Министерство образования Республики Беларусь
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Организация автомобильных перевозок и дорожного движения»

В.Н. Седюкевич

МАТЕМАТИЧЕСКИЕ МОДЕЛИ В ТРАНСПОРТНЫХ СИСТЕМАХ

Конспект лекций
для студентов специальности 1-44 01 01 «Организация перевозок и управление на автомобильном и городском транспорте»

Учебное электронное издание

Минск 2009

УДК 656.073:681.142 М

А в т о р :
В.Н. Седюкевич

Р е ц е н з е н т ы :
Г.И. Лебедева, доцент кафедры «Математика № 1» Белорусского национального технического университета, кандидат технических наук;

Н.Н. Пилипук, профессор кафедры "Организация и управление на транспорте", кандидат экономических наук

В конспекте лекций рассматриваются математические модели в транспортных системах, излагаются методы их исследования и принятия оптимальных решений. Приводятся алгоритмы и компьютерные программы для решения рассматриваемых задач. Пособие предназначено для студентов специальности 1-44 01 01 "Организация перевозок и управление на автомобильном и городском транспорте". Может быть использовано студентами специальности 1-44 01 02 "Организация дорожного движения" и других специальностей направления "Транспортная деятельность".

Белорусский национальный технический университет
пр-т Независимости, 65, г. Минск, Республика Беларусь
Тел.(017) 293-91-97 факс (017) 292-91-37
Регистрационный № БНТУ/АТФ18 – 4.2009

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель дисциплины "Математические модели в транспортных системах" – изучение методов исследования и оптимизации транспортных систем, в том числе с использованием вычислительной техники (компьютеров).

В результате изучения дисциплины должны быть освоены способы построения математических моделей, их исследование и моделирование на компьютерах; усвоены методы решения оптимизационных задач; развиты навыки и приобретено умение разработки алгоритмов и компьютерных программ, а также использования функционально и проблемно ориентированных пакетов прикладных программ для принятия решений.

Материал дисциплины базируется на ранее полученных математических, инженерно-технических знаниях, в частности, при изучении дисциплин "Математика" и "Информатика".

Изложение материала дисциплины производится применительно к организации перевозок и управлению на транспорте.

Изучаемые вопросы содержатся в рекомендуемой основной и дополнительной литературе.

Полученные знания используются в инженерной практике при транспортной деятельности. Задачи с использованием математических моделей и методов применяются на транспорте при решении перспективных вопросов, проектировании транспортно-технологических схем, оперативном планировании и управлении перевозками и разработке технических нормативов:

- прогнозирование объемов перевозок и технико-эксплутационных показателей;
- обоснование структуры парка транспортных средств;
- поиск кратчайших расстояний между пунктами транспортной сети;
- оптимизация распределения ресурсов;
- маршрутизация перевозок;
- выбор транспортных средств и схем перевозок;
- закрепление маршрутов перевозок за предприятиями транспорта;
- распределение транспортных средств по объектам перевозок;
- разработка графиков и расписаний, согласование работы транспортных средств и терминалов;
- обоснование норм времени на выполнение операций производственных процессов с учетом случайности, согласования и упорядочения работ;
- обоснование норм времени и норм материальных ресурсов на основе учета влияющих на них факторов.

Математические модели и методы закладываются в основу алгоритмов функционирования автоматизированных рабочих мест (АРМ) по организации перевозок и управлению на транспорте. Функции АРМ предусматривают решение ранее перечисленных и некоторых других задач (учетных, информационных). Применение математических методов и компьютерных технологий при научных исследованиях, решении задач планирования и управления производством способствуют ускорению научно-технического прогресса.

При изучении дисциплины разработка компьютерных программ ориентирована на такие алгоритмические языки как Бейсик и Паскаль. Кроме того, применяются готовые программные продукты для персональных компьютеров.

1. ОСНОВЫ ИССЛЕДОВАНИЯ СИСТЕМ И ПРИНЯТИЯ РЕШЕНИЙ

1.1. Постановка задач принятия решений и разработка моделей

При решении задач организации производственных процессов и управления ими используются методы математического моделирования и оптимизации на основе системного подхода. Принятие решений основывается на формализованном описании задачи, количественном анализе влияющих факторов и достигаемых целей и включает разработку математической модели задачи, исследование модели и нахождение оптимального решения, а также анализ полученных результатов. Решение получают на основе применения методов оптимизации. Ряд методов принятия решений объединяется под названием "исследование операций". При транспортной деятельности решения могут приниматься на основе логистических подходов.

Оптимальное – это такое решение, которое обеспечивает экстремум (максимум, минимум) целевой функции (критерия оптимальности) при выполнении заданной системы ограничений.

Критерии оптимальности должны быть представительными, чувствительными к изменениям оптимизируемых параметров и как можно более простыми. Цель может быть правильно сформулирована только с позиций надсистемы.

Под системой понимается множество подсистем (объектов, подразделений), которые функционируют как единое целое по выполнению поставленной цели.

Системный подход, опираясь на понимание функционирования системы как единого целого, предполагает учет всех факторов, влияющих на решение задачи, в том числе исследование внутренних связей между отдельными элементами и взаимодействие с другими системами и объектами.

Модели служат отображениями (прообразами) реальных систем, процессов, явлений и могут быть физические и математические. Математические модели представляют собой описание задачи в виде совокупности соотношений (уравнений, неравенств, логических условий), определяющих связи между параметрами функционирования исследуемой системы, ограничениями и критериями оптимальности.

Модель должна быть как можно более адекватна оригиналу. Понятие адекватности модели связано с такими общими кибернетическими терминами как "черный ящик", изоморфизм, гомоморфизм.

"Черный ящик" – это система, в которой доступны наблюдению только внешние входные и выходные параметры, а внутреннее устройство неизвестно. Исследуется по связям между значениями входных и выходных параметров.

Системы, характеризующиеся одинаковыми наборами входных и выходных величин (одинаково реагирующие на внешние воздействия), независимо от их внутренней структуры, называются изоморфными. В силу изоморфности систем исследование "черного ящика" не может привести к однозначному выводу о внутренней структуре системы. Однако любая из изоморфных систем может рассматриваться как модель остальных.

Система, полученная из исходной путем упрощения, является ее гомоморфной моделью.

Математическая модель, как правило, находится в гомоморфном отношении к реальному объекту. Она должна отражать связи между входными и выходными параметрами системы и является основой для вычисления значений критериев и проверки ограничений.

Если факторы, от которых зависит функционирование системы, разделить на известные $A = \{a_1, a_2, ..., a_m\}$, на которые влиять нельзя, и управляемые $X = \{x_1, x_2, ..., x_k\}$, учесть получаемые выходные параметры $Y = \{y_1, y_2, ..., y_n\}$, заданный вектор целевой функции $Z = \{z_1, z_2, ..., z_p\}$ и ограничения $O=\{o_1, o_2, ..., o_s\}$, то имеем:

$$Y = F(A,X);$$

$$Z = W(Y) = \underset{X}{\max(\min)}\,;$$

$$O = G(A,X,Y) <>= 0,$$

где F, Z, O – функции.

Формулировка задачи принятия решений следующая:

при заданных условиях A требуется найти такие значения элементов вектора X, при которых вектор целевых функций Z обращается в максимум (минимум) и выполняются ограничения O.

Когда не все условия, в которых происходит функционирование системы заранее известны, то имеется еще один набор факторов $U= \{u_1, u_2,...,u_r\}$. Это переводит задачу в другую категорию – принятие решения в условиях неопределенности. Формулировка задачи следующая:

при заданных условиях A с учетом неизвестных параметров (факторов) U найти такие значения элементов вектора X, которые дают экстремум вектора целевых функций Z при выполнении заданных ограничений O.

Ограничения, накладываемые на управляемые, неуправляемые и выходные параметры, могут быть связаны с лимитом ресурсов, обеспечением безопасности, являться следствием физических законов.

Решение поставленной задачи достигается по алгоритмам соответствующих методов оптимизации.

Для исследования математических моделей используются компьютеры. Необходимость применения последних возникает при обработке информации, использовании численных методов и методов случайного поиска, имитационном моделировании работы систем и решении других задач.

1.2. Классификация математических моделей и методов принятия решений

Математические модели могут быть статическими (рассматривается на конкретный момент времени) и динамическими (описывают процессы во времени). Если состояние системы описывается в каждый момент времени, то модель – непрерывная и если в фиксированные моменты времени – то дискретная.

Модели, в которых зависимости носят неслучайный характер, являются детерминированными, а в которых случайный характер – стохастическими.

По числу оптимизируемых параметров различают одномерные и многопараметрические задачи.

По возможным значениям оптимизируемых параметров решения могут быть вещественными или дискретными, например целочисленными.

Модели (задачи), в которых критерий оптимальности может иметь несколько локальных экстремумов, называют многоэкстремальными.

Задачи с ограничениями – это задачи условной оптимизации и без ограничений – безусловной. Первые относятся к задачам математического программирования. Задачи оптимизации при линейных критериях и ограничениях являются задачами линейного программирования, а при нелинейных – нелинейного, в т.ч. динамического, геометрического программирования.

В зависимости от условий внешней среды и степени информированности об ее состоянии различают следующие задачи принятия решений:

а) в условиях определенности;

б) в случайных условиях (в условиях риска);

г) в условиях неопределенности;

д) в условиях конфликтных ситуаций или противодействия (активного противника).

По способу исследования (оптимизации) различают следующие методы:

детерминированные – аналитические или численные методы;

методы случайного (статистического) поиска.

В зависимости от типа решаемых задач различают методы локальной оптимизации, позволяющие найти экстремум только унимодальной функции, и методы глобальной оптимизации, с помощью которых можно найти оптимум многоэкстремальной функции.

Кроме того методы оптимизации различают в зависимости от типа (вида) математической модели.

Типичными классами оптимизационных задач на транспорте являются:

управление запасами;

нахождение кратчайших путей;

распределение ресурсов;

массовое обслуживание;

сетевое планирование и управление;

замена оборудования.

1.3. Принятие решений в условиях определенности при векторном критерии

Принятие решений в условиях определенности характеризуется детерминированными связями между принятыми решениями и полученными результатами. Однако при этом может быть несколько критериев для принятия решения. Возникает задача принятия решений при "векторном критерии". Множество всех оптимальных точек по отдельным критериям называется областью компромиссов или областью решений по Парето.

Требование одновременной максимизации (минимизации) всех частных критериев обычно несовместимо, так как при увеличении (уменьшении) одного из них могут снижаться (увеличиваться) другие. Для нахождения решения при многокритериальной оптимизации переходят от задачи векторной оптимизации по частным критериям z_1, $z_2,\ldots,z_p$ к специально сконструированной скалярной функции Z_o, аргументами которой являются эти частные критерии. Процесс образования скалярной функции, являющейся обобщенным критерием для задач многокритериальной оптимизации, называется свертыванием или объединением векторного критерия.

Теория исследования операций предлагает ряд способов формирования обобщенного критерия Z_o по набору частных z_i.

Для **параметрических критериев** ниже приводится ряд способов их объединения.

Способ 1

Критерий Z_o является взвешенной суммой частных критериев z_i

$$Z_o = \sum_{i=1}^{P} k_i \; z_i = \min_{X} (\max) \, ,$$

где k_i – весовой коэффициент i-го критерия.

Неравнозначность частных критериев z_i оценивается весовыми коэффициентами k_i, что позволяет формировать с помощью данного критерия различные цели. Однако при применении такого критерия возможно, что при оптимальном решении экстремальное значение Z_o достигается при большом отклонении какого-то частного критерия z_i от его оптимального значения. Для исключения данной ситуации могут вводиться ограничения. Весовые коэффициенты могут быть: размерными (при различной размерности z_i) или безразмерными; положительными и отрицательными (при свертывании критериев типа $z_i \to \max$ и $z_i \to \min$).

Способ 2

Критерий Z_o основан на минимизации абсолютных отклонений частных критериев от их экстремальных значений

$$Z_o = \sum_{i=1}^{P} k_i \, abs(z_i - z_{io}) = \min_X ,$$

где $z_{io} = \min_X z_i$ – при минимизации и

$z_{io} = \max_X z_i$ – при максимизации.

Способ 3

Критерий Z_o состоит в минимизации относительных отклонений частных критериев от их экстремальных значений без учета весовых коэффициентов (применяется при различных видах экстремума, отсутствии информации о важности критериев или при различной их размерности)

$$Z_o = \sum_{i=1}^{P} abs((z_i - z_{io})/z_{io}) = \min_X .$$

Способ 4

Критерий Z_o формируется как взвешенная сумма частных критериев с учетом установленных ограничений. Тогда

$$Z_o = \sum_{i=1}^{P} k_i \; z_{i\,н} = \min_X(\max),$$

где $z_{iн} = \begin{cases} z_i, \text{ если ограничение не нарушено} \\ +\infty\ (-\infty) \text{ при нарушении ограничения.} \end{cases}$

Способ 5

Критерий Z_o является минимальным (максимальным) из частных критериев z_i (частные критерии должны быть одной размерности и вида экстремума)

$$Z_o = \min_X z_i \to \max \text{ или}$$

$$Z_o = \max_X z_i \to \min.$$

Для придания гибкости этому способу можно использовать весовые коэффициенты k_i при z_i. Изменяя значения k_i, можно определять необходимые цели.

Способ 6

Критерий Z_o является одним из множества частных критериев z_i (главным), отвечающим основной цели. По остальным критериям может проверяться выполнение наложенных на них ограничений.

Для **логических критериев** в зависимости от поставленной конечной цели возможны следующие способы их объединения:

цель достигается при выполнении всех частных целей одновременно

$$Z_o = \prod_{i=1}^{P} z_i = 1 \text{ (конъюнкция критериев);}$$

цель достигается при достижении хотя бы одной частной цели

$$Z_o = 1 - \prod_{i=1}^{P} (1 - z_i) = 1 \text{ (дизъюнкция критериев)},$$

$$z_i = \begin{cases} 1, \text{ если } i\text{-я цель достигнута} \\ 0, \text{ если } i\text{-я цель не достигнута} \end{cases}.$$

1.4. Принятие решений в условиях риска и неопределенности

Задача в условиях риска состоит в том, что из-за случайности влияния отдельных факторов, например, внешней среды (среды), при каждой принимаемой стратегии X_i имеет место множество возможных результатов Y_j с известными вероятностями $p(Y_j, X_i)$, $j = \overline{1, J}$, $i = \overline{1, I}$ (I – общее число стратегий; J – общее число результатов при каждой стратегии). При этом достигается эффект $V(Y_j, X_i)$.

Обобщенной оценкой стратегии X_i является величина ожидаемого эффекта $V_o(X_i)$, рассчитываемая по формуле

$$V_o(X_i) = \sum_{j=1}^{J} p(Y_j, X_i)\ V(Y_j, X_i)\ .$$

Если в качестве исходных данных определены вероятности различных состояний среды, то обобщенная оценка $V_o(X_i)$ стратегии X_i определяется по формуле:

$$V_o(X_i) = \sum_{r=1}^{R} p(U_r)\ V(X_i, U_r)\ ,$$

где R – общее число возможных состояний среды; $p(U_r)$– вероятность нахождения среды в состоянии U_r ($r = \overline{1, R}$); $V(X_i, U_r)$ – эффект, который складывается при стратегии X_i и состоянии среды U_r.

Принятие решений в условиях риска состоит в определении оптимальной i-й стратегии X_i как

$$\max_{Xi} V_o(X_i)\ ,$$

где $V_o(X_i)$ – оценки эффективности (полезности) для стратегии X_i, $i = \overline{1, I}$.

При принятии решений в условиях неопределенности информация о состоянии среды неизвестна принимающему решение.

Относительно состояния среды могут высказываться определенные гипотезы. Такие предположения о вероятном состоянии среды называется субъективными вероятностями $p(U_r)$, $r = \overline{1, R}$. В этом случае имеет место задача принятия решений в условиях риска.

В условиях неопределенности вероятности возможных состояний среды неизвестны. Для выбора оптимальной стратегии в условиях неопределенности предложен ряд критериев.

Критерий Вальда (критерий "осторожного наблюдателя") основывается на предположении, что среда будет находиться в неблагоприятном состоянии, и имеет решающее правило

$$\max_{Xi} \; (\min_{Ur} \; V(X_i , U_r)) \; .$$

Критерий Гурвица основывается на следующем решающем правиле:

$$\max_{Xi} \; (k_d \max_{Ur} V (X_i , U_r) + (1 - k_d) \min_{Ur} \; V (X_i , U_r)) \, ,$$

где k_d – коэффициент доверия.

По критерию Гурвица предполагается, что среда находится с вероятностью k_d в благоприятном состоянии и с вероятностью $1 - k_d$ – в неблагоприятном. При $k_d = 0$ получаем критерий Вальда, а при $k_d = 1$

$$\max_{Xi} \; (\max_{Ur} \; V(X_i , U_r)) \text{ – стратегия "здорового оптимиста".}$$

Критерий Лапласа (случай предположения о равновероятных состояниях среды) $P(U_1)=P(U_2)=\ldots =P(U_R)$ имеет решающее правило

$$\max_{Xi} (1/R \sum_{r=1}^{R} \; V(X_i , U_r)) \; .$$

Критерий Сэвиджа (критерий минимизации "сожалений") основывается на расчете "сожалений" $V_S (X_i \; , U_r)$, равных полезности результата $V (X_i , U_r)$ при данном состоянии среды U_r относительно наилучшего решения в зависимости от стратегии X_i, определяемых как $\max\limits_{X_i} \; V (X_i , U_r)$:

$$V_S (X_i \; , U_r) = V (X_i \; , U_r) - \max_{X_i} \; V (X_i , U_r) \; .$$

К рассчитанным сожалениям применяется решающее правило

$$\max_{X_i} \; (\min_{U_r} \; V_S (X_i \; , U_r)) \, .$$

Этот критерий минимизирует возможные потери при условии, что состояние среды неблагоприятное.

Выбор одного из вышеуказанных критериев в качестве решающего производится принимающим решение.

Пример.

Необходимо найти оптимальное число машиномест на проектируемой автомобильной стоянке. К рассмотрению приняты 3 типовых проекта на 200, 250 и 300 машиномест. Ожидаемая прибыль $V(X_i,U_r)$ в зависимости от числа мест X_i и состояния среды (числа занятых мест) U_r задана таблично. В этой же таблице приведены вероятности возможных состояний среды $p(U_r)$.

X_i	$V(X_i, U_r)$ и $V_s(X_i, U_r)$ (выделенные курсивом) при U_r / $p(U_r)$			
	150/0.1	200/0.1	250/0.6	300/0.2
200	-15	30	30	30
	0	*0*	*-20*	*-30*
250	-35	20	50	50
	-20	*-15*	*0*	*-15*
300	-55	-10	45	60
	-40	*-40*	*-5*	*0*

Решение.

Критерий Вальда

$$\max_{Xi} \left(\min_{Ur} V(X_i, U_r)\right) = \max_{Xi} \left(\min_{Ur} (-15,30,30,30) \text{ для } x_1 = 200;\right.$$

$$\min_{Ur} (-35,20,50,50) \text{ для } x_2 = 250; \min_{Ur} (-55,-10,45,60) \text{ для } x_3 = 300).$$

$$\max_{Xi}(-15 \text{ для } x_1 = 200, -35 \text{ для } x_2 = 250, -55 \text{ для } x_3 = 300) = -15 \text{ (при } x_1 = 200).$$

По этому критерию от строительства следует отказаться, так как при оптимальной стратегии имеет место отрицательный эффект.

Критерий Гурвица при K_d = 0.6

$$\max_{Xi} \left(k_d \max_{Ur} V(X_i, U_r) + (1-k_d) \min_{Ur} V(X_i, U_r)\right) =$$

$$= \max_{Xi} (0.6 \cdot 30 + 0.4(-15) \text{для } x_1 = 200; 0.6 \cdot 50 + 0.4(-35) \text{для } x_2 = 250; 0.6 \cdot 60 + 0.4(-55) \text{для } x_3 = 300).$$

$$\max_{Xi}(12.0 \text{ для } x_1 = 200; 16.0 \text{ для } x_2 = 250; 14.0 \text{ для } x_3 = 300) = 16.0 (\text{ при } x_2 = 250).$$

и **при Kd =1.0**

$$\max_{Xi} \left(\max_{Ur} V(X_i, U_r)\right) = 60.0 \; (x_3 = 300).$$

Критерий Лапласа

$$\max_{Xi}\left(1/R \sum_{r=1}^{R} V(X_i, U_r)\right) = \max_{Xi} (1/4 (-15+30+30+30) \text{ для } x_1 = 200;$$

$$1/4 (-35+20+50+50) \text{для } x_2 = 250; \; 1/4 (-55-10+45+60) \text{ для } x_3 = 300) =$$

$$= 85/4 (\text{при } x_2 = 250)$$

Критерий Сэвиджа

Результаты расчета сожалений по ранее приведенной формуле даны в таблице вторыми строками. Например, для 1-го столбца (U_r=150) сожаления определяются по выражению

$$V_s(X_i, U_1) = V(X_i, U_1) - \max_{Xi} V(X_i, U_1) =$$

$$= V(X_i, U_1) - \max_{Xi} V(-15, -35, -55) = V(X_i, U_1) + 15.$$

По рассчитанным сожалениям поиск оптимального решения проводится следующим образом

$$\max_{X_i}\left(\min_{U_r} V_s(X_i, U_r)\right) = \max_{X_i}\left(\min_{U_r}(0,0,-20,-30) \text{ для } X_1;\right.$$
$$\left.\min_{U_r}(-20,-10,0,-15) \text{ для } X_2; \min_{U_r}(-40,-140,-5,0) \text{ для } X_3\right) =$$
$$= \max_{X_i}(-30 \text{ для } X_1; -20 \text{ для } X_2; -40 \text{ для } X_3) = -20 \,(\text{при } X_2 = 250).$$

Если известны вероятности состояния среды, то решение производится в условиях риска:

$$\max_{X_i} V_o(X_i) = \sum_{r=1}^{R} p(U_r)\, V(X_i, U_r) = \max_{X_i}((0.1(-15)+0.1\cdot 30+0.6\cdot 30+0.2\cdot 30) \text{ для } X_1;$$
$$(0.1(-35)+0.1\cdot 20+0.6\cdot 50+0.2\cdot 50) \text{ для } X_2;\ (0.1(-55)+0.1(-10)+0.6\cdot 45+0.2\cdot 60) \text{ для } X_3) =$$
$$= 38.5 \,(\text{при } X_2 = 250)$$

Для условий примера по большинству критериев наиболее эффективно строительство стоянки на 250 машиномест.

Пример компьютерной программы для принятия решений в условиях риска и неопределенности приведен в приложении 1.

1.5. Программное компьютерное обеспечение исследования транспортных систем

Программное обеспечение компьютеров можно разделить на следующие виды: системное (операционные системы); системы программирования; прикладное.

Операционные системы (ОС) – это набор программ, осуществляющих управление работой компьютера.

Функции ОС:

связь с пользователем в реальном времени для подготовки устройств к работе, переопределение конфигурации и изменение состояния системы;

выполнение операций ввода-вывода с обработкой прерываний, запросов и распределением их между устройствами;

загрузка приложений в оперативную память (RAM/ОЗУ) и их выполнение, управление оперативной памятью, распределение ее между процессами, выделение виртуальной памяти;

управление доступом к данным с обеспечением их защиты и ограничений на доступ;

обработка исключительных условий во время выполнения задачи – ошибок, прерываний;

функции по обеспечению организации сетей, использованию служебных программ, выполнению сетевых операций.

Наибольшее распространение имеют семейства таких многозадачных многопользовательских ОС как Microsoft Windows, Mac OS и Linux.

Системное программирование, кроме непосредственно операционной системы, содержит также ряд внешних утилит, обеспечивающих сервисное обслуживание работы пользователя.

Программирование может осуществляться в машинных кодах и на символьных языках.

Наибольшее распространение получили следующие языки программирования: Ассемблер; Макроассемблер; Бейсик – варианты Quick, Turbo, Visual; Cobol (Кобол); Fortran(Фортран); Pascal (Паскаль); C (Си); Lisp (Лисп) – для машинной графики; Prolog (Пролог) – для обработки логической информации; объектноориентированная система программирования Delphi (Делфи).

Для удобной работы с компьютером кроме ОС используются оболочки (FAR manager, Norton Commander, DOS Navigator, Volkov Commander, Total Commander и др.). Большинство современных систем программирования также представляют собой среду со своим головным меню, редактором, транслятором, компоновщиком (редактором связей, сборщиком), отладчиком.

Прикладное программирование подразделяется на пакеты прикладных программ и программы пользователя.

Пакеты прикладных программ охватывают инструментальные средства, интегрированные, функционально ориентированные и проблемно ориентированные пакеты.

Инструментальные средства представляют собой диагностические, тестовые, антивирусные пакеты и т.п.

Для интегрированных пакетов характерно следующее:

- совместимость записи данных, дающая возможность их вызова различными средствами для различных целей;
- возможность продолжить выполнять свою функцию, если понадобилось на время переключиться на другую;
- преемственность различных типов команд и методов работы с меню.

Интегрированные пакеты позволяют работать с отдельными программами, базами данных, графикой, создавать прикладные программы, поддерживать связь с другими компьютерами. Примерами таких пакетов являются Windows Office, Lotus и др.

К функционально ориентированным пакетам относятся средства работы с текстом, обработки электронных таблиц, организации баз данных, поддержки интерактивной графики, функционирования экспертных систем и т.п. Примерами являются пакеты машинной графики (AutoCAD, Компос), графические редакторы (Adobe PhotoShop, CorelDraw и др.), электронные таблицы и деловая графика (SuperCalc, Excel, QuattroPro, Grapher), СУБД (Access, Clarion, Clipper, dBase, FoxBase, FoxPro, FoxGraph, Ingres, Paradox и др.), редакционно-издательские системы (PageMaker, Ventura Publisher), анимационные (3D StudioMAX и др.), презентационные (PowerPoint).

Проблемно-ориентированные пакеты охватывают различные сферы применения: математика, экономика, транспорт, бухгалтерский учет и др. Для разнообразных задач математической статистики могут служить пакеты программ Statistica и "Олимп". Программы Matlab, Gauss, Assyst, Eurica, Maple V, Mathematica, MathCad предназначены для решения задач матричной и векторной алгебры, векторного анализа, решения систем линейных и нелинейных уравнений. Некоторые из них позволяют выполнить преобразование математических выражений в символьной форме (упростить выражение или представить в другом виде), найти вид неопределенного интеграла.

Работа пользователя в пакетах производится с помощью "меню". Максимальное число альтернатив, содержащихся в "меню", различно. Обычно принимают равным 7±2 (7 – число по Миллеру).

Через меню могут запускаться программы из командного файла или из головной программы, а также ветвится выполнение программы (подпрограммы). Меню может быть одномерным, двумерным, аналогичным картотеке и представлено в виде алфавитно-цифровой информации и графических изображений. Активизация функций может производиться по набору ключевого символа, по нажатию клавиш ("Ввод", функциональных и др.) клавиатуры или кнопок манипуляторов ("мышки", джойстика и т.п.) при нахождении их указателя на месте соответствующего изображения.

При проектировании прикладных программ должны быть определены следующие характеристики:

1) состав исходного текста:

1.1) единый текст;

1.2) отдельные текстовые модули;

2) структура исполняемой программы:

2.1) единый модуль, полностью загружаемый в ОЗУ при запуске;

2.2) несколько сегментов, загружаемых в ОЗУ по мере необходимости;

2.3) резидентная часть, загружаемая в ОЗУ в начале сеанса, и одна или несколько нерезидентных частей, загружаемых по мере необходимости;

3) Способ хранения данных на внешнем постоянном запоминающем устройстве (ВПЗУ):

3.1) все данные располагаются в одном файле;

3.2) данные распределены по нескольким файлам.

Состав исходного текста оказывает влияние на способ разработки программ, структура исполняемой программы – на требования к ОЗУ и быстродействие, способ хранения данных на ВПЗУ – на быстродействие при доступе к данным и характер использования внешней памяти.

Применение подпрограмм, процедур, функций и других отдельных программных модулей обеспечивает структурирование программ на уровне исходных текстов, объектных модулей и выполняемых программ. Под объектным модулем понимается преобразованный в машинные коды (транслированный) текст программы. Может применяться подстановка – включение перед трансляцией в текст основной программы текстов других модулей. Исходные тексты модулей могут формироваться в виде библиотек. Отдельные модули можно также транслировать независимо друг от друга и связывать только на стадии компоновки исполняемой программы (загрузочного модуля). Выполняется сборка программы с помощью редактора связей (компоновщика). При таком подходе к программированию создаются библиотеки объектных модулей. В системах программирования могут иметься библиотеки стандартных процедур (функций и подпрограмм).

При создании перекрывающихся (оверлейных) сегментов программа состоит из отдельных частей, которые при ее выполнении загружаются в ОЗУ по мере необходимости. Корневой сегмент находится постоянно в ОЗУ. Он содержит обращения к процедурам, находящимся в оверлейных сегментах. Сегменты могут быть связаны в сложные древовидные структуры. Быстродействие системы падает из-за потерь времени на перезагрузку сегментов с внешнего накопителя.

Отдельные модули пакетов обычно создают (выделяют) по функциональному принципу: ввод данных, корректировка данных, расчетная часть, графическое представление результатов, вывод (печать) результатов.

Межмодульный информированный обмен может осуществляться через общие области ОЗУ и файлы на ВПЗУ. В случае необходимости обмена при разнесенном во времени исполнении программ или модулей применяется обмен через файлы на ВПЗУ.

Достоверность программного обеспечения отрабатывается и проверяется на контрольных примерах. Тестирование должно быть произведено для всех возможных вариантов расчетов и значений исходных данных. При наличии ограничений на исходные данные об этом должно сообщаться пользователю. Документация на программные продукты должна отвечать стандартам Единой системы программной документации (ЕСПД).

2. ПОСТРОЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

2.1. Детерминированные модели

2.1.1. Решение систем линейных уравнений

Исходный вид системы:

$$
\begin{aligned}
&a_{11}x_1 + \ldots + a_{1i}x_i + \ldots + a_{1p}x_p = b_1\\
&\ldots\\
&a_{j1}x_1 + \ldots + a_{ji}x_i + \ldots + a_{jp}x_p = b_j\\
&\ldots\\
&a_{p1}x_1 + \ldots + a_{pi}x_i + \ldots + a_{pp}x_p = b_p
\end{aligned}
$$

где a_{ji} – коэффициенты системы при неизвестных; b_j – свободные члены.

В свернутом виде система описывается следующим выражением:

$$\sum_{i=1}^{p} a_{ji}x_i = b_j;\ j = \overline{1, p},\ i = \overline{1, p}.$$

В матричном виде система имеет вид

$$A\,X = B,$$

где $A = \begin{pmatrix} a_{11} \ldots a_{1i} \ldots a_{1p} \\ \ldots \\ a_{j1} \ldots a_{ji} \ldots a_{jp} \\ \ldots \\ a_{p1} \ldots a_{pi} \ldots a_{pp} \end{pmatrix};$

$B = \begin{pmatrix} b_1 \\ \ldots \\ b_j \\ \ldots \\ b_p \end{pmatrix};$

$X = \begin{pmatrix} x_1 \\ \ldots \\ x_j \\ \ldots \\ x_p \end{pmatrix}.$

Требуется найти значения X, удовлетворяющие всем уравнениям системы.

Методами решения систем линейных уравнений являются: метод подстановок, метод последовательного исключения переменных, метод Крамера (матричный метод). Могут также применяться методы решения систем нелинейных уравнений.

Метод подстановок (последовательного выражения переменной из одного уравнения и подстановки в другое) не удобен для алгоритмизации расчетов при переменном числе переменных.

Метод последовательного исключения переменных достаточно удобен для машинной реализации. При этом методе с помощью преобразований строк текущей системы (выравнивания коэффициентов при первой переменной текущей системы и взаимного вычитания из одного уравнения другого) получается новая система без первой переменной.

Таким образом, на каждом этапе таких последовательных преобразований получаем понижение числа переменных (на последнем этапе до одной):

$$a_{1\,1}x_1 + \ldots + a_{1\,i}x_i + \ldots + a_{1\,p}x_p = b_1$$
$$a'_{2\,2}x_1 + \ldots + a'_{2\,i}x_i + \ldots + a'_{2\,p}x_p = b'_2$$
$$\ldots$$
$$x_p = b'_p\,,$$

где штрихом обозначены значения коэффициентов после преобразований.

Значение свободного члена для уравнения с одной переменной дает решение, например по переменной x_p ($x_p=b'_p$), Затем из одного из уравнений системы предпоследнего этапа находится x_{p-1} и т.д., для 1-го этапа – x_2 и из исходной системы – x_1. Разновидность – метод Гаусса с выбором главного элемента.

Метод Крамера основан на матричном исчислении и наиболее удобен с точки зрения составления алгоритма и его программной реализации на компьютере.

По методу Крамера

$$X = A^{-1}\,B$$

или

$$x_i = \det A_i \,/\, \det A\,,$$

где A^{-1} – матрица, обратная матрице А;

A_i – матрица, полученная по матрице А с заменой в ней i- го столбца столбцом свободных членов ($a_{ji} = b_j$), $j = \overline{1, p}$;

det – детерминант (определитель) матрицы.

Если det A равен нулю, то система не определена. При значениях det A близких к нулю система слабо обусловлена.

2.1.2. Решение систем нелинейных уравнений

Задача состоит в нахождении корней следующей системы уравнений

$$f_i(X) = 0\ \},\ i = \overline{1, m}\,,$$

где $X = \{\, x_1, x_2,\ldots,x_m\}$.

Для решения могут применяться следующие методы:

- метод простых итераций;
- метод Зейделя, отличающийся от метода простых итераций тем, что уточненные значения x_i сразу подставляются в последующие уравнения;
- на основе методов поиска экстремума многомерных функций и др.

Метод простых итераций состоит в реализации процесса по следующей формуле:

$$x_{i\,(k+1)} = F_i(X_k),$$

где $F_i(X) = f_i(X) + x_i = x_i$;

i – номер переменной;

k – номер итерации.

Итерации выполняются до тех пор, пока сохраняется хотя бы по одному из x_i условие

$$abs\ (x_{i(k+1)} - x_{i(k)}) > E,$$

где E – заданная точность.

Метод обеспечивает сходимость, если

$$\sum_{i=1}^{m} abs(\partial F_i(X) / \partial x_i) < 1,\ j = \overline{1,m},\ i = \overline{1,m}.$$

Графическая интерпретация метода приведена на рисунке 2.1, а схема алгоритма на рисунке 2.2.

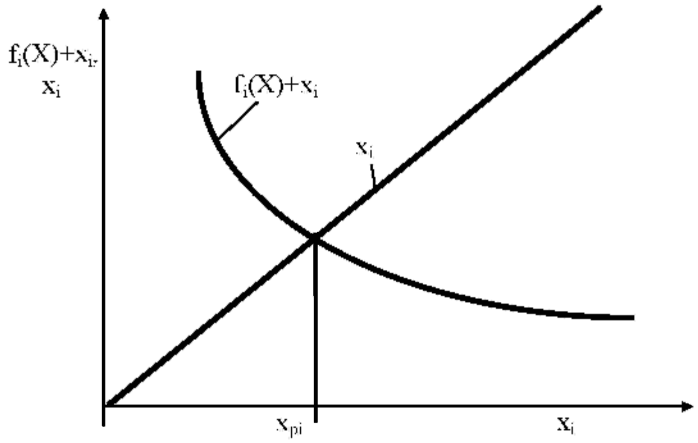


Рисунок 2.1 – Графическая интерпретация метода простых итераций (x_{pi} – решение)

Решение с применением методов поиска экстремума (оптимизации) многопараметрических функций основано на том, что формируются функции вида

$$Z = \sum_{i=1}^{m} (f_i(X))^2 = \min_X \qquad \text{или} \qquad (*)$$

$$Z = \sum_{i=1}^{m} abs(f_i(X)) = \min_X \qquad (**)$$

где $f_i(X) = 0,\ i = \overline{1,m}$.

Сходимость в большой степени определяется тем методом, который будет применен для поиска минимума функции Z. Для решения системы уравнений минимальное значение функции Z должно быть равно нулю. Это условие является необходимым и достаточным. Если Z не равно нулю, то это указывает или на отсутствие решения или на неэффективный метод поиска минимума. При применении свертывания уравнений в функцию (*) быстрее сходимость и ниже точность решения, а для функции (**) наоборот.

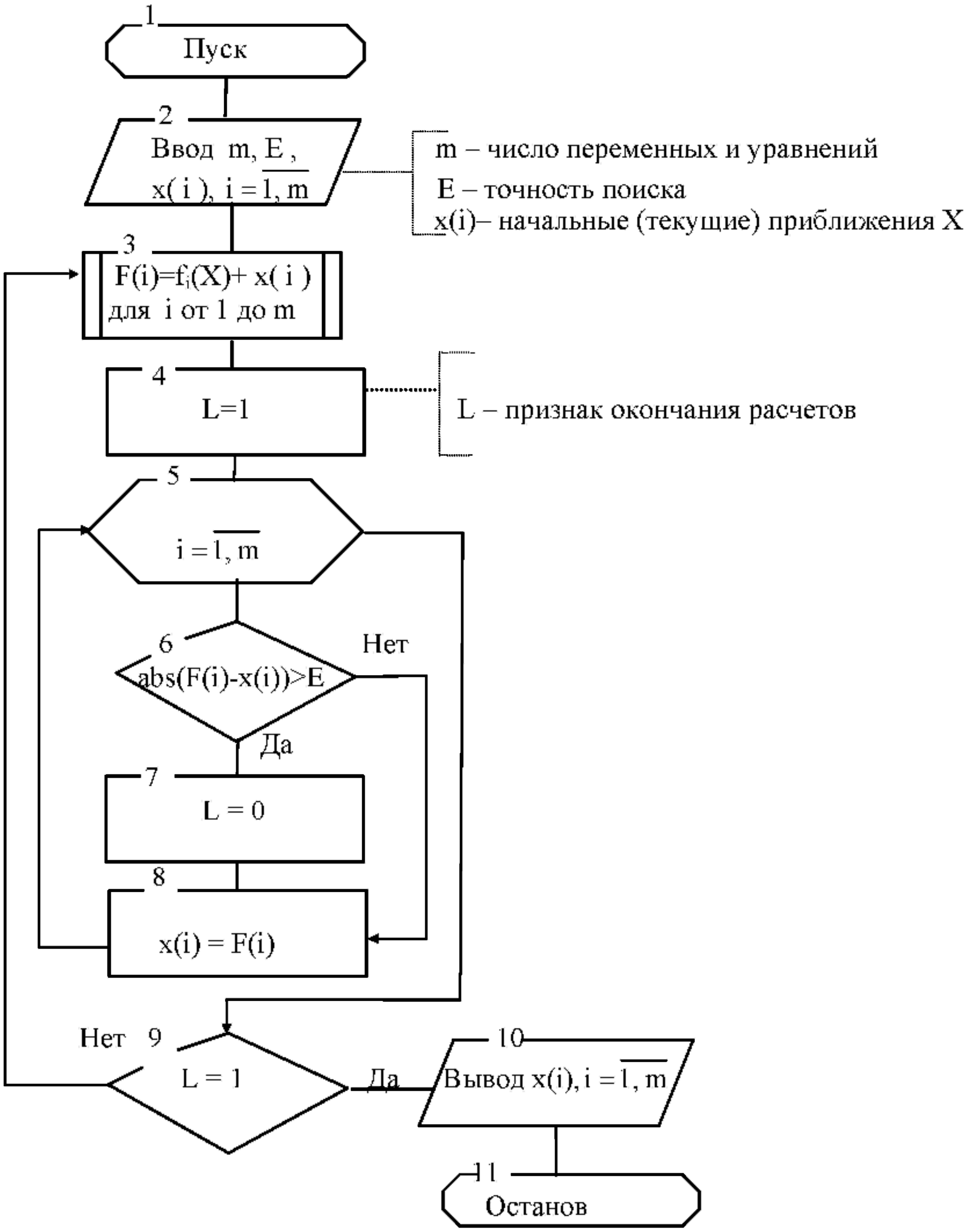


Рисунок 2.2 – Схема алгоритма метода простых итераций

2.1.3. Численное интегрирование

Численное интегрирование – это вычисление значения определенного интеграла

$$S = \int_a^b f(x)\,dx = F(b) - F(a)$$

на основе многократных вычислений значений f(x). Применяется, если функция F(x) не может быть определена аналитически.

Такой интеграл численно равен площади фигуры, ограниченной ординатами в точках a и b, осью абсцисс и линией графика подынтегральной функции f(x) (рисунок 2.3).

Для численного интегрирования отрезок [a, b] разбивается на m частей, к каждой из которых применяется аппроксимация выбранной функцией (прямой, параболой, полиномом и т.п.).

Существует ряд методов численного интегрирования: прямоугольников, модернизированный прямоугольников, трапеций, Ньютона-Котеса (аппроксимация полиномом Лагранжа), Симпсона (аппроксимация параболой), Уэдлля (разбивка каждого из m отрезков на 6 частей), Чебышева (с неравномерным разбиением аргумента), Симпсона (кубатурная аппроксимация), Гаусса (кубатурная аппроксимация), Ромберга, Бодэ и др.

По методу прямоугольников интеграл вычисляется по формуле

$$S = h\sum_{i=1}^{m} f(x(i)),$$

где $x(i) = a + (i-1)\,h$ или $x(i) = a + i\,h$;

$h = (b - a)/m$.

