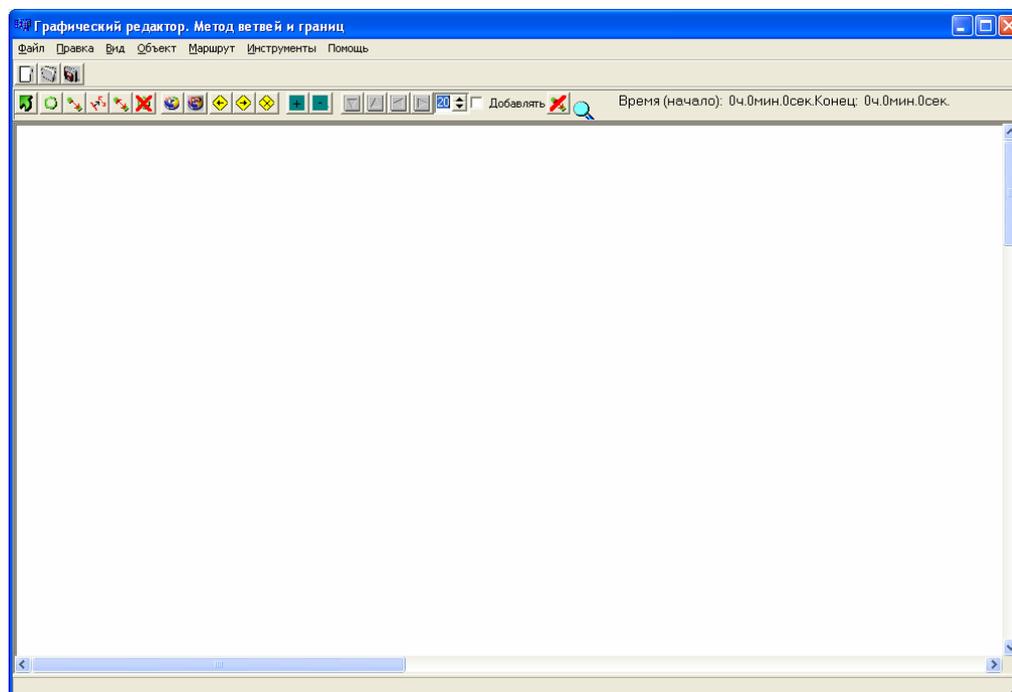


Рис. 2.1.1

Интерфейс программы после краткого ознакомления позволяет перейти к полноценной работе с программой.

2.1.1. Исходные данные для расчета

Для создания исходных данных для расчета требуется с использованием инструментальных средств программы сформировать граф, состоящий из вершин и связей между ними.

Перед созданием нового графа, необходимо выполнить команду «Новый» из меню «Файл». Открыть для редактирования ранее сохраненный граф можно с помощью команды «Файл/Открыть».

Параметры графического редактора устанавливаются на вкладке «Экран» (рис. 2.1.2) окна «Параметры», которое вызывается командой «Инструменты/Параметры».

В окне «Параметры» задаются параметры программы. На вкладке «Экран» можно задать следующие параметры:

1. ширина листа/(рабочей области для рисования, в пикселях);
2. высота листа (в пикселях);
3. цветовую схему.

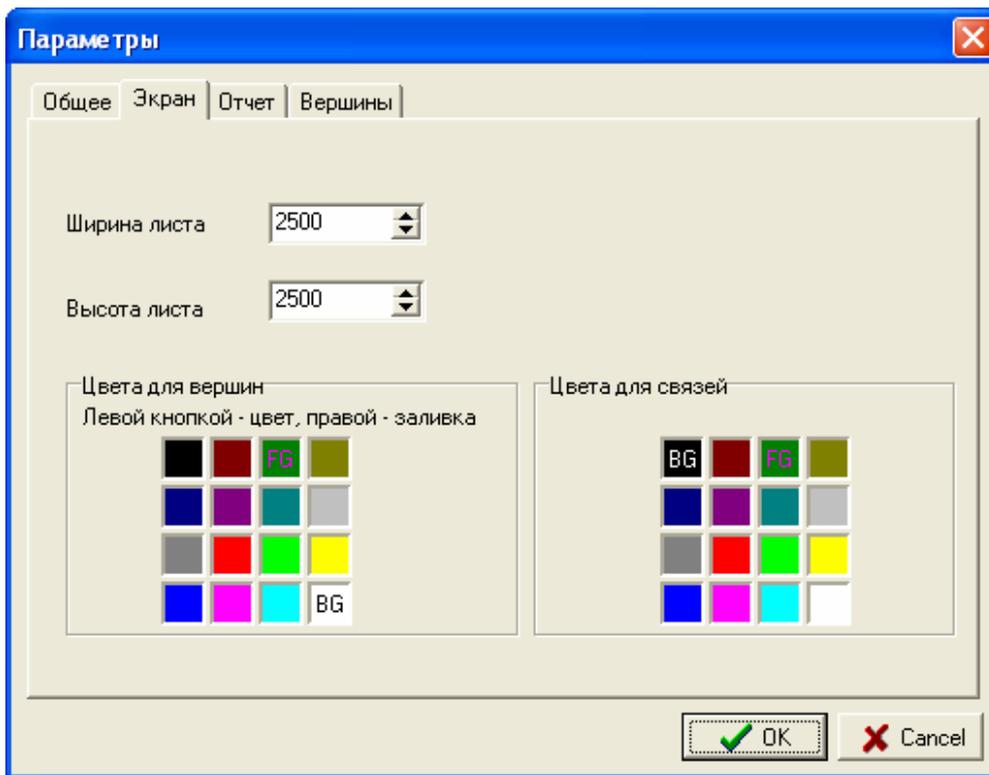


Рис. 2.1.2

Для создания новой вершины необходимо выполнить следующую команду «Добавить вершину» в меню «Объект» или нажать соответствующую кнопку на панели инструментов. После этого нажатием левой кнопки мыши необходимо в области рисования выбрать место расположения для новой вершины.

В появившемся окне (рис. 2.1.3) требуется задать следующие параметры вершины:

1. время заезда;
2. вес;
3. время разгрузки;
4. имя вершины;

Остальные параметры (номер вершины, радиус и контроль фиктивных узлов) являются не обязательными.

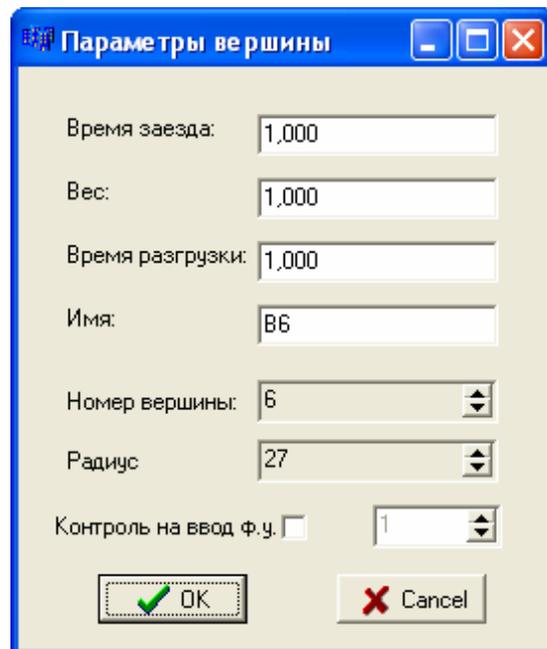


Рис. 2.1.



Рис. 2.1.4

Кнопка «Отмена» отменяет процедуру добавления вершины и переводит программу в предыдущее состояние. Для создания вершины с указанными параметрами следует нажать кнопку «ОК». Внешний вид созданной вершины показан на рис. 2.1.4. Двойной щелчок левой кнопкой мыши по созданной вершине открывает окно (рис. 2.1.3) для редактирования ее параметров.

Связь между вершинами устанавливается командой «Добавить связь» из меню «Объект» или соответствующей кнопкой на панели инструментов. После этого необходимо одним щелчком левой кнопкой мыши выбрать первую (исходящую) вершину и еще одним щелчком – вторую (входящую) вершину. После этой процедуры появляется окно для установки параметров связи (рис.2.1.5).

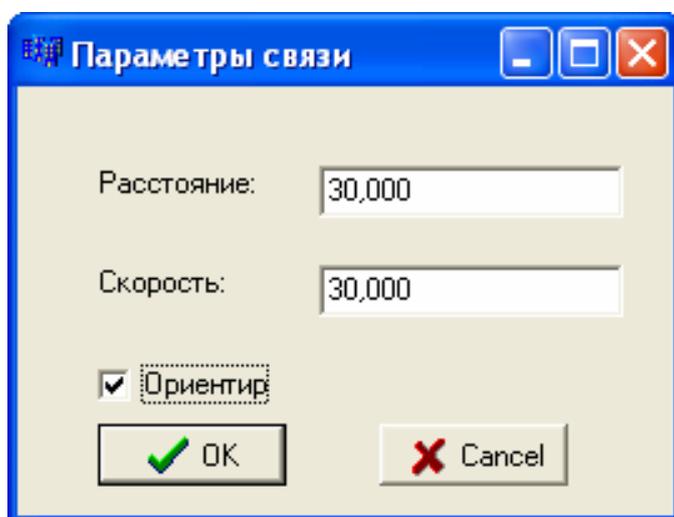


Рис. 2.1.5

В появившемся окне необходимо установить следующие параметры связи:

1. расстояние от одной вершины до другой;
2. скорость перемещения автомобиля;
3. признак ориентированности.

В случае ориентированной связи появляется соединение между выбранными вершинами, имеющее стрелку, направленную ко второй (входящей) вершине (рис. 2.1.6). В противоположном случае соединение не имеет стрелки.