Модернизированный метод прямоугольников отличается правилом выбора расчетных точек x(i):

$$x(i) = a + h/2 + (i-1)\,h.$$

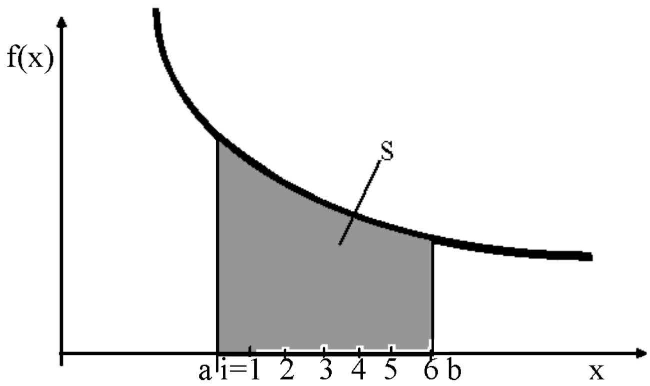


Рисунок 2.3 – Графическая интерпретация численного интегрирования

По методу трапеций

$$S = h\left(f(x(0))/2 + \sum_{i=1}^{m-1} f(x(i)) + f(x(m))/2\right),$$

где $x(i) = a + i\,h$.

Рассмотренные методы являются по сравнению с другими более простыми, но менее эффективными.

В качестве примера более эффективного (точного) метода рассмотрим метод Ньютона-Котеса. Этот метод базируется на применении к каждому i-му из m отрезков разбиения следующей формулы:

$$S_i = h/840(\sum_{j=0}^{6} d(j)\, f(x(j))),$$

где x(j) = x(i) + j h / 6 ;

x(i)= a + (i -1) h ;

d(0)= d(6)= 41;d(1)= d(5)= 216;d(2)= d(4)= 27;d(3)=272.

Значение интеграла S находится суммированием значений S_i :

$$S = \sum_{i=1}^{m} S_i \ .$$

Вариант программной реализации алгоритма последнего метода приводится ниже.

```
10 REM МЕТОД НЬЮТОНА-КОТЕСА
15 CLS
20 INPUT"a";a
30 INPUT"b";b
40 INPUT"m";m
45 DIM d(6)
47 d(0)=41:d(1)=216:d(2)=27:d(3)=272:d(4)=27:d(5)=216:d(6)=82
50 h=(b-a)/m:E=h/6
70 FOR I=1 TO m
72 XT=a+(I-1)*h
75 FOR J=0 TO 6
80 X=XT+J*E:GOSUB 200
90 SI=SI+d(J)*F
100 NEXT J
110 NEXT I
130 SI=SI*H/840
140 PRINT"SI="SI:GOTO 300
200 F=………………..:RETURN
300 END
```

Ошибка вычисления значения интеграла зависит от величины h и производной k-го порядка от интегрируемой функции в точке, где она максимальна (для первых трех методов k=2 и для Ньютона-Котеса k=8).

2.1.4. Вычисление специальных функций

Специальные функции – это такие, которые нельзя выразить аналитически через элементарные функции. Примерами таких функций являются гамма-функция, интегральная функция нормального закона распределения и др.

Значения специальных функций вычисляются в зависимости от их вида одним из следующих методов:

численным интегрированием;

по реккурентным соотношениям;

разложением в ряды;

на основе аппроксимаций.

Гамма-функция точно определяется по формуле

$$\Gamma(x) = \int_{t=0}^{\infty} e^{-t}\, t^{x-1}\, dt, \; x>0.$$

Для гамма-функции справедливы соотношения

$$\Gamma(x+1)=x\,\Gamma(x); \qquad (*)$$
$$\Gamma(1) = 1;$$
$$\Gamma(0{,}5)= \sqrt{\pi};$$
$$\Gamma(n+1)= n!,\; n= 0,1,2,\ldots;$$
$$\Gamma(x)\,\Gamma(-x) = -\pi /(x \sin \pi x). \qquad (**)$$

Рассчитывают с применением формулы Стирлинга или на основе аппроксимации.

Для x > -18 с погрешностью порядка 1E-04 гамма-функция может быть вычислена на основе 20-кратного преобразования следующим образом:

1) $z=21 + x$;

2) по формуле Стирлинга $\Gamma(z) \approx \sqrt{2\pi/z}\; e^{-z}\; z^{z}\left(1+\dfrac{1}{12z}\right)$;

3) последовательное уменьшение значения *z* на единицу до значения *x* по формуле $\Gamma(z) = \Gamma(z+1)/z$.

Преобразование обеспечивает вычисления для отрицательных чисел *x* и с высокой точностью при их малых значениях.

Пример программной реализации метода:

```
cls: input "Введите x= ";x:z=x+21: b=x
g=log(sqr(2*3.141592/z))-z+z*log(z)+log(1+1/12/z)
for i = 1 to 20
b=b*(z-i)
next i
g=exp(g)/b
print "Г("using"####.###";x;:print ")= "using"^.######^^^^";g
end
```

Гамма-функция на основе коррекции формулы Стирлинга вычисляется по формуле

$$\Gamma(x) = \sqrt{2\pi/x}\; e^{-x}\; x^{x}\, H(x),$$

где $x > 1.0$;

$$H(x) = 1 + \sum_{i=1}^{4} 1/(a_i\, x^{i}),$$

где $a_1= 12$; $a_2= 288$; $a_3= -139/51840$; $a_4= 571/2488320$.

При $0<x<1$ значение $\Gamma(x)$ с целью повышения точности находится с использованием формулы (*)

$$\Gamma(x) = \Gamma(x+1)/x.$$

Если x<0, то гамма-функция вычисляется на основе формулы (**) как

$$\Gamma(x) = -\pi/(z\,\Gamma(z)\sin \pi z)\ ,$$

где z=abs(x).

Ниже приведен пример программы на основе коррекции формулы Стирлинга:

```
cls:input "x";x
if x>1 then z=x:gosub pp:goto kon
if x>0  then z=x+1:gosub pp:g3=g3/x:goto kon
if x<=-1 then z=abs(x):gosub pp:goto 10
if x<0 then z=abs(x)+1:gosub pp:z=abs(x):g3=g3/007A
10 g3=-3.141592/z/sin(3.141592*z)/g3
goto kon
pp:
h=1+1/12/z+1/(288*z^2)-139/(51840*z^3)+571/(2488320*z^4)
g3= sqr(2*3.141592/z)*exp(-z)*z^z*h
return
kon:
print "Γ("using"####.###";x;:print ")= "using"^.######^^^^";g3
end
```

На основе аппроксимации определение Γ(z+1) для значений z от 0 до 1 может производиться с использованием степенного полинома

$$\Gamma(z+1) = 1 + \sum_{i=1}^{8} b_i\, z^i 10^{-8}\ ,$$

где b_1= -57719165, b_2= 98820589, b_3= -89705694, b_4= 91820688, b_5= -75670408, b_6= 48219934, b_7= -19352782, b_8=3586835.

Для расчета гамма-функции по аппроксимации необходимо вычислить гамма-функцию от абсолютной величины дробной части заданного аргумента и затем на основе использования выражения (*) и, при необходимости, выражения (**) найти значение гамма-функции исходного числа.

Для вычислений **интегральной функции нормального закона** распределения (рисунок 2.4) применяются численное интегрирование или аппроксимации.

Для вычисления F(x) = γ с точностью 0,0001 на основе численного интегрирования интервал интегрирования необходимо принимать от a-3,9σ до x,

где x – значение аргумента;

a и σ – параметры функций нормального закона распределения: a = $x_м$; σ = s,

$x_м$ –оценка математического ожидания случайной величины;

s – оценка среднеквадратического отклонения случайной величины.

Вычисление значения интегральной функции нормального распределения F(x) = γ на основе аппроксимации возможно по следующему алгоритму:

1) $z = (x- a)/\ \sigma$,

2) $P(abs(z)) = 1 - (1 + \sum_{i=1}^{6} c_i\, z^i)^{-16} / 2$,

где c_1= 49867347E-09; c_2= 21141006E-09; c_3= 32776261E-10;
c_4= 38004E-09; c_5= 48891E-09; c_6= 5383E-09;

3) $$F(x) = \gamma = \begin{cases} P(z), \text{ если } z > 0 \\ 1 - P(abs(z)), \text{ если } z < 0 \end{cases}.$$

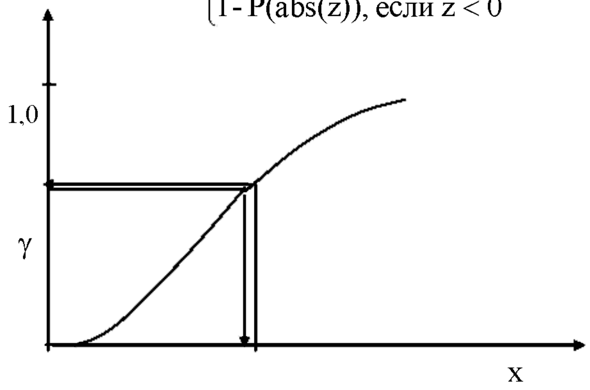


Рисунок 2.4 – Функция распределения (нормальный закон)

Нахождение **для нормального закона распределения по значению интегральной функции γ значения аргумента $x=F^{-1}(\gamma)$** возможно на основе аппроксимации по следующему алгоритму:

1) $$P = \begin{cases} \gamma, \text{ если } \gamma > 0.5 \\ 1 - \gamma, \text{ если } \gamma < 0.5 \end{cases}$$

2) $t = (\ln(1/P))^{0.5}$;

3) $$w = t - \frac{\sum_{i=0}^{2} a_i t^i}{\sum_{i=0}^{3} b_i t^i}$$

a_0 = 2.515517; a_1 = 0.802853; a_2 = 0.010328;
$в_0$ = 1.0; $в_1$ = 1.432788; $в_2$ = 0.189269; $в_3$ = 0.001308;
4) z = - w, если $\gamma \geq 0.5$, иначе z = w
5) $x = a + \sigma z$.

2.1.5. Сортировка чисел (символов)

При обработке числовой или символьной информации может требоваться или быть эффективна ее предварительная сортировка.

Наиболее часто используются следующие методы сортировки: по индексам, BUBBLE ("пузырька") и SHELL (Шелла). Наиболее простой из них – первый, наиболее эффективный в общем случае – третий и высокоэффективный для сортировки незначительно измененных ранее сортированных массивов – второй.

Ниже приводятся алгоритмы и программная реализация этих методов. Для данных, заданных в строке 95 программ (22 числа), число сравнений, необходимых для выполнения сортировки чисел, составляет при применении метода по индексам – 231, метода BUBBLE – 169 и метода SHELL – 105.

Алгоритм и программа метода сортировки по индексам приведена на рисунке 2.5.

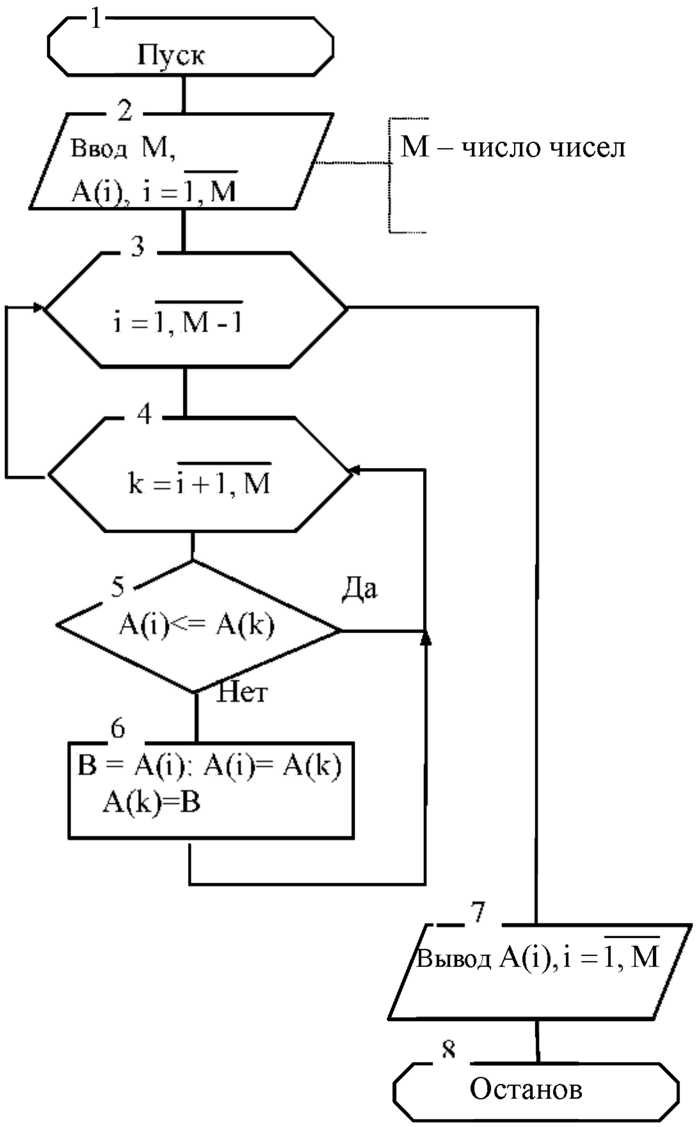


Рисунок 2.5 – Алгоритм программы метода сортировки по индексам

Программа сортировки по индексам

```
10 CLS:PRINT"СОРТИРОВКА ПО ИНДЕКСАМ"
20 DEFINT I-N: INPUT"ЧИСЛО ЧИСЕЛ";M:DIM A(M)
25 FOR I=1 TO M:READ A(I):NEXT I
30 FOR I=1 TO M-1
40 FOR K=I+1 TO M
50 IF A(I)<=A(K) THEN 70
60 B=A(I):A(I)=A(K):A(K)=B:GOTO 70
70 NEXT K
80 NEXT I
90 FOR I=1 TO M:PRINT A(I):NEXT I:GOTO 100
95 DATA 44,12,15,4.8.79.11.14.78.22.33.2,1.4.5,7.8,6,1,4,5,6
100 END
```

Число проходов блока 5 составляет ровно M (M-1)/2.

По методу "пузырька" в отличие от метода сортировки по индексам число необходимых сравнений для получения отсортированных чисел зависит от исходной последовательности чисел и равно, например, для ранее отсортированных чисел М-1. Ниже приведены алгоритм (рисунок 2.6) и программа для метода "пузырька".

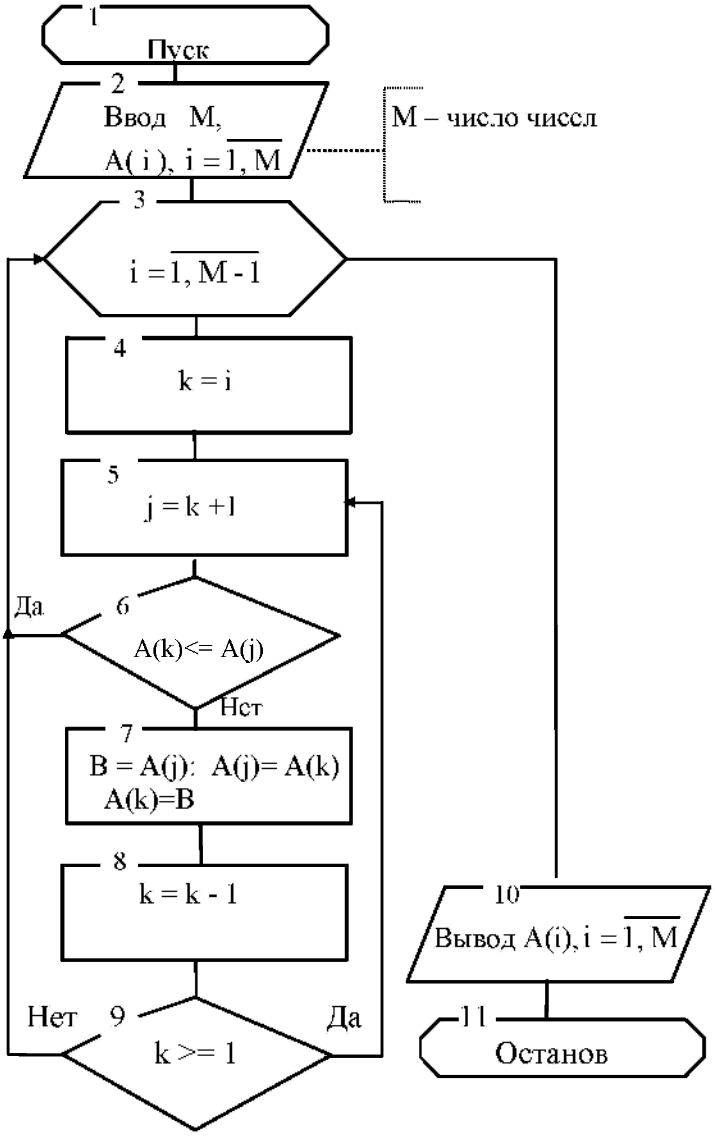


Рисунок 2.6 – Алгоритм сортировки по методу "пузырька"

Программа сортировки по методу "пузырька"

```
10 CLS:PRINT"СОРТИРОВКА ПО BUBBLE"
15 DEFINT I-M: INPUT"ЧИСЛО ЧИСЕЛ";M:DIM A(M)
25 FOR I=1 TO M:READ A(I):NEXT I
30 FOR I=1 TO M-1
40 K=I
44 J=K+1
50 IF A(K)<=A(J) THEN 80
60 B=A(K):A(K)=A(J):A(J)=B:K=K-1
65 IF K>=1 THEN 44
80 NEXT I
90 FOR I=1 TO M:PRINT A(I):NEXT I:GOTO 100
95 DATA 44,12,15,4,8,79,11,14,78,22,33,2,1,4,5,7,8,6,1,4,5,6
100 END
```

Ниже приведены алгоритм (рисунок 2.7) и программа сортировки по методу Шелла.

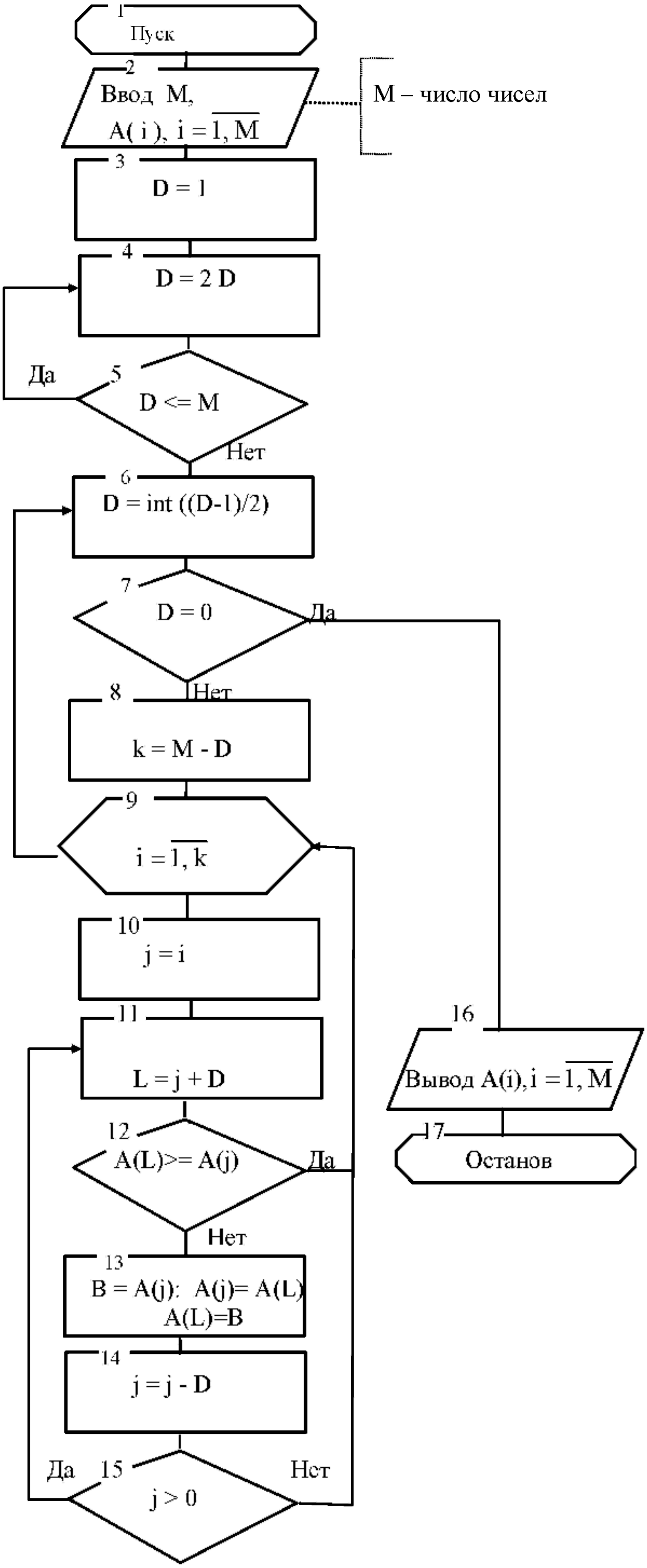


Рисунок 2.7 – Алгоритм сортировки по методу Шелла

Программа сортировки по методу Шелла

```
10 ‘СОРТИРОВКА SHELL
20 DEFINT I-M: INPUT "M";M:DIM A(M)
25 FOR I=1 TO M:READ A(I):NEXT I
30 D=1
35 D=D*2
40 IF D<=M THEN 35
45 D=INT((D-1)/2)
50 IF D=0 THEN 90
55 K=M-D
60 FOR I=1 TO K
65 J=I
70 L=J+D
75 IF A(L)>=A(J) THEN 88
80 X=A(J):A(J)=A(L):A(L)=X:J=J-D
85 IF J>0 THEN 70
88 NEXT I
89 GOTO 45
90 FOR I=1 TO M:PRINT A(I):NEXT I
95 DATA 44,12,15,4,8,79,11,14,78,22,33,2,1,4,5,7,8,6,1,4,5,6
100 END
```

2.2. Стохастические модели

2.2.1. Исследование распределения случайных величин

Случайная (стохастическая) величина может быть одной двух видов – дискретная или непрерывная.

Распределение случайных величин подчиняется определенным закономерностям, называемым законами распределения. Примерами законов распределения являются:

для дискретных – биномиальный, отрицательный биномиальный, Пуассона, гипергеометрический, Паскаля и его частный случай геометрический (Фарри), дискретный равномерный;

для непрерывных – нормальный, логарифмически нормальный, экспоненциальный (показательный), равномерный, Эрланга, Релея, Вейбулла.

Для практических целей находят применение кроме базовых законов распределения их усеченные и сдвинутые (со смещением) варианты.

Усеченные варианты законов распределения ограничивают интервал варьирования случайной величины слева или справа или слева и справа. Если функция распределения базового (неусеченного) закона имеет вид

$$F(x) = \int_{-\infty}^{x} f(x)\ dx,$$

то функции усеченного варианта этого же закона представляют следующие выражения

$$f_у(x) = \frac{1}{1 - F_н - F_в} f(x) \quad ;$$

$$F_у(x) = \frac{1}{1 - F_н - F_в} \int_{x_н}^{x} f(x)\ dx = \frac{F(x) - F_н}{1 - F_н - F_в} \quad \text{при } x_н \le x \le x_в,$$

$$F_н = \int_{-\infty}^{x_н} f(x)\ dx\ ;\ F_в = \int_{x_в}^{\infty} f(x)\ dx\ ,$$

где F(x), f(x) и $F_у(x)$, $f_у(x)$ – функции распределения (интегральные) F(x) и $F_у(x)$ и функции плотности распределения (дифференциальные) f(x) и $f_у(x)$ переменной x соответственно для базового и усеченного законов;

$x_н$ – нижняя (левая) граница усечения;

$x_в$ – верхняя (правая) граница усечения.

Пример графиков функций базового и усеченного распределений приведен на рисунке 2.8.

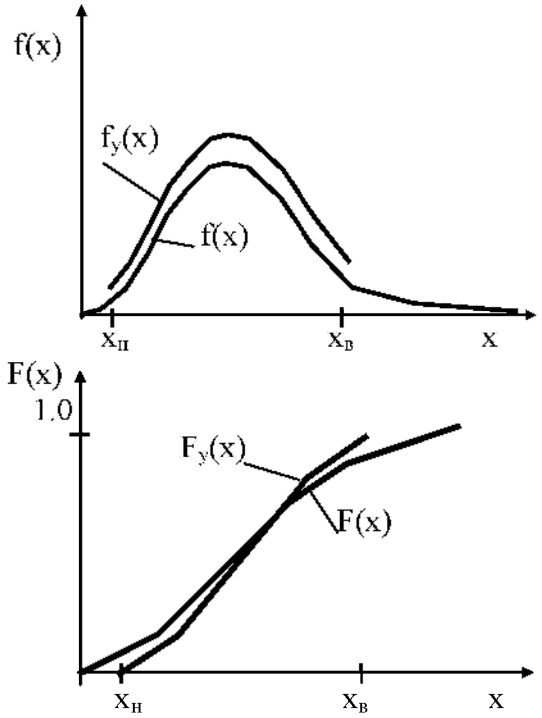


Рисунок 2.8 – Графики базового и усеченного законов распределения

Сдвинутое (смещенное) распределение характеризуется тем, что в отличие от базового, оно формируется не от начального значения $x_н$ для базового закона (например, нуля), а от минимально возможного значения $x_с$ ($x_н<x_с$). Сдвиг распределения не изменяет его форму, если базовое теоретическое распределение, например нормальное, определено на интервале от минус бесконечности.

Общий вид функции распределения со сдвигом имеет вид

$$F_c = \int_{x_c}^{x} f_c(x)\ dx,$$

где f_c (x) – функция плотности закона распределения со сдвигом.

Функция плотности закона распределения со сдвигом $f_c(x)$ отличается от функции базового закона f(x) тем, что в ней величина x заменяется x - x_c, а значения ее параметров определяются по зависимостям для базового закона распределения, в которые вместо оценки математического ожидания $x_м$ подставляется величина $x_м$ - x_c.

Например, для экспоненциального закона распределения имеем (рисунок 2.9):

для базового варианта закона

$$f(x) = \lambda\ \exp(-\lambda\, x);$$
$$F(x) = 1 - \exp(-\lambda\, x);$$
$$\lambda\ = 1/x_м;\ x > 0;$$

для сдвинутого

$$f_c(x) = \lambda_c \exp(-\lambda_c (x - x_c));$$
$$F_c(x) = 1 - \exp(-\lambda_c (x - x_c));$$
$$\lambda_c = 1/(x_м - x_c);\ x \ge x_c,$$

где $x_м$ – оценка математического ожидания случайной величины.

Математическая обработка выборки случайной величины производится с целью определения закономерностей изучаемого процесса (явления).

Для ее исследования необходимо сделать выборку из генеральной совокупности. Наблюдения случайной величины должны проводится в одинаковых условиях. Исследуемая совокупность должна быть однородной. Выборка должна быть репрезентативной (представительной).

Необходимый размер выборки, обеспечивающий оценку параметров распределения с заданной относительной точностью ε и вероятностью γ, зависит от закона распределения и его параметров.

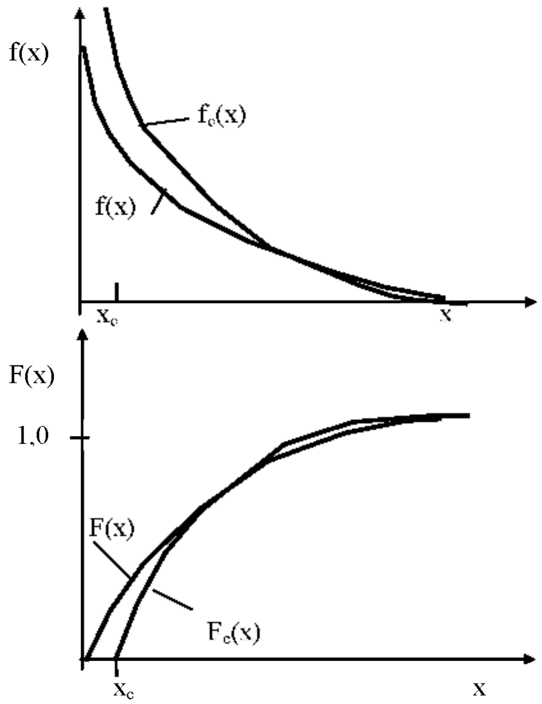


Рисунок 2.9 – Графики базового и смещенного (сдвинутого) законов распределения

Например, для нормального закона распределения размер выборки n определяется выражением

$$n > \frac{t^2_{1-\gamma,n-1}\, v^2}{\varepsilon^2} \approx \frac{U^2_{1-\gamma}\, v^2}{\varepsilon^2},$$

где $t_{1-\gamma,n-1}$ – квантиль распределения Стьюдента при вероятности 1-γ и числе степеней свободы n-1;

$U_{1-\gamma}$ – квантиль нормального закона распределения при вероятности 1-γ;
v – коэффициент вариации случайной величины;

ε – заданная относительная точность оценки математического ожидания случайной величины.

Для определения закономерностей распределения случайных величин рассчитываются характеристики эмпирического распределения, выдвигается гипотеза о теоретическом законе и находятся значения его параметров, производится оценка согласованности теоретического и эмпирического распределений.

Одним из возможных алгоритмов расчета характеристик **эмпирического распределения** непрерывной случайной величины является следующий:

а) по результатам наблюдения (замеров) необходимо получить заданное число n значений исследуемого параметра для процесса, явления, предмета;

б) составить интервальные статистические ряды распределения частот и частостей по значениям случайной величины:

1) найти в выборке минимальное X_{min} и максимальное X_{max} значения случайной величины и размах варьирования $X_p=X_{max}-X_{min}$;

2) определить число интервалов N разбиения случайной величины

$$N_п = 1 + int(3.32 \lg n);$$

$$N = \max (N_п; 5),$$

где n – размер выборки случайной величины;

3) рассчитать длину интервала h

$$h = X_p / N ;$$

4) определить границы X_j (верхнюю), X_{j-1} (нижнюю) и середину X_{cj} каждого j-го интервала распределения случайной величины ($j = \overline{1, N}$)

$$X_j = X_{min} + j\ h\ ;\ X_{j-1} = X_{min} + (j-1)\ h;$$

$$X_{cj} = (X_{j-1} + X_j)/2\ .$$

5) подсчитать число попаданий случайной величины в каждый j-й интервал (частоты M_j), для чего пересмотреть все числа x_i ($i = \overline{1, n}$) относительно границ интервалов:

$$M_j = M_j + 1\ , \text{если}\ X_{j-1} \le x_i < X_j\ \text{при}\ j = \overline{1, N-1}\ ;$$

$$M_j = M_j + 1\ , \text{если}\ X_{j-1} \le x_i \le X_j\ \text{при}\ j = N;$$

6) определить частости (эмпирические вероятности) $p_{эj}$ попадания значений случайной величины в каждый из интервалов путем деления соответствующих частот на объем выборки n, т.е. $p_{эj} = M_j / n$. Сумма всех частот равна объему выборки

$$\sum_{j=1}^{N} M_j = n\ ,$$

а сумма частостей $p_{эj}$ соответственно равна единице.

7) представить интервальные статистические ряды в виде массивов

Номер интервала j	1	2		N
Нижняя граница X_{j-1}	X_0	X_1		X_{N-1}
Верхняя граница X_j	X_1	X_2		X_N
Середина интервала X_{cj}	X_{c1}	X_{c2}		X_{cN}
Частоты M_j	M_1	M_2		M_N
Частости $p_{эj}$	$p_{э1}$	$p_{э2}$		$p_{эN}$

в) Построить гистограмму или полигон эмпирического распределения

Для построения гистограммы по оси x откладывают границы интервалов значений случайной величины и для каждого из интервалов строится прямоугольник, высота которого равна частному от деления частости для данного интервала на величину интервала: $f_{эj}= p_{эj}/h$, где $f_{эj}$ – эмпирическая функция плотности вероятности. Полигон строится также по значениям $f_{эj}$, но на серединах интервалов в виде ломаной линии (рисунок 2.10).

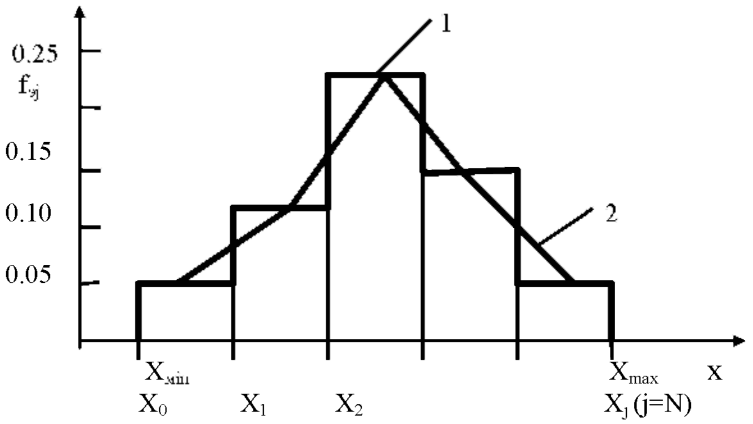


Рисунок 2.10 – Гистограмма (1) и полигон (2) эмпирического распределения (пример)

г) определить значения эмпирической функции распределения (кумулятивной кривой) и построить ее график (рисунок 2.11)

$$F_{эj} = \sum_{k=1}^{j} p_{эk} .$$

При этом $F_{э0} = 0$ (j=0).

д) Определить числовые характеристики выборки: начальные μ_k и центральные статистические μ_{ck} моменты k -го порядка и рассчитываемые через них параметры распределения – оценки математического ожидания x_m, выборочной дисперсии s^2, среднеквадратического (стандартного) отклонения s, коэффициентов вариации V и сдвинутого (смещенного) распределения V_c, коэффициентов асимметрии A и эксцесса E

$$\mu_k = \frac{1}{n}\sum_{i=1}^{n} x_i^k \text{ или } \mu_k = \frac{1}{n}\sum_{j=1}^{N} X_{cj}^k M_j ;$$

$$x_м = \mu_1 \text{, т.е. } x_м = \frac{1}{n}\sum_{i=1}^{n} x_i ;$$

$$\mu_{ck} = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_м)^k \text{ или } \mu_{ck} = \frac{1}{n}\sum_{j=1}^{N}(X_{cj} - x_м)^k M_j ;$$

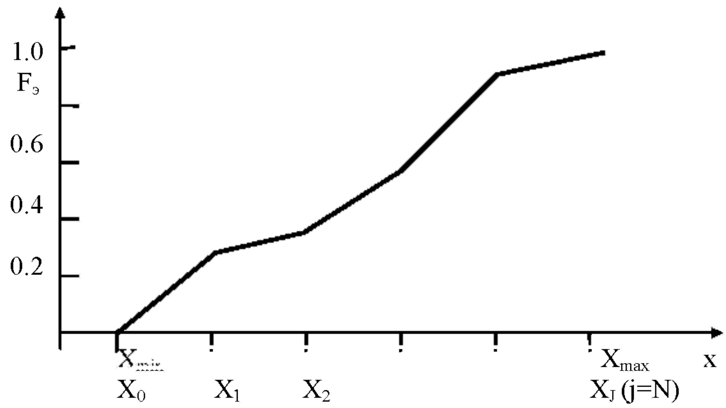


Рисунок 2.11 – График эмпирической функции распределения (пример)

$$s^2 = k_c \mu_{c2} \text{ или } s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - x_м)^2 \text{ или } s^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - nx_м^2\right);$$

$$s = \sqrt{s^2} ;$$

$$V = s / x_М ;$$

$$V_c = s / (x_М - x_c);$$

$$A = k_a \mu_{c3} / \mu_{c2}^{1,5} ;$$

$$E = k_e \mu_{c4} / \mu_{c2}^{2} - 3 .$$

Поправочные коэффициенты k_c, k_a и k_e служат для получения несмещенной оценки параметров и определяются по формулам:

$k_c = n / (n-1)$;
$k_a = n^2 / (n^2 - 3 n + 2)$;
$k_e \approx 1$.

Гипотеза о законе распределения исследуемой случайной величины выдвигается на основании учета следующей информации:

1) условия и факторы, влияющие на процесс формирования значений случайной величины;

2) форма гистограммы (полигона) эмпирического распределения;

3) значения коэффициента вариации V или V_c для сдвинутого распределения

Закон распределения случайной величины	Коэффициент вариации V или V_c	
	пределы изменения	среднее значение
Нормальный	0.08 - 0.40	0.25
Логнормальный	0.35 - 0.80	0.68
Экспоненциальный	0.70 - 1.30	1.0

4) значения выборочных коэффициентов асимметрии A и эксцесса E. Нормальный закон распределения допустимо выбирать в случае, если выполняются неравенства abs(A)<3S_a и abs(E)<3S_e ,

$$S_a = \sqrt{\frac{6(n-1)}{(n+1)(n+3)}};$$

$$S_e = \sqrt{\frac{24n(n-2)(n-3)}{(n-1)^2(n+3)(n+5)}}.$$

Для выбранного закона распределения определяются значения его параметров, записываются выражения для функции плотности вероятности f(x) и функции распределения F(x) исследуемой случайной величины, строятся графики функций f(x) и F(x). Для некоторых законов распределения ниже приведены вид функции плотности вероятности f(x) и функции распределения F(x), а также зависимости для вычисления значений параметров.

Нормальный закон распределения

$$f(x) = 1/(\sigma\sqrt{2\pi})\exp(-(x-a)^2/(2\sigma^2)),$$

где a и σ – параметры закона распределения; π = 3.1415... ;

$$F(x) = \int_{-\infty}^{x} 1/(\sigma\sqrt{2\pi})\exp(-(x-a)^2/(2\sigma^2))\,dx;$$

точечные оценки параметров нормального закона распределения равны: $a = x_м$, $\sigma = S$.

Логарифмически нормальный закон распределения

$$f(x) = 1/(x\sigma_1\sqrt{2\pi})\exp(-(\ln x - a_1)^2/(2\sigma_1^2)),\ x>0;$$

$$F(x) = \int_{0}^{x} 1/(x\sigma_1\sqrt{2\pi})\exp(-(\ln x - a_1)^2/(2\sigma_1^2))\,dx;$$

точечные оценки параметров закона распределения:

$$\sigma_1 = \sqrt{\ln(1+(S/x_м)^2)};$$

$$a_1 = \ln x_м - \sigma_1^2/2.$$

Логарифмически-нормальный закон можно описать функцией плотности вероятности нормального распределения, если вместо значений **x** использовать их логарифмы.

Экспоненциальный закон распределения

$$f(x) = \lambda\exp(-\lambda x),\ x \ge 0;$$

$$F(x) = 1 - \exp(-\lambda x),\ x \ge 0;$$

точечная оценка параметра закона распределения $\lambda = 1/x_м$.

Закон равномерной плотности

$$f(x) = 1/(b-a),\ a \le x \le b;$$

$$F(x) = (x-a)/(b-a),\ a \le x \le b;$$

точечная оценка параметра закона распределения:

$$a = x_м - S\sqrt{3};$$

$b = x_м + S\sqrt{3}$.

Закон распределения Релея

$$f(x) = x/\sigma_р^2 \exp(-x^2 / 2\sigma_р^2) \ , x \ge 0;$$
$$F(x) = 1 - \exp(-x^2 / 2\sigma_р^2) \ , x \ge 0;$$

точечная оценка параметра закона распределения:

$$\sigma_р = x_м \sqrt{2/\pi} \ .$$

Закон распределения Эрланга (гамма-распределение)

$$f(x) = \lambda_э^k x^{k-1} /(k-1)! \exp(-\lambda_э x) \quad , x \ge 0;$$
$$F(x) = 1 - \exp(-\lambda_э x) \sum_{i=0}^{k-1} (\lambda_э x)^i / i! \quad , x \ge 0;$$

точечная оценка параметров закона распределения:
$k' = x_м^2/s^2$ и по k' принимается k как ближайшее целое (k=1, 2, 3,...); $\lambda_э = k/ x_м$.

Закон распределения Вейбулла

$$f(x) = b / \lambda_в x^{b-1} \exp(-x^b \lambda_в) \quad , x \ge 0;$$
$$F(x) = 1 - \exp(-x^b / \lambda_в) , x \ge 0;$$

точечная оценка параметров закона распределения:

$$\lambda_в^{1/b} \Gamma(1+1/b) = x_м ;$$
$$\lambda_в^{1/b} \sqrt{\Gamma(1+2/b) - (\Gamma(1+1/b))^2} = s .$$

Для усеченных и смещенных законов распределения вид функций и расчет параметров находятся в соответствии с ранее приведенными соотношениями.

Укрупненный алгоритм программы для исследования случайных величин приведен на рисунке 2.12.

Оценка **согласованности эмпирического и теоретического** распределений может производиться по критериям Колмогорова, Пирсона, Романовского и Мизеса-Смирнова.

По критерию Колмогорова, Пирсона и Романовского оценка считается обоснованной при выборке случайной величины не менее 100 и по критерию Мизеса-Смирнова – не менее 50. При применении критерия Колмогорова для меньшего размера выборки необходимо использовать заранее известные значения математического ожидания и среднеквадратического отклонения случайной величины, а не их выборочные оценки.

Ниже приводится порядок проверки выдвинутой гипотезы о законе распределения случайной величины по различным критериям.

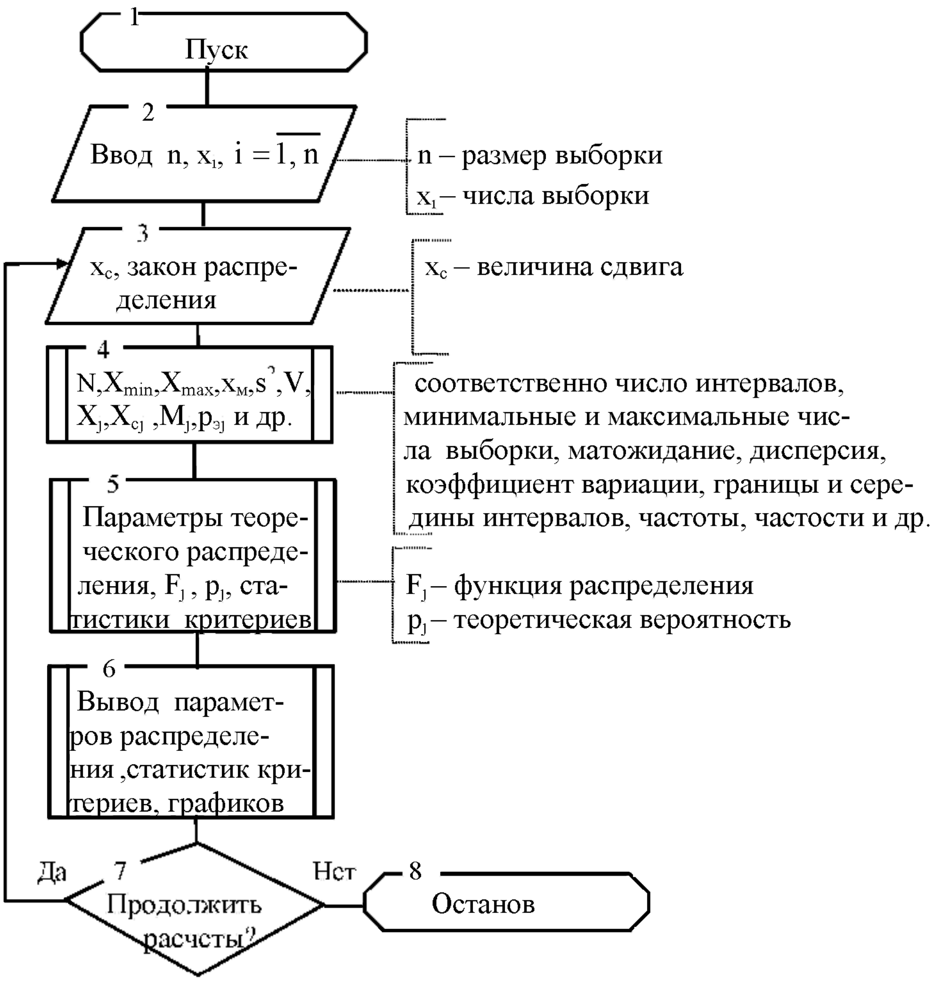


Рисунок 2.12 – Укрупненный алгоритм программы исследования случайных величин

1) Критерий хи - квадрат (Пирсона)

Для его применения вычисляют статистику хи - квадрат по формуле

$$\chi^2 = \sum_{j=1}^{N_J} (M_J - np_J)^2/(np_J),$$

где p_j – теоретическая вероятность попадания случайной величины в j-й интервал, которая распределена по выбранному закону распределения с найденными оценками параметров;

np_j – теоретическая частота попадания случайной величины в j-й интервал;

N_o – число интервалов с учетом их объединения для расчета статистики критерия Пирсона.

Для расчета статистики критерия Пирсона интервалы рекомендуется объединять на концах таким образом, чтобы $M_j > 5$ или $np_j > 10$. Однако число интервалов N_o должно быть не менее k+2, где k – число параметров рассматриваемой теоретической функции распределения (например, для нормального закона k = 2).

Вероятность p_j определяется по формуле

$$p_j = p(X_{j-1} < x < X_j) = F(X_j) - F(X_{j-1}),$$

где F(x) – значение функции распределения в точке x .

Значения F(x) определяются по интегральной функции рассматриваемого закона распределения. Для нормального закона распределения значение функции распределения в точке X_j может быть определено по таблицам или на основе аппроксимации по ранее рассмотренному алгоритму или на основе численного интегрирования. Для логарифмически нормального закона распределения определение теоретических вероятностей производится аналогично как для нормального, но только относительно логарифмов x.

Вычисленное значение критерия χ^2 необходимо сравнить с табличным $\chi^2_{\gamma,r}$ для заданного уровня значимости γ и числа степеней свободы r. Уровень значимости γ представляет собой вероятность отклонения верной гипотезы. Проверку соответствия теоретического и эмпирического распределений рекомендуется проводить при $\gamma = 0.05 - 0.1$. При больших значениях γ выше требования к согласованности распределений.

Число степеней свободы определяется по формуле $r = N_o - k - 1$.

После того, как по таблице квантилей распределения хи-квадрат при заданных γ и r найдено $\chi^2_{\gamma,r}$, проверяется условие $\chi^2 \le \chi^2_{\gamma,r}$. Если условие выполняется, то гипотеза о распределении случайной величины по рассматриваемому теоретическому закону не отклоняется.

Табличные значения критерия Пирсона приведены в таблице 2.1.

Таблица 2.1 - Табличные значения критерия Пирсона

Число степеней свободы		1	2	3	4	5	6	7	8	9
Уровень значимости	$\gamma = 0.05$	3.84	5.99	7.82	9.49	11.1	12.6	14.1	15.5	16.9
	$\gamma = 0.10$	2.71	4.61	6.25	7.78	9.24	10.6	12.0	13.4	14.7

2) Критерий Романовского

Критерий Романовского является производным от критерия χ^2, но не требует использования табличных значений распределения Пирсона.

По данному критерию для проверки гипотезы рассчитывается статистика R

$$R = (\chi^2 - r)/\sqrt{2r} .$$

Если параметр R > 3.0, то гипотеза о согласованности эмпирического и теоретического распределений отвергается, а если $R \le 3.0$, то считается, что нет оснований отклонять гипотезу.

3) Критерий Колмогорова

Статистика критерия Колмогорова определяется максимальным отклонением по границам интервалов между теоретической $F(X_j)$ и эмпирической $F_э(X_j)$ функциями распределения (рисунок 2.13).

Для этого для каждого значения X_j вычисляется модуль разности между эмпирической и теоретической функциями распределения abs(F (X_j)- $F_э$ (X_j)) и затем из всех рассчитанных значений находится максимальная величина

$$D = \max_j \text{abs}\,(F(X_j) - F_э(X_j))$$

и вычисляется значение статистики критерия Колмогорова

$$\lambda = D\sqrt{n} .$$

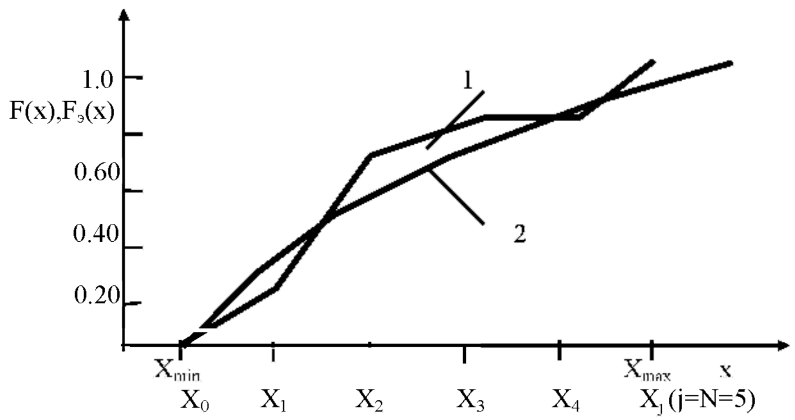


Рисунок 2.13 – Эмпирическая (1) и теоретическая (2) функции распределения

Полученное значение статистики λ необходимо сравнить с табличным. При принятых значениях уровня значимости γ (0.1–0.2) по таблице определяют критическое значение λ_γ и проверяют условие

$$\lambda < \lambda_\gamma .$$

При выполнении условия гипотеза о распределении случайной величины по предполагаемому теоретическому закону по критерию Колмогорова может быть принята, в противном случае – отклонена. При больших значениях γ требования к согласованности распределений повышаются.

Табличные значения критерия Колмогорова приведены в таблице 2.2.

Таблица 2.2 - Табличные значения критерия Колмогорова

Уровень значимости γ	0.40	0.30	0.20	0.10	0.05
Значение критерия λ_γ	0.89	0.97	1.07	1.22	1.36

4) Критерий Мизеса-Смирнова

Критерий Мизеса-Смирнова в отличие от критерия Колмогорова, который основывается на максимуме абсолютной величины разности между эмпирической и теоретической функциями распределения, использует статистику в виде суммы взвешенных через весовую функцию квадратов разностей между эмпирической и теоретической функциями по всем наблюдаемым значениям случайной величины

$$\varpi^2 = \int_{x_{min}}^{x_{max}} (F_э(x) - F(x))^2 \, g(F(x))\, d\,F(x) \quad ,$$

где F(x) – теоретическая функция распределения;

$F_э(x)$ – эмпирическая функция распределения;

g(F(x)) – весовая функция.

Обычно используют весовые функции двух видов:

$g(F(x)) = 1$, при которой все значения функции распределения обладают одинаковым весом, и

$g(F(x)) = 1/(F(x)(1 - F(x)))$, при которой увеличивается вес наблюдений на концах распределения.

Ниже рассматривается критерий при весовой функции второго вида.

После выполнения интегрирования выражение для расчета статистики критерия имеет вид

$$\varpi^2 = -n - 2\sum_{i=1}^{n}((2i-1)/(2n)\ln F(x_i) + (1-(2i-1)/(2n))\ln(1-F(x_i))) ,$$

где x_i – результаты наблюдений, отсортированные по величине ($x_i < x_{i+1}$).

Полученное значение статистики ω^2 сравнивается с табличным значением ω^2_γ. Значение γ принимается на уровне 0.1– 0.2. Табличное значение критерия при γ=0.1 составляет $\omega^2_{\gamma=0.1}$=1.94 и при γ=0.2 – $\omega^2_{\gamma=0,2}$ =1.42. Если рассчитанное значение статистики больше табличного, то гипотеза о согласованности отвергается, и если нет – то принимается.

Компьютерная программа для исследования распределения случайных величин приведена в приложении 2. Она состоит из головной программы и модулей для исследования распределения непрерывных и дискретных случайных величин.

2.2.2. Генерация случайных чисел по различным законам распределения

Генерация случайных чисел основана на том, что интегральная функция распределения F(x) ставит в соответствие любому заданному числу **x** вероятность от 0 до 1. Тогда наоборот некоторому значению F(x), равному, например r , соответствует определенная величина x (рисунок 2.14):

$$x = F^{-1}(r),$$

где F^{-1} – функция, обратная F.

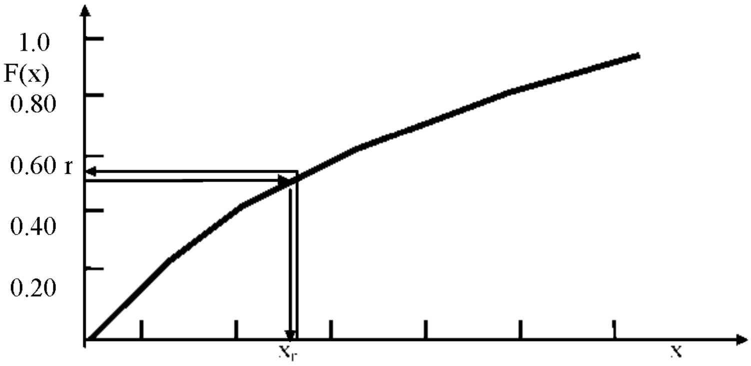


Рисунок 2.14 – Графическая интерпретация получения случайных чисел по заданному закону распределения

Отсюда следует способ формирования случайных чисел с заданным законом распределения, называемый методом обратных функций. Метод реализуется как по функциональным, так и аппроксимирующим зависимостям. При этом значения r должны быть распределены в интервале 0. – 1. случайно по равномерному закону.

Для нормального закона распределения применяется также метод, основанный на центральной предельной теореме теории вероятностей, согласно которой большое число **n** независимых случайных чисел с одним и тем же распределением вероятностей дает нормально распределенные числа с математическим ожиданием, равным сумме этих чисел и геометрической суммой среднеквадратических отклонений:

$$x = \sum_{i=1}^{n} x_i ; \quad s = \sqrt{\sum_{i=1}^{n} s_i^2} .$$

Формулы для получения случайных чисел по некоторым законам распределения приведены в таблице 2.3.

Псевдослучайные равномерно распределенные числа в интервале 0. – 1.0 можно получать по различным алгоритмам или применить стандартные функции и подпрограммы языков программирования – RND (Basic), RANDOM (Pascal), RAN, RANDU (Fortran). Генерируемая последовательность может задаваться с помощью оператора, например RANDOMIZE.

Таблица 2.3 – Получение случайных чисел

Закон распределения	Получение случайных чисел для базового закона	Получение случайных чисел для усеченных ($x_н<x<x_в$) или сдвинутых распределений ($x>x_c$)
Равномерной плотности	$x = a + r\,(b - a)$ $a = x_м - S\sqrt{3}$; $b = x_м + S\sqrt{3}$	–
Нормальный	$x = a + \sigma\sqrt{12/n}\,(\sum_{i=1}^{n} r_i - n/2)$, $n \ge 6$ $a = x_M$; $\sigma = S$	Для усеченного по выражению для базового, если полученное $x_н<x<x_в$, иначе попытка повторяется
Логарифмически нормальный	$x = \exp(a_1 + \sigma_1\sqrt{12/n}\,(\sum_{i=1}^{n} r_i - n/2))$, $n \ge 6$ $\sigma_1 = \sqrt{\ln(1 + (S/x_м)^2)}$ $a_1 = \ln x_м - \sigma_1^2/2$	Для сдвинутого $x = x_c + \exp(a_{1c} + \sigma_{1c}\sqrt{12/n}\,(\sum_{i=1}^{n} r_i - n/2))$, $n \ge 6$ $\sigma_{1c} = \sqrt{\ln(1 + (S/(x_м - x_c))^2)}$ $a_{1c} = \ln(x_м - x_c) - \sigma_{1c}^2/2$
Экспоненциальный	$x = -1/\lambda \ln r$ $\lambda = 1/x_M$	Для сдвинутого $x = x_c - 1/\lambda_c \ln r$ $\lambda_c = 1/(x_M - x_c)$
Релея	$x = \sigma_p\sqrt{-2\ln r}$ $\sigma_p = x_M\sqrt{2/\pi}$	Для сдвинутого $x = x_c + \sigma_{pc}\sqrt{-2\ln r}$ $\sigma_{pc} = (x_M - x_c)\sqrt{2/\pi}$
Вейбулла	$x = \sqrt[b]{-\lambda_в \ln r}$ $\lambda_в^{1/b}\,\Gamma(1 + 1/b) = x_м$; $\lambda_в^{1/b}\sqrt{\Gamma(1 + 2/b) - (\Gamma(1 + 1/b))^2} = s$	Для сдвинутого $x = x_c + \sqrt[bc]{-\lambda_{вс} \ln r}$ $\lambda_{вс}^{1/bc}\,\Gamma(1 + 1/b_c) = x_м - x_c$; $\lambda_{вс}^{1/bc}\sqrt{\Gamma(1 + 2/b_c) - (\Gamma(1 + 1/b_c))^2} = s$
Эрланга	$x = -1/\lambda_э \sum_{i=1}^{k} \ln r_i$ $k = x_м^2/s^2$, k=1, 2, 3,...; $\lambda_э = k/x_м$	Для сдвинутого $x = x_c - 1/\lambda_{эс} \sum_{i=1}^{kc} \ln r_i$ $k_c = (x_м - x_c)^2/s^2$, k_c=1, 2, 3,...; $\lambda_{эс} = k_c/(x_м - x_c)$

Наиболее распространенными способами получения псевдослучайных равномерно распределенных чисел в интервале 0. – 1.0 являются:

мультипликативный;

смешанный;

с использованием числа π;

с использованием тригонометрических функций.

Алгоритм смешанного метода приведен на рисунке 2.15.

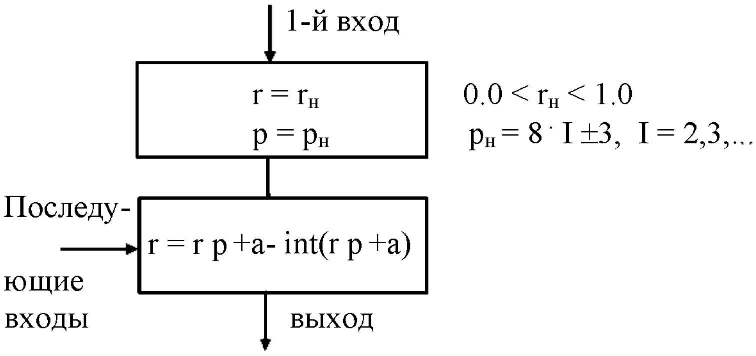


Рисунок 2.15 – Алгоритм смешанного метода

Наиболее часто в качестве a принимается число пи ($a = \pi$).

Мультипликативный метод отличается от смешанного тем, что $a = 0$. В этом случае начальное значение $r_н \neq 0{,}5$.

При большом числе сгенерированных случайных чисел оценка их математического ожидания должна стремится к 0.5 и среднеквадратическое отклонение к $1/(2\sqrt{3})$.

Пример.

Получить зависимость для генерации случайных чисел по экспоненциальному закону распределения.

Интегральная функция экспоненциального распределения имеет вид

$$F(x) = 1 - \exp(-\lambda x)\ ,\ x > 0$$

где λ – параметр распределения ($\lambda = 1/\ x_м$, $x_м$ – оценка математического ожидания случайной величины).

Для получения случайного числа **x** по r приравняем выражение для F(x) и величину r и выразим x:

$$F(x) = 1 - \exp(-\lambda x) = r\ ;$$
$$1 - r = e^{-\lambda x}\quad ;$$
$$\ln(1 - r) = -\lambda x\ ;$$
$$x = -1/\lambda \ln(1 - r) = -x_м \ln r\ .$$

2.2.3. Интервальная оценка параметров и определение интервалов распределения случайных величин

Интервальная оценка параметра распределения случайной величины определяется тем, что с вероятностью γ

$$abs(P - P_м) \leq \delta,$$

где P – точное (истинное) значение параметра;

$P_м$ – оценка параметра по выборке;

δ – точность (ошибка) оценивания параметра P.

Наиболее часто принимают γ от 0.8 до 0.99.

Доверительный интервал параметра $[P_м–\delta, P_м+\delta]$ – это интервал, в который попадает значение параметра с вероятностью γ. Например, на этой основе находится требуемый размер выборки случайной величины, который обеспечивает оценку математического ожидания при точности δ с вероятностью γ. Вид связи определяется законом распределения случайной величины.

Вероятность попадания случайной величины в заданный интервал $[X_1, X_2]$ определяется приращением интегральной функции распределения на рассматриваемом интервале $F(X_2) - F(X_1)$. Исходя из этого, при известной функции распределения можно найти ожидаемое гарантированное минимальное $X_{гн}$ ($x \geq X_{гн}$) или максимальное значение $X_{гв}$ ($x \leq X_{гв}$) случайной величины с заданной вероятностью γ (рисунок 2.16). Первое из них является тем значением, больше которого случайная величина будет с вероятностью γ, а второе – что случайная величина с вероятностью γ меньше этого значения. Гарантированное минимальное значение $X_{гн}$ с вероятностью γ обеспечивается при $F(x)= 1-\gamma$ и максимальное $X_{гв}$ при $F(x)=\gamma$. Таким образом, значения $X_{гн}$ и $X_{гв}$ находятся по выражениям:

$$X_{гн} = F^{-1} (1-\gamma);$$
$$X_{гв} = F^{-1} (\gamma).$$

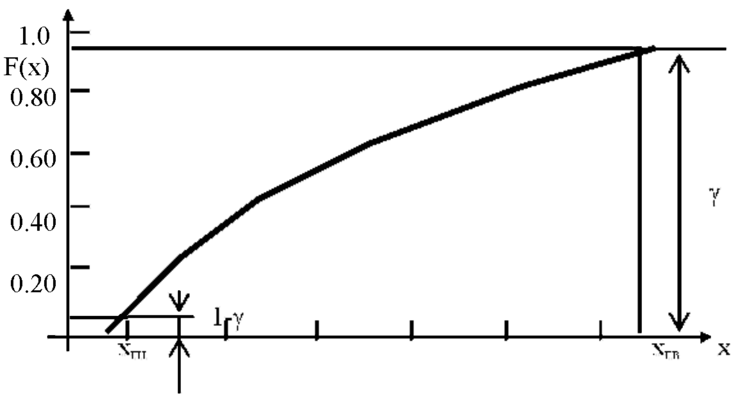


Рисунок 2.16 – Графическая интрепретация определения $X_{гн}$ и $X_{гв}$

Пример.

Случайная величина имеет экспоненциальное распределение с функцией $F(x) = 1 - e^{-0.01x}$.

Требуется найти значения $X_{гн}$ и $X_{гв}$, для которых случайная величина **x** с вероятностью γ=0.95 соответственно больше $X_{гн}$ и меньше $X_{гв}$.

Исходя из того, что $F^{-1} (\alpha) = -1/\lambda \ln(1- \alpha)$ (см.вывод ранее) и $\alpha = 1-\gamma = 0.05$ получаем

$X_{гн} = -1/\lambda \ln(1- \alpha) = -1/\lambda \ln \gamma = -1/0.01 \ln(0,95)=-100 (-.0513) = 5.13$.

Для $X_{гв}$ $\alpha = \gamma = 0.95$ аналогично имеем

$X_{гв} = -1/\lambda \ln(1- \alpha) = -1/\lambda \ln (1- \gamma) = -1/0.01 \ln(1-0.95)=-100 (-2.996)=299.6$.

Для нормального закона распределения значения $X_{гн}$ и $X_{гв}$ могут быть рассчитаны по формулам

$$X_{гн} = x_м + s\, U_{1-\gamma} = x_м - s\, U_\gamma ;$$
$$X_{гв} = x_м + s\, U_\gamma ,$$

где $x_м$ – математическое ожидание случайной величины;

s – среднеквадратическое отклонение случайной величины;

U_γ – односторонняя квантиль нормального закона распределения при вероятности γ.

2.2.4. Исследование статистических зависимостей между случайными величинами

При решении многих задач, в том числе и оптимизационных, возникает необходимость в нахождении модели связи зависимой переменной от различных влияющих на нее факторов (например, расход топлива автомобилем от его полной массы, удельной мощности, типа трансмиссии, удельного расхода топлива двигателем, скорости движения, суммарного дорожного сопротивления, температуры воздуха и других факторов). Такие зависимости называются уравнениями множественной регрессии (функциями отклика). Они описывают статистические зависимости между параметрами систем и процессов и могут быть использованы для обоснования нормативов, отыскания оптимальных решений и прогнозирования развития явлений.

Чтобы математические модели связи отражали объективные закономерности исследуемых процессов и явлений, они должны учитывать их физическую сущность, а также влияние внешних и внутренних факторов. Для этого планируются и проводятся эксперименты и обрабатывается полученная информация.

В общем виде уравнение регрессии имеет вид

$y_т = f(x_1,x_2,...,x_n)$,

где $y_т$ – зависимый параметр;

$x_1,x_2,...,x_n$ – факторы (аргументы), представляющие собой учитываемые (управляемые) переменные, значения которых регистрируются в ходе эксперимента;

n – число учитываемых факторов.

Уравнение регрессии может иметь аддитивный (*) или мультипликативный (**) вид

$$y_т = a_o + \sum_{j=1}^{n} f_j(x_j)\,; \qquad (*)$$

$$y_т = a_o \prod_{j=1}^{n} f_j\,(x_j)\,. \qquad (**)$$

В качестве функций f(x) наиболее часто применяются следующие:

$y = a\,x + b$;

$y = a / x + b$;

$y = x / (a\,x + b)$;

$y = 1 / (a\,x + b)$;

$y = a\,x^2 + b\,x + c$;

$y = a\,x^c + b$;

$y = a \ln(x) + b$;

$y = a \ln(x) + b\,x + c$;

$y = b\,a^x$;

$y = b\,x^{a}$;

$y = b\,a^{(1/x)}$;

$y = 1 / (a \exp(-x) + b)$;

$y = \exp(a\,x + b)$.

При множественной регрессии широко применяются аддитивная линейная

$$y_т = a_o + \sum_{j=1}^{n} a_j x_j$$

или мультипликативная степенная модель

$$y_т = a_o \prod_{j=1}^{n} x_j^{a_j}\,.$$

Компьютерная программа нахождения парных зависимостей между случайными величинами приведена в приложении 3.

Эксперимент может быть пассивным и активным. При пассивном эксперименте фиксируют складывающиеся при протекании реальных процессов значения факторов и зависимой переменной. Активный эксперимент проводится по заранее составленному плану.

При активном эксперименте факторы фиксируются на определенных уровнях (при определенных значениях). Совокупность уровней факторов называют факторным пространством или решеткой планирования. Принятые значения (уровни) факторов называют матрицей планирования. Каждой точке факторного пространства соответствует экспериментальное значение зависимой переменной.

Активный эксперимент может быть полнофакторным или дробнофакторным. При полнофакторном эксперименте рассматриваются все возможные сочетания различных уровней факторов. Число точек (m) решетки планирования полнофакторного эксперимента составляет

$$m = k^{n},$$

где к – число уровней, на которых зафиксирован каждый фактор;
n – число факторов.

Например, при k = 3 и n = 2 получаем m = 9 с планом испытаний по нижеследующей схеме (-1 , 0 и +1 – соответственно нижний, средний и верхний уровни значений факторов):

-1,+1	0,+1	+1,+1
-1,0	0,0	+1,0
-1,-1	0,-1	+1,-1

При k=3 и n=3 получаем нижеприведенный план эксперимента (- , 0 и + соответственно нижний, средний и верхний уровни значений факторов):

Факторы	**1**	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
	2	0	0	0	-	-	-	+	+	+	0	0	0	-	-	-	+	+	+	0	0	0	-	-	-	+	+	+
	3	-	0	+	-	0	+	-	0	+	-	0	+	-	0	+	-	0	+	-	0	+	-	0	+	-	0	+

Дробнофакторный эксперимент предусматривает проведение опытов в определенных точках, назначенных по специальным алгоритмам. В этом случае число опытов меньше чем при полнофакторном эксперименте. Примером дробнофакторного эксперимента является "латинский" квадрат, у которого по строкам и столбцам значения уровней факторов не повторяются. Для k = 3 и n = 3 в этом случае план эксперимента следующий:

+	-	0
0	+	-
-	0	+

При проведении пассивного эксперимента также необходимо фиксировать такие наборы значений факторов, которые как можно более полно охватывают исследуемое факторное пространство.

Для определения уравнения регрессии число опытов **m** должно быть не менее числа оцениваемых его коэффициентов (параметров).

Нахождение параметров уравнения регрессии (функции отклика) производится по методу наименьших квадратов:

$$Z = \sum_{i=1}^{m} (y_{эi} - y_{Ti})^2 = \min_{\{a_j\}},$$

где m – число зафиксированных значений зависимой переменной;

$y_{эi}$ – экспериментальные значения зависимой переменной при i-м опыте;

y_{Ti} – теоретические (выравненные) значения зависимой переменной при i-м опыте;

a_j – j-й коэффициент (параметр) уравнения регрессии.

Взяв, частные производные по искомым параметрам и приравняв их нулю, получаем систему из **n** уравнений по числу неизвестных. Решение полученной системы дает значения параметров уравнения регрессии.

Например, для зависимости y = ax + b получаем:

$$Z = \sum_{i=1}^{m} (y_{эi} - y_{Ti})^2 = \sum_{i=1}^{m} (y_{эi} - (ax_1 + b))^2 = \min .$$

Берем частные производные по искомым параметрам a и b:

$$\partial Z / \partial a = \sum_{i=1}^{m} 2(y_{эi} - (ax_1 + b))(-x_i) = 0;$$

$$\partial Z / \partial b = \sum_{i=1}^{m} 2(y_{эi} - (ax_1 + b))(-1) = 0.$$

После упрощений получаем систему:

$$\begin{cases} a\sum_{i=1}^{m} x_i^2 + b\sum_{i=1}^{m} x_1 = \sum_{i=1}^{m} y_{эi}x_i \\ a\sum_{i=1}^{m} x_1 + b\,m = \sum_{i=1}^{m} y_{эi} \end{cases}$$

из которой выражаются неизвестные параметры a и b:

$$a = (m\sum_{i=1}^{m} y_{эi}x_i - \sum_{i=1}^{m} x_1 \sum_{i=1}^{m} y_{эi})/(m\sum_{i=1}^{m} x_1^2 - (\sum_{i=1}^{m} x_i)^2) \quad ;$$

$$b = 1/m(\sum_{i=1}^{m} y_{эi} - a\sum_{i=1}^{m} x_i).$$

Изучение **статистических зависимостей** основывается на **корреляционно-регрессионном** анализе. Корреляционный анализ позволяет ответить на вопрос о существовании зависимости между случайными величинами, а также оценить степень тесноты статистической зависимости. Инструментом регрессионного анализа является уравнение регрессии. Исходными данными для проведения корреляционно-регрессионного анализа является статистическая информация, содержащая значения факторов и зависимого от них показателя (параметра).

Одной из возможных схем проведения многофакторного корреляционно-регрессионного анализа является следующая:

1) проводится взаимный парный корреляционный анализ между всеми возможными сочетаниями факторов и при существенности связи между факторами пары один из дублирующих друг друга факторов (зависимый фактор) исключается из дальнейших расчетов. Число возможных сочетаний равно n(n-1)/2, где n – число факторов;

2) принимается вид уравнения регрессии (модель связи);

3) рассчитываются параметры уравнения регрессии;

4) проверяется значимость отдельных факторов в модели и адекватность уравнения регрессии экспериментальным данным в целом. Если нет малозначимых факторов и уравнение регрессии согласуется с экспериментальными данными – исследование закончено, а иначе на п.5;

5) при наличии малозначимых факторов они исключаются из расчетов и переход на п. 3 и при неадекватности уравнения регрессии экспериментальным данным переход на п. 2 для принятия нового вида уравнения регрессии.

Если связь оказалась несущественной для различных видов уравнения регрессии, то необходимо считать, что зависимая переменная не описывается рассматриваемыми факторами.

Полученное уравнение регрессии является моделью связи между факторным пространством и зависимой переменной (параметром).

Статистикой, характеризующей тесноту связи между факторами и зависимой переменной, является коэффициент множественной корреляции.

Коэффициент множественной корреляции R (R^2 – коэффициент детерминации) показывает, какая часть дисперсии зависимой переменной объясняется полученным уравнением регрессии

$$R = \sqrt{S^2_{об}/S^2_{п}}\ ,$$

где $S^2_{об} = \sum_{i=1}^{m}(y_{тi} - y_{м})^2$ – объясненная сумма квадратов отклонений от оценки математического ожидания;

$S^2_{п} = \sum_{i=1}^{m}(y_{эi} - y_{м})^2$ – полная сумма квадратов отклонений от оценки математического ожидания;

$y_{м}$ – оценка математического ожидания случайной величины.

Разность между полной и объясненной суммой квадратов является остаточной (необъясненной) суммой отклонений между теоретическими и эмпирическими значениями зависимой переменной

$$S^2_{ост} = S^2_{п} - S^2_{об} = \sum_{i=1}^{m}(y_{эi} - y_{тi})^2\ .$$

Тогда через $S^2_{ост}$ значение коэффициента множественной корреляции рассчитывается по формуле

$$R = \sqrt{1 - S^2_{ост}/S^2_{п}}\ .$$

Значения R может быть в пределах от 0 до 1.0. При R = 0 связь между факторами и зависимой переменной отсутствует, а R = 1.0 указывает на функциональную зависимость.

Значимость факторов оценивается по критерию Стьюдента.

Статистика критерия Стьюдента t_j рассчитывается по формуле

$$t_j = abs(a_j / s_{aj}),$$

где a_j – значение j-го параметра (коэффициента) в уравнении регрессии;

s_{aj} – среднеквадратическое отклонение параметра a_j .

Если расчетное значение статистики критерия Стьюдента t_j для параметра (коэффициента) при рассматриваемом факторе больше табличного значения критерия $t_{т\,\gamma,k}$, то нет оснований считать данный фактор малозначимым. Табличное значение $t_{т\gamma,k}$ определяется заданным уровнем значимости γ и числом степеней свободы k. Число степеней свободы определяется по выражению

$$k = m - n - 1,$$

где m – число эмпирических точек, в которых определен зависимый параметр; n – число факторов, входящих в уравнение регрессии.

Уровень значимости рекомендуется принимать равным 0.01–0.1 (чем меньше γ, тем выше требования к значимости фактора).

Для проверки существенности коэффициента множественной корреляции и, таким образом, оценивания адекватности уравнения регрессии экспериментальным данным используется статистика критерия Фишера

$$F = s_1^2 / s_2^2 = \frac{S_{об}^2 / n}{S_{ост}^2 /(m - n - 1)} = \frac{S_{об}^2 (m - n - 1)}{S_{ост}^2\, n} \text{ или}$$

$$F = \frac{R^2 (m - n - 1)}{(1 - R^2)\, n},$$

где s_1^2 и s_2^2 – соответственно объясненная и остаточная дисперсия для зависимой переменной.

Чтобы не было оснований отвергнуть гипотезу, что экспериментальные данные согласуются с полученным уравнением регрессии, рассчитанная статистика критерия Фишера должна быть больше табличного значения ($F > F_т$). Табличное значение $F_т$ определяется в зависимости от уровня значимости γ и числа степеней свободы k_1 и k_2 :

$$k_1 = n \, ;$$
$$k_2 = m - n - 1 \, .$$

Уровень значимости (вероятность) рекомендуется принимать 0.01 – 0.05 (чем меньше, тем жестче требования к адекватности модели).

Если $F<F_т$, то считается, что уравнение регрессии не адекватно экспериментальным данным.

Статистику критерия Фишера можно использовать для оценки значимости отдельных факторов. Фактор следует считать малозначимым в том случае, если его исключение из модели не вызывает существенного снижения статистики критерия Фишера. При этом исключение малозначимого фактора может обеспечить увеличение статистики F. Например, если при m=7 и n=3 имели $S_{об}^2$=2.1, а $S_{ост}^2$=0.7, а при исключении одного из факторов (n=2) получили $S_{об}^2$=1.8 и $S_{ост}^2$=1.0, то

$$F = \frac{2{,}1(7-3-1)}{0{,}7 \cdot 3} = 3.0 \text{ при } n=3;$$

$$F = \frac{1{,}8(7-2-1)}{1{,}0 \cdot 2} = 3{,}6 \text{ при } n=2.$$

Увеличение статистики F в приведенном примере указывает на малозначимость исключаемого из модели фактора.

Мерой согласованности уравнения регрессии с экспериментальными данными может служить также средняя линейная ошибка аппроксимации E

$$E = 1/m \sum_{i=1}^{m} abs((y_{эi} - y_{Ti})/y_{Ti}).$$

Компьютерная программа проведения множественного корреляционно-регрессионного анализа приведена в приложении 4.

2.2.5. Исследование временных рядов

Временной ряд представляет собой значения показателя в зависимости от времени. Может представлять изменение показателя по годам, за год по месяцам, за неделю по дням, за сутки по часам и т.п. Временной ряд может быть моментным (показатель зафиксирован на какие-то моменты времени) или интервальным (показатель зафиксирован за промежутки времени).

Временной ряд может выравниваться различного рода зависимостями (степенными, показательными, параболическими, логарифмическими, гиперболическими и др.).

Временные ряды с периодическими изменениями наиболее часто описываются рядом Фурье. Таким периодическим изменениям подвержены многие физические и экономические явления, связанные с сезонной, недельной, суточной и другой изменчивостью. Например, имеется такая изменчивость по дням недели для транспортной подвижности населения и нет зависимости от дня недели для температуры воздуха.

Выражения многочлена ряда Фурье, связывающие фактор (время) и зависимую переменную, имеют следующий вид:

$$y_{Ti} = a_o + \sum_{k=1}^{n} (a_k \cos(2\pi k\, t_i / T) + b_k \sin(2\pi k\, t_i / T)) \text{ или}$$

$$y_{Ti} = a_o + \sum_{k=1}^{n} (a_k \cos(2\pi k\, i / m) + b_k \sin(2\pi k\, i / m)),$$

где y_{Ti} – значение теоретической функции в i-й расчетной точке;

a_o – свободный член уравнения;

n – верхнее значение номера гармоник ряда Фурье;

k – номер гармоники;

a_k и b_k – коэффициенты ряда Фурье при k-й гармонике соответственно при cos и sin;

t_i – значение фактора (времени) в i-й расчетной точке;

T – интервал времени, за который рассматривается временной ряд;

m – общее число чисел во временном ряду.

Вторая формула может применяться только для случая, когда показатели зафиксированы через равные интервалы времени, а первая – в любом случае.

Параметры (коэффициенты) уравнений определяются соответственно по следующим зависимостям:

$$a_o = \frac{1}{m}\sum_{i=1}^{m} y_{эi};$$

$$a_k = 2/m\sum_{i=1}^{m} y_{эi}\cos(2\pi k\, t_i / T) \text{ или } a_k = 2/m\sum_{i=1}^{m} y_{эi}\cos(2\pi k\, i / m);$$

$$b_k = 2/m\sum_{i=1}^{m} y_{эi}\sin(2\pi k\, t_i / T) \text{ или } b_k = 2/m\sum_{i=1}^{m} y_{эi}\sin(2\pi k\, i / m),$$

где $y_{эi}$ – экспериментальные значения зависимой переменной в i-х расчетных точках.

Проверка адекватности полученного уравнения (многочлена ряда Фурье) экспериментальным данным производится по критерию Фишера и средней линейной ошибке аппроксимации. При этом, при расчете числа степеней свободы под числом факторов понимается число использованных гармоник ряда Фурье.

При проведении расчетов номера гармоник, включаемые в уравнение, рекомендуется принимать адаптивно в зависимости от изменения значения статистики критерия Фишера F или средней линейной ошибки аппроксимации E. Гармоники, которые вызывают уменьшение значения F или увеличение значения E, не включаются в модель связи. При этом верхнее значение номера гармоник не должно быть больше чем m/2.

Компьютерная программа выравнивания динамических рядов с помощью многочлена ряда Фурье приведена в приложении 5.

2.2.6. Системы массового обслуживания

Системой массового обслуживания (СМО) является совокупность взаимодействующих входящих потоков требований на обслуживание и каналов (аппаратов), занятых обслуживанием. При этом могут возникать очереди требований и каналов в ожидании начала обслуживания.

Входящий поток представляет собой последовательность требований (заявок), прибывающих в систему обслуживания, и характеризуется частотой поступления требований в единицу времени (интенсивностью) и законом распределения интенсивности потока. Входящий поток может быть описан также интервалами времени между моментами поступления требований и законом распределения этих интервалов.

Требования в потоке могут поступать по одному (ординарные потоки) или группами (неординарные потоки).

Свойство ординарности потока заключается в том, что в любой момент времени может поступить только одно требование. Иными словами, свойство заключается в том, что вероятность поступления больше одного требования за малый промежуток времени есть бесконечно малая величина.

В случае группового поступления требований задается интенсивность поступления групп требований и закон ее распределения, а также размер групп и закон их распределения.

Интенсивность поступления требований может изменяться во времени (нестационарные потоки) или зависит только от единицы времени, принятой для определения интенсивности (стационарные потоки). Поток называется стационарным, если вероятность появления n требований за промежуток времени $(t_0, t_0+\Delta t)$ не зависит от t_0, а зависит только от Δt.

В нестационарном потоке интенсивность изменяется во времени по непериодической или периодической закономерности (например, процессы сезонного характера), а также может иметь периоды, соответствующие частичной или полной задержке потока.

В зависимости от того, имеется ли связь между числом требований, поступивших в систему до и после некоторого момента времени, поток бывает с последействием или с отсутствием последействия.

Ординарный, стационарный поток требований с отсутствием последействия является **простейшим.**

Системы массового обслуживания подразделяются в первую очередь на "разомкнутые" и "замкнутые". На вход разомкнутой системы массового обслуживания извне поступает поток заявок от бесконечно большого числа источников, которые в систему не входят и их состояние анализу не подвергается. В замкнутой системе массового обслуживания число источников заявок ограничено и соответственно интенсивность поступления требований зависит от числа источников и работы самой системы.

Кроме того, тип системы массового обслуживания определяется дисциплиной очереди (ожидания) и параметрами механизма обслуживания.

Дисциплина очереди предусматривает:

1) правило распределения требований по очередям перед каналами обслуживания;

2) ограничение на размер очередей или на время ожидания в них;

3) наличие или отсутствие приоритета у требований, виды приоритетов и правила их применения.

В зависимости от правила распределения требований по каналам различают системы с полной доступностью или системы с ограниченной доступностью. В первом случае требование может быть обслужено в любом канале системы и очередь может быть одна на всю систему. Во втором случае определенные требования могут быть обслужены только соответствующими каналами и очередь формируется отдельно для каждой группы каналов одинаковой доступности или для каждого канала группы раздельно. Кроме того, может быть такая система, когда часть каналов имеет ограниченную доступность, а часть – полную доступность. Правила формирования очередей определяются назначением каналов и могут иметь различные варианты.

В зависимости от того, как ведут себя требования в очереди, различают:

системы без потерь (с ожиданием), когда требование найдя все обслуживающие каналы занятыми неограниченно ожидает начала обслуживания;

системы с ограниченными потерями (смешанные системы), когда пребывание требования в очереди ограничивается по времени (требование находится в очереди ограниченное время) или длиной очереди (при достижении очередью определенной величины вновь прибывающее требование покидает систему);

системы с потерями (без ожидания), когда требование найдя все доступные ему каналы обслуживающей системы занятыми, покидает ее.