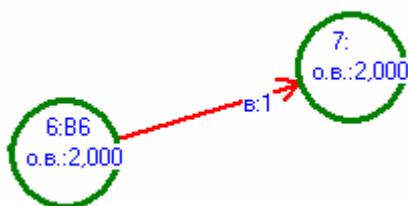


Рис. 2.1.6

Для удаления объекта (вершины или связи) необходимо выбрать следующие действия:

1. Нажатием левой кнопки мыши выбрать удаляемый объект;
2. Выполнить команду «Удалить» из меню «Объект» или выполнить аналогичное действие с использованием панели инструментов.

Для удаления всех вершин используется команда «Удалить вершины» из меню «Правка», а для удаления всех связей – «Удалить связи».

Изменить масштаб фигур можно командой «Увеличить» (для увеличения объектов) или «Уменьшить» (для уменьшения объектов) из меню «Вид» или соответствующими кнопками на панели инструментов.

Сохранить созданный граф можно с помощью команды «Файл/Сохранить», а для сохранения графического изображения графа – «Файл/Сохранить картинку».

2.1.2. Рабочий режим программы

Перед началом работы, связанной с поиском кратчайшего пути, необходимо задать параметры программы. Для этого необходимо выполнить команду «Инструменты/Параметры».

В появившемся окне (рис. 2.1.7) необходимо:

1. При необходимости, задать начальный ход (отключить флажок «Начать», указать первую и вторую вершину начального хода, выбрать режим управления фиктивными узлами);

2. Выбрать метод поиска оптимального маршрута (ветвей и границ или «метлы»);

3. При необходимости указать номер вершины, с которой следует рассчитывать затраченную работу;

4. Для увеличения скорости вычисления указать начальный порог. Если порог не известен, то указать максимальное число.

5. Указать количество маршрутов.

6. При необходимости включить контроль фиктивных узлов, который позволит уменьшить время поиска маршрута.

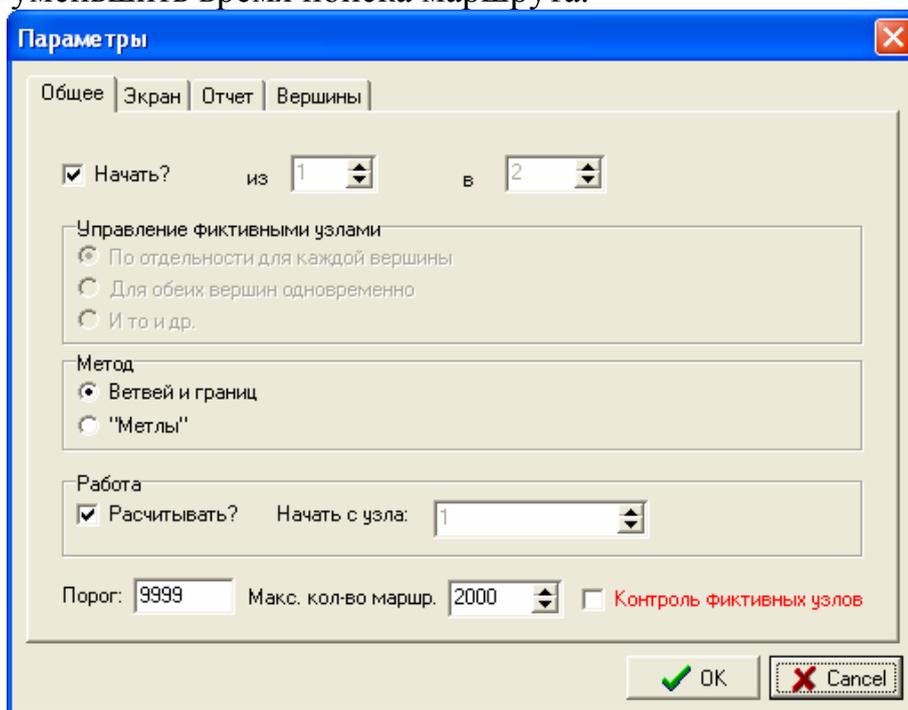


Рис. 2.1.7

Для поиска кратчайшего пути, в соответствии с установленными параметрами необходимо выполнить команду «Маршрут/Найти». После этого в появившемся окне (рис. 2.1.8) необходимо задать режим формирования матрицы поиска (по времени или по длине).

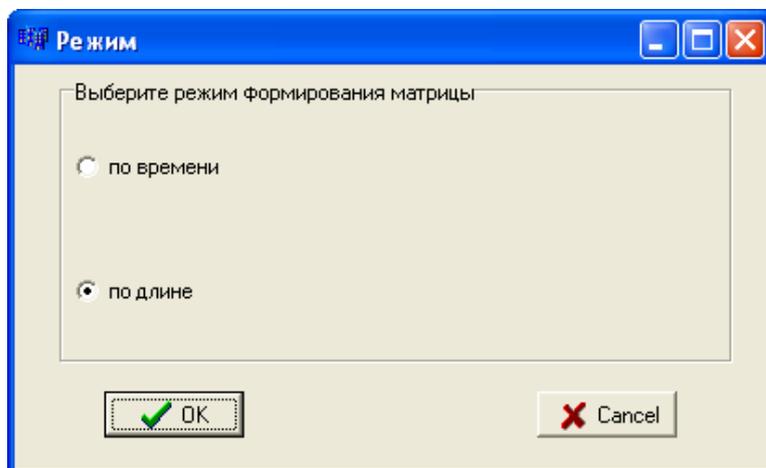


Рис. 2.1.8

Установив режим, программа рассчитывает кратчайший маршрут, отображает затраченное время и показывает отчеты поиска (рис. 2.1.9):

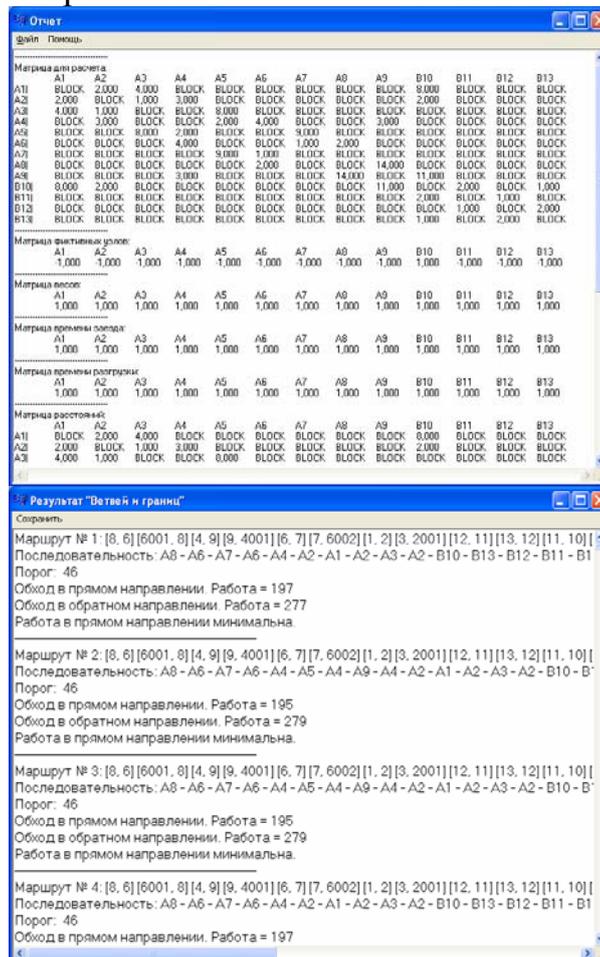


Рис. 2.1.9

Ознакомившись с отчетностью, можно посмотреть найденные маршруты непосредственно на графе. Для этого следует пользоваться командами «След. маршрут» (для отображения следующего маршрута) или «Пред. маршрут» (для отображения предыдущего маршрута) из меню «Маршрут» или нажать соответствующие кнопки на панели инструментов.

Исходная вершина и связи кратчайшего пути окрашиваются в черный цвет (рис. 2.1.10). У связей кратчайшего маршрута появляется информация о количестве посещений и порядковом номере заезда в виде следующей надписи:

$$(X1) X2,$$

где $X1$ – количество посещений, $X2$ – порядковый номер заезда.