В зависимости от правила отбора из очереди требований на обслуживание, системы массового обслуживания делятся на системы без приоритета требований и с приоритетами требований. В системе с приоритетами требований выбор требований из очереди на обслуживание отличается по каждому требованию или группам требований.

В системах без приоритета возможны следующие правила приема требований на обслуживание:

1) в порядке поступления в систему, то есть освободившийся аппарат принимает на обслуживание требование, поступившее ранее других;

2) требование, поступившее последним, принимается на обслуживание первым;

3) требования выбираются из очереди на обслуживание случайным образом (неупорядоченная очередь).

В системах с приоритетами требований различают относительный приоритет (без прерывания обслуживания), когда при поступлении требования с более высоким приоритетом оно принимается на обслуживание после окончания ранее начавшегося обслуживания требования с меньшим приоритетом, и абсолютный приоритет, когда канал освобождается немедленно для обслуживания поступившего требования с более высоким приоритетом.

Шкала приоритета может быть построена исходя из каких-то внешних относительно системы обслуживания критериев или на показателях, связанных с работой самой системы обслуживания. Практическое значение имеют следующие типы приоритетов:

разделение входящих требований по категориям приоритетности в зависимости от их источников;

приоритет у требований с наименьшим временем обслуживания. Эффективность данного приоритета может быть показана на следующем примере. Поступили последовательно два требования с длительностью обслуживания соответственно 6,0 и 1,0 ч. При приеме их на обслуживание освободившимся каналом в порядке поступления простой составит для 1-го требования 6,0 ч и для второго – 6,0+1,0 = 7,0 ч или суммарно для двух требований 13,0 ч. Если второе требование принять на обслуживание первым, то его простой составит 1,0 ч и простой другого– 1,0+6,0 = 7,0 ч или суммарно для двух требований 8,0 ч. Общее сокращение простоев требований в системе от принятия приоритета для второго требования составит 5,0 ч (13-8);

приоритет у требований с минимальным отношением времени обслуживания к мощности (производительности) источника требования, например, к грузоподъемности автомобиля.

Механизм обслуживания характеризуется параметрами отдельных каналов обслуживания, пропускной способностью системы в целом и другими данными об обслуживании требований. Пропускная способность системы определяется числом каналов (аппаратов) и производительностью каждого из них.

Производительность каждого отдельного канала (аппарата) определяется длительностью обслуживания одного требования или интенсивностью потока обслуженных требований – средними значениями и законом распределения.

В зависимости от организации работы механизма обслуживания системы могут следующих разновидностей:

– одноканальные (система состоит из одного канала) или многоканальные (система состоит из двух и более параллельных каналов), в том числе с постоянным или переменным числом каналов;

– однофазные, когда полный цикл обслуживания производится одним аппаратом, или многофазные, когда обслуживание производится в последовательных аппаратах и соответственно в канале может находиться несколько требований. Очередь требований перед каждым аппаратом может не разрешаться или ограничиваться;

– с обслуживанием в аппаратах одиночных или групп требований;

– с комбинациями параллельных и последовательных каналов и аппаратов (сети СМО);

– с каналами или аппаратами обслуживания, имеющими одинаковые или различные средние значения и законы распределения времени обслуживания;

– с приоритетом отдельных параллельных каналов и без приоритета. В системах без приоритета возможна очередность загрузки каналов по различным правилам: первым освободился – первым загружается; последним освободился – первым загружается (например, при длительной подготовке к возобновлению обслуживания); в случайном порядке, когда требование выбирает канал случайным образом из-за большого числа влияющих факторов. Приоритетность может отдаваться более производительным каналам и по другим соображениям. Требования могут выбирать канал обслуживания;

– с объединением каналов для обслуживания одного требования (взаимопомощь каналов) или раздельное обслуживание (отсутствие взаимопомощи).

Системы массового обслуживания могут быть исследованы аналитическими методами или на основе имитационного статистического моделирования. Аналитические зависимости имеются для определения параметров функционирования СМО в наиболее простых случаях, например, при простейшем потоке требований на обслуживание, экспоненциальном законе распределения длительности обслуживания, отсутствии приоритета требований и каналов друг перед другом и т.п.

Для сложных систем (сетей) отсутствуют аналитические зависимости и параметры их работы определяются на основе моделирования.

Основными параметрами функционирования системы массового обслуживания являются: вероятность того, что все каналы свободны p_b; вероятность p_k того, что в системе находится ровно k-требований; вероятность $p_з$ того, что все каналы заняты; вероятность $p(t_{ож}<t_з)$ того, что время пребывания требования в очереди ($t_{ож}$) не превысит заданной величины $t_з$; среднее время ожидания требованием начала обслуживания $t_{ожт}$; среднее время простоя каналов в ожидании требований на обслуживание $t_{ожк}$; средняя длина очереди $M_{ож}$; среднее время нахождения требования в системе t_c; среднее число требований в системе M_c; среднее число простаивающих каналов n_o; среднее число занятых каналов $n_з$; среднее число требований, простаивающих на обслуживании $M_{обс}$; коэффициент простоя каналов $К_п = n_o / n$; коэффициент занятости каналов $К_з = n_з / n$, где n – общее число каналов.

Между отдельными параметрами функционирования систем массового обслуживания существует связь:

$$n = n_o + n_з \;;$$

$$n_o = \sum_{k=0}^{n-1} (n - k) p_k \;;$$

$$n_з = \sum_{k=1}^{n} k p_k + n \sum_{k=n+1}^{\infty} p_k \;;$$

$$M_{обс} = n_з \;;$$

$$M_c = M_{обс} + M_{ож} \;;$$

$$M_{ож} = \sum_{k=n+1}^{\infty} (k - n) p_k \;;$$

$$M_c = \sum_{k=1}^{\infty} k\, p_k \;;$$

$t_c = t_{обс} + t_{ожт}$ ($t_{обс}$ – средняя продолжительность обслуживания в канале)

и др.

1) Аналитическое исследование систем массового обслуживания

Многоканальная разомкнутая система массового обслуживания

В качестве примера рассматривается многоканальная СМО с простейшим потоком требований и экспоненциальным распределением времени их обслуживания (рисунок 2.17). Система с ожиданием и без приоритетов требований и каналов друг перед другом.

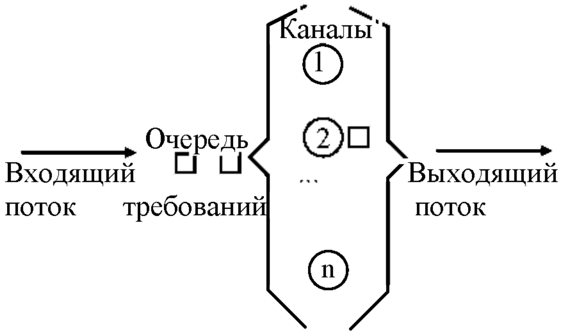


Рисунок 2.17 – Схема многоканальной разомкнутой системы массового обслуживания

Поток требований на обслуживание характеризуется средней интенсивностью L ($с^{-1}$, $мин^{-1}$, $ч^{-1}$, $сут^{-1}$) и имеет пуассоновский закон распределения. Доказано, что в этом случае интервалы между поступлениями требований распределены по экспоненциальному закону распределения. Длительность времени обслуживания требования характеризуется средней величиной $t_{обс}$ (потоком обслуживания $\nu=1/t_{обс}$). Число каналов в системе – n.

Основные показатели функционирования многоканальной разомкнутой системы массового обслуживания рассчитываются по формулам:

вероятность того, что все каналы обслуживания свободны

$$p_0 = (\sum_{k=0}^{n-1} x^k/k! + x^n/((n-1)!(n-x)))^{-1},$$

где $x = L\ t_{обс}$ – приведенный поток, физическая сущность которого – число каналов, необходимое для обслуживания требований при детерминированных их потоке и времени обслуживания. Должно соблюдаться условие $x < n$;

вероятность того, что в системе находится ровно k требований

$$p_k = \begin{cases} x^k p_o/k!, \text{если } k \le n \\ x^k p_o/(n^{k-n}\ n!), \text{если } k > n \end{cases}$$

вероятность того, что все каналы заняты

$$p_з = x^n p_o/((n-x)(n-1)!) = n\ p_n/(n-x);$$

вероятность того, что занято ровно n каналов

$$p_n = x^n p_o/n!;$$

вероятность того, что время ожидания требованием начала обслуживания $t_{ож}$ меньше или больше $t_з$

$$p(t_{ож} > t_з) = p_з\ \exp(-(n\nu - L)\,t_з) \text{ или}$$
$$p(t_{ож} < t_з) = 1 - p_з\ \exp(-(n\nu - L)\,t_з);$$

среднее число незанятых каналов обслуживания

$$n_0 = p_o \sum_{k=0}^{n-1} x^k (n-k)/k!;$$

среднее число требований, простаивающих в очереди на обслуживание

$$M_{ож} = x\ n\ p_o/(n-x)^2 = x\ p_з/(n-x);$$

среднее число требований на обслуживании

$$M_{обс} = n\,p_з + p_o \sum_{k=1}^{n-1} x^k/(k-1)!\,;$$

средняя длительность времени ожидания требованиями начала обслуживания $t_{ожт} = t_{обс}\ \ p_з/(n - x)$.

Многоканальная замкнутая система массового обслуживания

В качестве примера рассматривается многоканальная СМО с числом каналов n и числом источников, генерирующих требования, m (рисунок 2.18). При этом поток требований, создаваемый одним источником, простейший. Длительность времени обслуживания требований в канале имеет экспоненциальное распределение. Система с ожиданием и без приоритетов требований и каналов друг перед другом.

Поток требований, генерируемых одним источником во время нахождения его вне системы обслуживания, характеризуется средней интенсивностью λ ($с^{-1}$, $мин^{-1}$, $ч^{-1}$, $сут^{-1}$). Обратная величина λ является средней продолжительностью времени до последующего поступления требования от обслуженного источника (средний период до возврата в систему на обслуживание).

Время обслуживания характеризуется средней величиной $t_{обс}$ или потоком обслуживания $\nu = 1/t_{обс}$.

Основные показатели функционирования многоканальной замкнутой системы массового обслуживания рассчитываются по формулам:

вероятность того, что все каналы обслуживания свободны

$$p_o = \left(\sum_{k=0}^{n} x^k\ m!/(k!\,(m-k)!) + \sum_{k=n+1}^{m} x^k\ m!/(n^{k-n}\ n!\,(m-k)!)\right)^{-1};$$

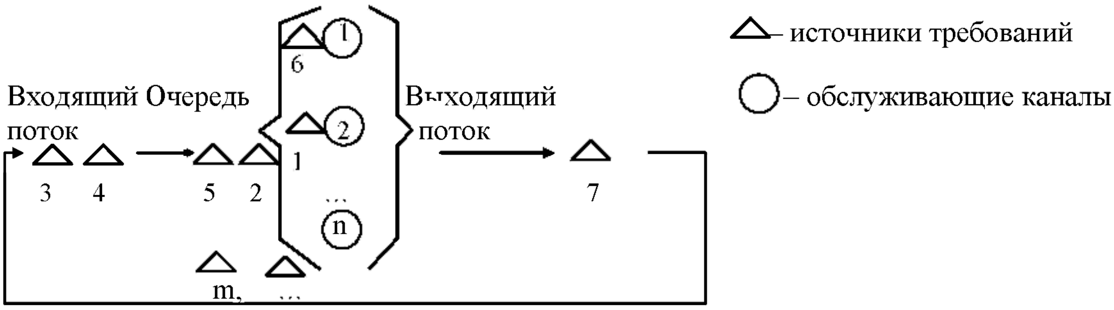


Рисунок 2.18 – Схема многоканальной замкнутой системы массового обслуживания

$x = \lambda\ t_{обс}$ или $x = \lambda/\nu$ – приведенный поток от одного источника требований при детерминированных потоке и времени обслуживания;

вероятность того, что в системе обслуживания находится ровно k требований

$$p_k = \begin{cases} x^k m!\,p_o/(k!\,(m-k)!), \text{если } k < n \\ x^k m!\,p_o/(n!\,(m-k)!\,n^{k-n}), \text{если } k > n \end{cases};$$

среднее число незанятых каналов обслуживания

$$n_o = p_o m! \sum_{k=0}^{n-1} x^k (n-k)/(k!\,(m-k)!)\,;$$

среднее число требований, простаивающих в очереди на обслуживание

$$M_{ож} = p_o m!/n! \sum_{k=n+1}^{m} x^k (k-n)/((m-k)!n^{k-n});$$

среднее число требований, находящихся на обслуживании

$$M_{обс} = n_з; \qquad n_з = n - n_o.$$

Для одноканальной замкнутой СМО (n = 1) имеют место следующие зависимости:
вероятность того, что все каналы свободны

$$p_o = (\sum_{k=0}^{m} x^k m!/(m-k)!)^{-1};$$

средняя продолжительность ожидания требованием начала его обслуживания

$$t_{ожт} = t_{обс}(m/(1-p_o)-1)-1/x;$$

средняя продолжительность простоя канала в ожидании очередного требования на обслуживание

$$t_{ожк} = p_o t_{обс}/(1-p_o);$$

вероятность того, что канал занят $p_з = 1 - p_o$.

2) Статистическое имитационное моделирование

Статистическое имитационное моделирование основывается на генерации случайных величин, имитации функционирования системы и статистической обработке результатов моделирования. Методом моделирования может быть исследована СМО любой степени сложности.

Для проведения моделирования могут использоваться как универсальные языки программирования так и проблемно-ориентированные – GPSS, SIMULA и др.

Параметры функционирования системы оцениваются при моделировании по результатам многократного обслуживания требований (многократных испытаний). При имитации работы системы случайные величины (длительность обслуживания в каналах, интервалы между поступлениями требований, время возврата требований в систему, моменты возникновения отказов каналов и их длительность и др.) получают генерацией по ранее приведенным алгоритмам в зависимости от вида распределения (закон, усечение, смещение).

Число обслуживаний (опытов) необходимо принимать таким, чтобы обеспечить оценку интересующих параметров с заданной точностью при принятой доверительной вероятности.

Таким образом, определение числа опытов производится по аналогии с расчетом размера выборки для исследования случайных величин. При этом это число рекомендуется определять в ходе моделирования на основе оценки точности рассчитываемых параметров.

Алгоритмы моделирования ранее рассмотренных систем массового обслуживания приведены на рисунках 2.19 и 2.20. Число моделируемых обслуживаний определяется на основе формулы для нормального закона распределения, а в качестве интересующего показателя принята средняя продолжительность ожидания требованием начала обслуживания. Относительная точность оценивания задана равной ε с односторонней доверительной вероятностью γ= 0.95 (квантиль равна 1.645).

Структура алгоритмов следующая:

блок 2 – ввод и вывод на принтер исходных данных;

блоки 3-6 – формирование начальных условий моделирования;

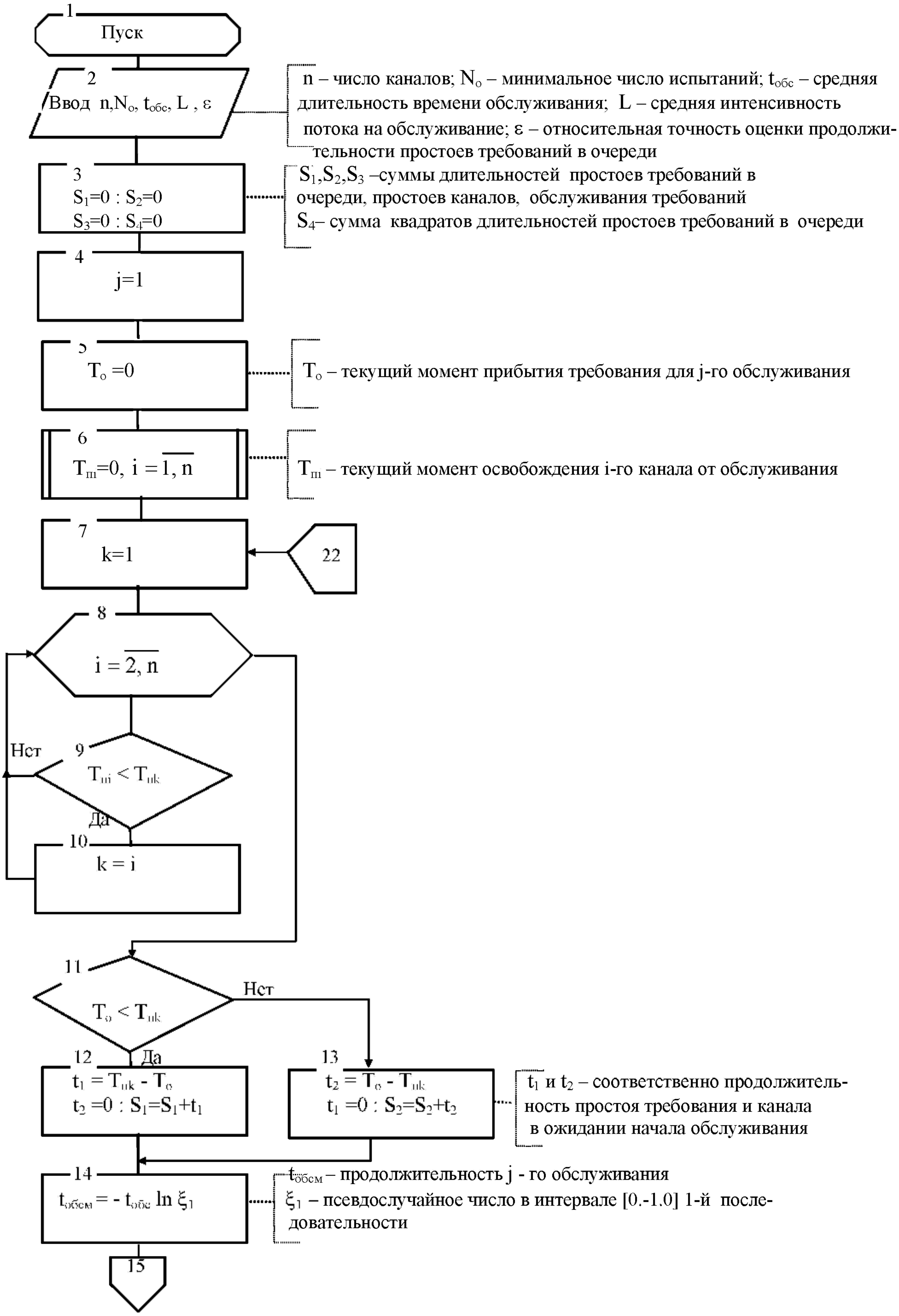
блоки 7-10 – поиск канала (источника) с минимальным значением момента времени освобождения от предыдущего обслуживания (прибытия на обслуживание);

блоки 11-18 – имитация обслуживания требований и накопление сумм длительностей времени простоев и обслуживания;

блоки 19-21 – принятие решения об окончании моделирования или его продолжении;

блок 22 – наращивание номера опыта (испытания);

блоки 23-24 – вычисление средних значений параметров и вывод их на монитор (принтер).


1 Пуск
2 Ввод n,N₀, tобс, L , ε
n – число каналов; N₀ – минимальное число испытаний; tобс – средняя длительность времени обслуживания; L – средняя интенсивность потока на обслуживание; ε – относительная точность оценки продолжительности простоев требований в очереди
3 S1=0 : S2=0 S3=0 : S4=0
S1,S2,S3 –суммы длительностей простоев требований в очереди, простоев каналов, обслуживания требований
S4– сумма квадратов длительностей простоев требований в очереди
4 j=1
5 To =0
To – текущий момент прибытия требования для j-го обслуживания
6 Tпi=0, i = 1, n
Tпi – текущий момент освобождения i-го канала от обслуживания
7 k=1
22
8 i = 2, n
9 Tпi < Tпk
Нет
Да
10 k = i
11 To < Tпk
Нет
Да
12 t1 = Tпk - To t2 =0 : S1=S1+t1
13 t2 = To - Tпk t1 =0 : S2=S2+t2
t1 и t2 – соответственно продолжительность простоя требования и канала в ожидании начала обслуживания
14 tобсм = - tобс ln ξ1
tобсм – продолжительность j - го обслуживания
ξ1 – псевдослучайное число в интервале [0.-1.0] 1-й последовательности
15

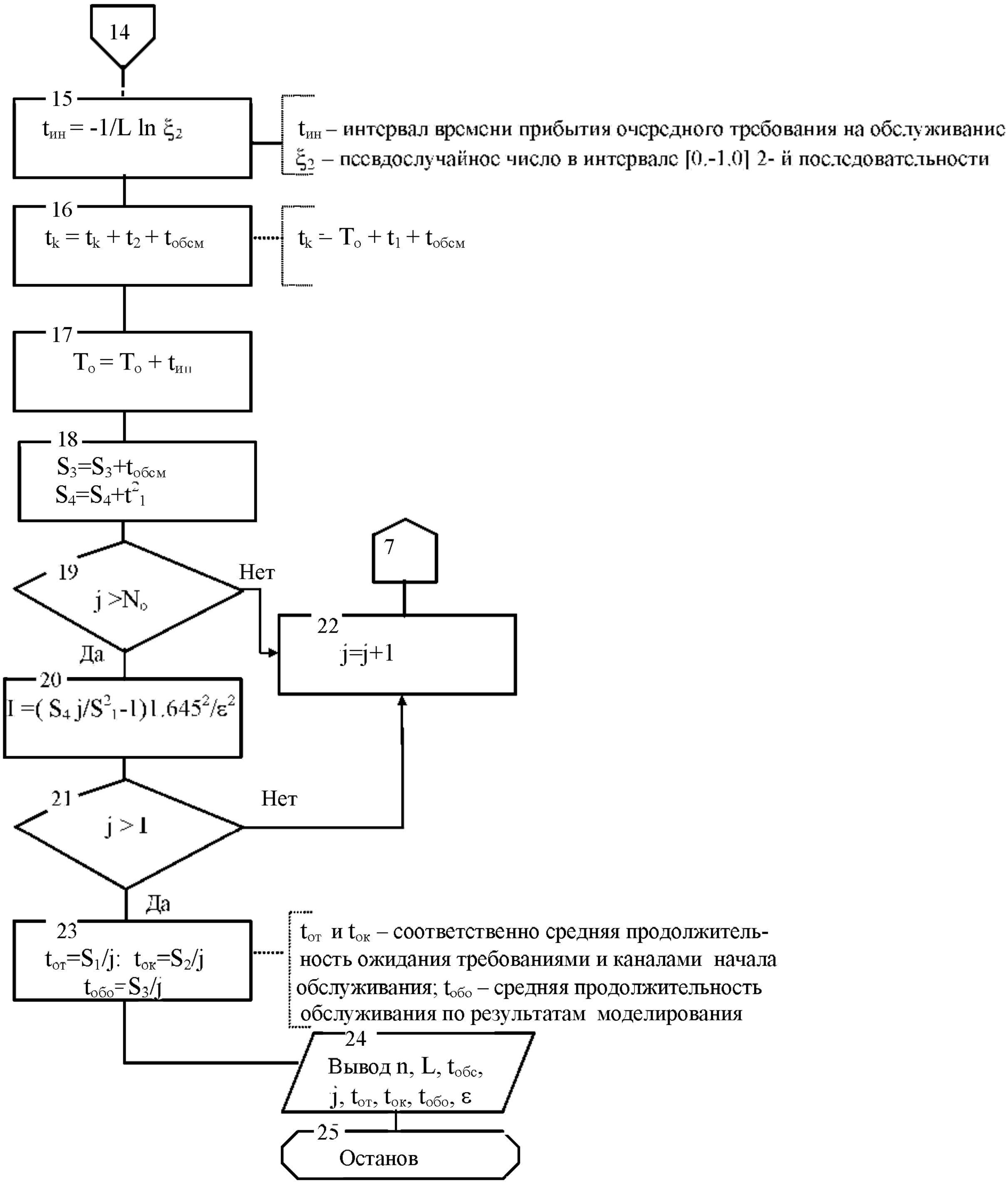
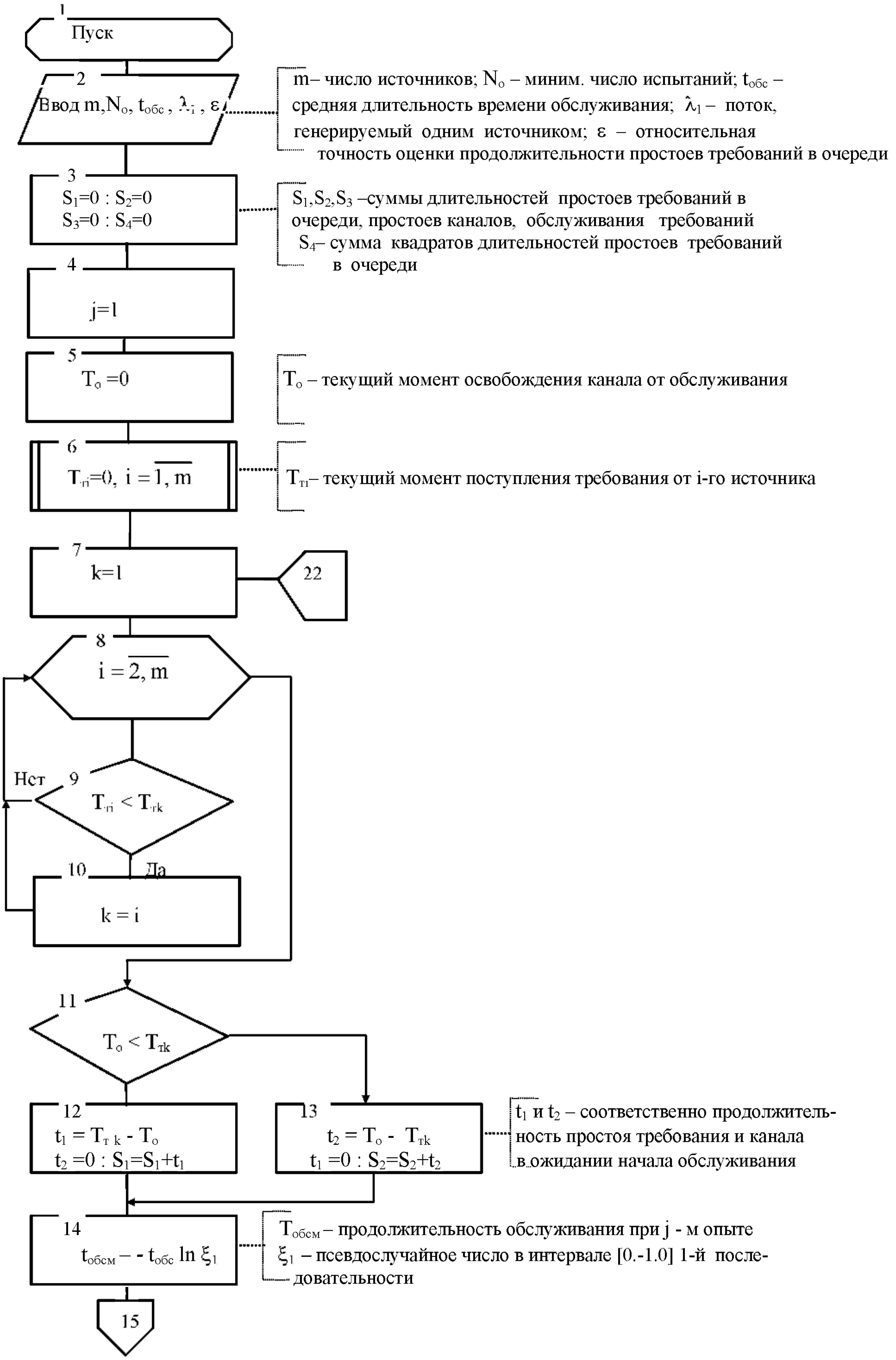


Рисунок 2.19 – Алгоритм моделирования многоканальной СМО разомкнутого типа с ожиданием


1
Пуск
2
Ввод m,N$_o$, t$_{обс}$, λ_i , ε
m– число источников; N$_o$ – миним. число испытаний; t$_{обс}$ – средняя длительность времени обслуживания; λ_1 – поток, генерируемый одним источником; ε – относительная точность оценки продолжительности простоев требований в очереди
3
S$_1$=0 : S$_2$=0
S$_3$=0 : S$_4$=0
S$_1$,S$_2$,S$_3$ –суммы длительностей простоев требований в очереди, простоев каналов, обслуживания требований
S$_4$– сумма квадратов длительностей простоев требований в очереди
4
j=1
5
T$_o$ =0
T$_o$ – текущий момент освобождения канала от обслуживания
6
T$_{тi}$=0, i = $\overline{1, m}$
T$_{тi}$– текущий момент поступления требования от i-го источника
7
k=1
22
8
i = $\overline{2, m}$
9
T$_{тi}$ < T$_{тk}$
Нет
Да
10
k = i
11
T$_o$ < T$_{тk}$
12
t$_1$ = T$_{т k}$ - T$_o$
t$_2$ =0 : S$_1$=S$_1$+t$_1$
13
t$_2$ = T$_o$ - T$_{тk}$
t$_1$ =0 : S$_2$=S$_2$+t$_2$
t$_1$ и t$_2$ – соответственно продолжительность простоя требования и канала в ожидании начала обслуживания
14
t$_{обсм}$ – - t$_{обс}$ ln ξ$_1$
T$_{обсм}$ – продолжительность обслуживания при j - м опыте
ξ$_1$ – псевдослучайное число в интервале [0.-1.0] 1-й последовательности
15

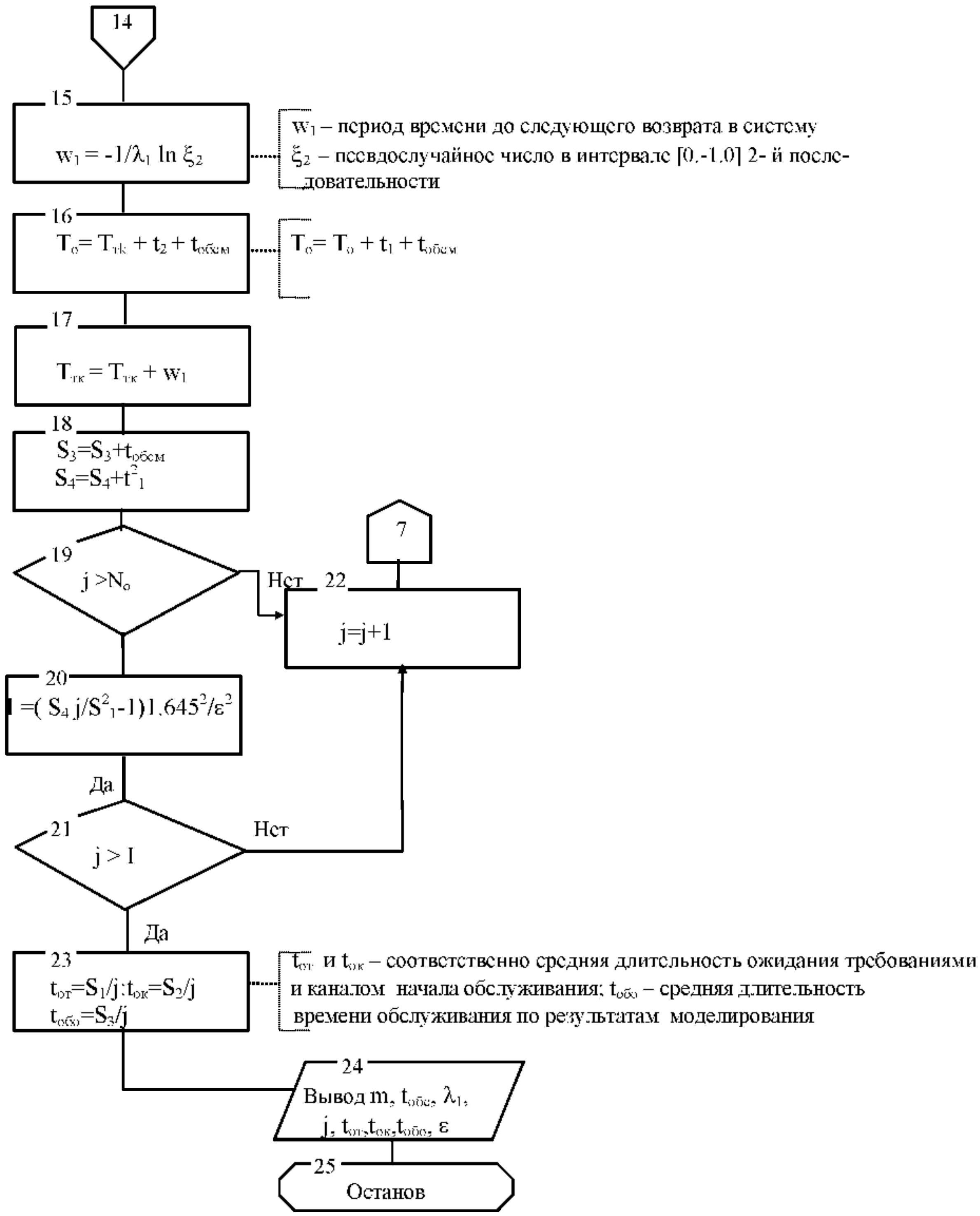


Рисунок 2.20 – Алгоритм моделирования одноканальной СМО замкнутого типа с ожиданием

3. ОПТИМИЗАЦИОННЫЕ ЗАДАЧИ И МЕТОДЫ ИХ РЕШЕНИЯ

Оптимизационные задачи состоят в отыскании таких значений управляемых параметров, при которых достигается экстремум (минимум или максимум) целевой функции, а также выполняются заданные ограничения в случае условной оптимизации.

3.1. Безусловная оптимизация одномерной унимодальной целевой функции

Унимодальной называется функция, имеющая один экстремум.

Задача поиска экстремума сводится к нахождению значения $\mathbf{x_o}$, соответствующего максимуму или минимуму f(x).

Для решения задачи могут применяться аналитический метод, численные методы и методы случайного поиска. Реализация численного метода или метода случайного поиска позволяет найти максимум или минимум. Для нахождения противоположного вида экстремума, например, максимума по алгоритму решения на минимум, необходимо значения оптимизируемой функции f(x) умножить на (-1).

Применение **аналитического** метода возможно в случае дифференцируемости функции f(x) и находится как $f'(x) = 0$. Решение полученного уравнения дает оптимальное значение $\mathbf{x_o}$. Значение второй производной определяет вид экстремума. Если $f''(\mathbf{x_o}) > 0$, то имеем минимум, а если $f''(\mathbf{x_o}) < 0$, то максимум (рисунок 3.1).

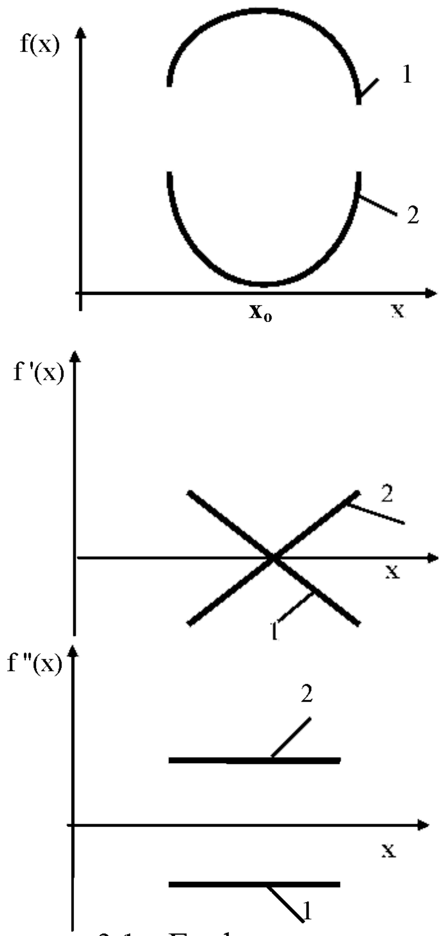


Рисунок 3.1 – Графическая интерпретация поиска экстремума дифференцируемой функции

Из приведенной графической интерпретации метода следует, что функция (1) имеет максимум и функция (2) – минимум.

Среди **численных** находят применение следующие методы: дихотомии, золотого сечения, Фибоначчи, шаговые и аппроксимации кривыми.

Метод дихотомии (половинного деления) является одним из методов одномерной безусловной оптимизации унимодальной целевой функции. Алгоритм метода основывается на выборе исходного отрезка поиска решения [a, b] и последующем делении текущего отрезка пополам:

1) $x_c=(b+a)/2$;

2) $x_1 = x_c - \varepsilon/2$; $x_2 = x_c + \varepsilon/2$, где ε – точность поиска экстремума;

3) если при минимизации $f(x_1) < f(x_2)$, то $b = x_c$, иначе $a = x_c$;

4) при $b - a <= \varepsilon$, $x_{опт} = (b + a)/2$ и решение получено, иначе на п. 1.

Ниже приведена графическая интерпретация (рисунок 3.2) и один из возможных алгоритмов метода дихотомии (рисунок 3.3).

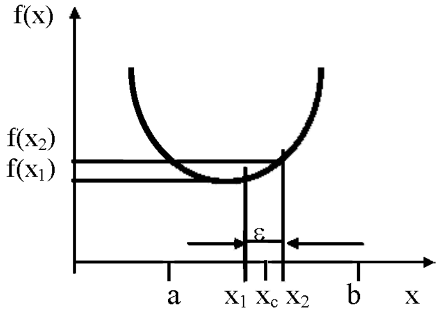


Рисунок 3.2 – Графическая интерпретация метода дихотомии

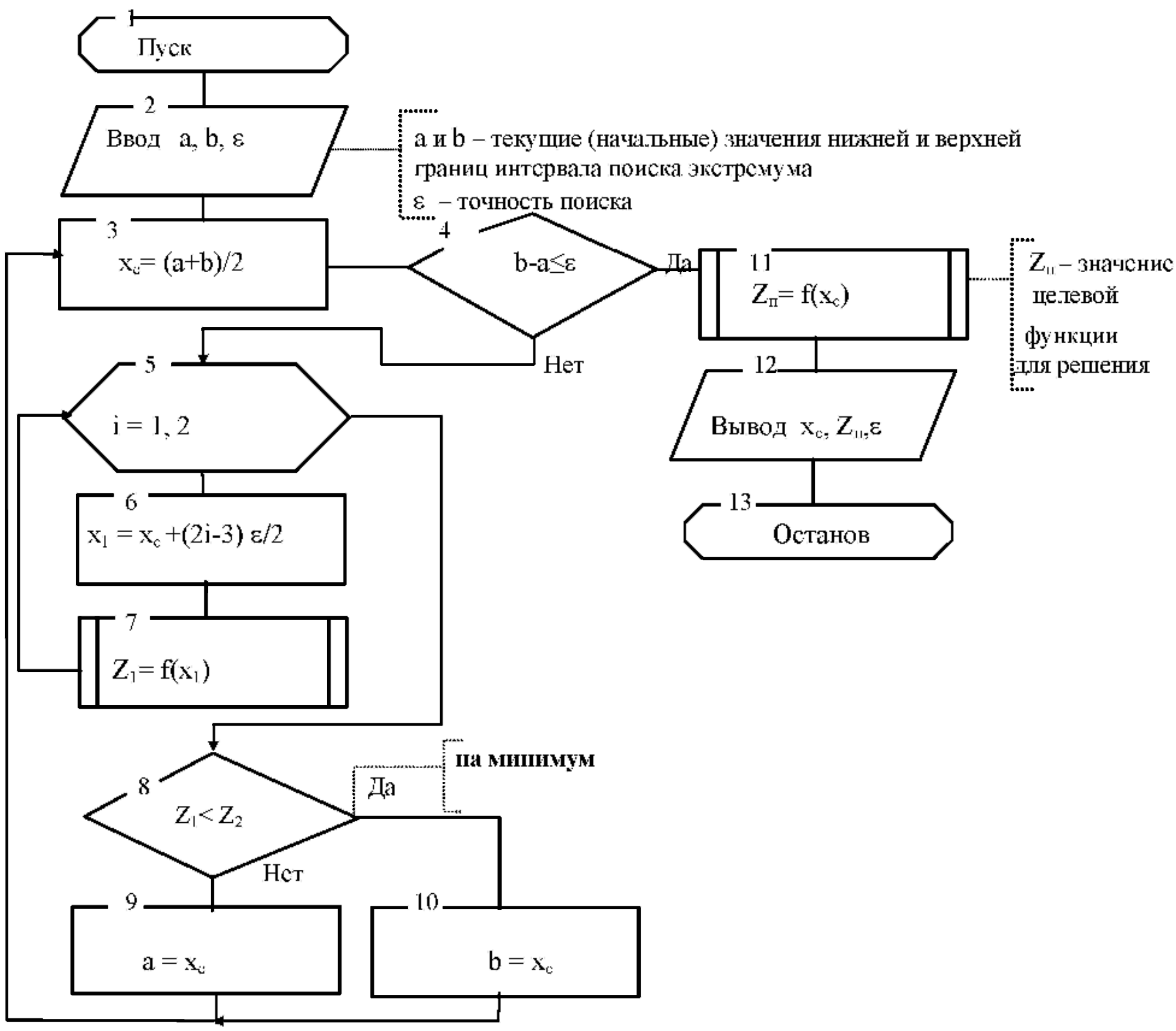


Рисунок 3.3 - Схема алгоритма программы по методу дихотомии

Метод золотого сечения основан на делении отрезка [a,b] по правилу "золотого" сечения, когда отношение большего отрезка к меньшему const. Такое отношение определяется выражением $(\sqrt{5}-1)/2 \approx 0.62$. При этом методе в отличие от метода дихотомии на каждой итерации требуется расчет только одного значения целевой функции. В результате находится решение $x_п$ и соответствующее ему значение целевой функции $Z_п$ (рисунки 3.4, 3.5).

На минимум:

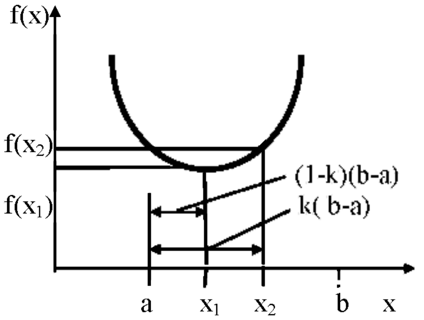


Рисунок 3.4 – Графическая интерпретация метода золотого сечения

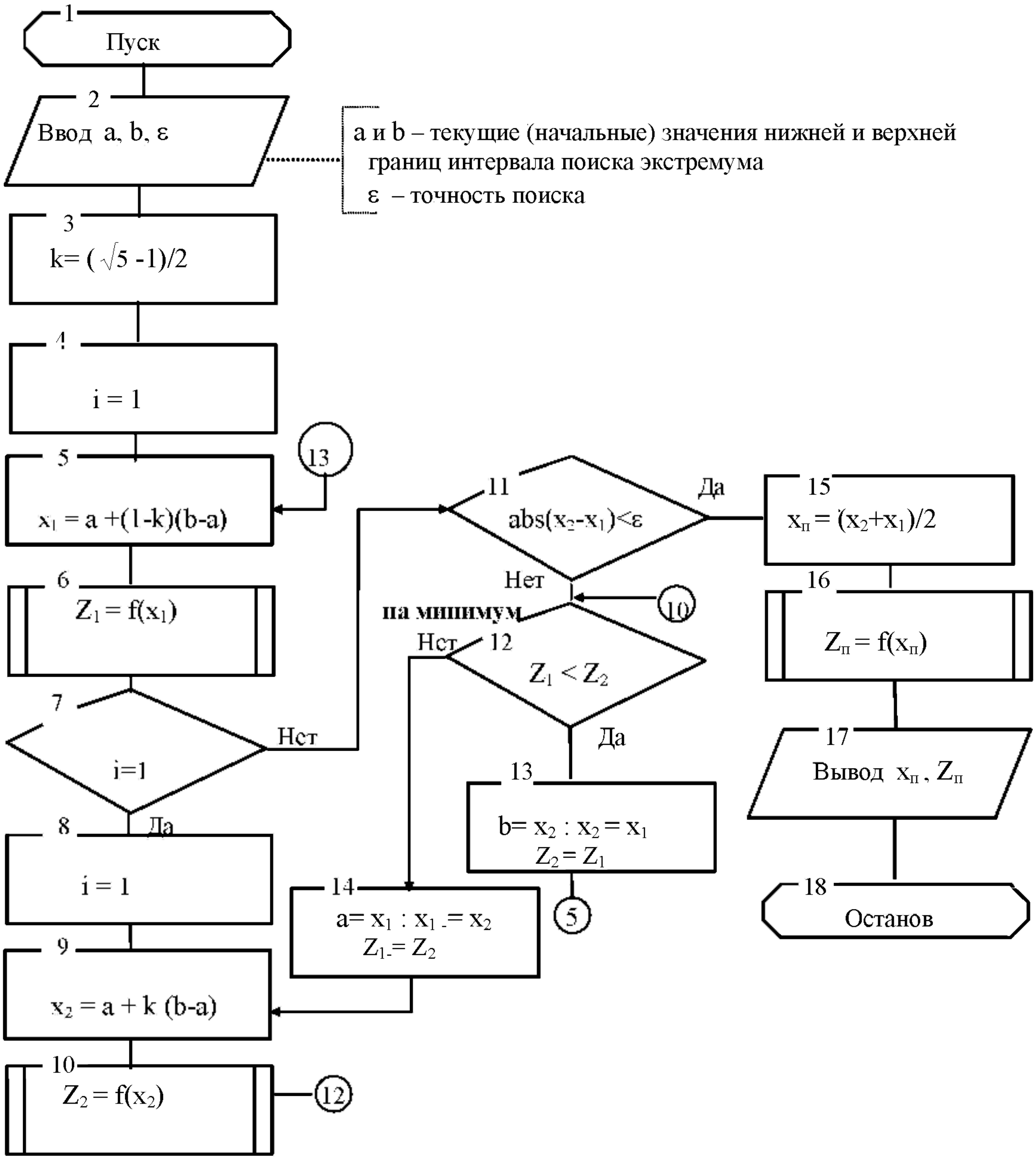


Рисунок 3.5 - Схема алгоритма программы по методу золотого сечения

Метод Фибоначчи основан на делении отрезка [a, b] с использованием чисел Фибоначчи, представляющих ряд, у которого последующее число равно сумме двух предыдущих (1,1,2,3,5,8 и т.д.).

Шаговые методы основаны на том, что текущему приближению к решению $x_п$ на каждом новом шаге дается приращение h как $x_п = x_п + h$ и вычисляется $f(x_п)$. Если новое значение целевой функции "лучше" предыдущего, то переменной x дается новое приращение. Если функция "ухудшилась", то поиск в данном направлении завершен.

Имеется ряд разновидностей шагового метода поиска экстремума целевых функций (прямой поиск, поразрядного приближения, Зейделя и др.).

Графическая интерпретация и алгоритм поиска экстремума функции на основе поразрядного приближения приведены на рисунках 3.6, 3.7.

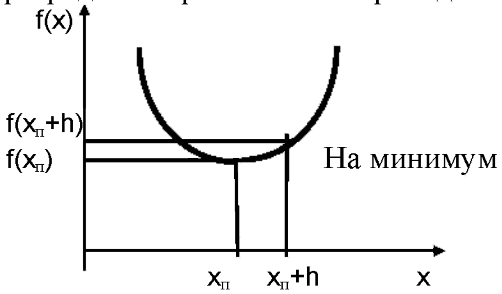


Рисунок 3.6 – Графическая интерпретация метода поразрядного приближения

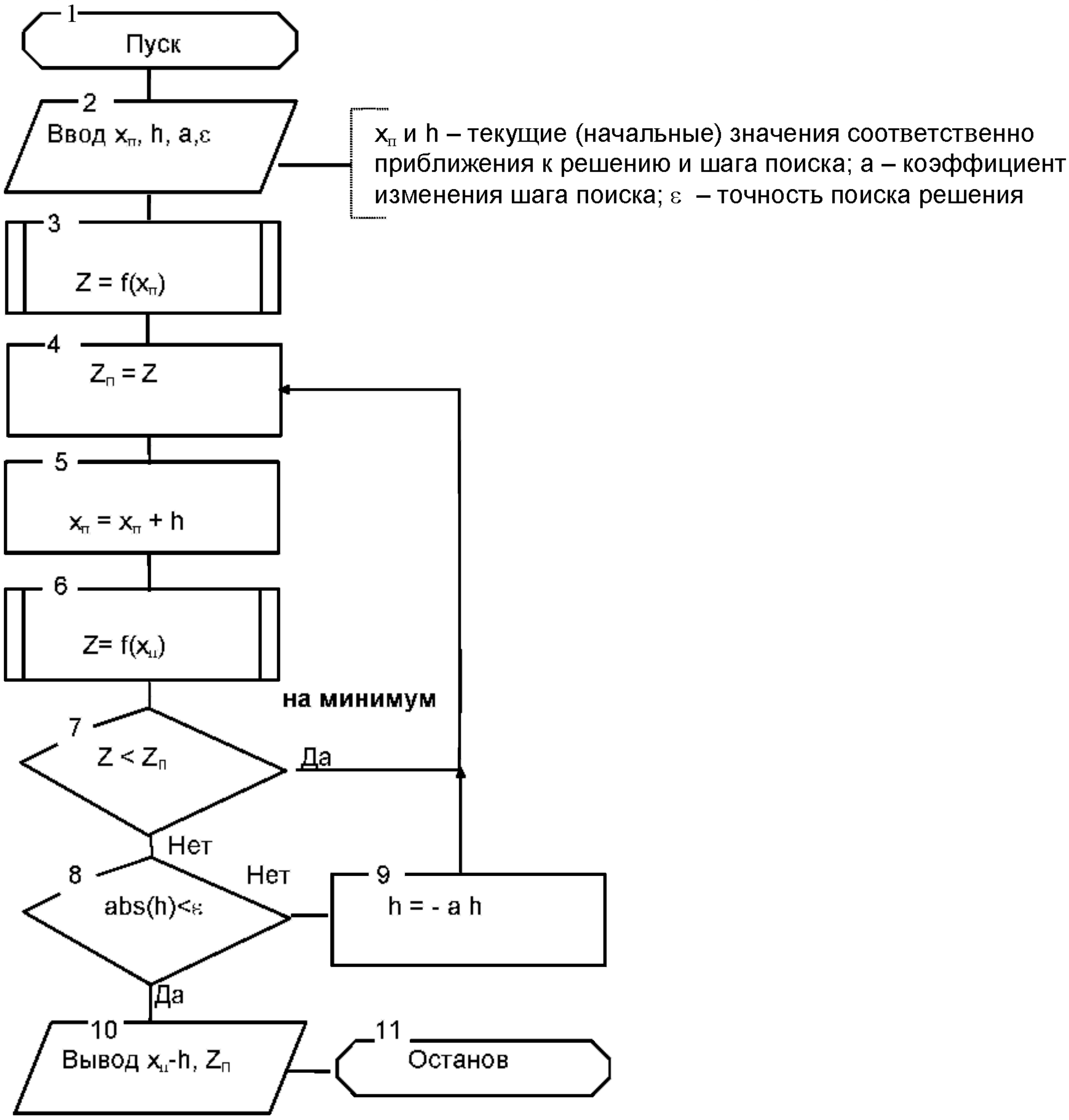


Рисунок 3.7 – Алгоритм поиска экстремума по методу поразрядного приближения

Метод квадратичной интерполяции-экстраполяции является одним из методов аппроксимации кривыми и базируется на описании целевой функции квадратичной параболой по трем точкам – текущее приближение $x_1= x_п$ и точки, лежащие от нее слева x_0 и справа x_2 на удалении h и нахождении экстремума аналитически. Процесс проводится до тех пор, пока предыдущее и последующее приближения различаются более, чем на заданную точность поиска. Алгоритм метода следующий (рисунок 3.8, 3.9):

1) находятся $x_0 = x_п - h$; $y_0 = f(x_0)$; $x_1 = x_п$; $y_1 = f(x_1)$; $x_2 = x_п + h$; $y_2 = f(x_2)$;

2) находятся параметры параболы, проходящей через три выбранных точки:

$a = (y_o- 2y_1 + y_2) / (2h^2)$;

$b = (-y_o (2x_1 + h) + 4y_1 x_1 - y_2 (2x_1 - h)) / (2h^2)$;

3) вычисляется очередное приближение $x_п$ на основе аналитической оптимизации аппроксимирующей функции как $x_п = - b/(2a)$;

4) проверяется условие: $abs(x_п - x_1) < \varepsilon$. Если условие выполняется, то оптимум найден и решение закончено, иначе переходим к 1-му пункту алгоритма.

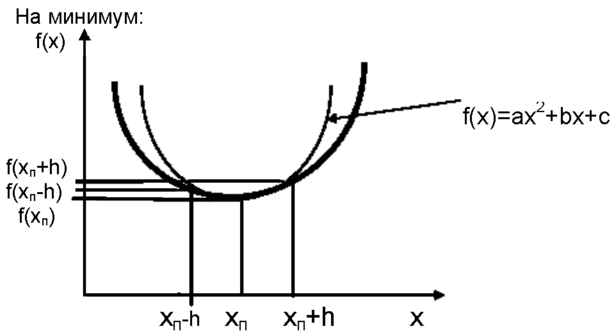


Рисунок 3.8 – Графическая интерпретация метода на основе квадратичной аппроксимации

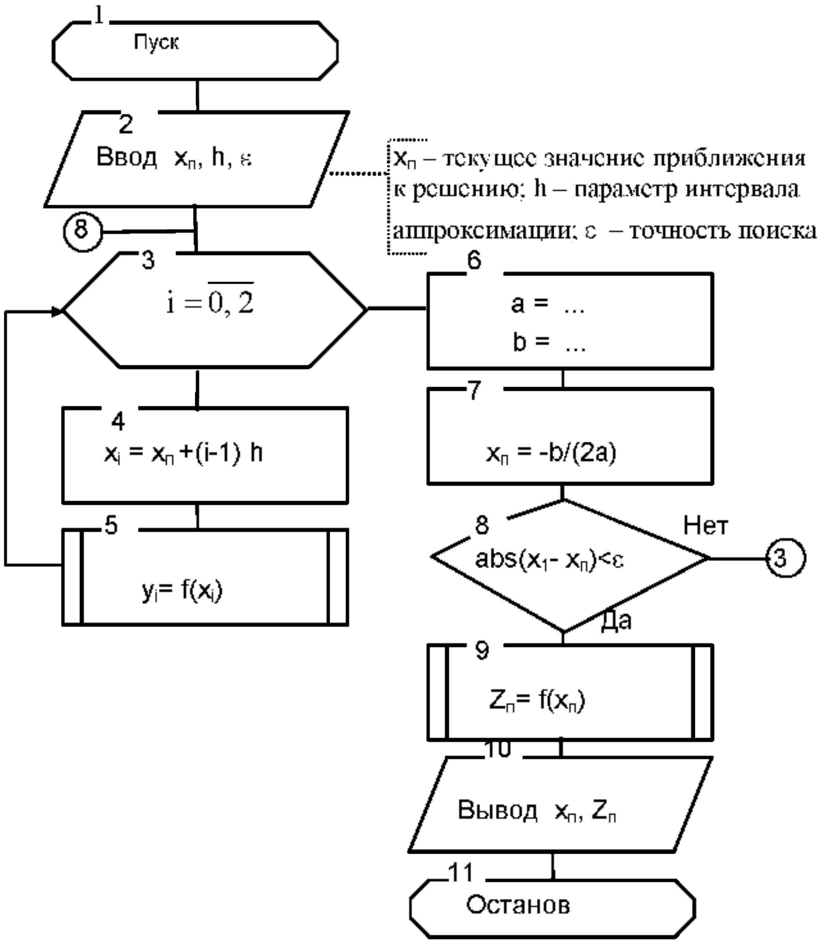


Рисунок 3.9 – Алгоритм на основе квадратичной аппроксимации

Методы случайного поиска основаны на формировании на отрезке поиска случайным образом расчетных точек, вычислении в этих точках значений функции и нахождении точки, соответствующей экстремуму целевой функции. Точность определяется числом точек поиска (числом испытаний) n.

Повторение испытаний описывается формулой Бернулли (биноминальным распределением)

$$P(k) = C_n^k \, p^k \, q^{n-k}, \; k = 0, 1, 2, \ldots, n,$$

где k – число благоприятных случаев;

n – общее число испытаний;

p – вероятность благоприятного исхода при одном испытании;

q – вероятность, противоположная p (q=1-p).

Вероятность, что событие наступит n раз

$$P(n) = p^n \; ;$$

что не наступит ни разу

$$P(0) = (1 - p)^n \; ;$$

наступит хотя бы один раз

$$P_1 = 1 - p(0) = 1 - (1 - p)^n \; .$$

Если абсолютную точность поиска решения на участке (b - a) обозначить через ε, то вероятность решения при одном испытании p = ε/(b - a). При этом вероятности P_1 получения решения с заданной точностью ε в зависимости от числа испытаний n при различных p приведены в таблице 3.1.

Таблица 3.1 – Оценка точности поиска

p	n	p_1
0.1	10 20 50 100	0.651 0.878 0.995 0.99999
0.01	100 200 500 1000	0.634 0.866 0.993 0.99996
0.0001	10000 20000 50000 100000	0.632 0.865 0.993 0.99996

Например, если p = ε /(b - a)=0.01, то вероятность получения решения с точностью ε при числе испытаний n = 100 равна 0.634; при n =200 – 0.866; при n = 500 – 0.993; при n = 1000 – 0.99996, т.е. требуется 10 (b - a) /ε испытаний для надежного решения (P_1> 0.9999) с заданной точностью ε. Аналогичная зависимость, как видно из таблицы, имеет место и при других точностях решения.

Имеется большое множество методов оптимизации на основе случайного поиска и их разновидностей. Ниже приведена иллюстрация простейшего метода поиска (рисунок 3.10), состоящая в последовательном формировании на отрезке от a до b случайным образом расчетной точки $x_т$, расчете в ней целевой функции $f(x_т)$, сравнении значения функции $f(x_т)$ и ранее найденного "лучшего" значения функции $f(x_п)$, присвоении $x_п = x_т$ и $f(x_п) = f(x_т)$, если текущая точка "лучше". Формирование случайной точки производится по выражению $x_т = a + r \cdot (b-a)$, где r – случайное число, равномерно распределенное в интервале от 0 до 1.0.

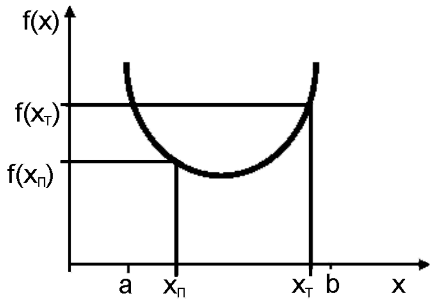


Рисунок 3.10 – Графическая интерпретация метода случайного поиска (на минимум)

Более эффективным является метод случайного поиска с пересчетом, алгоритм которого приведен на рисунке 3.11. Принимаемые значения показателей $a_ш$, $b_ш$, h и L должны находится в определенном соотношении, например, начальные значения могут быть приняты как $b_ш = 2$; $h= b_ш /5$; $a_ш = 0.25$; L=10. Формирование случайной расчетной точки $x_т$ (блок 9) производится относительно текущего приближения $x_п$ на основе использования случайных чисел r, равномерно распределенных в интервале от 0 до 1.0. Произведение sh определяет отрезок, в пределах которого формируется случайная точка.

3.2. Многомерная безусловная оптимизация

Многомерная оптимизация заключается в поиске экстремума функции нескольких (n) переменных $Z= f(X) = f(x_1, x_2,..., x_n)= \min_{\{x_i\}}(\max)$, $i = \overline{1, n}$.

Для решения данной задачи применяются аналогично одномерной оптимизации аналитический метод, численные методы и методы случайного поиска.

Аналитический метод основывается на 1-х и 2-х частных производных по оптимизируемым переменным:

$$\partial Z/\partial x_i = 0 \},\quad i = \overline{1, n} .$$

Решение системы уравнений $\partial Z/\partial x_i = 0 \}$, $i = \overline{1, n}$ дает оптимальное решение $X_п$, если оно существует.

Вид экстремума определяется значением определителя det G матрицы Гессе

$$G = \partial^2 Z/(\partial x_i \partial x_j) ,\ i = \overline{1, n},\ j = \overline{1, n} .$$

Если решение существует в точке $X_п$ и в этой точке det G > 0, то оптимизируемая функция имеет минимум, а иначе максимум.

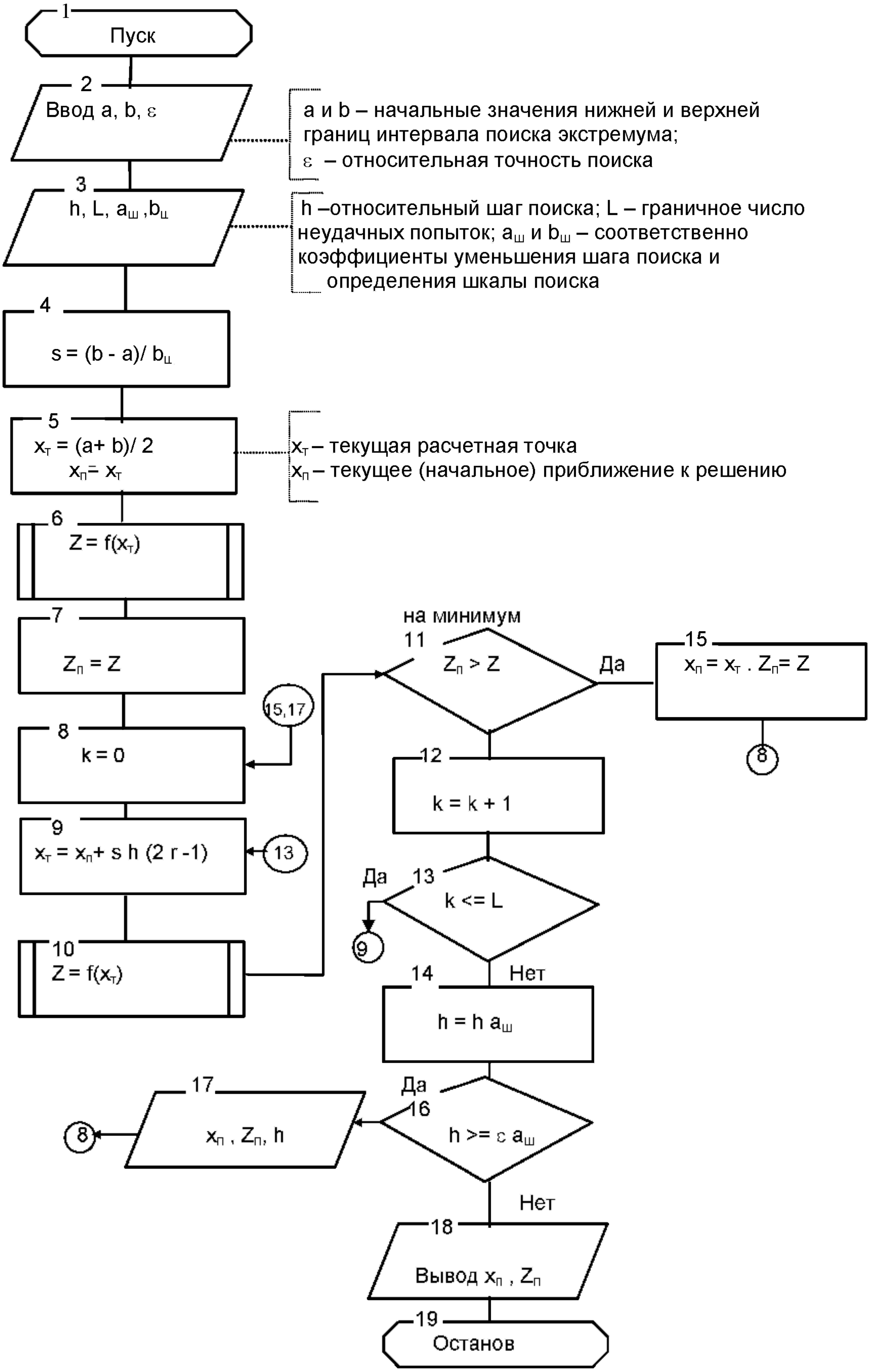


Рисунок 3.11 – Алгоритм одномерного случайного поиска экстремума с пересчетом

Среди численных методов различают методы нулевого порядка и градиентные (1-го и 2-го порядка). Первые основаны на вычислениях только значений оптимизируемой функции. Вторые используют частные производные соответствующего порядка. Эффективность процедуры поиска оптимума (возможность отыскания решения и скорость сходимости к решению) зависит от применяемого метода оптимизации и вида целевой функции.

Наиболее известны следующие методы ***нулевого порядка***:

координатного спуска – поочередная оптимизация переменных вдоль осей одним из известных одномерных методов (одна из реализаций – метод Гаусса-Зейделя на основе шагового метода);

координатного спирального спуска;

вращающихся координат (метод Розенброка);

конфигураций;

Хука-Дживса с поиском по образцу;

параллельных касательных (метод Пауэлла);

Нелдера-Мида – поиска по деформируемому многограннику за счет его отражения, растяжения и сжатия и др.

Наиболее известными являются такие ***градиентные методы*** как наискорейшего спуска и Давидона-Флетчера-Пауэлла (ДФП) с использованием кубической интерполяции.

Для ***случайного поиска*** применяются метод с пересчетом, метод с парными пробами, метод по наилучшей пробе и др.

Эффективная оптимизационная процедура должна успешно решать тестовые задачи, в качестве которых применяются:

функция Розенброка

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

$$X_п = (1;\ 1);$$

функция Пауэлла

$$f(X) = (x_1 + 10x_2^2)^2 + 5\,(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10\,(x_1 - x_4)^4$$

$$X_п = (0;\ 0;\ 0;\ 0);$$

двумерная экспоненциальная функция :

$$f(x_1, x_2) = \sum_{i=1}^{k} \left((e^{-kx_1} - e^{-kx_2}) - (e^{-k} - e^{-10k})\right)^2 ,$$

$$при\ k = 1 \quad X_п = (1;10).$$

Один из методов нулевого порядка – метод **координатного спирального спуска** основывается на поочередном приближении к решению с текущим значением шага по всем оптимизируемым переменным (рисунок 3.12). Если с текущим шагом в данном направлении поиска по всем переменным происходит "ухудшение" целевой функции, то шаг уменьшается и направление поиска меняется на противоположное. Коэффициент $a_ш$ рекомендуется принимать равным 0.25 – 0.40. Вычисления продолжаются до тех пор, пока шаг по модулю не станет равным или менее заданной точности решения. Алгоритм метода координатного спирального спуска приведен ниже.

Метод координатного спирального спуска недостаточно эффективен для поверхностей с "оврагами", так как в этом случае получение решения с требуемой точностью не гарантировано. Это вызвано тем, что в случае "оврага", повернутого относительно осей, попытка продвижения в любом направлении может вызывать "ухудшение" целевой функции. В то же время продвижение вдоль "оврага" может давать "улучшение" целевой функции. Для таких случаев необходимо применять другие методы, например, метод Розенброка или метод Пауэлла.

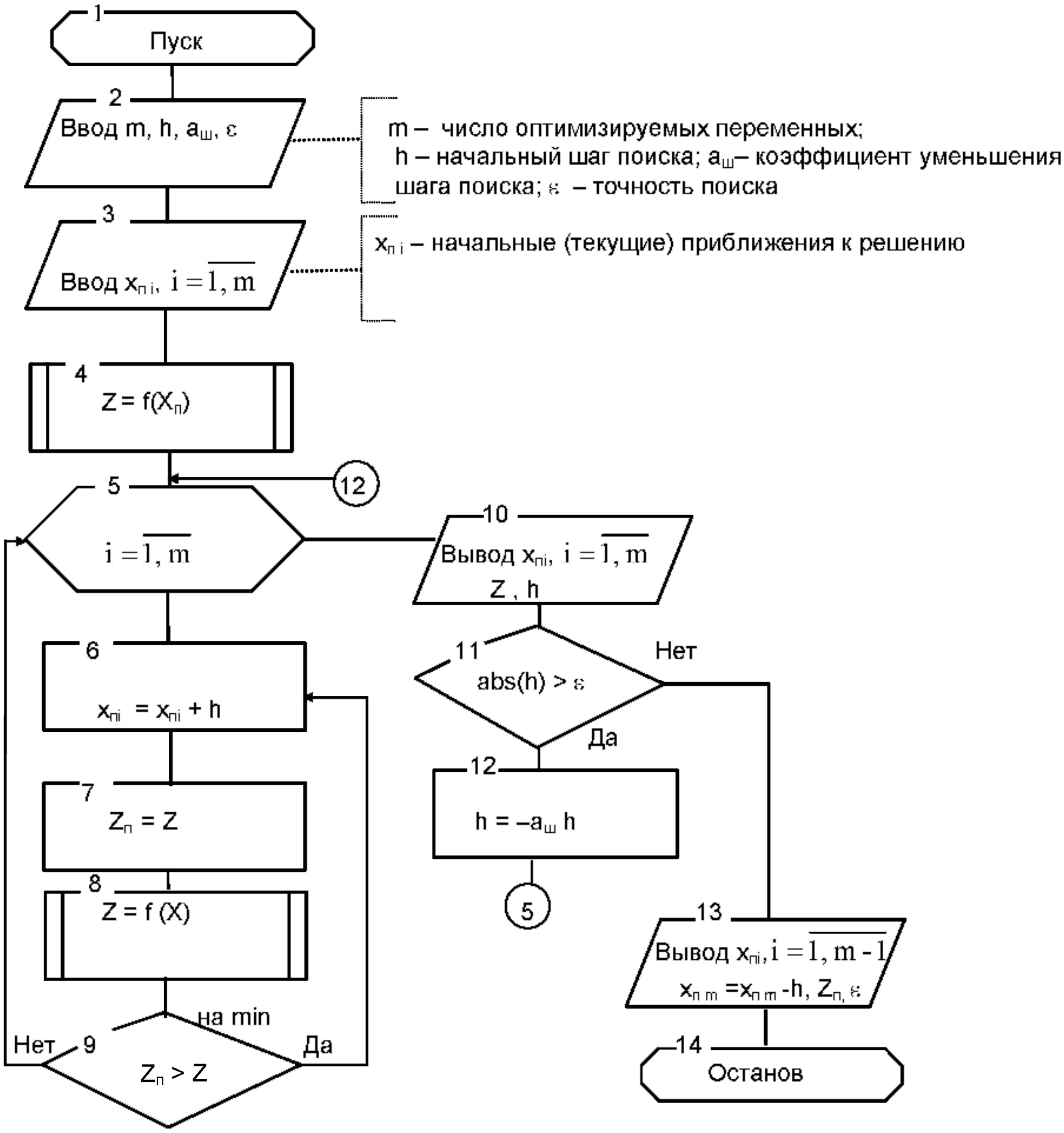


Рисунок 3.12 – Алгоритм координатного спирального спуска

Метод вращающихся координат (метод Розенброка) имеет следующий алгоритм (рисунок 3.13):

1) принимается начальное (текущее) приближение к решению – точка $X_{п1}$ и начальный (текущий) шаг поиска;

2) находится одним из известных методов при текущем значении шага следующее приближение – точка $X_{п2}$;

3) по линии $X_{п1}$ – $X_{п2}$ принимается новое направление поиска (производят поворот осей);

4) модифицируется шаг поиска и вдоль новых осей Z_1 и Z_2 находится следующее текущее приближение;

5) если результат с заданной точностью получен, то решение закончено или иначе по последнему и предпоследнему приближениям находится новое направление поиска и делается переход на пункт 4.

Для выполнения поворота осей используют ортогонализацию по Граму-Шмидту.

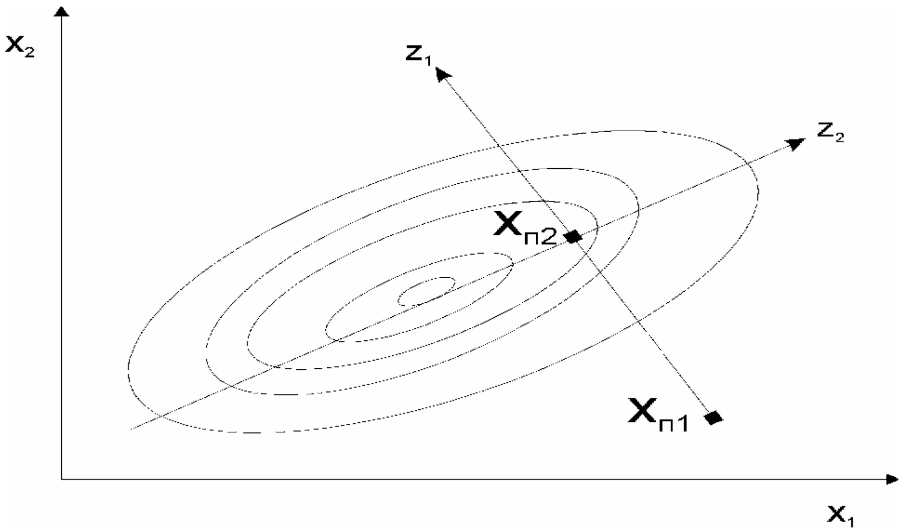


Рисунок 3.13 – Графическая интерпретация метода Розенброка

Метод Пауэлла основан на онределении направлений поиска путем проведения касательных к изолиниям (изоповерхностям), параллельных друг другу. Линия, соединяющая точки касания определяет текущее нанравлепие поиска Z_1. Графическое нредставлепие метода приведено на рисунке 3.14.

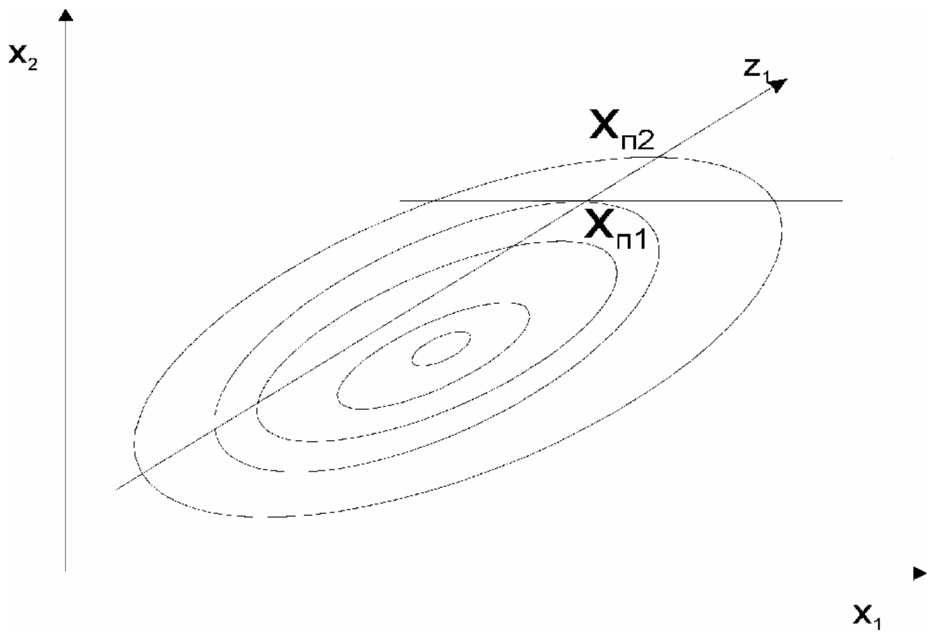


Рисунок 3.14 – Графическая интерпретация метода Пауэлла

Итерациоппые процессы оптимизации, при которых нанравлепие поиска на каждом шаге совпадает с аптиградиептом функции, называются **градиентными методами**. Отличаются способами выбора величины шагов поиска.

Метод наискорейшего спуска – одип из методов с переменным шагом (рисупок 3.15). Шаг принимается с учетом степени изменения целевой функции в направлении поиска:

$$H_i = h\, S_i\, G_i,$$

где i – номер переменной;

H_i – величина шага поиска;

h – относительное значение шага поиска;

S_i – принятая шкальность изменения переменных;

G_i – значение антиградиента функции по x_i.

Шаг H_i остается постоянным до тех пор, пока значение функции убывает (растет).

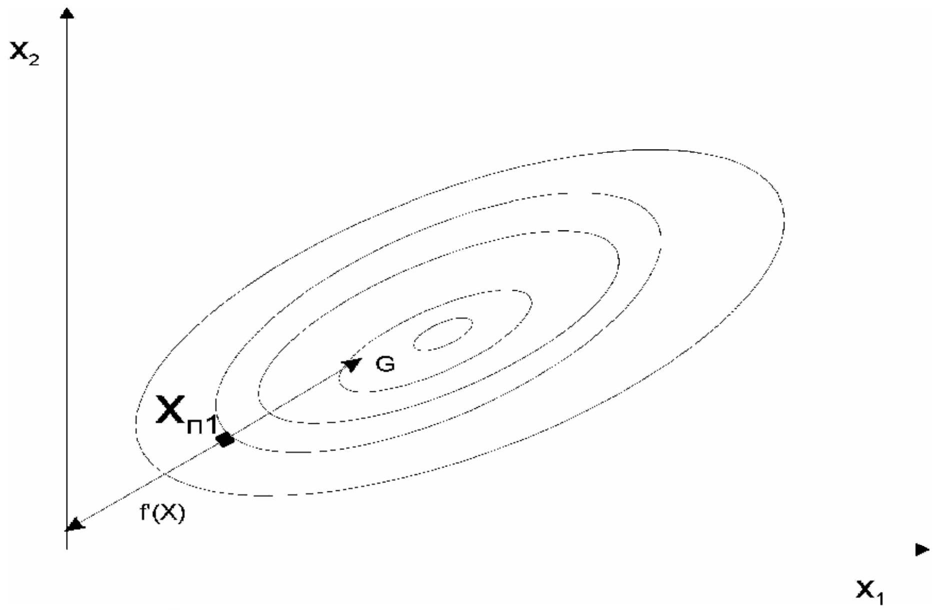


Рисунок 3.15 – Графическая интерпретация метода наискорейшего спуска

Шкальность определяется зависимостью $S_i = (x_в - x_н)_i\, b_ш$, где $x_н$, $x_в$ – соответственно принятая нижняя и верхняя граница интервала поиска; $b_ш$ – коэффициент, определяющий шкальность по переменным.

Имеется ряд методов оптимизации на основе **случайного поиска**. Ниже приведен алгоритм метода случайного поиска с пересчетом. В алгоритме приняты обозначения: r – псевдослучайное число с равномерным законом распределения в интервале от 0 до 1.0; S_i – шкальность изменения x_i ; $X_т = \{x_{тi}\}$ – текущее значение X; $X_п = \{x_{пi}\}$ – текущее приближение к решению X.

3.3. Оптимизация при наличии ограничений

В отличие от безусловной оптимизации в данном случае установлены ограничения в виде равенств и (или) неравенств в зависимости от X.

В этом случае задача состоит в поиске экстремума функции

$$Z = f(X) = \max_{X}(\min), \quad X = \{x_i\}, \ i = \overline{1, m}$$

при выполнении ограничений вида

$$O_j = \varphi_j(X) <=> b_j, \ j = \overline{1, n}.$$

Аналитическое решение задачи при ограничениях типа равенства и дифференцируемости оптимизируемой функции возможно по методу Лагранжа. Для этого вводятся множители Лагранжа λ_j и с их применением формируется функция L по выражению:

$$L = f(X) - \sum_{j=1}^{n} \lambda_j \varphi_j(X),$$

где $\varphi_j(X)=0$.

Оптимальные значения вектора X определяются системой m+n уравнений:

$\partial L/\partial x_i = 0$ } m – уравнений

$\partial L/\partial \lambda_j$ } n – уравнений

Пример.

$Z = 0.5\,(x_2 - x_1)^2 + (1 - x_1)^2 = \min$

$x_1 + x_2 = 4$ или

$x_1 + x_2 - 4 = 0$

$L = 0.5\,(x_2 - x_1)^2 + (1 - x_1)^2 - \lambda\,(x_1 + x_2 - 4) = 0$

$$\begin{cases} \partial L/\partial x_1 = 0.50 \cdot 2 \cdot (x_2 - x_1)(-1) + 2 \cdot (1-x_1)(-1) - \lambda = 0 \\ \partial L/\partial x_2 = 0.50 \cdot 2 \cdot (x_2 - x_1) - \lambda = 0 \\ \partial L/\partial \lambda = -(x_1 + x_2 - 4) = 0 \end{cases}$$

Решение системы уравнений дает следующее решение:

$x_1 = 1.67$; $x_2 = 2.33$ и $\lambda = 0.67$.

Без учета ограничений аналитический метод дает решение в точке $x_1 = 1.0$; $x_2 = 1.0$.

Графическое представление полученных решений приведено на рисунке 3.16.

В случае, если ограничения имеют вид неравенств, необходимые условия нахождения экстремума (минимума) функции f(X) при ограничениях $\varphi_j(X) < b_j$ следующие:

ограничения в виде неравенств преобразуются в ограничения в виде равенств путем добавления дополнительных (ослабляющих) переменных u_i^2 ($u_i^2 > 0$) к виду:

$$\varphi_j^* = \varphi_j(X) - b_j + u_j^2 = 0 \;;$$

формируется функция L с применением множителей Лагранжа

$$L = f(X) - \sum_{j=1}^{n} \lambda_j \varphi_j^* \;;$$

и система уравнений, решение которой определяет точку оптимума

$\partial L/\partial x_i = 0$ } m – уравнений

$\partial L/\partial \lambda_j$ } n – уравнений

$\partial L/\partial u_j$ } n – уравнений.

Условия решения такой задачи известны как условия Куна-Такера.

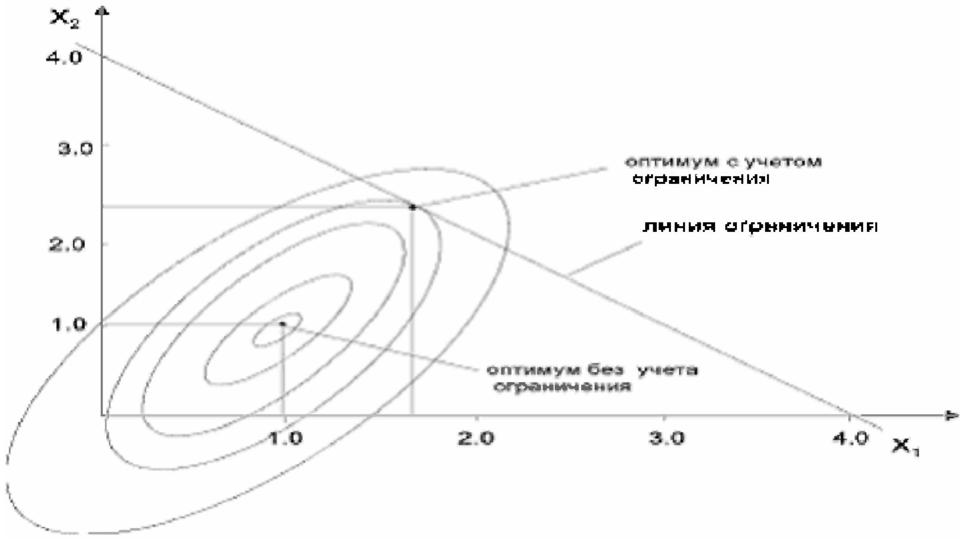


Рисунок 3.16 – Графическое представление решения по методу Лагранжа

При использовании для условной оптимизации численных методов **нулевого порядка** может применяться прямой учет ограничений. Для этого достаточно при оптимизации присваивать целевой функции значение далекое от экстремального, если ограничения нарушаются, например, при поиске минимума – бесконечно большие значения. Однако такой метод не всегда обеспечивает получение решения с требуемой точностью, т.к. ограничения создают "овражность".