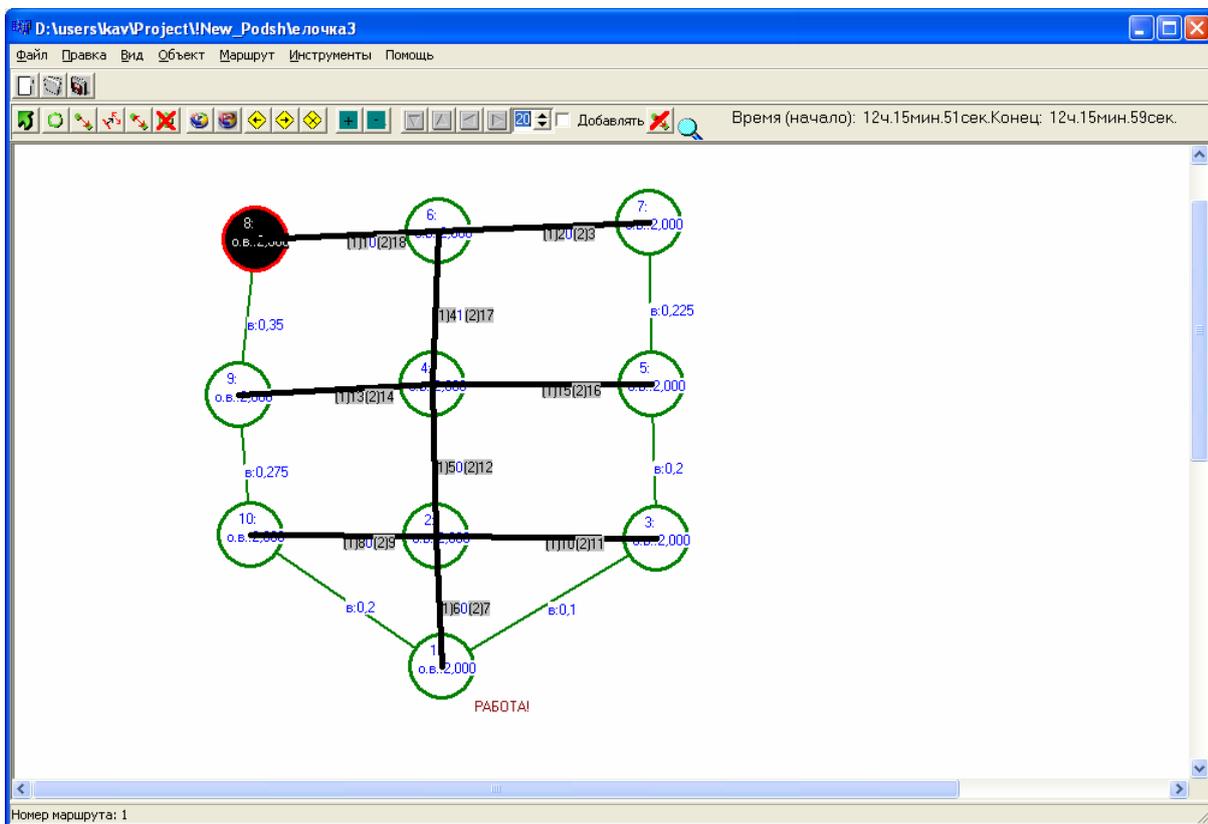


Рис. 2.1.10

Выйти из режима просмотра кратчайших путей можно с помощью кнопки «Обновить».

Если необходимо рассчитывать кратчайший маршрут в текстовом режиме, следует воспользоваться командой «Выгрузить в матрицу» из меню «Инструменты». В появившемся окне (рис. 2.1.11) при необходимости отредактировать полученную матрицу и произвести расчет кратчайшего пути, либо методом ветвей и границ (кнопка «Выполнить «Виг»»), либо методом метлы (кнопка «Выполнить «Метла»»).

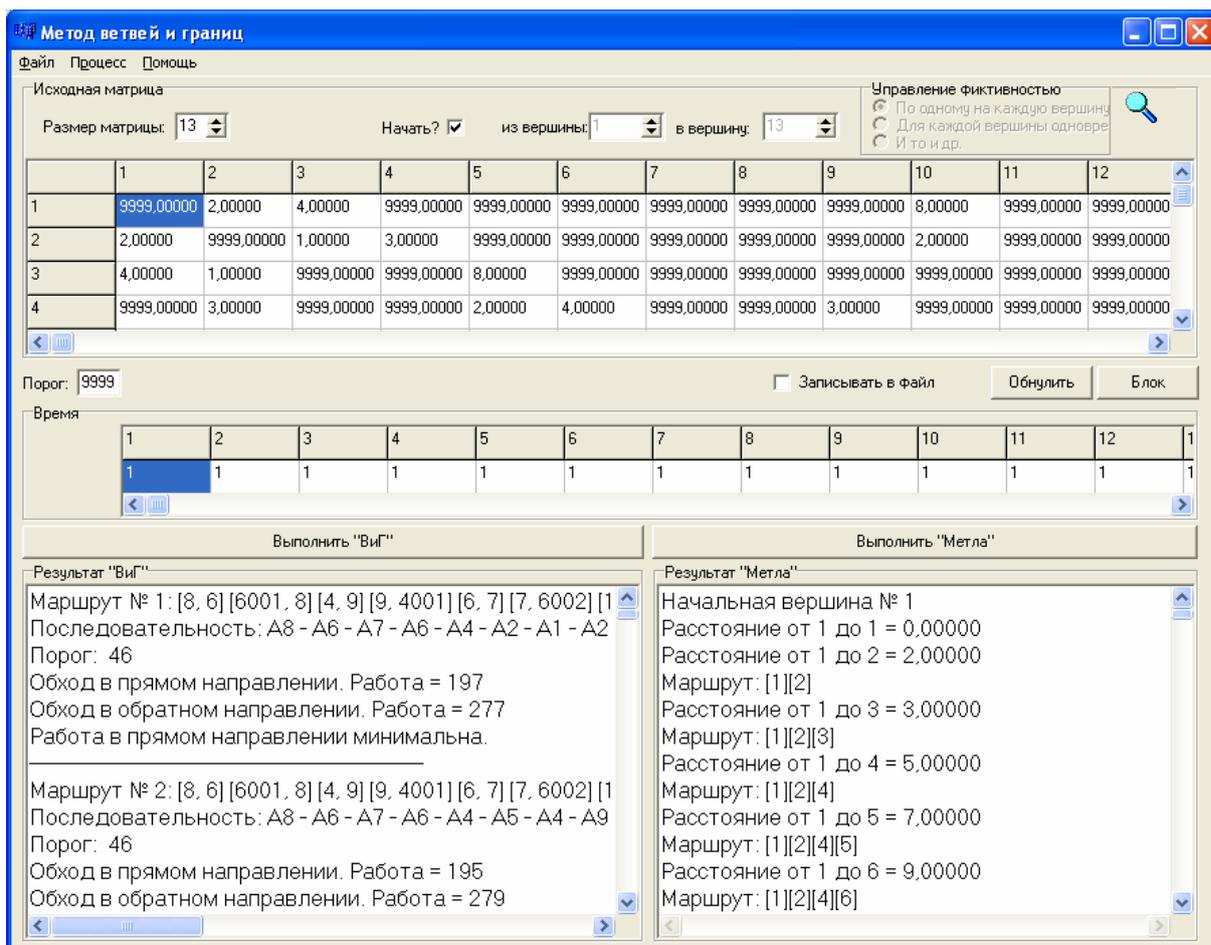


Рис. 2.1.11

Выход из программы осуществляется командой «Выход» из меню «Файл».

2.2. Техническое описание программы

Программа функционирует под управлением операционной системы Windows 95, 98, XP. Все модули программы реализованы на языке C++ в среде программирования Borland C++ Builder 6.0.

Структура меню приведена в табл. 2.2.1.

Таблица 2.2.1

Название пункта	Описание
1	2
Меню «Файл»	
Новый	Создание нового графа
Открыть	Открыть граф
Сохранить	Сохранить структуру графа
Сохранить картинку	Сохранить изображение графа
Меню «Правка»	
Удалить все вершина	Удалить все вершины графа
Удалить все связи	Удалить все связи графа

Окончание табл. 2.2.1

1	2
Меню «Вид»	
Увеличить	Увеличить масштаб
Уменьшить	Уменьшить масштаб
Меню «Объект»	
Добавить вершину	Добавить вершину к графу
Добавить связь	Добавить связь между вершинами
Удалить	Удалить выбранный объект
Меню «Маршрут»	
Найти	Поиск кратчайших маршрутов
Пред. маршрут	Показать предыдущий найденный кратчайший маршрут
След. маршрут	Показать следующий найденный кратчайший маршрут
Меню «Инструменты»	
Выгрузить в матрицу	Перейти из графического в текстового режим работы
Показать окно результата «ВиГ»	Показать отчет результатов поиска кратчайшего пути методом «Ветвей и границ»
Показать окно результата «Метла»	Показать отчет результатов поиска кратчайшего пути методом «Метла»
Сформировать отчет	Сформировать отчет о исходных данных в виде матриц
Показать отчет	Показать отчет исходных данных
Параметры	Изменить параметры программы
Меню «Помощь»	
О программе	Информация о разработанной программе

Компоненты основных форм программы описаны в табл. 2.2.2.

Таблица 2.2.2

Название компоненты	Описание
1	2
Форма «Form1» (Главная форма)	
MainMenu1;	Основное меню
ToolBar1;	Панель инструментов
StatusBar1;	Строка состояния
ImageList1;	Список картинок для панели инструментов
OpenDialog1;	Диалог для открытия файлов

Окончание табл. 2.2.2

1	2
SaveDialog1;	Диалог для сохранения файлов
PanelHolst;	Панель для рабочей области
ImageHolst;	Рабочая область
ScrollBar1;	Панель с полосой прокрутки, на которой размещается рабочая область
SavePictureDialog1;	Диалог для сохранения изображения
Animate1;	Анимация
Labels	Метки
Buttons	Кнопки
Форма «FormOptions» (Параметры программы)	
BitBtn1;	Кнопка «ОК»
BitBtn2;	Кнопка «Отмена»
TabbedNotebook1;	Вкладки
CheckBoxMode;	Флаг начального хода
SpinEditBeg;	Номер первой вершины начального хода
SpinEditEnd;	Номер второй вершины начального хода
Labels;	Метки
CSpinEditWigth;	Ширина холста
CSpinEditHeight;	Высота холста
RadioMetod;	Переключатель метода расчета
EditPorog;	Поле ввода порога
RadioGroupModeFU;	Метод расчета фиктивных узлов для начального хода
CheckBoxRabota;	Флаг расчета работы
CSpinEditRabotaBeg;	Номер вершины, с которой следует рассчитывать работу
GroupBoxs;	Панели для группирования компонентов
CColorGridTop;	Таблица цветовой схемы для вершин
CColorGridConnect;	Таблица цветовой схемы для связей
Форма «FormOptionsTop» (Параметры вершин)	
Labels;	Метки
Edits	Поля для ввода
Buttons	Кнопки
Форма «FormOptionsCon» (Параметры связи)	
Labels;	Метки
Edits	Поля для ввода
Buttons	Кнопки

Остальные окна также построены из стандартных компонентов.

2.3. Описание классов и интерфейсных функций

Назначение основных классов приведено в табл. 2.3.1.

Т а б л и ц а 2.3.1

Класс	Назначение
Модуль «UnitGraph»	
TForm1	Главная форма
Модуль «UnitOptions»	
TFormOptions	Форма для параметров программы
Модуль «UnitOptTop»	
TFormOptionsTop	Форма для параметров вершины
Модуль «UnitConnect»	
TFormOptionsCon	Форма для параметров связи
Модуль «UnitAbout»	
TFormAbout	Форма информации о программе
Модуль «UnitTree»	
CTree	Класс, реализующий метод ветвей и границ
Модуль «UnitMetla»	
CMetla	Класс, реализующий метод метлы
Модуль «UnitFigures»	
CFigures	Абстрактный класс для фигур
CTop	Класс вершины (наследник класса CFigures)
CConnect	Класс cdzptq (наследник класса CFigures)
Модуль «UnitMatrix»	
TFormMatrix	Форма для текстового режима работы
Модуль «UnitReport»	
TFormReport	Форма для отчета найденных маршрутов методом «Ветвей и границ»
Модуль «UnitReportMetla»	
TFormReportMetla	Форма для отчета найденных маршрутов методом «Метлы»
Модуль «UnitResult»	
TFormResult	Форма отчета результатов

Рассмотрим подробнее класс CTree, реализующий метод ветвей и границ. Описание основных атрибутов и методов этого класса приведено соответственно в табл. 2.3.2. и 2.3.3.

Т а б л и ц а 2.3.2

Атрибут	Описание
float** matrix;	исследуемая матрица
int size_matrix;	размер матрицы
s_ver* ver_vertical;	вертикальные вершины
s_ver* ver_horizont;	горизонтальные вершины
int level;	уровень в дереве "0" - корень
char f_fict;	флаг фиктивности узла в дереве
char f_end;	флаг конца обработки
s_mar* marshrut;	пройденные вершины в маршруте
int cnt_marshall;	размер маршрута
s_mar* cut;	"склеенный" маршрут
int cnt_cut;	размер "склеенного" маршрута
s_ver ver_beg;	начальная вершина
s_ver ver_end;	конечная вершина
int* ind_i;	массив найденных вычеркиваемых строк
int* find_j;	массив найденных вычеркиваемых столбцов
float porog;	Граница вычислений
CTree** L;	массив указателей на нормальную ветвь
CTree** F1;	массив указателей на фиктивную ветвь №1
CTree** F2;	массив указателей на фиктивную ветвь №2
CTree* Parent;	указатель на родителя "0, NULL" – корень
int cnt_max;	количество максимальных значений

Т а б л и ц а 2.3.3

Метод	Описание
1	2
CTree();	конструктор
~CTree();	деструктор
int Init(float* mas[], int size, s_ver* ver_ver, s_ver* ver_hor, int lev, char f, s_mar* mar, int cnt_mar, s_ver mar_beg, s_ver mar_end, CTree* pred);	Инициализация класса
int InitPorog(float p);	Инициализация порога
int InitCut(s_mar* mar_cut, int cnt_mar_cut);	Инициализация «склеенного» маршрута

Окончание табл. 2.3.3

1	2
int Process_matrix();	Обработка матрицы
int CalcPorog(s_ver ver_beg, s_ver ver_end, float* n_porog);	Вычислить порог
int Resize_matrix(float** mas, int ii, int jj);	Удалить из матрицы строку и столбец
int Resize_matrix_str(float** mas, int ii, int jj, int bjj, int size);	Добавить к матрице строку
int Resize_matrix_stolb(float** mas, int ii, int jj, int bii, int size);	Добавить к матрице столбец
int Find(s_ver* mas_ver, int ver);	Найти вершину
int Check_block();	Проверить матрицу на блокировку
int Resize_mas_ver(s_ver* mas_ver, int ii, s_ver* new_mas_ver);	Удалить элемент из массива вершин
int AddFictVer(s_ver* mas_ver_beg, s_ver* mas_ver_end, int i, int size);	Добавить фиктивную вершину к массиву вершин
int Delete();	Удалит дерево поиска
int DeleteMatrix();	Удалить матрицу
int Sim();	Запустить алгоритм поиска (блок-схема алгоритма приведена в приложении Б)
float GetPorog();	Вычислить порог

Реализация классов приведено в прил. А.

Приложение А

А.1. Файл «UnitTree.h»

```

#ifndef UnitTreeH
#define UnitTreeH
#include <math.h>
#include <alloc.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <io.h>
#include <errno.h>
#define BLC 9999
#define FICT 1
#define NOT_FICT 0
// #define CNT_MAS_MAX 100
#define CNT_MAS_CUT 300
// #define CNT_TABLE 1000
#define ADD_FICT_NOT 0
#define ADD_FICT_STR 1
#define ADD_FICT_STOLB 2
#include <Classes.hpp>

```

```

#include <ComCtrls.hpp>
#include <DSTRING.H>
#define FILE_NAME_GLOB_TABLE      "gltab.bin"
#define FILE_NAME_GLOB_TABLE_LEC  "ghtablec.bin"
//структура вершины
typedef struct ver{ int num;      //номер вершины
    char f;      //признак фиктивности вершины int f_num;  //если вершина фиктивная, то указывается номер вершины int cnt_f;  //счетчик фиктивности int max_cnt_f; //максимальное значение счетчика
    bool super;  //"правительственная" связь      } s_ver;
//структура для маршрута
typedef struct mar{
    s_ver beg;  //начальная вершина
    s_ver end;  //конечная вершина} s_mar;
//структура для таблицы результатов
typedef struct table_res{
    s_mar mar[1000]; //маршрут int size;      //длина float por;} s_table_res;
int InitSignal();
int InitPorog(float por);
int InitStartMode(char mode);
int InitMode(char m);
int InitControlFict(bool m);
int StartWriteSim(FILE **handle, char* name);
int StartReadSim(FILE **handle, char* name);
int StopSim(FILE **handle);
int FileWriteMar(FILE *handle, s_mar* marshrut, int cnt_marshall, float porog);
int FileReadMar(FILE *handle, s_mar* marshrut, int cnt_marshall);
int FileReadSizePor(FILE *handle, int* cnt_marshall, float* porog);
//int SaveTable(s_table_res* tab, int* size);
int CalckRabotaPlus(s_ver* mar, int cnt, int i_start, float* rab);
int CalckRabotaMinus(s_ver* mar, int cnt, int i_start, float* rab);
int CalkLength(s_ver* mar, int cnt, int i_start, float* len);
//класс "дерево решений"
class CTree{
public:
    CTree();
    ~CTree();
    //Инициализация класса int Init(float* mas[], int size, s_ver* ver_ver, s_ver* ver_hor, int lev, char f, s_mar* mar, int cnt_mar, s_ver mar_beg, s_ver mar_end, CTree* pred); int InitPorog(float p); int InitCut(s_mar* mar_cut, int cnt_mar_cut);
    //Обработка матрицы int Process_matrix(); int CalcPorog(s_ver ver_beg, s_ver ver_end, float* n_porog); int Resize_matrix(float** mas, int ii, int jj); int Resize_matrix_str(float** mas, int ii, int jj, int bjj, int size); int Resize_matrix_stolb(float** mas, int ii, int jj, int bii, int size); int Find(s_ver* mas_ver, int ver); int Check_block(); int Resize_mas_ver(s_ver* mas_ver, int ii, s_ver* new_mas_ver); int AddFictVer(s_ver* mas_ver_beg, s_ver* mas_ver_end, int i, int size); int LoadMatrix(float** mas, float* ver, int size, s_ver* ver_ver, s_ver* ver_hor, float** mas_dist); int Delete(); int DeleteMatrix();
    //Симуляция int Sim(); float GetPorog();
    protected: float** matrix;      //исследуемая матрица int size_matrix;      //размер матрицы
    s_ver* ver_vertical;      //вертикальные вершины
    s_ver* ver_horizont;      //горизонтальные вершины int level;      //уровень в дереве
"0" - корень
    char f_fict;      //флаг фиктивности узла в дереве
    char f_end;      //флаг конца обработки
    s_mar* marshrut;      //пройденные вершины в маршруте int cnt_marshall;      //размер маршрута
    s_mar* cut;      //"склеенный" маршрут int cnt_cut;      //размер "склеенного" маршрута
    s_ver ver_beg;      //начальная вершина

```

```

        s_ver ver_end;          //конечная вершина int*   find_i;          //массив найденных вычер-
киваемых строк int*   find_j;          //массив найденных вычеркиваемых столбцов float   porog;
//Граница вычислений
    CTree** L;                //массив указателей на нормальную ветвь
    CTree** F1;               //массив указателей на фиктивную ветвь №1
    CTree** F2;               //массив указателей на фиктивную ветвь №2
    CTree* Parent;            //указатель на родителя "0, NULL" - корень int   cnt_max;};
#endif