Для получения эффективных и надежных методов требуются более сложные процедуры. Одним из методов решения является модификация метода Нелдера-Мида, разработанная Боксом (назван комплексным методом). Метод основан на выборе 2m точек, удовлетворяющих ограничениям, и называемых комплексом.

Метод прямого учета ограничений успешно может быть применен при оптимизации **случайным поиском** (рисунок 3.17). Для **прямого учета ограничений** достаточно в подпрограмме расчета целевой функции присваивать ей, если хотя бы одно из ограничений нарушено, бесконечно большое (при минимизации) или бесконечно малое (при максимизации) значение. При этом данная попытка при случайном поиске не должна учитываться счетчиком неудачных проб. Если ни одно из ограничений не нарушено, то вычисляется реальное значение целевой функции.

Например, на Бейсике подпрограмма без учета ограничений и с учетом ограничений будет при оптимизации на минимум следующей:

```
m1:
' п/п вычисления Z без учета ограничений
Z= …
Return
```

```
m1:
' п/п вычисления Z с учетом ограничений
if (ограничение 1 нарушено) then m2
if (ограничение 2 нарушено) then m2
……….
if (ограничение n нарушено) then m2
Z= …:goto m3
m2:
Z= 1E20
m3:
return
```

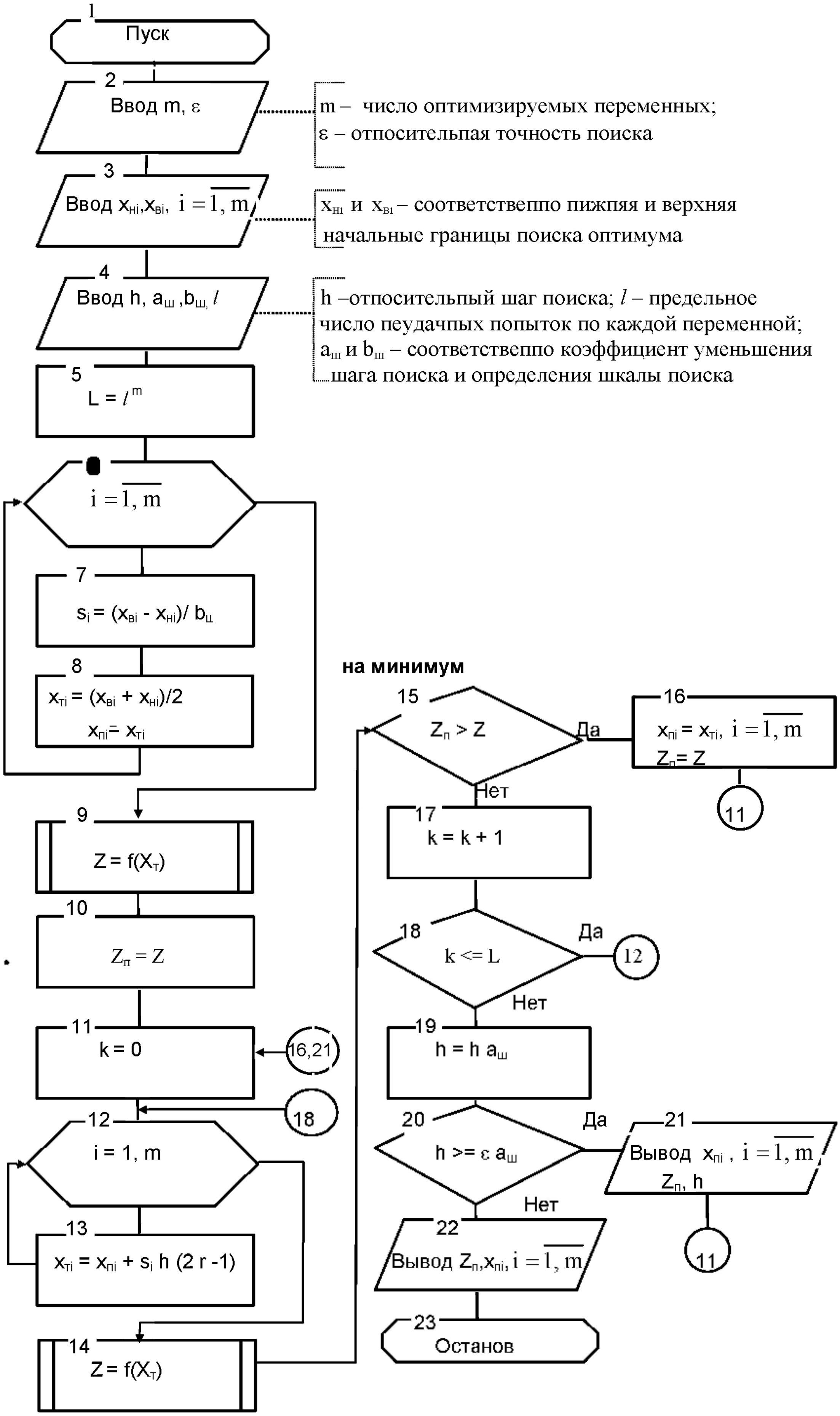


Рисунок 3.17 – Алгоритм многомерной оптимизации случайным поиском с пересчетом

Задача с ограничениями может быть сведена к задаче без ограничений с помощью **"штрафных" функций**. Идея внутренних штрафных функций состоит в формировании целевой функции $Z_ш$ по $Z = f(X) = \min$ и ограничений $\varphi_j(X) > 0$ ($j = \overline{1, n}$):

$$Z_ш = f(X) + P_ш = \min; \quad P_ш = r\sum_{j=1}^{n} 1/\varphi_j(X);$$

$$Z_ш = f(X) + r\sum_{j=1}^{n} 1/\varphi_j(X) = \min,$$

где r – положительная малая величина; $P_ш$ – дополнительная штрафная функция.

Если значения X близкие или равные ограничению, то происходит резкое изменение (увеличение) функции $Z_ш$ (образуется "гребень" с крутыми краями). Графическая интерпретация приведена ниже на рисунке 3.18.

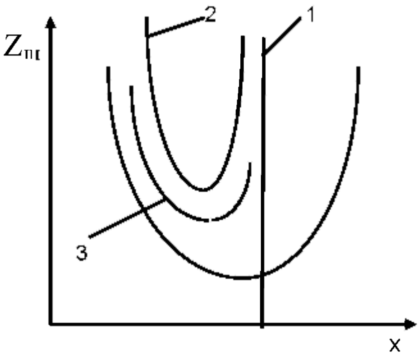


Рисунок 3.18 – Графическая интерпретация метода штрафных функций:

1 – линия ограничения; 2 – функция $Z_ш$ при большем значении r; 3 – функция $Z_ш$ при меньшем значении r

Значения **r** принимаются малыми, чтобы влияние дополнительной функции $P_ш$ было меньшим в точке оптимума. При меньших значениях **r** хуже сходимость, при больших значениях – ниже точность решения. Поэтому **r** необходимо изменять в ходе оптимизации.

Существует метод внешних штрафных функций, когда штрафуется удаление от допускаемой области.

3.4. Задача линейного программирования

Задача линейного программирования – это оптимизационная задача с линейной целевой функцией и линейными ограничениями в виде равенств или неравенств:

целевая функция

$$Z = \sum_{i=1}^{m} c_i x_i = \max_{x_i}(\min), \; i = \overline{1, m}$$

и ограничения

$$\sum_{i=1}^{m} a_{ij} x_i \le = \ge b_j, \quad j = \overline{1, n}$$

$$x_i > 0,$$

где $x_i,\ i = \overline{1, m}$ – вектор оптимизируемых управляемых параметров;

c_i – удельные эффект или затраты, приходящиеся на единицу x_i;

a_{ij} и b_i – параметры (коэффициенты) ограничений;

m – общее число оптимизируемых параметров;

n – общее число ограничений.

Такой вид имеет формализованная постановка общей задачи линейного программирования.

Пример.

Автомобиль при работе на объекте 1 имеет производительность 20 единиц, на объекте 2 – 25 единиц. На этих объектах необходимо освоить объем перевозок не более чем 175 единиц. Общее число автомобилей, задействованных на перевозках не должно превышать 8 единиц. Эффект от работы автомобиля на 1-м объекте составляет 5 единиц; на 2-м – 6 единиц. Требуется найти оптимальный вариант распределения автомобилей по объектам перевозок.

Пусть x_1 – число автомобилей для работы на 1-м объекте и x_2 – на 2-м объекте.

Тогда задача в математической форме имеет вид:

$$\left.\begin{array}{l} 20x_1 + 25x_2 <= 175 \\ x_1 + x_2 <= 8 \\ x_1 >= 0 \\ x_2 >= 0 \end{array}\right\} \text{ограничения}$$

$5x_1 + 6x_2 = \max$ – целевая функция.

Двухмерная задача линейного программирования может быть решена графическим методом. Этот метод основан на получении выпуклого многоугольника допустимых решений и нахождении на нем точки (точек) экстремума.

Алгоритм **графического метода** следующий:

1) строятся линии ограничений и исключаются из рассмотрения запрещенные области по всем ограничениям. В результате получается выпуклый многоугольник допустимых решений;

2) строится изолиния целевой функции для произвольного ее значения;

3) изолиния целевой функции перемещается параллельно таким образом, чтобы она только касалась полученного многоугольника допускаемых решений по точке или линии при экстремуме целевой функции.

Рассмотрим графическое решение на вышеприведенном примере (рисунок 3.19).

Строим линии ограничений и исключаем запрещенные области.

1 ограничение:

$x_1 = 0;\ x_2 = 175/25 = 7;$

$x_2 = 0;\ x_1 = 175/20 = 8.75.$

2 ограничение:

$x_1 = 0;\ x_2 = 8;$

$x_2 = 0;\ x_1 = 8.$

Кроме того, ограничениями являются оси координат, так как $x_1 > 0;\ x_2 > 0$.

Полученная незапрещенная область (на рисунке затемненная) – область допустимых решений.

Строим изолинию целевой функции.

Пусть Z = 30, тогда

$x_1 = 0;\ x_2 = 5$;

$x_2 = 0;\ x_1 = 6.$

Перемещая изолинию параллельно все выше и выше, находим наибольшее значение Z, когда изолиния только касается многоугольника допустимых решений и имеется хотя бы одна общая точка с ним. Точка "B", в которой $x_1 = 5$ и $x_2 = 3$ соответствует оптимальному решению поставленной задачи. Если линия уровня целевой функции касалась бы многоугольника допустимых решений не в точке, а по линии – то мы имели бы случай, когда задача имеет множество решений. Если ограничения задачи противоречивы, то задача не имеет решения (например, 1-е ограничение $20\ x_1 + 25\ x_2 \geq 500$).

При изменениях коэффициентов целевой функции оптимальное решение может изменяться. Например, при целевой функции $4\ x_1 + 6\ x_2 = \max$ решение в точке А, при функции $5x_1 + 6.25\ x_2 = \max$ – множество точек отрезка [A,B], при функции $5x_1 + 5\ x_2 = \max$ – множество точек отрезка [B, C] и при функции $10x_1 + 6\ x_2 = \max$ решение в точке С.

При числе оптимизируемых параметров более двух мы имеем дело с выпуклым многогранником ограничений (при m = 3 ограничения определяются плоскостями, допустимая область – выпуклым многогранником, уровень целевой функции – плоскостью, решение – касание допустимой области плоскостью целевой функции).

Для решения задачи в m-мерном пространстве применяется симплекс-метод.

Идея метода – последовательный пересмотр вершин многогранника допустимых значений и выбор одной из них, соответствующей экстремуму целевой функции.

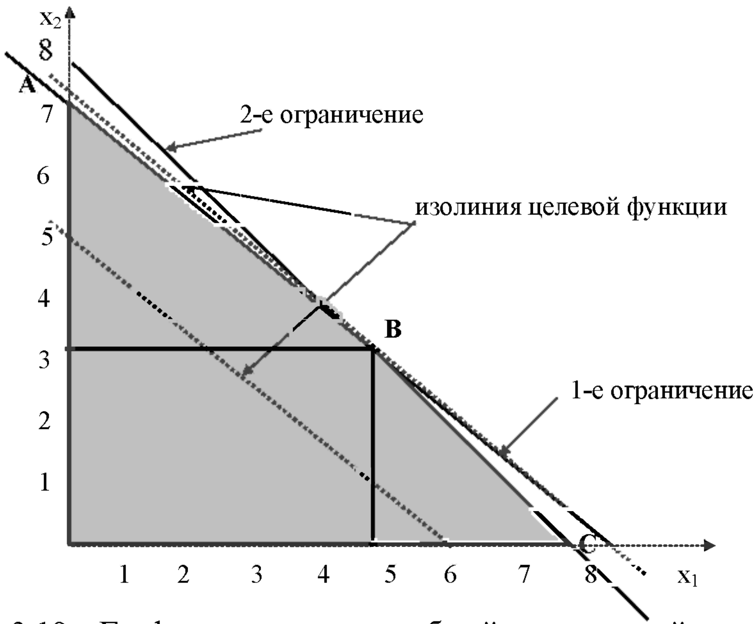


Рисунок 3.19 – Графическое решение общей задачи линейного программирования

Для применения симплекс-метода исходная задача приводится к каноническому (стандартному) виду путем ввода дополнительных переменных по числу ограничений в виде неравенств из общего числа n. Цель – заменить ограничения типа неравенств ограничениями типа равенств. Стандартная форма задачи имеет вид:

целевая функция

$$Z = \sum_{i=1}^{M} c_i x_i = \max (\min),$$

$$c_i = 0 \text{ для } i = \overline{m+1, M};$$

ограничения

$$\sum_{i=1}^{m} a_{ij} x_i + \sum_{i=m+1}^{M} d_{ij} x_i = b_j, \ j = \overline{1, n},$$

где M = m+n

и

$$d_{ij} = \begin{cases} +1\,(-1)\,\text{если } i = m+j \\ 0, \text{если } i \neq m+j \end{cases}.$$

Для неравенств типа $\leq$ $d_{ij} = 1$ и для типа $\geq$ $d_{ij} = -1$.

Например, стандартная форма для ранее приведенной задачи следующая:

$$20\,x_1 + 25\,x_2 + 1\,x_3 + 0\,x_4 = 175$$
$$x_1 + x_2 + 0\,x_3 + 1\,x_4 = 8$$
$$Z = 5\,x_1 + 6\,x_2 + 0\,x_3 + 0\,x_4 = \max.$$

Основные шаги решения задачи (после представления исходной системы в стандартном виде):

1) формируется первоначальное базисное решение;
2) выражается Z через небазисные переменные;
3) проверяется базисное решение на оптимальность. Если оптимально, то на п.10;
4) проверяется задача на наличие решения. Если решения нет, то выход;
5) выбирается из небазисных переменная, которая способна при введении ее в базис в большей степени улучшить значение целевой функции, и вводится в базис;
6) определяется базисная переменная, которая должна выводится из базиса;
7) алгебраически выражается вводимая в базис переменная через переменную, выводимую из базиса и другие небазисные переменные;
8) алгебраически выражаются другие базисные переменные через небазисные;
9) переход на п. 2;
10) определяются значения базисных переменных. Они являются решением задачи.

Итерационный процесс (шаги 2–9) повторяются до тех пор, пока не произойдет выход на шаге 3 или 4.

Алгоритм реализации отдельных шагов при решении задачи на **максимум** и ограничениях типа $\leq$ следующий:

Шаг 1 состоит в назначении базисных переменных по числу ограничений в задаче. На данном этапе решения в качестве базисных принимаются дополнительные переменные, а в качестве небазисных – основные, т.е. $u = \overline{m+1, M}$ и $p = \overline{1, m}$. Базисные переменные x_u выражаются через небазисные x_p из равенств, в которые они входят через значимые коэффициенты. При этом небазисные переменные принимаются нулевыми. Тогда первое базисное решение

$$x_{m+1} = b_1;$$
$$x_{m+2} = b_2;$$
$$\ldots$$
$$x_M = b_n;$$
$$x_1 = 0;$$
$$x_2 = 0;$$
$$\ldots$$
$$x_m = 0.$$

Шаг 2 – алгебраически выражается целевая функция Z через небазисные переменные

$$Z = \sum_p c_p x_p = \max .$$

Шаг 3 – проверяется все ли $c_p \le 0$. Если да, то решение оптимально.

Шаг 4 – оценка наличия решения. Если при x_p, имеющем $c_p > 0$, во всех уравнениях $a_{pj} < 0$ (j=1, 2, ..., n), то решение отсутствует (выход из программы с соответствующим сообщением).

Шаг 5 – определяется та небазисная переменная, которую наиболее целесообразно ввести в базис, по максимуму положительного коэффициента в текущем выражении для целевой функции $\max_p c_p = c_s$, где s– номер переменной, вводимой в базис.

Шаг 6 – определяется одна из базисных переменных для вывода ее из базиса. Для этого во всех текущих j-х равенствах для x_s вычисляется отношение свободного члена b_j к соответствующему коэффициенту a_{sj} ($a_{sj} > 0$), т.е. b_j/a_{sj}. Минимальное из отношений указывает на j-е равенство и соответственно на выводимую переменную из базиса.

Шаг 10 – формируется окончательное решение в виде численных значений искомых переменных, которые входят в базис. Вычисляются из последних выражений для них при значениях небазисных переменных, равных нулю. С практической точки зрения определение численных значений базисных переменных, которые являются дополнительными, не требуется. Основные переменные, которые не входят в базис, равны нулю.

Для ручного решения задачи могут применяться симплекс-таблицы.

Решение задачи при ограничениях одновременно типа $<$, $=$ и $>$ отличается от приведенного алгоритма введением искусственных переменных [2].

Рассмотрим решение симплекс-методом ранее приведенного примера задачи.

1-я итерация:

1) базисные переменные $x_3 = 175$ и $x_4 = 8$ и небазисные $x_1 = 0$ и $x_2 = 0$;
2) целевая функция $Z = 5x_1 + 6x_2$;
3) поскольку 5>0 и 6>0, то решение неоптимально;
4) так как $a_{1.1} > 0$, $a_{1.2} > 0$, $a_{2.1} > 0$ и $a_{2.2} > 0$, то не следует, что решения нет;
5) поскольку 6>5 и поиск на max, то вводим в базис x_2;
6) исходя из минимума соотношений (175/25=7 в первом уравнении и 8/1=8 во втором) необходимо выводить из базиса переменную x_3 (1-е уравнение);
7) выражаем $x_2 = (175-20 \cdot x_1 - x_3)/25 = 7-0.8 \cdot x_1 - 0.04 \cdot x_3$ (уравнение $0.8 \cdot x_1 + 0.04 \cdot x_3 + x_2 = 7$);
8) выражаем другие базисные через небазисные $x_4 = 8 - x_1 - x_2 = 8 - x_1 - (7 - 0.8 \cdot x_1 - 0.04\,x_3) =$ $= 1 - 0.2\,x_1 + 0.04\,x_3$ (уравнение $0.2\,x_1 - 0.04\,x_3 + x_4 = 1$);
9) переход на шаг 2.

2-я итерация:

2) целевая функция $Z = 5x_1 + 6x_2 = 5x_1 + 6(7 - 0.8 \cdot x_1 - 0.04\,x_3) = 42 + 0.2 \cdot x_1 - 0.24\,x_3$;

3) поскольку 0.2>0, то решение неоптимально;

4) так как при x_1 ($c_1 > 0$) $a_{1.1} > 0$ и $a_{1.2} > 0$ (для текущих уравнений, приведенных к основному виду со свободным членом в правой части), то не следует, что решения нет;

5) поскольку 0.2>-0.24 и поиск на max, то вводим в базис x_1;

6) исходя из минимума соотношений 7/0.8=8.75 в первом уравнении и 1/0.2=5 во втором необходимо выводить из базиса переменную x_4;

7) выражаем x_1: $x_1 = (1 + 0.04\,x_3 - x_4)/0.2 = 5 + 0.2\,x_3 - 5 \cdot x_4$;

8) выражаем другие базисные переменные через небазисные

$x_2 = 7 - 0.8\,x_1 - 0.04\,x_3 = 7 - 0.8 \cdot (5 + 0.2\,x_3 - 5\,x_4) - 0.04\,x_3 = 3 - 0.20\,x_3 + 4\,x_4$;

9) переход на шаг 2.

3-я итерация:

2) целевая функция $Z = 5x_1 + 6x_2 = 5(5 + 0.2\,x_3 - 5 \cdot x_4) + 6(3 - 0.20\,x_3 + 4 \cdot x_4) = 43 - 0.2\,x_3 - 1 \cdot x_4$;

3) поскольку -0.2 <0 и -1 <0, то решение оптимально;

10) базисные переменные $x_1 = 5$ и $x_2 = 3$.

Таким образом, как и при графическом методе, получаем, что оптимальное решение: $x_1 = 5$ и $x_2 = 3$. При этом значение целевой функции равно Z=43.

Компьютерная программа решения задачи линейного программирования на основе одного из алгоритмов симплекс-метода [2] приведена в приложении 6.

3.5. Отыскание кратчайших расстояний и путей между пунктами транспортной сети. Кратчайшая связывающая сеть

Транспортная сеть района (региона) представляет систему путей сообщения, которые предназначены для передвижения транспортных средств. Модель транспортной сети – это описание местонахождения ее вершин (пунктов), а также параметров (характеристик) соединяющих вершины звеньев. Модель может быть задана в графическом или табличном виде. Звеном транспортной сети является ее часть, соединяющая две смежные вершины. Параметр звена может быть выражен в единицах длины, приведенных единицах длины, временных и денежных затратах на движение по ним (в дальнейшем – длина). В качестве дополнительных характеристик звеньев может быть задана техническая категория пути сообщения, скорость движения по ним и др. Вершины транспортной сети соответствуют местонахождению ресурсообразующих и ресурсопоглащающих точек, терминалов, пересечений путей.

Длина звеньев транспортной сети определяется по справочным таблицам, атласам дорог, с помощью курвиметра по масштабным картам и планам, а также непосредственным замером на местности, например по показаниям одометра (счетчика пройденного пути) автомобиля.

Отыскание кратчайших расстояний между пунктами транспортной сети возможно по методу потенциалов, методу "метлы", динамическому методу, на основе теории графов (метод Флойда) и др.

Например, задача отыскания кратчайших расстояний между пунктами транспортной сети **методом потенциалов** решается по следующему алгоритму:

1) начальному пункту, от которого требуется определить кратчайшие расстояния, присваивается потенциал $v_i = 0$.

2) находятся все звенья, для которых начальным пунктам i присвоены потенциалы v_i, а конечным пунктам j не присвоены. Если таких звеньев нет, то решение закончено (на п.6), а иначе на следующий п.3.

3) для найденных звеньев по п.2 рассчитываются значения потенциалов конечных пунктов j по следующей формуле:

$$u_{j(i)} = v_i + l_{ij} ,$$

где $u_{j(i)}$ – потенциал конечного пункта j звена i-j; l_{ij} – длина звена i-j.

4) из всех рассчитанных потенциалов по п. 3 выбирается потенциал с наименьшим значением, т.е. определяется:

$$\min_{i,j}\{u_{j(i)}\} = u_{r(s)} ,$$

где $\{ u_{j(i)} \}$ – множество значений потенциалов конечных пунктов j звеньев i-j, i-м начальным пунктам которых ранее присвоены потенциалы;

$u_{r(s)}$ – потенциал конечного пункта r звена s-r, являющийся наименьшим по значению элементом множества $\{ u_{j(i)} \}$.

Потенциал $u_{r(s)}$ присваивается соответствующему конечному пункту ($v_r = u_{r(s)}$), а звено s - r отмечается стрелкой.

В случае если несколько значений потенциалов множества $\{u_{j(i)}\}$ окажутся равными и наименьшими, то необходимо установить, относятся они к одному и тому же конечному пункту или нет.

Если наименьшие равные значения потенциалов относятся к различным пунктам r (у потенциалов не совпадают цифровые индексы без скобок), эти значения потенциалов присваиваются всем соответствующим конечным пунктам и стрелками отмечаются соответствующие звенья.

Если наименьшие равные значения потенциалов относятся к одному и тому же конечному пункту r (у потенциалов совпадают цифровые индексы без скобок), то пункту r присваивается это наименьшее значение потенциала и отмечается стрелкой то звено s-r, которому соответствует потенциал $u_{r(s)}$ с большим удельным весом в его составе длин звеньев с лучшими условиями перемещения. При одинаковых дорожных условиях кратчайшее расстояние реализуется по одному из звеньев s-r исходя из других предпочтений.

5) переход на п. 2

6) формируется окончательное решение. Величина потенциалов пунктов показывает кратчайшие расстояния от выбранного начального пункта до этих пунктов, а цепочки звеньев с последовательно входящими друг в друга стрелками образуют кратчайший путь движения от интересующего пункта до исходного. Если любых два пункта сети соединены такой цепочкой, то кратчайшее расстояние между ними равно разности их потенциалов.

Принимая за начальный последовательно каждый из пунктов сети и выполняя действия по вышеописанному методу, получается таблица кратчайших расстояний между всеми пунктами (таблица 3.2). Кратчайшие расстояния необходимо знать для оптимизации грузопотоков, маршрутизации перевозок, закрепления маршрутов за перевозчиками и т.п.

Таблица 3.2 – Кратчайшие расстояния между пунктами транспортной сети (км)

Пункты	1	. . .	j	. . .	n
1	0		l_{1j}		l_{1n}
⋮					
i	l_{i1}		l_{ij}		l_{in}
⋮					
n	l_{n1}		l_{nj}		$l_{nn}=0$

Пример.

Необходимо найти кратчайшие расстояния от пункта 1 до остальных пунктов нижеприведенной транспортной сети (рисунок 3.20).

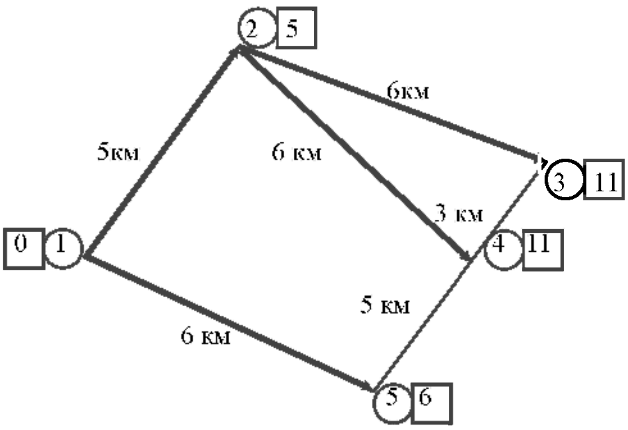


Рисунок 3.20 – Схема транспортной сети

Решение.

Шаг 1. Начальному пункту 1 присваивается потенциал $v_1 = 0$ (потенциалы указаны в квадратах).

Шаг 2.

1-й этап. Звенья, начальным пунктам i которых присвоен потенциал v_i, а для конечных пунктов j потенциалы не присвоены: 1-2 и 1-5. Рассчитываются значения потенциалов конечных пунктов j :

$$u_{2(1)} = v_1 + l_{1\text{-}2} = 0 + 5 = 5$$
$$u_{5(1)} = v_1 + l_{1\text{-}5} = 0 + 6 = 6$$

Из рассматриваемых на данном этапе потенциалов находится минимальный

$$\min_{i,\,j}\{u_{2(1)}, u_{5(1)}\} = u_{2(1)} = 5.$$

Потенциал $u_{2(1)}$ присваивается соответствующему конечному пункту ($v_2 = u_{2(1)} = 5$), а звено 1-2 отмечается стрелкой.

2-й этап. Звенья, начальным пунктам i которых присвоен потенциал v_i, а для конечных пунктов j потенциалы не присвоены: 1-5, 2-3 и 2-4. Рассчитываются значения потенциалов конечных пунктов j :

$$u_{5(1)} = v_1 + l_{1\text{-}5} = 0 + 6 = 6$$
$$u_{3(2)} = v_2 + l_{2\text{-}3} = 5 + 6 = 11$$
$$u_{4(2)} = v_2 + l_{2\text{-}4} = 5 + 6 = 11$$

Из рассматриваемых на данном этапе потенциалов находится минимальный

$$\min_{i,\,j}\{u_{5(1)}, u_{3(2)}, u_{4(2)}\} = u_{5(1)} = 6.$$

Потенциал $u_{5(1)}$ присваивается соответствующему конечному пункту ($v_5 = u_{5(1)}$), а звено 1 - 5 отмечается стрелкой.

3-й этап. Звенья, начальным пунктам i которых присвоен потенциал v_i, а для конечных пунктов j потенциалы не присвоены: 2-3, 2-4 и 5-4. Рассчитываются значения потенциалов конечных пунктов j:

$$u_{3(2)} = v_2 + l_{2\text{-}3} = 5 + 6 = 11$$
$$u_{4(2)} = v_2 + l_{2\text{-}4} = 5 + 6 = 11$$
$$u_{4(5)} = v_4 + l_{5\text{-}4} = 6 + 5 = 11$$

Из рассматриваемых на данном этапе потенциалов находится минимальный

$$\min_{i,\,j}\{u_{3(2)}, u_{4(2)}, u_{4(5)}\} = u_{3(2)} = u_{4(2)} = u_{4(5)} = 11.$$

Минимальное значение потенциалов, равное 11, присваивается соответствующим конечным пунктам $v_3 = 11$ и $v_4 = 11$ (первый частный случай), и звено 2-3, а также, например, 2-4 (одно из двух 2-4 и 5-4 – второй частный случай) отмечаются стрелкой.

4-й этап. Так как потенциалы присвоены всем пунктам, то решение закончено.

Алгоритм метода отыскания кратчайших расстояний, который легко реализуется на компьютере, и поясняющая схема приведены на рисунке 3.21. Реализация данного алгоритма в виде компьютерной программы приведено в приложении 7.

Кратчайшая связывающая сеть (КСС) представляет собой "дерево" графа минимальной длины, т.е. она не должна содержать контуров (циклов) и включать звенья, которые дают суммарную минимальную длину. Применяется для нахождения сети маршрутов минимальной длины, соединяющей заданные пункты (маршруты перевозок пассажиров или мелких партий грузов, сеть путей с определенными параметрами и т.п.).

Один из алгоритмов отыскания кратчайшей связывающей сети следующий:

1) находится звено минимальной длины и включается в КСС;

2) последовательно включаются звенья в КСС:

2.1) находятся звенья, соединенные только одним своим концом (началом) с имеющейся частью КСС. Если нет ни одного такого звена, то решение закончено. Иначе переход на пункт 2.2;

2.2) из найденных по п. 2.1 звеньев определяется одно звено минимальной длины и включается в КСС;

2.3) переход на 2.1.

Кратчайшая связывающая сеть включает ровно m-1 звеньев при числе вершин m.

Пример. Необходимо найти кратчайшую связывающую сеть для ранее приведенной схемы транспортной сети.

Решение.

1) находим из всех звеньев звено минимальной длины и включаем в КСС сеть (звено 3-4);

2) последовательно включаем звенья в КСС:

1-итерация

2.1.1) выбираем звенья, соединенные только одним своим концом с имеющейся частью КСС. Это звенья 3-2, 4-2 и 4-5.

2.2.1) из выбранных звеньев находим звено минимальной длины и включаем в КСС. Это звено 4-5;

2.3.1) переход на 2.1.2)

2- итерация

2.1.2) выбираем звенья, соединенные только одним своим концом с имеющейся частью КСС. Это звенья 3-2, 4-2 и 5-1.

2.2.2) из выбранных звеньев находим звено минимальной длины и включаем в КСС. Поскольку все звенья равной минимальной длины, то выбираем любое из них, например звено 5 - 1;

2.3.2) переход на 2.1.3)

3- итерация

2.1.3) выбираем звенья, соединенные только одним своим концом с имеющейся частью КСС. Это звенья 3-2, 4-2 и 1-2.

2.2.3) из выбранных звеньев находим звено минимальной длины и включаем в КСС. Это звено 1 - 2;

переход на 2.1.4)

4- итерация

2.1.4) выбираем звенья, соединенные только одним своим концом с имеющейся частью КСС. Таких звеньев нет. Решение закончено.

Звенья КСС выделены «жирными» линиями (рисунок 3.22).

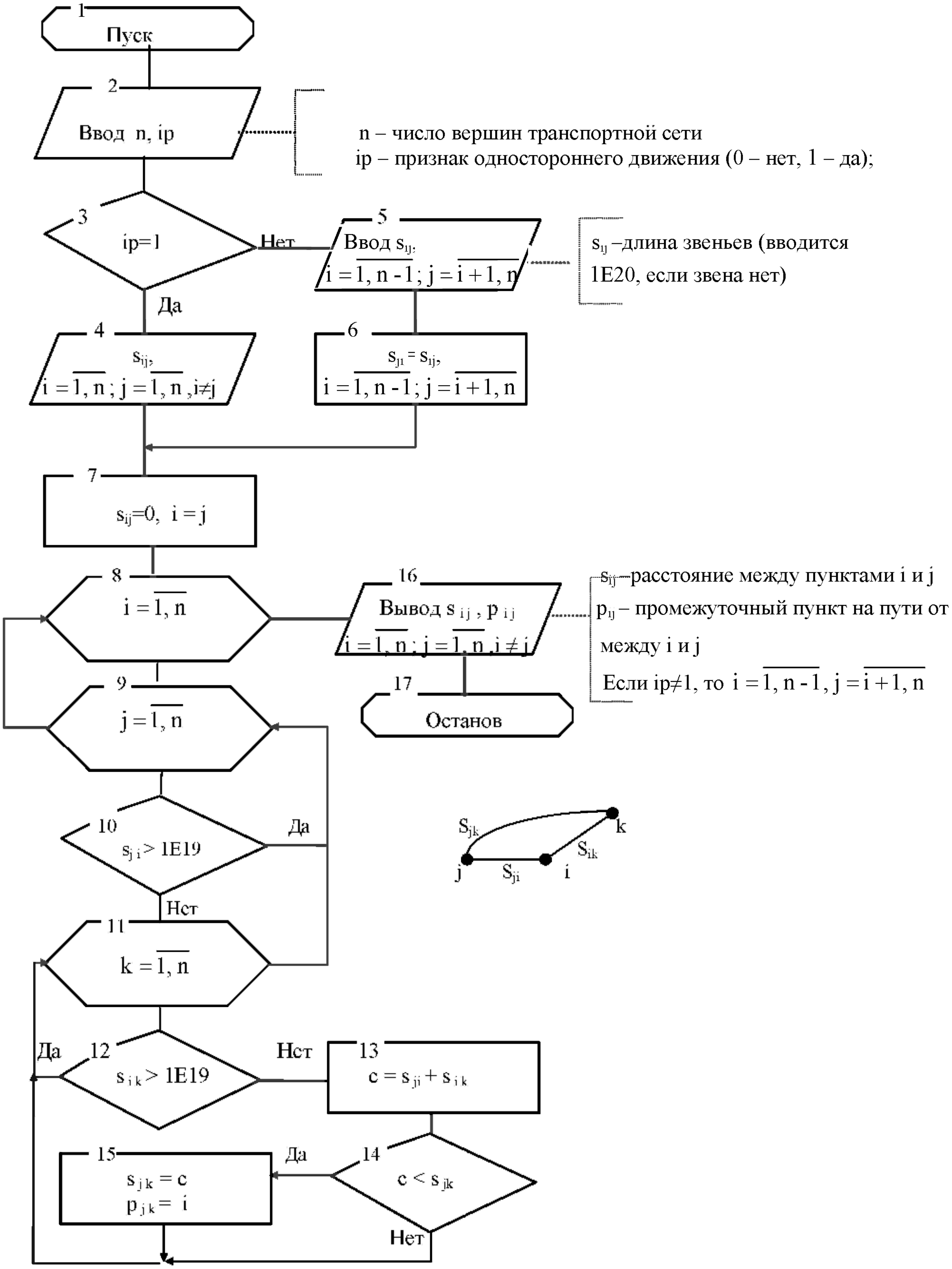


Рисунок 3.21 – Алгоритм поиска кратчайших расстояний между пунктами транспортной сети

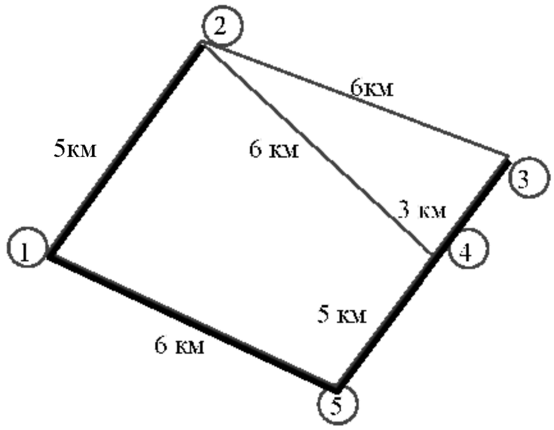


Рисунок 3.22 – Схема кратчайшей связывающей сети («жирные» линии)

3.6. Транспортная задача линейного программирования

При оперативном управлении и перспективном планировании в ряде случаев требуется оптимизация поставок, состоящая в определении наилучшего плана корреспонденций однородных (взаимозаменяемых) ресурсов (грузов, автомобилей и т.п.) от поставщиков A_i $(i = \overline{1, m})$ к потребителям B_j $(j = \overline{1, n})$. Такой план должен обеспечивать соблюдение ограничений по наличию и потребностям ресурсов, а также минимальные транспортные издержки. Вместо стоимости при поиске оптимального решения могут приниматься за основу расстояния поставки, затраты времени на доставку, условные расстояния и другие показатели. Если целевая функция и ограничения имеют линейный вид, то имеет место транспортная задача линейного программирования.

Задача оптимизации закрепления потребителей за поставщиками применяется при оптимизации грузопотоков (грузоотправители – поставщики, грузополучатели – потребители), возврата порожних транспортных средств (грузополучатели – поставщики порожних транспортных средств, грузоотправители – получатели порожних транспортных средств), подачи транспортных средств в начальные пункты маршрутов (предприятия-перевозчики – поставщики, начальные пункты маршрутов – получатели транспортных средств). Первая задача – необходимо получить грузопотоки, обеспечивающие минимальные затраты (при учете расстояний – грузооборот); вторая и третья – обеспечить минимум непроизводительных перемещений транспортных средств.

На основании полученных оптимальных планов закрепления потребителей ресурса за его поставщиками может строиться картограмма поставок. Для этого эпюры потоков ресурса накладываются на схему транспортной сети. При построении необходимо учитывать, что ресурс корреспондирует по пути с минимальной стоимостью перемещения (кратчайшему пути) между пунктами i и j транспортной сети.

Если обозначить размер ресурса (вывоза) от поставщика A_i через X_{Ai}, требуемый размер поставки потребителю B_j через X_{Bj}, размер корреспонденции от i-го поставщика j-му получателю (потребителю) через X_{ij} и стоимость поставки единицы ресурса между ними через l_{ij}, то задача в математической форме примет вид:

$$\sum_{j=1}^{n} X_{ij} \le X_{Ai} \, ; \tag{3.1}$$

$$\sum_{i=1}^{m} X_{ij} \le X_{Bj} \, ; \tag{3.2}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij}\, l_{ij} = \min_{\{X_{ij}\}} \, ; \tag{3.3}$$

$$X_{ij} \ge 0. \tag{3.4}$$

В случае, если запасы у поставщиков равны общим потребностям получателей, то

$$\sum_{i=1}^{m} X_{Ai} = \sum_{j=1}^{n} X_{Bj} \, . \tag{3.5}$$

Поставленная таким образом задача (ограничения 3.1, 3.2, 3.4 и целевая функция 3.3) является сбалансированной (условие 3.5) классической транспортной задачей линейного программирования, в результате решения которой по известным значениям X_{Ai}, X_{Bj}, l_{ij} находятся оптимальные значения корреспонденций X_{ij}.

Задача может быть поставлена с дополнительными условиями:

1) имеются ограничения на размер поставок от каждого i-го поставщика каждому j-му потребителю в объемах X_{oij} , $i = \overline{1, m}$; $j = \overline{1, n}$;

2) запрещена поставка ресурса от i-го поставщика j-му потребителю, т.е. должно быть обеспечено условие X_{ij}=0;

3) должна быть обеспечена поставка от i-го поставщика j-му потребителю в гарантированном объеме X_{rij} ($X_{rij} \le X_{Ai}$ и $X_{rij} \le X_{Bj}$);

4) при несбалансированной задаче необходимо назначить по какому-то из пунктов корреспонденцию реального или фиктивного ресурса.

Для решения транспортная задача может быть представлена в виде распределительной таблицы (таблица 3.3).

Одна из возможных общих схем алгоритма решения несбалансированной транспортной задачи линейного программирования с учетом ограничений на размер корреспонденций представлена на рисунке 3.23. Ниже приведены алгоритмы реализации отдельных этапов решения поставленной задачи.

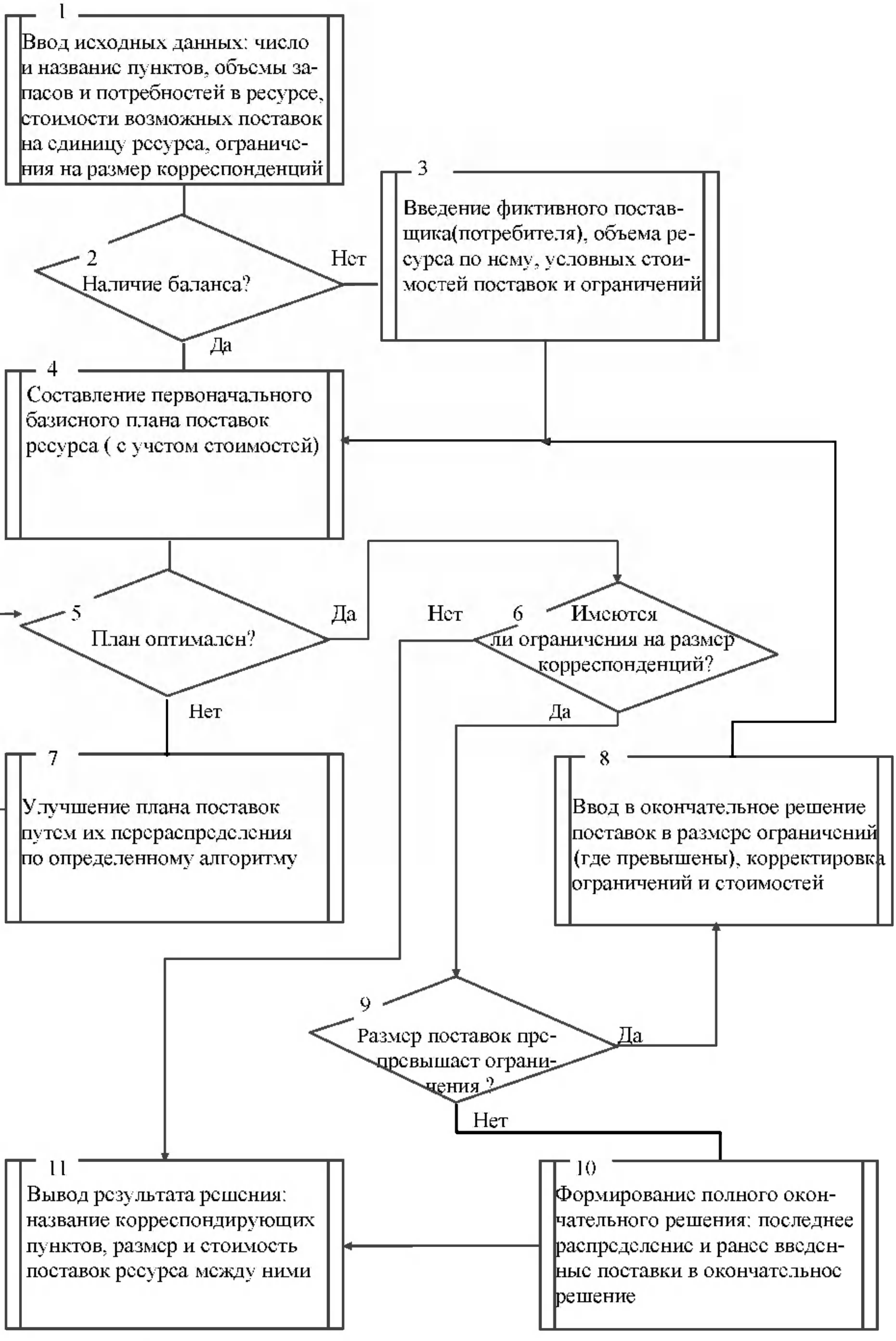


Рисунок 3.23 – Общая схема алгоритма решения транспортной задачи линейного программирования

Приведение несбалансированной задачи к сбалансированному виду производится по алгоритму:

1) рассчитываются суммарный объем наличия ресурса X_{cA} и суммарный объем потребности в ресурсе X_{cB}

$$X_{cA} = \sum_{i=1}^{m} X_{Ai} \; ; \; X_{cB} = \sum_{j=1}^{n} X_{Bj} \; ;$$

2) сравниваются объемы X_{cA} и X_{cB}. Если они равны, то задача сбалансирована, иначе на пункт 3;

3) если $X_{cA} > X_{cB}$, то вводится фиктивный потребитель (n=n+1) и если $X_{cA}<X_{cB}$ – фиктивный поставщик (m=m+1);

4) по фиктивному потребителю или поставщику назначается объем ресурса в размере abs(X_{cA} - X_{cB});

5) назначаются стоимости по фиктивному пункту равными постоянной величине, например $l_ф=10\max\limits_{i,j} l_{ij}$ (для фиктивного потребителя $l_{in} = l_ф$, $i=\overline{1,m}$ или для фиктивного поставщика $l_{mj}=l_ф$, $j=\overline{1,n}$;

6) назначаются ограничения по фиктивному пункту, равными постоянной величине, например $X_{оф}$ >abs(X_{cA} - X_{cB}) (для фиктивного потребителя $X_{oij} = X_{оф}$, j=n, $i=\overline{1,m}$ или для поставщика $X_{oij} = X_{оф}$, i=m, $j=\overline{1,n}$).

Для отыскания оптимального закрепления потребителей за поставщиками необходимо сделать в распределительной таблице первоначальное закрепление, т.е. получить произвольный план закрепления (опорный), удовлетворяющий ограничениям (3.1), (3.2), (3.4) и (3.5) при числе загруженных связей m+n-1 и отсутствии циклов (контуров). План, содержащий ровно m+n-1 загруженных связей без циклов, называется базисным.

Цикл (контур) в распределительной таблице – это условная замкнутая ломаная линия, образованная прямыми отрезками, углы между которыми равны 90°, а вершины углов лежат в загруженных связях. Контур может быть четырехугольным, шестиугольным, восьмиугольным и т.д. Если число загруженных связей более m+n-1, то среди них есть цикл. Примеры циклов приведены на рисунке 3.24 (точками показаны загруженные связи):

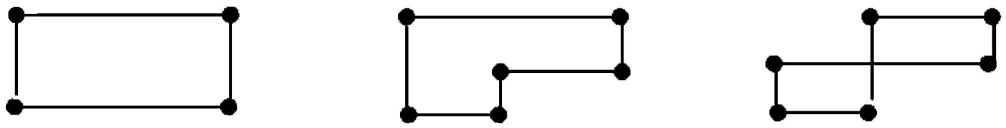


Рисунок 3.24 – Примеры циклов

Существует несколько методов получения опорного плана – метод северо-западного угла (диагональный) и ряд эффективных, дающих решение более близкое к оптимальному,– метод минимального элемента, метод абсолютного двойного предпочтения, метод Фогеля (наиболее эффективный) и др.

Таблица 3.3 – Распределительная таблица (общий вид)

Поставщики A_i	Потребители B_j					Запас ресурса X_{Ai}	Потенциалы u_{Ai}
	B_1	. . .	B_j	. . .	B_n		
A_1							
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
A_i			s_{ij} l_{ij} X_{ij} X_{oij}			X_i	u_{Ai}
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
A_m							
Потребность в ресурсе X_{Bj}							
Потенциалы u_{Bj}			u_{Bj}				

Для **получения первоначального базисного решения** рассмотрим применение метода минимального элемента. В соответствии с этим методом распределение составляется по следующему алгоритму:

1) формируются дополнительные массивы стоимостей l'_{ij} ($l'_{ij} = l_{ij}$), объемов ресурса X'_{Ai} ($X'_{Ai}=X_{Ai}$) и X'_{Bj} ($X'_{Bj}=X_{Bj}$), $i=\overline{1,m}$; $j=\overline{1,n}$;

2) находится минимальная стоимость из всех связей

$$l_{rw} = \min_{i,j} l_{ij}.$$

Если $l_{rw} = \infty$, то первоначальное базисное решение получено. Иначе на 3);

3) связь rw, соответствующую минимальной стоимости, загружают корреспонденцией X_{rw}. Объем корреспонденции X_{rw}, назначаемый для связи rw, определяется как минимум от объемов запаса и потребности с учетом ранее назначенных других поставок:

$X_{rw} = \min(X'_{Ar}, X'_{Bw})$,

где X'_{Ar} – количество ресурса r-го поставщика с учетом ранее назначенных корреспонденций другим, кроме w-го, потребителям; X'_{Bw} – количество ресурса, требуемого w-му потребителю с учетом ранее назначенных корреспонденций от других, кроме r-го поставщика;

4) из сравниваемых величин X'_{Ar} и X'_{Bw} вычитается значение X_{rw}, в результате чего получаются измененные ограничения $X'_{Ar}=X'_{Ar} - X_{rw}$ и $X'_{Bw}=X'_{Bw} - X_{rw}$;

5) изменяется массив l'_{ij} по следующему правилу:

если $X'_{Bw}=0$, то $l'_{iw} = \infty$ ($i=\overline{1,m}$),

иначе ($X'_{Ar}=0$) $l'_{rj} = \infty$ ($j=\overline{1,n}$);

6) переход на пункт 2)

Пункт 5 алгоритма обеспечивает исключение из дальнейшего рассмотрения поставщика (если $X'_{Ar}=0$), либо получателя (если $X'_{Bw}=0$). Остаток объема ресурса X'_{Ar} или X'_{Bw} к дальнейшему распределению может иметь значение, равное нулю.

Полученное таким образом закрепление потребителей ресурса за поставщиками является одним из возможных базисных распределений.

Для оценки **оптимальности текущего решения – поставок ресурса от потребителей поставщикам** применяется ряд методов: распределительные методы (метод Хичкока, метод Креко, метод МОДИ), методы с разрешающими элементами (метод разрешающих слагаемых и метод разрешающих множителей) и др.

Наиболее широкое применение получил модифицированный метод (МОДИ). В распределительную таблицу вводят строку и столбец для расчета потенциалов. Метод, как и другие распределительные методы, основан на том, что если к стоимостям любой строки (столбца) распределительной таблицы прибавить (или отнять) одно и то же произвольное число, то оценка оптимальности относительно не изменится. Если, например, от стоимостей поставок от каждого i-го поставщика отнять число u_{Ai} , а от стоимостей поставок каждому j-го потребителю – u_{Bj}, то относительной оценкой любой связи ij может служить оценочный параметр s_{ij} (вместо l_{ij}):

$$s_{ij} = l_{ij} - u_{Ai} - u_{Bj}. \tag{3.6}$$

Принимая для загруженных связей $s_{ij} = 0$ и используя выражение (3.6), определяются потенциалы u_{Ai} и u_{Bj} по следующему правилу:

потенциал для одного пункта, например первого поставщика принимается равным нулю;

по расстояниям загруженных связей подбираются потенциалы для других поставщиков и потребителей таким образом, чтобы соблюдалось принятое условие $l_{ij} - u_{Ai} - u_{Bj} = 0$, т.е. стоимость для каждой загруженной связи должна быть равна сумме потенциалов по поставщику и потребителю.

Затем по вычисленным потенциалам u_{Ai} , u_{Bj} определяется, используя формулу (3.6), значение оценочного параметра s_{ij} для каждой незагруженной связи (не входящей в базисный план). Величина параметра s_{ij} характеризует общее изменение целевой функции, получаемое при включении в план единичной корреспонденции для связи ij по сравнению с рассматриваемым планом.

Если значение оценочного параметра свободной связи будет меньше нуля (s_{ij} <0), то это значит, что перераспределение корреспонденций с загрузкой такой связи, называемой потенциальной, уменьшит значение целевой функции на величину abs($s_{ij}X_{ij}$), где X_{ij} – размер корреспонденции, которой будет загружена связь ij. Отсутствие связей со значением параметра s_{ij} <0 означает, что проверяемый план закрепления потребителей за поставщиками является оптимальным. Если для какой-то свободной связи значение s_{ij} равно нулю, то это означает, что можно получить другой план с тем же минимальным значением целевой функции.

Если решение не оптимально, то из всего множества свободных связей выявляется наиболее потенциальная с минимальным значением s_{ij} . Загрузка корреспонденцией такой связи дает наибольшее уменьшение целевой функции на каждую единицу перераспределенного объема ресурса.

При перераспределении загрузок по связям в распределительной таблице для наиболее потенциальной связи, как для загруженной, строится контур. Введение к m+n-1 загруженным связям еще одной образует ровно один определенный контур, присущий вводимой связи. После чего связи, соответствующие вершинам контура, нумеруются: номер 1 присваивается выбранной потенциальной связи, а дальнейшая нумерация ведется в порядке следования вершин по контуру (могут присваиваться связям контура положительные и отрицательные знаки).

Затем производится перераспределение по контуру корреспонденций следующим образом:

1) выявляется связь с четным номером, которой соответствует наименьшее значение корреспонденции $X_м$;

2) значение объема ресурса $X_м$ вычитается от значений объемов корреспонденций всех связей с четными номерами вершин. Если среди связей с четными номерами вершин окажутся две (или более) с одинаковой минимальной корреспонденцией, то из плана исключается только одна из них, для связи с большим расстоянием поставки, а вместо остальных оставляют условную корреспонденцию с нулевым значением, чтобы не допустить вырождения;

3) для всех связей с нечетными номерами вершин (включая и потенциальную) к значениям объемов корреспонденций прибавляется величина $X_м$.

В результате одна связь становится свободной, а наиболее потенциальная – получает загрузку. Измененные значения корреспонденций и значения объемов корреспонденций для связей, которые не входили в контур, переносятся в таблицу нового улучшенного плана закрепления потребителей за поставщиками.

Полученный новый план проверяется на оптимальность. Действия по проверке на оптимальность и улучшению распределения повторяются до тех пор, пока не будет найден оптимальный план закрепления потребителей за поставщиками.

При решении транспортной задачи линейного программирования на основе существующего базисного решения возможны варианты, когда число загруженных связей в таблице меньше m+n-1 (случай вырождения) или больше m+n-1 (случай наличия циклов). В первом случае нельзя оценить оптимальность решения, так как невозможно определить потенциалы для некоторых поставщиков и (или) потребителей, во втором – потенциалы определяются неоднозначно.

В случае **вырождения** необходимое число свободных связей загружают нулями с соблюдением правила: нулевые объемы поставок заносятся для связи поставщиков (потребителей), для которых нельзя рассчитать потенциал, с поставщиками (потребителями), для которых потенциалы уже определены. Целесообразно выбрать из данных связей такие, которым соответствуют наименьшие стоимости поставок. Связи, загруженные нулями, не должны образовывать контуров (циклов) с другими загруженными связями.

В случае **наличия циклов для некоторых связей** исключают корреспонденции по следующему алгоритму:

1) строится один из возможных контуров;

2) нумеруются углы контура;

3) определяется сумма стоимостей поставок для связей, в которых лежат углы контура с четными номерами и затем с нечетными номерами;

4) среди связей (с четными или нечетными номерами), для которых сумма стоимостей перевозок больше, выявляется связь с наименьшим значением размера корреспонденции $X_м$;

5) размер корреспонденции $X_м$ вычитается от загрузки каждой из связей с четными или нечетными номерами, для которых сумма стоимостей поставок больше, и прибавляется к загрузке каждой из связей с нечетными или четными номерами, для которых сумма стоимостей поставок меньше.

Указанные действия повторяются до полного исключения циклов (контуров) в распределительной таблице. При этом нельзя допускать вырождения за счет одновременного исключения загрузок для нескольких связей одного цикла. Исключение контуров в распределительной таблице по вышеописанной методике уменьшает одновременно и значение целевой функции.

При отсутствии дополнительных условий полученное оптимальное решение является окончательным.

При наличии **ограничений на размер корреспонденций**, задача в дальнейшем решается по следующей схеме:

1) проверяется выполнение условия $X_{ij} \leq X_{oij}$ для связей ij. Если условие выполняется для всех связей, то решение закончено (переход на блок 10 общего алгоритма), иначе переход на пункт 2;

2) для каждой связи ij, по которой имеет место условие $X_{ij} > X_{oij}$, последовательно должен быть реализован алгоритм:

2.1) в окончательное решение вводится корреспонденция $X_{ij} = X_{oij}$;

2.2) корректируются соответствующие размеры наличия и потребности в ресурсе

$X_{Aj} = X_{Aj} - X_{ij}$;
$X_{Bj} = X_{Bj} - X_{ij}$;

2.3) корректируется соответствующая стоимость поставки l_{ij} путем замены на большое число, например $l_{ij} > 100 \max_{i,j} l_{ij}$;

3) переход на блок 4 укрупненного алгоритма решения задачи.

Решение задачи **с запрещением поставки ресурса** от i-го поставщика j-му потребителю решается по общей схеме с предварительной заменой реальной стоимости l_{ij} на большое число, например $l_{ij} > 100 \max_{i,j} l_{ij}$. Тем самым обеспечивается $X_{ij} = 0$.

Решение задачи **с необходимостью обеспечения поставки** от i-го поставщика j-му потребителю в гарантированном объеме X_{rij} производится следующим образом:

1) предварительно для каждой такой связи ij, по которой имеет место условие $X_{ij} > X_{rij}$:

1.1) в окончательное решение вводится корреспонденция $X_{ij} = X_{rij}$;

1.2) корректируются соответствующие размеры наличия и потребности в ресурсе

$X_{Aj} = X_{Aj} - X_{ij}$; $X_{Bj} = X_{Bj} - X_{ij}$;

2) решается задача с новыми условиями по общей схеме.

Полное окончательное решение будет представлять собой множество корреспонденций, полученных решением по измененным условиям, и корреспонденций, назначенных по пункту 1.1).

При решении **несбалансированной транспортной задачи** возможно за счет назначения различных стоимостей по фиктивному пункту распределение по какому-то из пунктов **корреспонденции реального или фиктивного ресурса**. Так, если по одному из пунктов в корреспонденции с фиктивным поставить относительно большее значение стоимости, то это явится предпосылкой, что по нему будет назначена корреспонденция только реального ресурса (по пункту будет обеспечена корреспонденция реального ресурса). Наоборот, если поставить относительно меньшее значение стоимости, то это вызовет назначение корреспонденции в виде фиктивного ресурса (не будет обеспечена корреспонденция реального ресурса). При одинаковом значении стоимости корреспонденций в связях с фиктивным пунктом фиктивные поставки будут назначены исходя из минимума общих затрат на поставки, что не всегда отвечает реальным требованиям производства.

Компьютерная программа решения несбалансированной транспортной задачи линейного программирования с ограничениями на размер корреспонденций приведена в приложении 8. Программа состоит из головного модуля, модуля ввода данных и модуля выполнения расчетов.

Пример.

На предприятиях A_i пассажирского транспорта имеется следующее число X_{Ai} транспортных средств для работы на маршрутах: на $A_1 - X_{A1} = 10$; $A_2 - X_{A2} = 30$; $A_3 - X_{A3} = 40$. Для работы на маршрутах B_j требуется следующее число транспортных средств: на $B_1 - X_{B1} = 10$; $B_2 - X_{B2} = 10$; $B_3 - X_{B3} = 20$; $B_4 - X_{B4} = 30$. Имеются дополнительные условия:

1) заданы ограничения на размер корреспонденций для всех связей (кроме A_1B_3) – в размере 30 ед, для A_1B_3 – в размере 5 ед.;

2) транспортные средства 2-го предприятия необходимо использовать в полном объеме;

3) поставка транспортных средств 2-го предприятия на 2-й маршрут должна быть не менее 5 единиц;

4) поставка транспортных средств 1-го предприятия на 4-й маршрут запрещена.

Требуется оптимальным образом распределить транспортные средства по маршрутам перевозок. Расстояния между предприятиями и маршрутами (нулевые пробеги транспортных средств) приведены ниже в таблице 3.4.

Таблица 3.4 – Данные по нулевым пробегам транспортных средств

A_i	Расстояние между предприятиями A_i и маршрутами B_j			
	B_1	B_2	B_3	B_4
A_1	4	8	2	5
A_2	5	4	4	7
A_3	5	3	3	4

Решение.

Проверяем задачу на сбалансированность, для чего рассчитываем суммарный объем наличия ресурса X_{cA} и суммарный объем потребности в ресурсе X_{cB}

$$X_{cA} = \sum_{i=1}^{m} X_{Ai} = 15+30+40=80; \quad X_{cB} = \sum_{j=1}^{n} X_{Bj} = 10+10+20+30=70.$$

После этого сравниваем объемы X_{cA} и X_{cB}. Поскольку они не равны, то задача **несбалансирована**. Так как $X_{cA} > X_{cB}$, то вводится фиктивный потребитель B_5 (n=n+1) и по не нему назначается объем ресурса в размере X_{B5} = abs(X_{cA} - X_{cB})=80-70=10. Для фиктивного потребителя назначаются стоимости (расстояния) равными постоянной величине, например, $l_ф=10\max\limits_{i,j} l_{ij}$ = 8*10=80(l_{i5} = 80, i=1,2,3). Однако в связи с тем, что транспортные средства 2-го предприятия необходимо использовать в полном объеме, то расстояние $l_{2\text{-}5}$ следует принять значительно больше других, например, $l_{2\text{-}5}$=800. Затем по фиктивному пункту назначаются ограничения равными постоянной величине, например, $X_{оф} \geq$ abs(X_{cA} - X_{cB}) $\geq$ 10 (для фиктивного потребителя X_{oi5} = 10, i = 1, 2, 3).

Для учета **запрещения** поставки ресурса от поставщика A_1 потребителю B_4 заменяем реальную стоимость (расстояние) $l_{1\text{-}4}$ на большое число, например $l_{1\text{-}4} \geq 100\max\limits_{i,j} l_{ij} \geq 800$. Принимаем $l_{1\text{-}4}$ = 800.

Обеспечение поставки от поставщика A_2 потребителю B_2 в гарантированном объеме $X_{г2\text{-}2}$=5 производится следующим образом:

в окончательное решение вводится корреспонденция $X_{2\text{-}2}$ = 5;

корректируются соответствующие размеры наличия и потребности в ресурсе

$X_{A2} = X_{A2} - X_{2\text{-}2}$ =30-5=25; $X_{B2} = X_{B2} - X_{2\text{-}2}$ =10-5=5.

С учетом исходных условий и последующих изменений составляем распределительную таблицу (таблица 3.5) и производим первоначальное базисное распределение по методу минимального элемента. Для этого назначаем корреспонденции для связей в порядке возрастания по ним стоимостей (расстояний). Связи загружаются в следующей последовательности: A_1-B_3 объемом 10, A_3-B_2 объемом 5, A_3-B_3 объемом 10; A_3-B_4 объемом 25, A_2-B_1 объемом 10, A_2-B_4 объемом 5, A_2-B_5 объемом 10.

Полученное первоначальное распределение проверяем на оптимальность, для чего рассчитываем значения потенциалов u_{Ai} и u_{Bj}. При этом значение одного из потенциалов, например для A_1, задаем равным нулю (u_{A1} =0). После того, как определены значения потенциалов, для каждой свободной связи рассчитываем оценочный параметр s_{ij} по формуле (3.6).

Таблица 3.5 – Распределительная таблица (первоначальное базисное распределение)

Поставщики A_i	Потребители B_j					X_{Ai}	u_{Ai}
	B_1	B_2	B_3	B_4	B_5		
A_1	+3 4 30	+6 8 30	2 **10** 5	+797 800 30	-716 80 10	10	0
A_2	5 **10** 30	-2 4 30	-2 4 30	7 + **5** 30	800 **10** – 10	25	4
A_3	+3 5 30	3 **5** 30	3 **10** 30	4 – **25** 30	-717 80 + 10	40	1
X_{Bj}	10	5	20	30	10	80	
u_{Bj}	1	2	2	3	796		

Если для свободной связи $s_{ij}>0$, то связь не является потенциальной, $s_{ij}<0$ – потенциальна. В первоначальном распределении потенциальными оказались связи A_1-B_5, A_3-B_5, A_2-B_2 и A_2-B_3 соответственно с оценочными параметрами, равными (-716), (-717), (-2) и (-2).

Для связи A_3-B_5, как для наиболее потенциальной, строится контур, и по нему перераспределяются корреспонденции с загрузкой потенциальной связи. Связи A_2-B_4 и A_3-B_5 (связи со знаком +) догружаются объемом по 10 ед. (минимальный для связей со знаком -), а для связей A_2-B_5 и A_3-B_4 (связи со знаком -) снимается объем по 10 ед. В результате получаем улучшенный план закрепления маршрутов за предприятиями (1-е улучшенное базисное распределение) (таблица 3.6).

Таблица 3.6 – Распределительная таблица (1-е улучшенное базисное распределение)

Поставщики A_i	Потребители B_j					X_{Ai}	u_{Ai}
	B_1	B_2	B_3	B_4	B_5		
A_1	+3 4 30	+6 8 30	2 **10** 5	+797 800 30	+1 80 10	10	0
A_2	5 **10** 30	-2 4 + 30	-2 4 30	7 – **15** 30	+717 800 10	25	4
A_3	+3 5 30	3 – **5** 30	3 **10** 30	4 + **15** 30	80 **10** 10	40	1
X_{Bj}	10	5	20	30	10	80	
u_{Bj}	1	2	2	3	79		

В улучшенном распределении потенциальными оказались связи A_2-B_2 и A_2-B_3 соответственно с оценочными параметрами, равными (-2) и (-2).

Для связи A_2-B_2, как одной из наиболее потенциальных, строится контур, и по нему перераспределяются корреспонденции с загрузкой потенциальной связи. Связи A_2-B_2 и A_3-B_4 (связи со знаком +) догружаются объемом по 5 ед. (минимальный для связей со знаком -), а для связей A_3-B_2 и A_2-B_4 (связи со знаком -) снимается объем по 5 ед. В результате получаем 2-е улучшенное базисное распределение (таблица 3.7).

Таблица 3.7 – Распределительная таблица (2-е улучшенное базисное распределение)

Поставщи- Ки A_i	Потребители B_j					X_{Ai}	u_{Ai}
	B_1	B_2	B_3	B_4	B_5		
A_1	+3 4 30	+8 8 30	2 **10** 5	+797 800 30	+1 80 10	10	0
A_2	5 **10** 30	-2 4 **5** 30	-2 4 + 30	7 – **10** 30	+717 800 10	25	4
A_3	+3 5 30	3 30	3 – **10** 30	+ 4 **20** 30	80 **10** 10	40	1
X_{Bj}	10	5	20	30	10	80	
u_{Bj}	1	0	2	3	79		

Производим проверку полученного распределения на оптимальность.

В улучшенном распределении потенциальной оказалась связь A_2-B_3 с оценочным параметром, равным (-2).

Для связи A_2-B_3, как потенциальной, строится контур, и по нему перераспределяются корреспонденции с загрузкой потенциальной связи. Связи A_2-B_3 и A_3-B_4 (связи со знаком +) догружаются объемом по 10 ед. (минимальный для связей со знаком -), а для связей A_3-B_3 и A_2-B_4 (связи со знаком -) снимается объем по 10 ед. В результате получаем 3-е улучшенное базисное распределение (таблица 3.8).

Производим проверку полученного распределения на оптимальность. Так как в улучшенном распределении для всех свободных связей оценочные параметры $s_{ij} > 0$, то последнее решение является оптимальным.

Проверяем выполнение ограничений на размер корреспонденций. Сравнивая в последней таблице значения X_{ij} (центр клетки связи) и X_{oij} (правый нижний угол клетки связи) устанавливаем, что нарушено ограничение только для связи A_1-B_3 ($X_{1\text{-}3}=10>X_{o1\text{-}3}=5$). Для связи A_1-B_3 реализуем следующие действия:

в окончательное решение вводится корреспонденция $X_{1\text{-}3}=5$;

корректируются соответствующие размеры наличия и потребности в ресурсе

$X_{A1}=X_{A1}–X_{1\text{-}3}=10\text{-}5=5$;

$X_{B3}=X_{B3}–X_{1\text{-}3}=20\text{-}5=15$;

корректируется соответствующая стоимость (расстояние) поставки $l_{1\text{-}3}$ путем замены на большое число, например $l_{1\text{-}3} > 100 \max\limits_{i,j} l_{ij}$. Принимаем $l_{1\text{-}3}=800$.

Таблица 3.8 – Распределительная таблица (3-е улучшенное базисное распределение)

Поставщики A_i	Потребители B_j					X_{Ai}	u_{Ai}
	B_1	B_2	B_3	B_4	B_5		
A_1	+1 4 30	+6 8 30	2 **10** 5	+797 800 30	+1 80 10	10	0
A_2	5 **10** 30	4 **5** 30	4 **10** 30	+2 7 30	+719 800 10	25	2
A_3	+1 5 30	0 3 30	3 **0** 30	4 **30** 30	80 **10** 10	40	1
X_{Bj}	10	5	20	30	10	80	
u_{Bj}	3	2	2	3	79		

Согласно укрупненному алгоритму решения задачи следует с новыми данными вернуться на составление первоначального базисного решения (таблица 3.9).

Производим проверку полученного распределения на оптимальность. Так как в данном распределении для отдельных свободных связей оценочные параметры $s_{ij} \leq 0$, то данное решение является неоптимальным.

Действуя в ранее изложенном порядке в результате многократных улучшений получаем нижеприведенное оптимальное решение (таблица 3.10).

Согласно общему алгоритму проверяем выполнение ограничений на размер корреспонденций. Поскольку объемы корреспонденций не превышают заданные ограничения на размер поставок, то решение закончено.

Таблица 3.9 – Первоначальное базисное распределение для измененных данных

Поставщики A_i	Потребители B_j					X_{Ai}	u_{Ai}
	B_1	B_2	B_3	B_4	B_5		
A_1	0 4 **5** 30	+3 8 30	+795 800 5	+794 800 30	-719 80 10	5	0
A_2	5 **5** 30	-2 4 30	-2 4 30	7 **10** 30	800 **10** 10	25	1
A_3	+3 5 30	3 **5** 30	3 **15** 30	4 **20** 30	-721 80 10	40	-2
X_{Bj}	10	5	15	30	10	80	
u_{Bj}	4	5	5	6	799		

Таблица 3.10 – Оптимальное распределение для измененных данных

Поставщики A_i	Потребители B_j					X_{Ai}	u_{Ai}
	B_1	B_2	B_3	B_4	B_5		
A_1	4 **5** 30	+5 8 30	+797 800 5	+796 800 30	0 80 10	5	0
A_2	5 **5** 30	4 **5** 30	4 **15** 30	+2 7 30	+719 800 10	25	1
A_3	+1 5 30	0 3 30	3 **0** 30	4 **30** 30	80 **10** 10	40	0
X_{Bj}	10	5	15	30	10	80	
u_{Bj}	4	3	3	4	80		

Полное окончательное решение содержит:
ранее включенные корреспонденции в окончательное решение
$X_{2\text{-}2} = 5$;
$X_{1\text{-}3} = 5$;
и последнее решение для реальных пунктов
$X_{1\text{-}1} = 5$;
$X_{2\text{-}1} = 5$;
$X_{2\text{-}2} = 5$;
$X_{2\text{-}3} = 15$;
$X_{3\text{-}4} = 30$.

Ответ: исходя из минимума нулевых пробегов и других дополнительных условий оптимально закрепить за 1-м предприятием 1-й и 3-й маршрут с выделением по 5 единиц, за 2-м – 1-й, 2-й и 3-й с выделением соответственно 5, 10 и 15 единиц, за 3-м – 4-й маршрут с выделением 30 единиц транспортных средств.

3.7. Однопродуктовая задача динамического программирования

Задачи динамического программирования – это разновидность задач нелинейного программирования, когда решение может быть получено динамическиметодом.

Одной из наиболее простых является следующая нелинейная однопродуктовая распределительная задача:

необходимо оптимальным образом распределить ресурс в количестве x_{co} по n вариантам (объектам, схемам, этапам) при целевой функции

$$Z = \sum_{i=1}^{n} f_i(x_i) = f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n) = \max_{\{x_i\}}(\min)$$

и ограничениях

$$\sum_{i=1}^{n} x_i \le x_{co},$$

где x_i – количество ресурса, назначаемое по i-му варианту ($0 \le x_i \le x_{co}$);

f(x) – нелинейная функция, определяющая эффект (затраты) в зависимости от значений x;

n – общее число вариантов;

x_{co} – общее количество распределяемого ресурса.

Этой моделью описывается задача оптимального распределения однородного ограниченного по количеству ресурса (сырья, машин, денежных средств и т.п.).

Обозначим через x_{ci} – количество ресурса, назначаемого суммарно по i-му варианту и всем предыдущим этапам (вариантам). При этом количество ресурса, входящее в x_{ci}, распределено по предыдущим вариантам оптимально исходя из минимума затрат или максимума эффекта.

Тогда целевая функция Z_i при рассмотрении i-го этапа решения определяется по выражению

$$Z_i(x_{ci}, x_i) = f_i(x_i) + f^*_{i-1}(x_{ci} - x_i),$$

где $0 \le x_{ci} \le x_{co}$ при $i \le n-1$ и $x_{ci} = x_{co}$ при $i = n$;

f^*_{i-1} – функция, определяющая эффект (затраты) оптимальным образом по предыдущим вариантам в зависимости от количества ресурса x_{ci-1}.

Значение f^*_i определяется по Z_i:

$$f_1^*(x_{c1}) = f_1(x_{c1}) \quad \text{для } i = 1 \ (x_{c1} = x_1);$$

$$f_i^*(x_{ci}) = \max_{xi}(\min) Z_i(x_{ci}, x_i) \text{ при } x_i^*, \ i = \overline{2, n}.$$

Определение $f_i^*(x_{ci})$ и соответствующих им x_i^* при данном x_{ci} производятся поэтапно от i=2 до i=n.

По результатам завершенных поэтапных расчетов формируется оптимальное решение $\{x_{\text{опт}i}\}$:

$$x_{oi} = x_i^* \quad (\text{для } i = n);$$

$$x_{oi} = x_i^* \text{ при } x_{ci} = x_{co} - \sum_{k=i+1}^{n} x_{ok} \ (\text{для } i = \overline{n, 2, -1});$$

$$x_{oi} = x_{co} - \sum_{k=2}^{n} x_{ok} \quad (\text{для } i=1).$$

Вариант алгоритма решения однопродуктовой задачи динамического программирования приведен на рисунке 3.25. Решение предусмотрено для целевой функции, подлежащей оптимизации на максимум. Для решения задачи на минимум необходимо ввести исходные значения $s_{вji}$ с отрицательным знаком или внести изменения в блоках 12 (заменить 1E+12 на -1E+12) и 15 (условие < на >).

Выводимые j, $k_{ji}d_x$, s_{nm} представляют соответственно номер варианта, количество ресурса по варианту и значение целевой функции для оптимального решения.

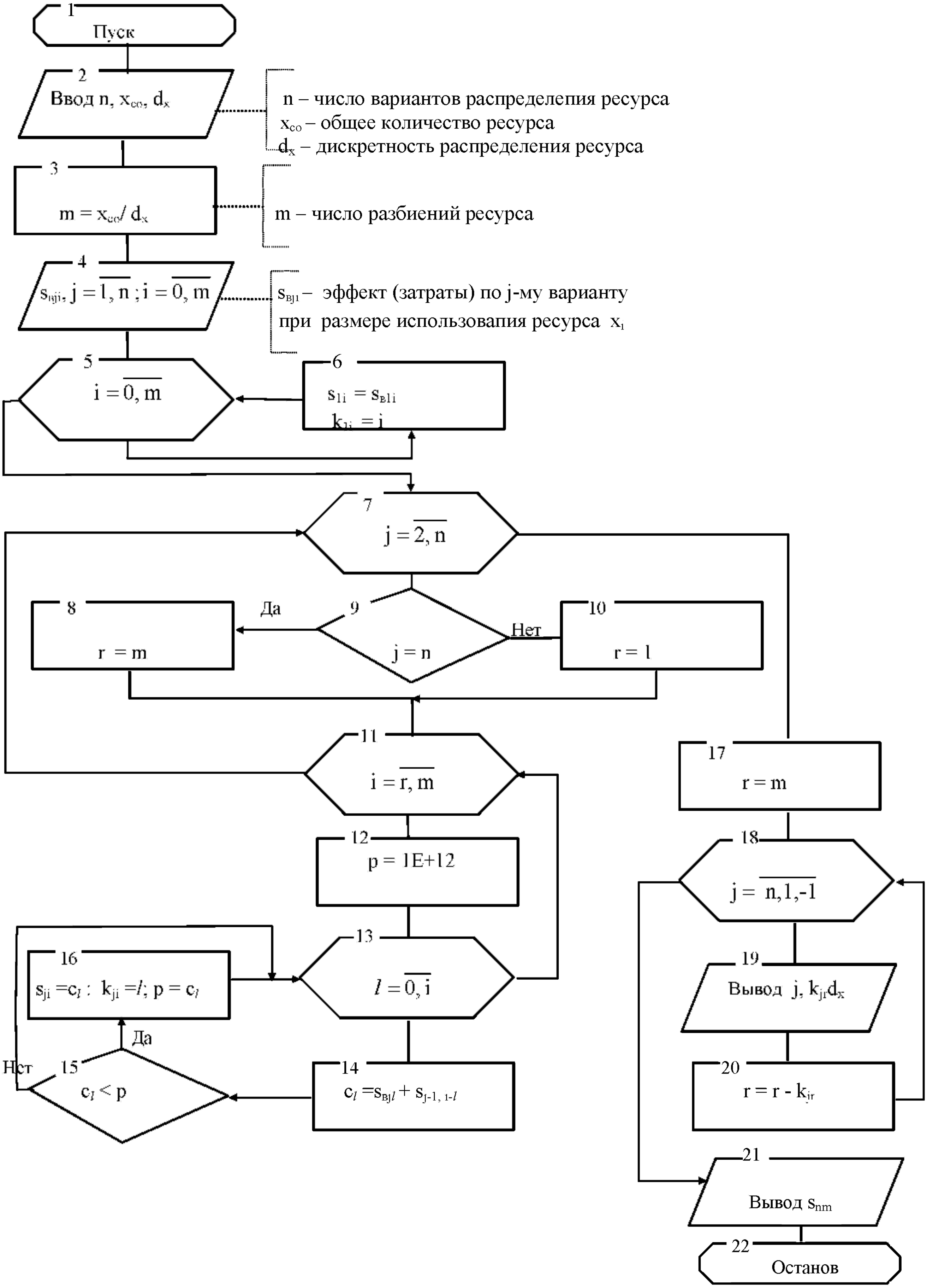


Рисунок 3.25 – Алгоритм решения однопродуктовой задачи динамического программирования

Пример.

Банковский кредит может быть использован по 4-м вариантам (n = 4). Общий размер кредита x_{co} = 800 тыс.ед. Дискретность распределения кредита по вариантам использования dx = 200 тыс.ед. Зависимость эффекта по вариантам использования кредита нелинейна и задана таблично (таблица 3.11).

Таблица 3.11 – Значения эффектов в зависимости от номера варианта i и размера кредита x

Номер варианта i	Эффект f_i (x) при размерах использования кредита x				
	x=0	x=200	x=400	x=600	x=800
1	0	12	21	31	37
2	0	11	22	32	40
3	0	9	20	31	38
4	0	11	21	30	39

Необходимо распределить общую сумму кредита, чтобы суммарный эффект от его использования был максимальным.

Решение.

Имеем целевую функцию $Z = \sum_{i=1}^{n} f_i(x_i) = f_1(x_1) + f_2(x_2) + f_2(x_3) + f_4(x_4) = \max;$

ограничения $x_{общ} = x_1 + x_2 + x_3 + x_4 = 800;\ 0 \le x_i \le x_{общ},$

а также $f_1^*(x_{c1}) = f_1\ (x_{c1})$ – для одного первого варианта.

На втором этапе (i=2) рассматриваем одновременно распределение ресурса для первого и второго варианта. Результаты расчетов сводим в таблицу 3.12. Значком (*) отмечен столбец, соответствующий оптимальному значению x_i^* при данном x_{ci}.

Таблица 3.12 – Расчет при i=2

x_{ci}	200		400			600				800				
x_i	0	200	0	200	400	0	200	400	600	0	200	400	600	800
$x_{ci} - x_i$	200	0	400	200	0	600	400	200	0	800	600	400	200	0
$f_i(x_i)$	0	11	0	11	22	0	11	22	32	0	11	22	32	40
$f_{i-1}^*(x_{ci} - x_i)$	12	0	21	12	0	31	21	12	0	37	31	21	12	0
$f_i(x_i) + f_{i-1}^*(x_{ci} - x_i)$	12*	11	21	23*	22	31	32	34*	32	37	44*	43	44*	40
$f_i^*(x_{ci})$	12		23			34				44				

На третьем этапе (таблица 3.13) решения рассматриваются одновременно оптимальный вариант по 1-му и 2-му вариантам совместно с третьим вариантом (i=3).

Таблица 3.13 – Расчет при i=3

x_{ci}	200		400			600				800				
x_i	0	200	0	200	400	0	200	400	600	0	200	400	600	800
$x_{ci} - x_i$	200	0	400	200	0	600	400	200	0	800	600	400	200	0
$f_i(x_i)$	0	9	0	9	20	0	9	20	31	0	9	20	31	38
$f_{i-1}^*(x_{c1} - x_i)$	12	0	23	12	0	34	23	12	0	44	34	23	12	0
$f_i(x_i) + f_{i-1}^*(x_{c1} - x_i)$	12*	9	23*	21	20	34*	32	32	31	44*	43	43	43	38
$f_i^*(x_{ci})$	12		23			34				44				

Переходим к последнему этапу (i=4). При этом $x_{ci} = x_o$ (таблица 3.14).