```

А.2. Файл «UnitTree.cpp»

```

#pragma hdrstop
#include "UnitTree.h"
#include <string.h>
#pragma package(smart_init)
float** global_matrix;
int global_matrix_size = 0;
float** global_matrix_dist;
s_ver* global_vertical;
s_ver* global_horizont;
float* g_vershins_weigth;
int cnt_table;
int cnt_table_lec;
int CNT_TABLE = 1000;
float global_porog = 9999;
char f_start_mode;          //0 - с одним ф.у. в начале
char f_mode;                //режим 0 - по всем вершинам, 1 - с заданной начальной и конечной
bool f_control_fict;        // контроль фиктивных узлов
int handle;
FILE *hGlobTable;          // дискриптор файла таблицы решений
FILE *hGlobTableLec;        // дискриптор файла таблицы решений с лекалом
char signal_for_end = 0;
int StartWriteSim(FILE **handle, char* name){
    AnsiString file;
    file = ExtractFilePath(Application->ExeName) + name;
    *handle = fopen(file.c_str(), "wb");
    if (*handle == NULL){ perror("Error:"); return 1;} return 0;}
int StartReadSim(FILE **handle, char* name){
    AnsiString file;
    file = ExtractFilePath(Application->ExeName) + name;
    *handle = fopen(file.c_str(), "rb");
    if (*handle == NULL) {
        perror("Error:"); return 1; } return 0;}
int StopSim(FILE **handle){ fclose(*handle); return 0;}
int FileWriteMar(FILE *handle, s_mar* marshrut, int cnt_marshrut, float porog){ int i;
    // Записываем размер
    fwrite(&cnt_marshrut, sizeof(int), 1, handle);
    // Записываем порог
    fwrite(&porog, sizeof(float), 1, handle);
    fwrite(marshrut, sizeof(s_mar), cnt_marshrut, handle); return 0;}
int FileReadMar(FILE *handle, s_mar* marshrut, int cnt_marshrut){ int i;
    fread(marshrut, sizeof(s_mar), cnt_marshrut, handle); return 0;}
int FileReadSizePor(FILE* handle, int* cnt_marshrut, float* porog){
    // Считываем размер
    fread(cnt_marshrut, sizeof(int), 1, handle);
    // Считываем порог
    fread(porog, sizeof(float), 1, handle); return 0;}
int InitSignal(){ signal_for_end = 0; return 0;}
int InitPorog(float por){ global_porog = por; return 0;}
int InitMode(char m){ f_mode = m; return 0;}

```

```

int InitWriteFile(char m){ f_write_file = m; return 0;}
int InitStartMode(char mode){ f_start_mode = mode; return 0;}
int InitControlFict(bool m){ f_control_fict = m; return 0;}
int CalckRabotaPlus(s_ver* mar, int cnt, int i_start, float* rab){ int i, j, beg, tek; float l, r; float *weigth;
    l = 0;
    r = 0;
    weigth = new float[cnt];
    for(i = 0; i < cnt; i++) {weigth[i] = g_vershins_weigth[i]; }
    j = i_start; beg = mar[j].f_num - 1;
    if (j < (cnt-1)) {j++; } else {j = 0; }
    for(i = 1; i < cnt; i++) {
        tek = mar[j].f_num - 1;
        if (!((weigth[tek] == 0) && (i == 1))) {
            l += global_matrix_dist[beg][tek]; }
        r += (l * weigth[tek]);
        weigth[tek] = 0; //Сбор рпыза
        beg = tek;
        if (j < (cnt-1)) {j++; } else {j = 0; } } delete weigth; *rab = r; return 0;}
int CalckRabotaMinus(s_ver* mar, int cnt, int i_start, float* rab){ int i, j, beg, tek; float l, r; float *weigth;
    l = 0; r = 0;
    weigth = new float[cnt];
    for(i = 0; i < cnt; i++) {
        weigth[i] = g_vershins_weigth[i]; }
    j = i_start; beg = mar[j].f_num - 1;
    if (j > 0) {j--; } else {j = cnt - 1; }
    for(i = (cnt - 2); i >= 0; i--) {
        tek = mar[j].f_num - 1;
        if (!((weigth[tek] == 0) && (i == (cnt - 2)))) {
            l += global_matrix_dist[beg][tek]; }
        r += (l * weigth[tek]);
        weigth[tek] = 0; //Сбор рпыза
        beg = tek;
        if (j > 0) {
            j--; } else {j = cnt - 1; } } delete weigth; *rab = r; return 0;}
int CalkLength(s_ver* mar, int cnt, int i_start, float* len){ int i, j, beg, tek; float l;
    l = 0; j = i_start; beg = mar[j].f_num - 1;
    if (j < (cnt-1)) {j++; } else {j = 0; }
    for(i = 1; i <= cnt; i++) {
        tek = mar[j].f_num - 1; l += global_matrix_dist[beg][tek]; beg = tek;
        if (j < (cnt-1)) {j++; } else {j = 0; } } *len = l; return 0;}
int GlobalFind(s_ver* mas_ver, int ver){ int i;
    for(i = 0; i < global_matrix_size; i++) { if (mas_ver[i].num == ver) { return i; } } return -1;}
int FullMemo(){ return 0;}
CTree::CTree(){ int i_max;
    matrix = 0; size_matrix = 0;
    ver_horizont = 0; ver_vertical = 0; level = 0;
    f_fict = 0; f_end = 0; cnt_marshrut = 0; ver_beg.num = 0;
    ver_beg.f = NOT_FICT; ver_end.num = 0;
    ver_end.f = NOT_FICT; marshrut = 0;
    cut = 0; cnt_cut = 0; Parent = 0; porog = 0;
    cnt_max = 0;
}
CTree::~CTree(){ int i;
    if (size_matrix != 0) {
        for (i = 0; i < size_matrix; i++) { delete matrix[i]; } delete matrix; delete ver_horizont; delete
ver_vertical; }
    if (marshrut != 0) { delete marshrut; }
    if (cnt_cut != 0) { delete cut; } }

```

```

int CTree::Init(float* mas[], int size, s_ver* ver_ver, s_ver* ver_hor, int lev, char f, s_mar* mar, int
cnt_mar, s_ver mar_beg, s_ver mar_end, CTree* pred){ int i, j;
    if (matrix == 0) {
        matrix = new float*[size]; size_matrix = size; ver_horizont = new s_ver[size];
        ver_vertical = new s_ver[size]; }
    if (size != size_matrix) {
        for (i = 0; i < size_matrix; i++) { delete matrix[i]; } delete matrix; delete ver_horizont; delete
ver_vertical;
        matrix = new float*[size]; size_matrix = size; ver_horizont = new s_ver[size];
        ver_vertical = new s_ver[size]; }
    for(i = 0; i < size; i++) {
        matrix[i] = new float[size];
        for(j = 0; j < size; j++) {
            matrix[i][j] = mas[i][j]; }
        ver_vertical[i] = ver_ver[i];
        ver_horizont[i] = ver_hor[i]; }
    level = lev;
    f_fict = f;
    marshrut = new s_mar[cnt_mar + 10];
    if (level != 0) {
        for(i = 0; i < cnt_mar; i++) { marshrut[i] = mar[i]; }
        cnt_marshall = cnt_mar;
        marshrut[cnt_marshall].beg = mar_beg; marshrut[cnt_marshall].end = mar_end; cnt_marshall++;
    }
    ver_beg = mar_beg; ver_end = mar_end; Parent = pred; return 0;
}
int CTree::InitPorog(float p){ porog = p; return 0;}
int CTree::InitCut(s_mar* mar_cut, int cnt_mar_cut){ int i, j, t_i;
    char flag_update; int block2_i, block2_j;
    AnsiString str1;
    cut = new s_mar[cnt_mar_cut + 2];
    for(i = 0; i < cnt_mar_cut; i++) {
        cut[i] = mar_cut[i]; }
    cnt_cut = cnt_mar_cut;
    cut[cnt_cut].beg = ver_beg;
    cut[cnt_cut].end = ver_end;
    cnt_cut++;
    flag_update = 1;
    while (flag_update == 1) {
        flag_update = 0;
        for (i = 0; i < cnt_cut; i++) {
            for (j = 0; j < cnt_cut; j++) {
                if (i != j) {
                    if (cut[i].beg.num == cut[j].end.num) {
                        //Блокируем цикличную ветвь
                        block2_i = Find(ver_vertical, cut[i].end.num);
                        block2_j = Find(ver_horizont, cut[j].beg.num);
                        if ((block2_i != -1) && (block2_j != -1)) {
                            matrix[block2_i][block2_j] = BLC;
                        }
                        //Склеиваем маршрут
                        cut[i].beg = cut[j].beg;
                        for (t_i = j; t_i < cnt_cut; t_i++) {
                            cut[t_i] = cut[t_i + 1];
                        }
                    } cnt_cut--; flag_update = 1; break; } } }
                if (flag_update == 1) {break; } } } return 0;
    }
}
int CTree::Process_matrix(){ float* min_str; float* min_stolb; float** zero; float min_a, min_b, max; int i,
j, a, b, t_i;
    min_str = new float[size_matrix]; min_stolb = new float[size_matrix];

```

```

zero = new float*[size_matrix];
for(i = 0; i < size_matrix; i++) { zero[i] = new float[size_matrix]; }
//Найти минимальный элемент в каждой строке
for(i = 0; i < size_matrix; i++) { min_str[i] = matrix[i][0];
    for(j = 0; j < size_matrix; j++) {
        if(matrix[i][j] < min_str[i]) {min_str[i] = matrix[i][j]; } } }
// ... и вычтеть его из элементов строк
for(i = 0; i < size_matrix; i++) {
    for(j = 0; j < size_matrix; j++) {
        if (matrix[i][j] != BLC) {matrix[i][j] = matrix[i][j] - min_str[i]; } } }
//Найти минимальный элемент в каждом столбце
for(j = 0; j < size_matrix; j++) {
    min_stolb[j] = matrix[0][j];
    for(i = 0; i < size_matrix; i++) {
        if(matrix[i][j] < min_stolb[j]) {min_stolb[j] = matrix[i][j]; } } }
//... и вычтеть его из элементов столбцов
for(j = 0; j < size_matrix; j++) {
    for(i = 0; i < size_matrix; i++) {
        if (matrix[i][j] != BLC) {matrix[i][j] = matrix[i][j] - min_stolb[j]; } } }
//Оценка нулей
for (i = 0; i < size_matrix; i++) {
    for(j = 0; j < size_matrix; j++) {
        zero[i][j] = -1;
        if (matrix[i][j] == 0) { //встретили "ноль"
            //Находим минимальный элемент в строке
            min_a = BLC;
            for(a = 0; a < size_matrix; a++) {
                if(a != j) {
                    if (matrix[i][a] < min_a) {
                        min_a = matrix[i][a];
                    } } }
            //Находим минимальный элемент в столбце
            min_b = BLC;
            for(b = 0; b < size_matrix; b++) {
                if(b != i) {
                    if (matrix[b][j] < min_b) {
                        min_b = matrix[b][j]; } } }
            zero[i][j] = min_a + min_b; } } } //Оцениваем...
//Находим элемент в матрице с максимальной оценкой
max = zero[0][0];
cnt_max = 0;
for (i = 0; i < size_matrix; i++) {
    for(j = 0; j < size_matrix; j++) {
        if(zero[i][j] == max) {
            cnt_max++;
        } else if(zero[i][j] > max) {
            max = zero[i][j]; cnt_max = 0; cnt_max++; } } }
//Нашли количество максимальных значений
//Выделяем память
find_i = new int[cnt_max]; find_j = new int[cnt_max];
L = new CTree*[cnt_max]; F1 = new CTree*[cnt_max]; F2 = new CTree*[cnt_max];
t_i = 0;
for (i = 0; i < size_matrix; i++) {
    for(j = 0; j < size_matrix; j++) {
        if(zero[i][j] == max) {
            find_i[t_i] = i; find_j[t_i] = j; t_i++; } } } delete min_str; delete min_stolb;
for(i = 0; i < size_matrix; i++) { delete zero[i]; } delete zero; return 0;}
int CTree::CalcPorog(s_ver ver_beg, s_ver ver_end, float* n_porog){ int i, j;
if (ver_beg.f == FICT) {
    i = GlobalFind(global_vertical, ver_beg.f_num); } else {

```

```

        i = GlobalFind(global_vertical, ver_beg.num); }
    if (ver_end.f == FICT) {
        j = GlobalFind(global_horizont, ver_end.f_num); } else {
        j = GlobalFind(global_horizont, ver_end.num); }
    *n_porog = porog + global_matrix[i][j]; return 0;}
int CTree::Find(s_ver* mas_ver, int ver){ int i;
    for(i = 0; i < size_matrix; i++) {
        if (mas_ver[i].num == ver) {
            return i; } } return -1;}
int CTree::Resize_matrix(float** mas, int ii, int jj){ int i, j, a, b;
    a = 0; b = 0;
    for(i = 0; i < size_matrix; i++) {
        for(j = 0, b = 0; j < size_matrix; j++) {
            if ((i != ii) && (j != jj)) {
                mas[a][b] = matrix[i][j]; b++;
            } }
        if ((i != ii) && (j != jj)) {a++; } } return 0;}
int CTree::Resize_matrix_str(float** mas, int ii, int jj, int bjj, int size){ int i, a, b;
    for (i = 0, a = 0, b = 0; i < size_matrix; i++) {
        if (i != ii) {
            mas[a++][size - 1] = matrix[i][bjj]; }
        if (i != jj) {
            mas[size - 1][b++] = matrix[ii][i]; } }
    mas[size - 1][size - 1] = BLC; return 0;}
int CTree::Resize_matrix_stolb(float** mas, int ii, int jj, int bii, int size){ int i, a, b;
    for (i = 0, a = 0, b = 0; i < size_matrix; i++) {
        if (i != ii) {
            mas[a++][size - 1] = matrix[i][jj]; }
        if (i != jj) {
            mas[size - 1][b++] = matrix[bii][i]; } }
    mas[size - 1][size - 1] = BLC; return 0;}
int CTree::Resize_mas_ver(s_ver* mas_ver, int ii, s_ver* new_mas_ver){ int i, a;
    for (i = 0, a = 0; i < size_matrix; i++) {
        if(i != ii) {
            new_mas_ver[a++] = mas_ver[i]; } } return 0;}
int CTree::AddFictVer(s_ver* mas_ver_beg, s_ver* mas_ver_end, int i, int size){
    mas_ver_beg[size - 1].f = FICT;
    mas_ver_beg[size - 1].max_cnt_f = mas_ver_end[i].max_cnt_f;
    mas_ver_beg[size - 1].super = mas_ver_end[i].super;
    if (mas_ver_end[i].f == NOT_FICT) {
        mas_ver_beg[size - 1].num = (mas_ver_end[i].num * 1000) + 1;
        mas_ver_beg[size - 1].f_num = mas_ver_end[i].num;
        mas_ver_beg[size - 1].cnt_f = 1; } else {
        mas_ver_beg[size - 1].num = mas_ver_end[i].num + 1;
        mas_ver_beg[size - 1].f_num = mas_ver_end[i].f_num;
        mas_ver_beg[size - 1].cnt_f = mas_ver_end[i].cnt_f + 1; }
    if (mas_ver_beg[size - 1].cnt_f > (mas_ver_beg[size - 1].max_cnt_f - 1)) { return 1; } else { return 0;
}}
int CTree::Check_block(){ int i, j; int flag = 0;
    //Проверка строк
    for (i = 0; i < size_matrix; i++) {
        flag = 0;
        for (j = 0; j < size_matrix; j++) {
            if (matrix[i][j] != BLC) {flag = 1; break; } }
        if (flag == 0) { return 1; } }
    //Проверка столбцов
    for (j = 0; j < size_matrix; j++) {
        flag = 0;
        for (i = 0; i < size_matrix; i++) {

```

```

        if (matrix[i][j] != BLC) {flag = 1; break; } }
        if (flag == 0) {return 1; } } return 0;}
int CTree::Sim(){ float** temp_mas; float** temp2_mas;
    s_ver* temp_ver_horizont;
    s_ver* temp_ver_vertical;
    int i, i_max; int block_i, block_j, block2_i, block2_j, bii, bjj, ii, jj; int j, k, m, cnt; float beg_num,
tek_num;
    char* mas_f_ver; float new_porog; int res_L = 0, res_F1 = 0, res_F2 = 0;
    char flag_add_Ver = 0, flag_add_Hor = 0, f_rodn_stolb = 0, f_rodn_str = 0, f_route_str = 0,
f_route_stolb = 0;
    char f_add_fict_str = 0, f_add_fict_stolb = 0;
    if (sygnal_for_end == 1) { return 100; }
    temp_mas = new float*[size_matrix + 1];
    for(i = 0; i < size_matrix + 1; i++) {
        temp_mas[i] = new float[size_matrix + 1]; }
    temp_ver_horizont = new s_ver[size_matrix + 1];
    temp_ver_vertical = new s_ver[size_matrix + 1];
    for(i = 0; i < size_matrix; i++) {
        temp_ver_horizont[i] = ver_horizont[i];
        temp_ver_vertical[i] = ver_vertical[i]; }
    //Граничные условия
    //1. если матрица 2*2
    if (size_matrix == 2) {
        //Записать оставшуюся матрицу в массив маршрутов
        if ((matrix[0][0] == BLC) && (matrix[0][1] == BLC) ||
            (matrix[1][0] == BLC) && (matrix[1][1] == BLC) ||
            (matrix[0][1] == BLC) && (matrix[1][1] == BLC) ||
            (matrix[0][0] == BLC) && (matrix[1][0] == BLC)) {
            f_end = 0;
            for(i = 0; i < (size_matrix + 1); i++) {delete temp_mas[i]; }
            delete temp_mas; delete temp_ver_horizont; delete temp_ver_vertical;
            return 1; }
        if ((matrix[0][0] == BLC) || (matrix[1][1] == BLC)) {
            marshrut[cnt_marshrut].beg = ver_vertical[1];
            marshrut[cnt_marshrut].end = ver_horizont[0]; cnt_marshrut++;
            CalcPorog(ver_vertical[1], ver_horizont[0], &porog);
            marshrut[cnt_marshrut].beg = ver_vertical[0];
            marshrut[cnt_marshrut].end = ver_horizont[1]; cnt_marshrut++;
            CalcPorog(ver_vertical[0], ver_horizont[1], &porog);
        } else if ((matrix[1][0] == BLC) || (matrix[0][1] == BLC)) {
            marshrut[cnt_marshrut].beg = ver_vertical[0];
            marshrut[cnt_marshrut].end = ver_horizont[0]; cnt_marshrut++;
            CalcPorog(ver_vertical[0], ver_horizont[0], &porog);
            marshrut[cnt_marshrut].beg = ver_vertical[1];
            marshrut[cnt_marshrut].end = ver_horizont[1]; cnt_marshrut++;
            CalcPorog(ver_vertical[1], ver_horizont[1], &porog); } else {
            marshrut[cnt_marshrut].beg = ver_vertical[1];
            marshrut[cnt_marshrut].end = ver_horizont[0]; cnt_marshrut++;
            CalcPorog(ver_vertical[1], ver_horizont[0], &porog);
            marshrut[cnt_marshrut].beg = ver_vertical[0];
            marshrut[cnt_marshrut].end = ver_horizont[1]; cnt_marshrut++;
            CalcPorog(ver_vertical[0], ver_horizont[1], &porog); }
        if (porog <= global_porog) {
            mas_f_ver = new char[global_matrix_size];
            for (i = 0; i < global_matrix_size; i++) {mas_f_ver[i] = 0; }
            beg_num = marshrut[0].beg.num;
            if (marshrut[0].beg.f == FICT) {mas_f_ver[marshrut[0].beg.f_num - 1] = 1;
        } else { mas_f_ver[marshrut[0].beg.num - 1] = 1; }
            tek_num = marshrut[0].end.num;

```

```

        if (marshrut[0].end.f == FICT) {
            mas_f_ver[mashrut[0].end.f_num - 1] = 1;
        } else { mas_f_ver[mashrut[0].end.num - 1] = 1; }
        j = 0; k = 1; cnt = 0;
        while ((tek_num != beg_num) && (k == 1)) {
            tek_num = marshrut[j].end.num;
            if (marshrut[j].end.f == FICT) {
                mas_f_ver[mashrut[j].end.f_num - 1] = 1;
            } else { mas_f_ver[mashrut[j].end.num - 1] = 1; }
        } j = 0; k = 0; while (j < cnt_marshrut) {
            if (marshrut[j].beg.num == tek_num) {
                k = 1; cnt++; break; } j++; } }
        //Проверка: По всем ли вершинам мы прошли?
        m = 1;
        for (i = 0; i < global_matrix_size; i++) {if (mas_f_ver[i] == 0) {m = 0; } }
        f_end = 4;
        if ((k == 1) && (m == 1)) {
            //Сохранить маршрут в глобальной таблице
            if (porog < global_porog) {cnt_table = 0;
                // Очищаем файл
                StopSim(&hGlobTable);
                StartWriteSim(&hGlobTable, FILE_NAME_GLOB_TABLE);
                FileWriteMar(hGlobTable, marshrut, cnt_marshrut, porog);
            }
            cnt_table++;
            global_porog = porog; f_end = 1;
            if(cnt_table >= (CNT_TABLE)) {signal_for_end = 1;
        } } else { } delete mas_f_ver; } else { }
        for(i = 0; i < (size_matrix + 1); i++) {
            delete temp_mas[i]; } delete temp_mas; delete temp_ver_horizont; delete temp_ver_vertical;
        if (f_end == 1) {
            return 0; } else {return 1; } }
        if (Check_block() == 1) {
            for(i = 0; i < (size_matrix + 1); i++) {
                delete temp_mas[i]; } delete temp_mas; delete temp_ver_horizont; delete temp_ver_vertical;
            f_end = 0; return 1; }
        if (level == 0) {
            cnt_table = 0;
            if (f_mode == 1) {
                cnt_max = 2; i_max = -1; find_i = new int[cnt_max]; find_j = new int[cnt_max];
                L = new CTree*[cnt_max]; F1 = new CTree*[cnt_max]; F2 = new CTree*[cnt_max];
                if ((f_start_mode == 0) || (f_start_mode == 2)) {
                    i_max++;
                    find_i[i_max] = ver_beg.num - 1; //вначале совпадают номера индексов с номером вершины
                    find_j[i_max] = ver_end.num - 1;
                    CalcPorog(ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], &new_porog);
                    block_i = Find(ver_vertical, ver_horizont[find_j[i_max]].num);
                    block_j = Find(ver_horizont, ver_vertical[find_i[i_max]].num);
                    if (block_i == -1) {
                        f_end = 0;
                        return 1; } if (block_j == -1) {f_end = 0; return 1; }
                    Resize_matrix(temp_mas, find_i[i_max], find_j[i_max]);
                    Resize_mas_ver(ver_vertical, find_i[i_max], temp_ver_vertical);
                    Resize_mas_ver(ver_horizont, find_j[i_max], temp_ver_horizont);
                    block_i = Find(temp_ver_vertical, ver_horizont[find_j[i_max]].num);
                    block_j = Find(temp_ver_horizont, ver_vertical[find_i[i_max]].num);
                    if ((block_i != -1) && (block_j != -1)) {
                        temp_mas[block_i][block_j] = BLC; }
                    L[i_max] = new CTree;

```

```

L[i_max]->Init(temp_mas, size_matrix - 1, temp_ver_vertical, temp_ver_horizont,
level + 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], this);
L[i_max]->InitPorog(new_porog);
res_L += L[i_max]->Sim();
delete L[i_max]; L[i_max] = 0; if (level == 0 ) {
f_end = 100; };
F1[i_max] = new CTree;
bii = Find(ver_vertical, ver_horizont[find_j[i_max]].num);
Resize_matrix_stolb(temp_mas, find_i[i_max], find_j[i_max], bii, size_matrix);
AddFictVer(temp_ver_vertical, ver_horizont, find_j[i_max], size_matrix);
AddFictVer(temp_ver_horizont, ver_horizont, find_j[i_max], size_matrix);
F1[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical, temp_ver_horizont, level
+ 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], this);
F1[i_max]->InitPorog(new_porog);
res_F1 += F1[i_max]->Sim();
delete F1[i_max];
F1[i_max] = 0;
F2[i_max] = new CTree;
bjj = Find(ver_horizont, ver_vertical[find_i[i_max]].num);
Resize_matrix_str(temp_mas, find_i[i_max], find_j[i_max], bjj, size_matrix);
AddFictVer(temp_ver_horizont, ver_vertical, find_i[i_max], size_matrix);
AddFictVer(temp_ver_vertical, ver_vertical, find_i[i_max], size_matrix);
F2[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical, temp_ver_horizont, level
+ 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], this);
F2[i_max]->InitPorog(new_porog);
res_F2 += F2[i_max]->Sim();
delete F2[i_max];
F2[i_max] = 0;
}
if ((f_start_mode == 1) || (f_start_mode == 2)) {
i_max++;
find_i[i_max] = ver_beg.num - 1; //вначале совпадают номера индексов с номером
вершины
find_j[i_max] = ver_end.num - 1;
CalcPorog(ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], &new_porog);
//Удалим найденную строку и столбец из матрицы и из массивов вершин
Resize_matrix(temp_mas, find_i[i_max], find_j[i_max]);
Resize_mas_ver(ver_vertical, find_i[i_max], temp_ver_vertical);
Resize_mas_ver(ver_horizont, find_j[i_max], temp_ver_horizont);
//Блокируем в L3 элемент на пересечении строки find_j и столбца find_i
block_i = Find(temp_ver_vertical, ver_horizont[find_j[i_max]].num);
block_j = Find(temp_ver_horizont, ver_vertical[find_i[i_max]].num);
if ((block_i != -1) && (block_j != -1)) {
temp_mas[block_i][block_j] = BLC;
}
//Создаем две ветви и инициализируем их
if (block_i == -1) {
F1[i_max] = 0; } else {
bii = Find(ver_vertical, ver_horizont[find_j[i_max]].num);
Resize_matrix_stolb(temp_mas, find_i[i_max], find_j[i_max], bii, size_matrix);
flag_add_Ver = AddFictVer(temp_ver_vertical, ver_horizont, find_j[i_max],
size_matrix);
flag_add_Hor = AddFictVer(temp_ver_horizont, ver_horizont, find_j[i_max],
size_matrix);
bjj = Find(ver_horizont, ver_vertical[find_i[i_max]].num);
Resize_matrix_str(temp_mas, find_i[i_max], find_j[i_max], bjj, size_matrix + 1);
temp_mas[size_matrix - 1][size_matrix] = matrix[find_i[i_max]][bii];
temp_mas[size_matrix][size_matrix - 1] = matrix[bjj][find_j[i_max]];
flag_add_Hor = AddFictVer(temp_ver_horizont, ver_vertical, find_i[i_max],
size_matrix + 1);

```

```

        flag_add_Ver = AddFictVer(temp_ver_vertical, ver_vertical, find_i[i_max],
size_matrix + 1);
        F1[i_max] = new CTree;
        F1[i_max]->Init(temp_mas, size_matrix + 1, temp_ver_vertical,
temp_ver_horizont, level + 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]],
ver_horizont[find_j[i_max]], this);
        F1[i_max]->InitPorog(new_porog);
        F1[i_max]->InitCut(cut, cnt_cut);
        res_F1 += F1[i_max]->Sim();
        delete F1[i_max];
        F1[i_max] = 0;
    }
    F2[i_max] = 0;
} delete find_i; delete find_j; delete L; delete F1; delete F2;
L = 0; F1 = 0; F2 = 0; cnt_max = 0;
//Удаляем найденные маршруты
for(i = 0; i < (size_matrix + 1); i++) {
    delete temp_mas[i];
} delete temp_mas; delete temp_ver_horizont; delete temp_ver_vertical;
return 0; } }
Process_matrix();
for (i_max = 0; i_max < cnt_max; i_max++) {
    //Вычислим порог
    CalcPorog(ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], &new_porog);
    if ((ver_vertical[find_i[i_max]].num == 5) && (ver_horizont[find_j[i_max]].num == 4) &&
(cnt_max == 14)) {
        f_end = 10; }
    //2. если порог (граница) превышена
    if (new_porog > global_porog) { f_end = 2; continue; }
    //3. если удаляется ранее введенный фиктивный узел
    if ((f_control_fict == true) && (f_fict == FICT) && (find_i[i_max] != (size_matrix - 1) &&
find_j[i_max] != (size_matrix - 1))) { f_end = 3; continue; }
    //Удалим найденную строку и столбец из матрицы и из массивов вершин
    Resize_matrix(temp_mas, find_i[i_max], find_j[i_max]);
    Resize_mas_ver(ver_vertical, find_i[i_max], temp_ver_vertical);
    Resize_mas_ver(ver_horizont, find_j[i_max], temp_ver_horizont);
    //Блокируем в L3 элемент на пересечении строки find_j и столбца find_i
    block_i = Find(temp_ver_vertical, ver_horizont[find_j[i_max]].num);
    block_j = Find(temp_ver_horizont, ver_vertical[find_i[i_max]].num);
    if ((block_i != -1) && (block_j != -1)) {
        temp_mas[block_i][block_j] = BLC; }
    //Блокируем ветвь ведущую к заикливаннию
    if(cnt_marshrut > 0) {
        if (ver_vertical[find_i[i_max]].num == marshrut[cnt_marshrut - 1].end.num) {
            block2_i = Find(temp_ver_vertical, ver_horizont[find_j[i_max]].num);
            block2_j = Find(temp_ver_horizont, marshrut[cnt_marshrut - 1].beg.num);
            if ((block2_i != -1) && (block2_j != -1)) {
                temp_mas[block2_i][block2_j] = BLC; } } }
    //Поиск родных ветвей
    f_rodn_stolb = 0; f_rodn_str = 0; f_route_str = 0; f_route_stolb = 0;
    if (cnt_marshrut > 0) {
        if ((marshrut[cnt_marshrut - 1].end.num == ver_vertical[find_i[i_max]].num) &&
(marshrut[cnt_marshrut - 1].beg.f_num == ver_horizont[find_j[i_max]].f_num) &&
(marshrut[cnt_marshrut - 1].beg.num < ver_horizont[find_j[i_max]].num) ) {
            f_rodn_stolb = 1; } else {
            if ((marshrut[cnt_marshrut - 1].beg.num == ver_horizont[find_j[i_max]].num) &&
(marshrut[cnt_marshrut - 1].end.f_num == ver_vertical[find_i[i_max]].f_num) &&
(marshrut[cnt_marshrut - 1].end.num < ver_vertical[find_i[i_max]].num) ) {
                f_rodn_str = 1; } } }
    //Создаем три ветви и инициализируем их

```

```

if((f_rodn_stolb == 0) || (f_rodn_str == 0)) //||
// ((f_add_fict_str == 0) && (f_add_fict_stolb == 0)))
{
    L[i_max] = new CTree;
    L[i_max]->Init(temp_mas, size_matrix - 1, temp_ver_vertical, temp_ver_horizont, level +
1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], this);
    L[i_max]->InitPorog(new_porog);
    L[i_max]->InitCut(cut, cnt_cut);
    res_L += L[i_max]->Sim(); delete L[i_max];
    L[i_max] = 0; } else {
    L[i_max] = 0; }
if((block_i == -1) || (f_rodn_str == 1) || (f_route_str == 1)) {
    F1[i_max] = 0; } else {
    bii = Find(ver_vertical, ver_horizont[find_j[i_max]].num);
    Resize_matrix_stolb(temp_mas, find_i[i_max], find_j[i_max], bii, size_matrix);
    flag_add_Ver = AddFictVer(temp_ver_vertical, ver_horizont, find_j[i_max], size_matrix);
    flag_add_Hor = AddFictVer(temp_ver_horizont, ver_horizont, find_j[i_max], size_matrix);
    if((flag_add_Ver == 1) || (flag_add_Hor == 1)) {
        f_end = 3;
        F1[i_max] = new CTree;
        if (f_rodn_stolb == 1) {
            F1[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical, temp_ver_horizont,
level + 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]], ver_horizont[find_j[i_max]], this);
        } else {
            if ((temp_ver_vertical[size_matrix - 1].cnt_f > 1) || (temp_ver_vertical[size_matrix
- 1].super == true)) {
                F1[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical,
temp_ver_horizont, level + 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]],
ver_horizont[find_j[i_max]], this);
                f_route_stolb = 1; } else {
                    F1[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical,
temp_ver_horizont, level + 1, FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]],
ver_horizont[find_j[i_max]], this); } }
                F1[i_max]->InitPorog(new_porog); F1[i_max]->InitCut(cut, cnt_cut);
                res_F1 += F1[i_max]->Sim(); delete F1[i_max]; F1[i_max] = 0;
            } }
        } }
    if((block_j == -1) || (f_rodn_stolb == 1) || (f_route_stolb == 1)) {
        F2[i_max] = 0; } else {
        bjj = Find(ver_horizont, ver_vertical[find_i[i_max]].num);
        Resize_matrix_str(temp_mas, find_i[i_max], find_j[i_max], bjj, size_matrix);
        flag_add_Hor = AddFictVer(temp_ver_horizont, ver_vertical, find_i[i_max], size_matrix);
        flag_add_Ver = AddFictVer(temp_ver_vertical, ver_vertical, find_i[i_max], size_matrix);
        if((flag_add_Ver == 1) || (flag_add_Hor == 1)) {
            f_end = 3; } else {
            F2[i_max] = new CTree;
            //Если ветвь родная...
            if (f_rodn_str == 1) {
                F2[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical,
temp_ver_horizont, level + 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]],
ver_horizont[find_j[i_max]], this); } else {
                if ((temp_ver_horizont[size_matrix - 1].cnt_f > 1) ||
(temp_ver_horizont[size_matrix - 1].super == true)) {
                    F2[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical,
temp_ver_horizont, level + 1, NOT_FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]],
ver_horizont[find_j[i_max]], this);
                    f_route_str = 1; } else {
                        F2[i_max]->Init(temp_mas, size_matrix, temp_ver_vertical,
temp_ver_horizont, level + 1, FICT, marshrut, cnt_marshrut, ver_vertical[find_i[i_max]],
ver_horizont[find_j[i_max]], this);

```

```

    } }
        F2[i_max]->InitPorog(new_porog);
        F2[i_max]->InitCut(cut, cnt_cut);
        res_F2 += F2[i_max]->Sim();
        delete F2[i_max];
        F2[i_max] = 0;
    } } //i_max
if (f_write_file == 1) {
    if (level == 0) {
        close(handle); } } delete find_i; delete find_j; delete L; delete F1; delete F2;
L = 0; F1 = 0; F2 = 0; cnt_max = 0;
//Удаляем найденные маршруты
for(i = 0; i < (size_matrix + 1); i++) { delete temp_mas[i]; } delete temp_mas; delete
temp_ver_horizont; delete temp_ver_vertical;
if ((res_L == 0) || (res_F1 == 0) || (res_F2 == 0)) { return 0; } else { return 1; } }
int CTree::Delete(){ int i_max;
for (i_max = 0; i_max < cnt_max; i_max++) {
    if (L[i_max] != 0) {
        L[i_max]->Delete(); delete L[i_max]; }
    if (F1[i_max] != 0) {
        F1[i_max]->Delete(); delete F1[i_max]; }
    if (F2[i_max] != 0) {
        F2[i_max]->Delete(); delete F2[i_max]; } }
if (cnt_max != 0) { delete find_i; delete find_j; delete L; delete F1; delete F2;
L = 0; F1 = 0; F2 = 0; cnt_max = 0; } return 0;}
int CTree::LoadMatrix(float** mas, float* ver, int size, s_ver* ver_ver, s_ver* ver_hor, float** mas_dist){
int i, j;
if (global_matrix_size == 0) {
    global_matrix = new float*[size]; global_matrix_dist = new float*[size];
    global_matrix_size = size; global_vertical = new s_ver[size];
    global_horizont = new s_ver[size]; g_vershins_weigth = new float[size]; }
if (size != global_matrix_size) {
    for (i = 0; i < global_matrix_size; i++) { delete global_matrix[i]; delete global_matrix_dist[i]; }
delete global_matrix; delete global_matrix_dist; delete global_vertical; delete global_horizont; delete
global_matrix; delete g_vershins_weigth;
    global_matrix = new float*[size]; global_matrix_dist = new float*[size];
    global_matrix_size = size; global_vertical = new s_ver[size];
    global_horizont = new s_ver[size]; g_vershins_weigth = new float[size]; }
for(i = 0; i < size; i++) {
    global_matrix[i] = new float[size]; global_matrix_dist[i] = new float[size];
    for(j = 0; j < size; j++) {
        global_matrix[i][j] = mas[i][j]; global_matrix_dist[i][j] = mas_dist[i][j]; }
    global_vertical[i] = ver_ver[i]; global_horizont[i] = ver_hor[i];
    g_vershins_weigth[i] = ver[i]; } return 0;}
int CTree::DeleteMatrix(){ int i;
for (i = 0; i < global_matrix_size; i++) { delete global_matrix[i]; delete global_matrix_dist[i]; } delete
global_matrix; delete global_matrix_dist;
global_matrix_size = 0; return 0;}
float CTree::GetPorog(){ return global_porog;}

```

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	3
ВВЕДЕНИЕ	4
1. СОВРЕМЕННОЕ СОСТОЯНИЕ ТЕОРИИ И ПРАКТИКИ ПЕРЕВОЗОК ГРУЗОВ ПО РАЗВОЗОЧНЫМ МАРШРУТАМ	5
1.1. Обзор научных исследований в области перевозок грузов мелкими партиями	5
1.2. Анализ методов решения задач маршрутизации при перевозке мелких партий грузов	10
2. СОВЕРШЕНСТВОВАНИЕ МЕТОДА «ВЕТВЕЙ И ГРАНИЦ»	20
2.1. Разработка алгоритма решения задачи	20
2.2. Примеры решения задач по методике «фиктивных ветвей»	27
3. РАЗРАБОТКА МОДЕЛИ СОВМЕЩЕНИЯ МАРШРУТОВ ПРИ ДОСТАВКЕ МЕЛКОПАРТИОННЫХ ГРУЗОВ	39
3.1. Совершенствование теории совместной развозки грузов от двух производителей	39
3.2. Определение пункта заезда и экономическая оценка при совмещении маршрутов	54
3.3. Оценка эффективности организации перевозочного процесса	60
4. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ РАЗРАБОТАННОЙ МОДЕЛИ	67
4.1. Цель и задачи эксперимента	67
4.2. Выбор и описание объекта исследования	67
4.3. Методика проведения экспериментальных исследований	69
4.4. Моделирование совмещённого маршрута и оценка его эффективности	74
4.5. Оценка применения методики «фиктивных ветвей»	84
ЗАКЛЮЧЕНИЕ	87
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	88
ПРИЛОЖЕНИЕ	98

Научное издание

Подшивалова Кристина Сергеевна

**ДОСТАВКА МЕЛКОПАРТИОННЫХ ГРУЗОВ
АВТОМОБИЛЬНЫМ ТРАНСПОРТОМ**

Монография

В авторской редакции

Верстка Н.А. Сазонова

Подписано в печать 14.11.14. Формат 60×84/16.

Бумага офисная «Снегурочка». Печать на ризографе.

Усл. печ. л. 7,2. Уч.-изд. л. 7,75. Тираж 500 экз. 1-й завод 100 экз.

Заказ №404.

Издательство ПГУАС.
440028, г.Пенза, ул. Германа Титова, 28.