Таблица 3.14 – Расчет при i=4

x_{ci}	800				
x_i	0	200	400	600	800
x_{ci} - x_i	800	600	400	200	0
$f_i(x_i)$	0	11	21	30	39
$f^*_{i-1}(x_{c1} - x_i)$	44	34	23	12	0
$f_i(x_i) + f^*_{i-1}(x_{c1} - x_i)$	44	45*	44	42	39
$f^*_i(x_{ci})$	45				

Исходя из результатов выполнения последнего этапа расчетов следует, что кредит можно использовать с максимальным эффектом, равных 45 тыс.ед., при этом, $\mathbf{x_{опт4}}$ = 200 тыс.ед.

Тогда на 3-й и все предыдущие варианты остается 600 тыс.ед. Оптимальный размер использования кредита по 3-му варианту, как следует из этапа 3 решения, при $\mathbf{x_{ci}}$ = 600 равен $\mathbf{x_{опт3}}$ = 0;

на 2-й и все предыдущие остается по прежнему 800-200-0=600. Оптимальный размер использования кредита по 2-му варианту, как следует из этапа 2 решения, при $\mathbf{x_{ci}}$ = 600 равен $\mathbf{x_{опт2}}$ = 400.

Оптимальный размер кредита к использованию по первому варианту определяется как остаток $\mathbf{x_{опт1}}$ =800-200-0-400 = 200.

Ответ: Оптимальное распределение кредита по вариантам использования следующее:

1-й – 200 тыс.ед; 2-й – 400 тыс.ед., 3-й – 0 тыс.ед; 4-й – 200 тыс.ед.

3.8. Эвристические методы решения транспортных задач

Эвристические методы – это приближенные методы решения оптимизационных задач, позволяющие принимать рациональные решения. Преимуществом таких методов является ускорение получения решения с одновременным учетом всех установленных ограничений.

Такие методы применяются для решения нижеуказанных и других задач:

маршрутизация перевозок ресурсов помашинными отправками – метод гарантированного эффекта и методы на основе расчета выигрышей;

маршрутизация перевозок ресурсов мелкими отправками по сборочно-развозочным маршрутам – метод на основе кратчайшей связывающей сети, метод на основе расчета выигрышей (метод Кларка-Райта) или метод на основе решения задачи о коммивояжере.

3.8.1. Маршрутизация перемещения ресурсов помашинными отправками

Метод **гарантированного эффекта** основывается на формировании множества замкнутых маршрутов из отдельных перевозок (производительных ездок) и непроизводительных ездок, удовлетворяющих установленным ограничениям (по числу ездок на маршруте, длине, времени на движение и т.п.) и следующему условию

$$Z = \sum_{i=1}^{m} l_{гi} - K_г \sum_{j=1}^{n} l_{xj} \geq 0\,,$$

где $l_{гi}$ – длина (стоимость) i –й производительной ездки;

l_{xj} – длина (стоимость) j –й непроизводительной ездки;

m – число производительных перевозок (ездок) на маршруте;

n – число непроизводительных ездок на маршруте;

$K_г$ – коэффициент гарантированного эффекта.

Большие значения $K_г$ принимаются при местных перевозках (порядка 1,15) и меньшие при магистральных перевозках (порядка 1,05).

Из множества образованных возможных рациональных маршрутов включаются в окончательное решение маршруты по максимуму значения Z. При этом на принятом к включению в решение маршруте перевозок количество ресурса при отдельных ездках должно удовлетворять условию

$$\min_i Q_i / \gamma_{ci} = Q_{пi} / \gamma_{ci} = \underset{i}{\mathrm{const}},$$

где Q_i – количество ресурса при i-й перевозке (с учетом ранее окончательно принятых маршрутов);

$Q_{пi}$ – количество ресурса i-й перевозки, включаемое в данный маршрут;

γ_{ci} – коэффициент использования вместимости транспортных средств при i-й перевозке.

Если количество ресурса при какой-то перевозке освоено полностью, то все рациональные маршруты, не включенные в окончательное решение и содержащие такую перевозку, исключаются из дальнейшего рассмотрения. Если перевозка ресурса или его части не входит ни в один из окончательно принятых рациональных маршрутов, то она осваивается на маятниковом маршруте с обратным непроизводительным пробегом.

Например, требуется решить задачу маршрутизации для исходных данных, приведенных на нижеследующей схеме (рисунок 3.26).

Из схемы следует, что возможен маршрут А-В-С-Д-А с Z= (11+10) – 1.15 (6.0+5.0) = 4.9>0 (при $K_г$ =1.15).

Если принять, что дополнительных ограничений нет, то этот маршрут вводится в окончательное решение с объемом перевозок $\min_i \{100/1.0; 10\,0/0.8 = 125\} = 100$, т.е. при 1-й ездке назначается объем 100 и при 2-й ездке 100*0.8=80.

Объем 2-й перевозки ресурса в объеме 20 ед. (остаток от осваиваемого на принятом маршруте) назначается на маятниковый маршрут с обратным непроизводительным пробегом.

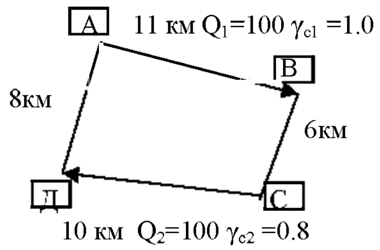


Рисунок 3.26 – Пример схемы транспортной сети и корреспонденций ресурса

Метод **на основе расчета выигрышей** основывается на определении значений сокращения пробега (стоимости, времени на проезд и т.п.) для всех возможных вариантов объединений исходных перевозок (ездок) по две или по две и по три (объединение большего числа ездок практически не применяется) по сравнению с перевозками на маятниковых маршрутах без обратной загрузки. Число возможных сочетаний и перестановок составляет: по две перевозки m(m-1)/2 и по три – m(m-1)(m-2)/3, где m – общее число перевозок ресурса.

Выигрыш от объединения определяется как разница между производительным и непроизводительным пробегом на маршруте $\Delta Z = \sum_{i=1}^{m} l_{гi} - \sum_{j=1}^{n} l_{xj}$. Например, при рассмотрении объединения по две перевозки выигрыши рассчитываются по формуле (см. рисунок 3.27):

$$\Delta Z_{i\text{-}j,k\text{-}n} = (l_{i\text{-}j} + l_{k-n}) - (l_{j-k} + l_{n-i}).$$

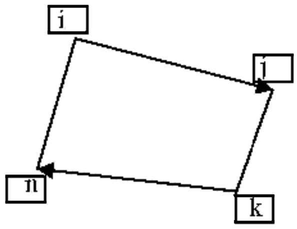


Рисунок 3.27 – Схема транспортной сети и корреспонденций ресурса (перевозок)

В качестве критерия очередности включения сформированных объединенных маршрутов в окончательное решение может приниматься максимум выигрыша ΔZ , а также экстремум других оценочных параметров:

максимум коэффициента использования пробега

$$\beta = \sum_{i=1}^{m} l_{ri} / (\sum_{i=1}^{m} l_{ri} + \sum_{j=1}^{n} l_{xj})$$

или максимум отношения производительных пробегов к непроизводительным

$$z_{пн} = \sum_{i=1}^{m} l_{ri} / \sum_{j=1}^{n} l_{xj}$$

или минимум отношения непроизводительных пробегов к общим (коэффициент непроизводительных пробегов)

$$z_{но} = \sum_{j=1}^{n} l_{xj} / (\sum_{i=1}^{m} l_{ri} + \sum_{j=1}^{n} l_{xj})$$

или минимум отношения непроизводительных пробегов к производительным

$$z_{нп} = \sum_{j=1}^{n} l_{xj} / \sum_{i=1}^{m} l_{ri} \; .$$

Значения необходимо строго обеспечить, чтобы $\Delta Z > 0$, $\beta > 0.5$, $z_{пн} > 1.0$, $z_{но} < 0.5$ и $z_{нп} < 1{,}0$. Рекомендуется принимать рациональные маршруты со значениями коэффициента использования пробега β не ниже 0.55-0.60 (меньшие допускаемые значения соответствуют магистральным и большие местным перевозкам) и значениями $z_{но}$ не выше 0.40-0.45 (большие допускаемые значения соответствуют магистральным и меньшие местным перевозкам).

Порядок окончательного формирования системы маршрутов для освоения перемещения ресурсов соответствует ранее приведенному для метода гарантированного эффекта.

3.8.2. Маршрутизация перемещения мелких партий ресурсов

Задача маршрутизации перевозок мелких партий ресурсов состоит в том, чтобы найти такое множество маршрутов, на которых достигались бы минимальные издержки на транспортирование (минимальный общий пробег транспортных средств, минимальное общее время работы транспортных средств, минимальная общая сумма произведений разрешенной максимальной массы на пробег транспортных средств, минимальное общее время на доставку ресурсов и т.п.). Наиболее часто в качестве критерия оптимальности решения задачи принимается минимум общего пробега транспортных средств

$$\sum_{j=1}^{n} l_{oj} = \min ,$$

где l_{oj} – пробег за оборот транспортного средства на j-м маршруте;

n – общее число маршрутов для освоения перевозок заданного количества ресурсов.

При этом могут быть заданы ограничения, зависящие от конкретных условий перевозок – число пунктов заезда, длина оборота на маршруте, время доставки ресурса с момента погрузки и другие.

Решение задачи маршрутизации перевозок мелких партий ресурсов, обеспечивающее абсолютный минимум вышеуказанной целевой функции, находится только при пересмотре всех возможных вариантов маршрутов. Точное решение задачи по оптимальному варианту объезда пунктов сформированного маршрута дает метод ветвей и границ решением задачи о коммивояжере. При большом числе пунктов решение этих задач затруднено. Поэтому разработаны приближенные, более "быстрые" методы формирования маршрутов, позволяющие получать результаты, близкие к оптимальным. Наиболее распространенными из них являются метод составления сборочных (развозочных) маршрутов по кратчайшей связывающей сети и метод Кларка-Райта.

В общем случае метод **маршрутизации по кратчайшей связывающей сети** состоит из четырех этапов.

Этап 1. Находится кратчайшая связывающая сеть

Кратчайшая связывающая сеть – это незамкнутая сеть, связывающая две и более вершины, с минимальной суммарной длиной всех соединяющих их звеньев. Нахождение кратчайшей связывающей сети рассмотрено ранее.

Этап 2. Набор пунктов в маршруты

По каждой ветви кратчайшей связывающей сети, начиная с той, которая имеет наибольшее число звеньев, набирают пункты в маршруты. В каждый маршрут группируют пункты с учетом количества ввозимого и вывозимого ресурса, вместимости транспортного средства, а также имеющих место других ограничений. Процесс необходимо начинать от пункта, наиболее удаленного от базового (начального) пункта. Если все пункты данной ветви не могут быть включены в один маршрут, то ближайшие к другой ветви пункты группируются вместе с пунктами этой другой ветви.Результаты набора пунктов в маршруты могут быть приведены в табличной форме (таблице 3.15).

Таблица 3.15 – Набор корреспонденций (перевозок) в маршруты

Маршрут ...			Маршрут ...		
Пункты	Количество ресурса, ед.(т)		Пункты	Количество ресурса, ед. (т)	
	Ввоз	вывоз		ввоз	Вывоз
Всего			Всего		

Этап 3. Определение очередности объезда пунктов маршрута

Этот этап имеет целью связать все пункты маршрута, начиная с начального, такой замкнутой линией, которая соответствует кратчайшему пути объезда этих пунктов. Одним из наиболее простых методов нахождения рационального объезда пунктов маршрута является метод сумм.

Для нахождения пути объезда заданных пунктов с помощью метода сумм строится таблица в виде симметричной матрицы (таблица 3.16). По главной диагонали в ней расположены пункты, включаемые в маршрут. Цифры показывают кратчайшее расстояние между пунктами. Кратчайшие расстояния находятся одним из изученных методов, например, методом потенциалов. Дополнительно в этой матрице имеется итоговая строка – строка сумм. В ней проставляют сумму расстояний по каждому столбцу.

Таблица 3.16 – Таблица для применения метода сумм

Пункт	A	1	...	i	...	m
A	0	l_{A1}		l_{Ai}		l_{Am}
1	l_{1A}	0		l_{1i}		l_{1m}
...			0			
J	l_{jA}	l_{j1}		l_{ji}		l_{jm}
...					0	
M	l_{mA}	l_{m1}		l_{mi}		0
Сумма						

На основании результатов расчетов, указанных в вышеприведенной таблице, строят начальный маршрут, состоящий из базового и двух промежуточных пунктов, имеющих максимальную сумму по своим столбцам. Затем в маршрут включают следующий пункт с максимальной суммой. Чтобы определить, между какими пунктами его следует вставить, надо поочередно рассматривать его включение между каждой парой соседних пунктов. При этом для каждой пары рассматриваемых пунктов находят величину прироста пробега транспортного средства при включении в маршрут дополнительного пункта. Величина этого прироста находится по формуле

$$\Delta l_{kp} = l_{ki} + l_{ip} - l_{kp}\,,$$

где l – расстояние (стоимость, время и т.п.);

k – первый соседний пункт;

p – второй соседний пункт;

i – индекс включаемого пункта.

Из всех получаемых величин Δl_{kp} выбирают минимальную $\min\limits_{kp} \Delta l_{kp}$ и между пунктами k и p включают в маршрут пункт **i**.

Действия продолжают до включения всех пунктов маршрута. Получается в результате последовательность объезда пунктов, дающая путь движения с минимальной или близкой к минимальной длиной.

Этап 4. Определение возможности одновременного развоза и сбора ресурсов на маршруте

Проверка возможности использования транспортного средства для одновременного развоза и сбора ресурсов на маршруте производится в той последовательности объезда пунктов, которая получена на 3-м этапе расчетов. Для проверки составляем таблицу нижеприведенного вида (таблица 3.17).

Таблица 3.17 – Расчет загрузки транспортного средства при перевозке на маршруте

Пункт	Количество ресурса, ед.		
	Разгружено	Загружено	Всего

Количество ресурса на транспортном средстве при выходе из пункта i определяется по формуле

$$Q_{i,i+1} = Q_{i-1,i} - Q_{pi} + Q_{ci},$$

где $Q_{i,i+1}$ – объем перевозимого ресурса на участке маршрута (при выходе из пункта i);

$Q_{i-1,i}$ – объем перевозимого ресурса на предшествующем участке i-1, i маршрута;

Q_{pi} – объем выгрузки ресурса в пункте i;

Q_{ci} – объем погрузки ресурса в пункте i.

Одним из возможных способов избежать перегруза транспортного средства является изменение направления объезда пунктов маршрута. Если изменение порядка объезда не приводит к желаемому результату, то возможно в один или несколько пунктов только завозить ресурс, а затем уже на обратном пути вывозить ресурс в этих пунктов.

В этом случае на кратчайшей связывающей сети выбирают один или несколько пунктов, ближайших по одной ветви к базовому пункту и назначают их для повторного заезда. Число пунктов для повторного заезда нужно принять таким, чтобы общий объем отправления из них ресурса был не менее, чем максимальный перегруз транспортного средства на исходном маршруте. Как только условие выполнилось, дальнейшее включение пунктов для повторного заезда, как правило, нецелесообразно.

Схема маршрута в этом случае выглядит следующим образом:

базовый пункт – загрузка;

пункты повторного заезда – разгрузка;

остальные пункты – разгрузка-загрузка;

пункты повторного заезда – загрузка;

базовый пункт – разгрузка.

Пример.

Составить развозочный маршрут для следующих исходных данных:

транспортная сеть района перевозок, заданная в примере при отыскании кратчайших расстояний;

базовым является пункт 3;

объемы завоза ресурса по пунктам 1 – 3 ед., 2 – 3 ед., 4 – 4 ед., 5 – 1 ед.;

вместимость транспортного средства – 7 ед.;

предельно-допустимое число пунктов заезда – 3.

Решение.

Этап 1. Используем ранее найденную кратчайшую связывающую сеть.

Этап 2. Набор пунктов в маршруты. В соответствии с ранее приведенным алгоритмом производим набор пунктов в маршруты (таблица 3.18).

Таблица 3.18 – Набор корреспонденций (перевозок) в маршруты для данных примера

Маршрут 1			Маршрут 2		
Пункты	Количество ресурса,ед. (т)		Пункты	Количество ресурса, ед. (т)	
	Ввоз	вывоз		ввоз	вывоз
2	3	–	4	4	–
1	3	–	–	–	–
5	1	–	–	–	–
Всего	7	–	Всего	4	–

Этап 3. Для нахождения порядка объезда пунктов строим таблицу по методу сумм для маршрута 1 (таблица 3.19). На маршруте 2 никаких вариантов объезда нет.

Затем строим начальный маршрут из трех пунктов, имеющих максимальную сумму по своим столбцам. В него включаем исходный пункт А, пункт 5 и один из пунктов 2 или 1 (принимаем пункт 2). Получаем маршрут А-5-2-А.

После этого включаем в маршрут пункт 1, который можно вставить между пунктами А и 5, 5 и 2, А и 2.

Таблица 3.19 – Таблица по методу сумм для маршрута 1

Пункт	А	2	1	5
А	0	6	11	8
2	6	0	5	11
1	11	5	0	6
5	8	11	6	0
Сумма	25	22	22	25

Расчет приращений Δl_{kp} для i=1 выполняем в расчетной таблице 3.20.

Таблица 3.20 – Расчет приращений для маршрута 1

Пункт k	Пункт p	l_{ki}	l_{ip}	l_{kp}	Δl_{kp}
А	5	11	6	8	9
5	2	6	5	11	0
А	2	11	5	6	10

Из всех получаемых величин Δl_{kp} выбираем минимальную $\min\limits_{kp} \Delta l_{kp}=0$ (k=5 и p=2) и между пунктами k=5 и p=2 вставляем включаемый пункт 1.

В результате получаем два маршрута:

Маршрут 1 – А-5-1-2-А (протяженность 25) и маршрут 2 – А-4-А (протяженность 6) суммарной протяженностью 31.

Этап 4. Поскольку маршрут только развозочный, то проверка на возможность одновременного развоза и сбора не требуется и маршруты остаются без изменений.

Метод Кларка-Райта предусматривает решение задачи маршрутизации перевозок по сборочным или развозочным маршрутам, осуществляемых в общем случае парком транспортных средств различной вместимости.

Основой решения являются следующие исходные данные:

число K видов транспортных средств различной k-й вместимости и значения вместимостей q_k, $k=\overline{1,K}$;

число промежуточных пунктов (m), в которые доставляется или из которых вывозится ресурс;

количество ресурса (Q_{pi}, $i=\overline{1,m}$), подлежащего завозу или вывозу по промежуточным пунктам;

стоимость перевозок ресурса (расстояния, время перевозок) между пунктами транспортной сети (c_{ij}, $i=\overline{0,m}$; $j=\overline{0,m}$), включающими исходный (индекс 0) и промежуточные пункты;

различного рода ограничения – по числу промежуточных пунктов на маршруте ($n_п$), использованию вместимости транспортных средств, длине маршрута, времени оборота на маршруте и т.п.

Алгоритм одной из реализаций метода следующий:

1) строится система маятниковых маршрутов, на каждом из которых предусматривается обслуживать один пункт. Для каждого такого i-го маршрута ($i=\overline{1,m}$) назначается объем перевозок $Q_i = Q_{pi}$, число пунктов заезда n_i=1 и транспортное средство возможно минимальной вместимости, но не менее Q_i, т.е. $q_i = \min\limits_k q_k$ (для $q_k \geq Q_i$);

2) рассчитываются выигрыши для всех возможных вариантов попарного объединения маршрутов, образованных согласно пункту 1 (см. ниже схему). Выигрыши рассчитываются по формуле

$$\Delta c_{ij} = c_{Ai} + c_{Aj} - c_{ij},\ i = \overline{1, m-1},\ j = \overline{i+1, m},$$

где Δc_{ij} – величина сокращения пробега транспортного средства при объединении маршрутов А-i-А и А-j-А ;

c_{Ai} , c_{Aj} – стоимость перемещения от исходного пункта А соответственно до пунктов i и j;

c_{ij} – расстояние между пунктами i и j.

Вариантность попарного объединения пунктов описывается треугольной матрицей (рисунок 3.28)и соответственно общее число вариантов определяется выражением (m (m-1))/2, где m – число промежуточных пунктов;

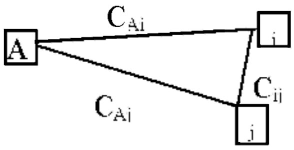


Рисунок 3.28 – Треугольная матрица

3) последовательно производится объединение текущих маршрутов следующим образом:

3.1) находится максимальный выигрыш от возможного попарного объединения исходных маршрутов $\max\limits_{i,j} \Delta c_{ij} = \Delta c_{rs}$, где r и s – соответственно пункты, по которым может быть рассмотрено объединение маршрутов ($i = \overline{1, m-1},\ j = \overline{i+1, m}$). Если максимальный выигрыш Δc_{rs} нулевой или отрицательный – то решение закончено;

3.2) оценивается возможность объединения маршрутов с учетом наличия транспортных средств необходимой вместимости и выполнения других заданных ограничений. Для этого необходимо рассчитать общий объем перевозимого ресурса как сумму ресурсов объединяемых маршрутов $Q_т=Q_r+Q_s$, число пунктов заезда на объединенном маршруте $n_т = n_r+n_s$ и др. Если такое объединение невозможно ($Q_T > \max\limits_k q_k$ или $n_т > n_п$ и др.), то переход на пункт 3.5. Иначе – на 3.3.

3.3) формируется новый объединенный маршрут, состоящий из двух объединяемых по пунктам r и s. Полученный маршрут имеет вид А-p-...-r-s-...-u-А;

3.4) производится корректировка текущего решения в связи с принятием объединения маршрутов по пунктам r и s:

- маршруты r и s, вошедшие в сформированный маршрут, аннулируются и соответственно присваиваются $Q_{r(...)}= 0$ или $Q_{(...)r}= 0$ и $Q_{s(...)}= 0$ или $Q_{(...)s}= 0$;
- формируется индекс нового маршрута, определяемый номерами крайних пунктов (пункты p и u);
- назначается на маршруте с индексом p(u) объем перевозок $Q_{p(u)}= Q_т$ и число промежуточных пунктов заезда $n_{p(u)}=n_т$;
- назначается транспортное средство, удовлетворяющее условию $q_{p(u)} = \min\limits_k q_k$ (для $q_k \geq Q_{u(p)}$);
- если $n_т>2$, то на отрицательный, например, -1 заменяется выигрыш между конечными пунктами маршрута p и u (если p<u, то $\Delta c_{pu} = -1$, иначе $\Delta c_{up} = -1$);
- если $n_r>1$, то на отрицательные заменяются выигрыши всех других пунктов с пунктом r ($\Delta c_{ri}= -1,\ i = \overline{r+1, m}$ и если r>1, то $\Delta c_{ir}= -1,\ i = \overline{1, r-1}$) и если $n_s>1$ – то и с пунктом s ($\Delta c_{si}= -1,\ i = \overline{s+1, m}$ и если s>1, то $\Delta c_{is}= -1,\ i = \overline{1, s-1}$);

3.5) значение выигрыша Δc_{rs} заменяется отрицательным (Δc_{rs}= -1) независимо от того состоялось формирование нового маршрута или нет;

3.6) переход на 3.1.

При реализации алгоритма возможно многократное присвоение отрицательного значения выигрышу для одной и той же пары пунктов, что не влияет на конечный результат.

Пример.

Рассмотрим решение приведенной ранее задачи по методу Кларка-Райта:

1) строим систему маятниковых маршрутов, на каждом из которых предполагается обслуживать один пункт. Для каждого такого маршрута назначаем транспортное средство требуемой вместимости (таблица 3.21).

2) рассчитываются выигрыши для всех возможных вариантов попарного объединения маршрутов, образованных согласно пункту 1. Число вариантов попарного объединения маршрутов равно m(m-1)/2 = 3 · 2 = 6 (6 вариантов). Расчет выигрышей для данных примера приведен в таблице 3.22.

Таблица 3.21 – Данные по исходным маршрутам

Шифр маршрута i	Схема А-i-А	Объем ресурса Q_i	Число промежуточных пунктов n_i	Вместимость транспортного средства q_i
1	3-1-3	3	1	7
2	3-2-3	3	1	7
4	3-4-3	4	1	7
5	3-5-3	1	1	7

Таблица 3.22 – Расчет выигрышей от попарного объединения маршрутов

Маршрут i	Маршрут j	c_{Ai}	c_{Aj}	c_{ij}	Выигрыш Δc_{ij}
1	2	11	6	5	~~12~~ -1 **
1	4	11	3	10	~~4~~ -1 **
1	5	11	8	6	~~13~~ -1 *
2	4	6	3	6	~~3~~ -1 ****
2	5	6	8	11	~~3~~ -1 **
4	5	3	8	5	~~6~~ -1 ***

Примечание. Число символов "*" показывает номер итерации, при которой происходит замена выигрыша на -1

3) последовательно производится объединение маршрутов следующим образом (последняя цифра в подпунктах – номер итерации):

3.1.1) находим максимальный выигрыш от возможного попарного объединения исходных маршрутов. Это два маршрута r =1 и s=5. Поскольку максимальный выигрыш Δc_{rs}=13 ненулевой – то переход на 3.2.1;

3.2.1) оцениваем возможность объединения маршрутов с учетом наличия транспортных средств необходимой вместимости и выполнения других заданных ограничений. Для возможного нового маршрута определяем общий объем перевозимого ресурса как сумму ресурсов рассматриваемых маршрутов $Q_т$=Q_r+Q_s,=Q_1 +Q_5 =3+1=4 и число пунктов заезда на объединенном маршруте $n_т$ = n_r+n_s = n_1+n_5 =1+1=2. В нашем случае полученные параметры не превышают допускаемых и объединение возможно.

3.3.1) формируется новый объединенный маршрут, состоящий из двух объединяемых по пунктам с найденным максимальным выигрышем. Полученный маршрут имеет вид А-1-5-А;

3.4.1)

- маршруты с шифрами 1 и 5, вошедшие в сформированный маршрут, аннулируются (Q_1=0; Q_5=0);
- формируется шифр маршрута 1(5), определяемый номерами крайних пунктов (пункты 1 и 5);
- назначается объем перевозок $Q_{1(5)} = Q_т = 4$ и число промежуточных пунктов заезда $n_{1(5)} = n_т = 2$;
- назначается транспортное средство, удовлетворяющее условию

$$q_{1(5)} = \min_k q_k \,(\text{для } q_k \geq Q_{p(u)} = 4) = 7;$$

- поскольку $n_т$ =2, то далее;
- поскольку не выполняется $n_r>1$ и $n_s>1$, то на 3.5.1;

3.5.1) значение выигрыша $c_{1\text{-}5}$ заменяется отрицательным ($c_{1\text{-}5}$= -1);

3.6.1) переход на 3.1.2.

3.1.2) находим максимальный выигрыш от попарного объединения исходных маршрутов. Это два маршрута r =1 и s=2. Поскольку выигрыш $\Delta c_{1\text{-}2}$=12 ненулевой – то решение не закончено;

3.2.2) оцениваем возможность объединения маршрутов с учетом наличия транспортных средств необходимой вместимости и выполнения других заданных ограничений. Для возможного нового маршрута определяем общий объем перевозимого ресурса как сумму ресурсов объединяемых маршрутов $Q_т=Q_r+Q_s=Q_1+Q_2=4+3=7$ и число пунктов заезда на объединенном маршруте $n_т = n_r+n_s = n_1+n_2 = 2+1=3$. Полученный параметр $Q_т$=7 и $n_т$ =3 не превышают заданных ограничений и поэтому на 3.3.2;

3.3.2) принимаем маршрут А-2-1-5-А;

3.4.2)
- маршруты с шифрами 1(5) и 2, вошедшие в сформированный маршрут, аннулируются ($Q_{1(5)}$=0; Q_2=0);
- формируется шифр маршрута 2(5), определяемый номерами крайних пунктов (пункты 2 и 5);
- назначается объем перевозок $Q_{2(5)}$= 7 и число промежуточных пунктов заезда $n_{2(5)}$=3;
- назначается транспортное средство, удовлетворяющее условию

$$q_{2(5)} = \min_k q_k \,(\text{для } q_k \geq Q_{u(p)} = 7) = 7;$$

- поскольку $n_т>2$, то на -1 заменяется выигрыш между конечными пунктами маршрута 2 и 5 ($\Delta c_{2,5}$= -1);
- поскольку выполняется $n_r>1$, то $\Delta c_{1,i}$= -1, $i = \overline{2, m}$ (для всех выигрышей с объединением по пункту 1) и на 3.5.2;

3.5.2) реальное значение выигрыша $c_{1,2}$ заменяется отрицательным ($c_{1\text{-}2}$= -1);

3.6.2) переход на 3.1.3.

3.1.3) находим максимальный выигрыш от попарного объединения исходных маршрутов. Это маршруты с конечными пунктами r =4 и s=5. Поскольку выигрыш ненулевой– то решение не закончено;

оцениваем возможность объединения маршрутов с учетом наличия транспортных средств необходимой вместимости и выполнения других заданных ограничений. Для возможного нового маршрута определяем общий объем перевозимого ресурса как сумму ресурсов объединяемых маршрутов $Q_т=Q_r+Q_s=Q_2+Q_4=7+4=11$ и число пунктов заезда на объединенном маршруте $n_т = n_r+n_s = n_4+n_5 = 3+1=4$. Полученный параметр $Q_т$=11 превышает вместимость транспортного средства, а также $n_т$ = 4 больше допустимого числа $n_п$ и поэтому маршрут не возможен. Переходим на 3.5.3.

3.5.3) реальное значение выигрыша $c_{4,5}$ заменяется отрицательным ($c_{4,5}$= -1);

3.6.3) переход на 3.1.4)

3.1.4) находим максимальный выигрыш от попарного объединения исходных маршрутов. Это два маршрута r =2 и s=4. Максимальный выигрыш ненулевой и поэтому решение не закончено;

3.2.4) оцениваем возможность объединения маршрутов с учетом наличия транспортных средств необходимой вместимости и выполнения других заданных ограничений. Для возможного нового маршрута определяем общий объем перевозимого ресурса как сумму ресурсов объединяемых маршрутов $Q_T=Q_r+Q_{s,}=Q_2+Q_4=7+4=11$ и число пунктов заезда на объединенном маршруте $n_T = n_r+n_s = n_1+n_3 =3+1=4$. Поскольку объединение невозможно по ограничениям на вместимость и число промежуточных пунктов, то переход на 3.5.4.

3.5.4) реальное значение выигрыша $c_{2,4}$ заменяется отрицательным ($c_{2\text{-}4}= -1$);

3.6.4) переход на 3.1.5)

3.1.5) находим максимальный выигрыш от попарного объединения исходных маршрутов. Это два маршрута r =1 и s=2. При этом максимальный выигрыш отрицательный и поэтому решение закончено.

В результате получены следующие маршруты: А-2-1-5-А или А-5-1-2-А (протяженность 25) и А-4-А (протяженность 6), которые совпали с составленными на основе кратчайшей связывающей сети.

Алгоритм Кларка-Райта, как и метод маршрутизации по кратчайшей связывающей сети, не гарантирует получения оптимального варианта. Для улучшения решения может быть рекомендован поиск оптимального порядка объезда пунктов, входящих в сформированные маршруты, для чего при небольшом числе пунктов представляется возможным выполнить перебор всех возможных вариантов объезда. Оптимальный порядок объезда может быть определен также на основе точных методов решения задачи о коммивояжере.

Осуществляя маршрутизацию мелкопартионных перевозок, следует сгруппировать ресурсы с учетом одновременности доставки в течение суток, недели, месяца и допустимости совместной перевозки.

Компьютерная программа для составления маршрутов на основе изложенного алгоритма дана в приложении 9.

3.9. Задачи дискретной оптимизации

Примерами таких задач являются задачи целочисленного линейного программирования, в том числе задачи о ранце, о назначениях и о коммивояжере.

3.9.1. Целочисленная задача линейного программирования

Задача состоит в следующем:

необходимо найти оптимальные значения управляемых параметров $\{K_i\}$, дающие экстремум целевой функции

$$Z = \sum_{i=1}^{m} c_i K_i = \max_{\{K_i\}}(\min)$$

при ограничениях

$$\sum_{j=1}^{n}\sum_{i=1}^{m} a_{ij} K_i \le=\ge b_j, \quad i = \overline{1,m}; \; j = \overline{1,n}$$

где $K_i = 0, 1, 2, ...;$

c_i – эффект от одной единицы i-го параметра;

K_i – значение i-го оптимизируемого параметра;
a_{ij} и b_j – параметры ограничений;
m – общее число оптимизируемых параметров;
n – общее число ограничений.

Решение задачи без учета целочисленности с последующим округлением, отбрасыванием дробной части и т.п. не обеспечивает получение правильного результата.

Например, для нижеприведенной задачи (рисунок 3.29), как следует из ее графического решения, нецелочисленное решение в точке А (K_1 = 1.6, K_2 = 2.6). При округлении K_1=2, K_2=3 нарушается заданное ограничение, при отбрасывании дробной части K_1 = 1, K_2 = 2 не гарантировано оптимальное решение. Если провести линию дополнительного ограничения, которое не отсекает ни одного целочисленного решения, то очевидно, что максимум Z достигается в точке K_1 =3, K_2=0.

Решение задачи производится по следующему алгоритму:

1) решается задача линейного программирования (ЗЛП) без учета целочисленности оптимизируемых параметров;

2) полученное решение проверяется на целочисленность. Если целочисленно, то решение получено (ВЫХОД), иначе на п. 3;

3) строится дополнительное ограничение, отсекающее часть области допускаемых нецелочисленных решений;

4) решается ЗЛП с дополнительным ограничением;

5) возврат к п.2.

Формирование дополнительных ограничений осуществляется по алгоритму Гомори, называемым правильным отсечением. Отсекающая плоскость должна быть линейной и исключать найденное оптимальное нецелочисленное решение и не отсекать ни одной из допускаемых целочисленных точек [14].

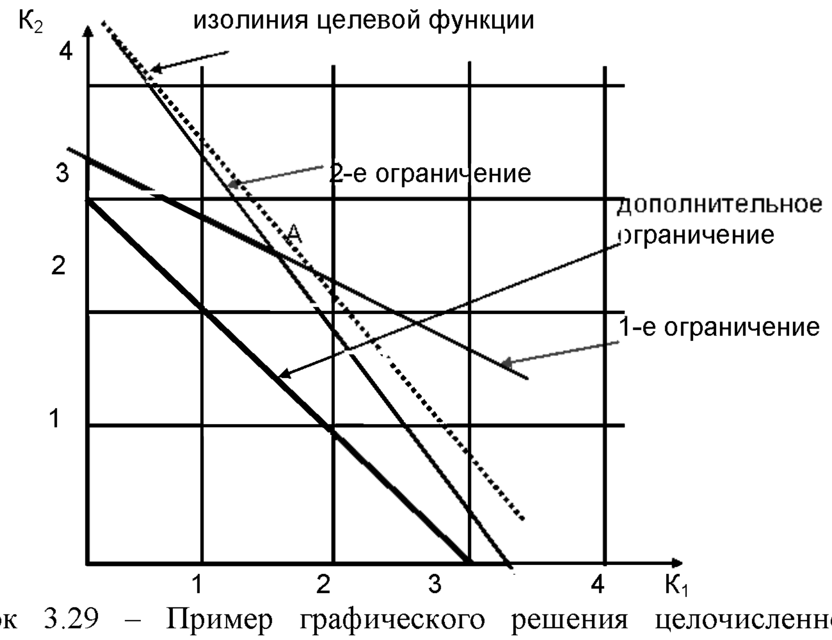


Рисунок 3.29 – Пример графического решения целочисленной задачи линейного программирования

3.9.2. Задача о назначениях

Задача состоит в том, что требуется найти такое множество назначений i-х исполнителей (претендентов) на j-е работы K_{ij} $(i=\overline{1,m}; j=\overline{1,n})$, при которых достигается максимум эффекта

$$Z = \sum_{j=1}^{n}\sum_{i=1}^{m} c_{ij}K_{ij} = \max_{\{K_{ij}\}}$$

и выполняются ограничения

$$\sum_{i=1}^{m} K_{ij} \le 1,\quad i = \overline{1, m};$$

$$\sum_{j=1}^{n} K_{ij} \le 1,\quad j = \overline{1, n};$$

$$K_{ij} = \begin{cases} 1, \text{если } i\text{-й претендент назначен на } j\text{-ю работу} \\ 0, \text{ в противном случае} \end{cases}.$$

Если число претендентов и работ равны между собой, то $m = n$.

Задача решается венгерским методом или как транспортная задача линейного программирования, но на максимум целевой функции.

3.9.3. Задача о ранце (рюкзаке)

Сущность задачи состоит в том, что в ранце можно разместить набор различных предметов общей массой В. Этот набор может включать n видов предметов, каждый предмет типа j имеет массу m_j, $j = \overline{1, n}$. Ценность каждого предмета c_j. Обозначим через K_j – число в наборе предметов j-го типа. Необходимо найти оптимальный набор предметов в ранце, чтобы был максимальный эффект Z

$$Z = \sum_{j=1}^{n} c_j K_j = \max_{\{K_j\}},\quad j = \overline{1, n}$$

при ограничениях

$$K_j = 0,1,2,\ldots \text{ (целочисленно) и}$$

$$\sum_{j=1}^{n} m_j K_j \le B.$$

Задача является целочисленной задачей линейного программирования и решается соответствующим способом.

3.9.4. Задача о коммивояжере

Имеется **n** пунктов, в которых должен побывать коммивояжер. Задана матрица стоимостей (расстояний, времени и т.п.) на переход между пунктами c_{ij}, $i = \overline{1, n}$ и $j = \overline{1, n}$. Коммивояжер должен выйти из одного из пунктов, побывать во всех остальных пунктах по одному разу и вернуться в начальный пункт.

Необходимо найти порядок обхода, чтобы получить минимальную суммарную стоимость посещения всех пунктов.

Введем переменную K_{ij}

$$K_{ij} = \begin{cases} 1, \text{при переходе между пунктами } i \text{ и } j \\ 0, \text{нет перехода между пунктами } i \text{ и } j \end{cases}.$$

Необходимо найти множество $\{K_{ij}\}$, $i = \overline{1, n}$ и $j = \overline{1, n}$, дающее минимум

$$Z = \sum_{j=1}^{n}\sum_{i=1}^{n} c_{ij}K_{ij} = \min_{\{K_{ij}\}}$$

и выполнение ограничений

$$\sum_{j=1}^{n} K_{ij} = 1 \; ; \qquad (*)$$

$$\sum_{i=1}^{n} K_{ij} = 1 \; ; \qquad (**)$$

$$U_i - U_j + nK_{ij} \le n-1, \;\; i \ne j, \qquad (***)$$

где U_i, U_j – целые неотрицательные числа, представляющие собой номера этапов, на которых посещаются соответственно пункты i и j.

Условие (*) означает, что коммивояжер выходит из каждого пункта один раз, а условие (**) – что он входит в каждый пункт только один раз. Условие (***) обеспечивает замкнутость цикла (контура) только на n-м этапе решения задачи.

Задача без учета условия (***), представляет постановку, аналогичную задаче о назначениях и отличается тем, что целевая функция стремится к минимуму (Z→min). Если при ее решении получен замкнутый контур, то это оптимальное решение, а иначе полученное значение целевой функции является той оценкой, которая всегда меньше или равна целевой функции (длине пути) с учетом условия (***).

Для решения задачи применяются приближенные методы – метод ближайшего соседа, метод сумм (изложен ранее для составления сборочно-развозочных маршрутов), метод Кларка-Райта и точный метод – метод ветвей и границ.

Алгоритм метода ближайшего соседа (один из вариантов):

1) создается рабочий массив стоимостей переходов $c^p_{ij} = \begin{cases} c_{ij}, \text{если } i \ne j \\ 1E12, \text{если } i = j \end{cases}$, $i = \overline{1, n}$; $j = \overline{1, n}$;

2) текущее множество переходов коммивояжера L задается нулевым (число переходов m=0). В итоге решения элементы массива L будут представлять перечень пунктов l_k, $k = \overline{1, n+1}$;

3) находится переход коммивояжера максимальной стоимости из массива c^p_{ij} как $\max\limits_{ij} c^p_{ij} = c_{rs}\,(i{\ne}j)$;

4) изменяется m (m=m+1) и один из пунктов r или s вводится во множество L (l_m=l_1=r или l_m=l_1=s);

5) составляется путь переходов коммивояжера:

5.1) рассматривается множество M стоимостей переходов, соединенных с пунктами l_1 и l_m, т.е. рассматриваются стоимости переходов $c^p_{l_1 j}$ и $c^p_{l_m j}$ (j не принадлежит множеству L);

5.2) находится переход минимальной стоимости из массива M как $\min\limits_j M = c_{rs}$. Если c_{rs}=1E12, то решение закончено (на 7), а иначе на 5.3;

5.3) изменяется m (m=m+1);

5.4) текущее множество переходов коммивояжера L дополняется переходом rs. Если l_1=r, то l_k=l_{k-1}, $k = \overline{m, 2, -1}$ и l_1= s , а если l_{m-1}=r, то l_m= s;

5.5) если m=2, то $c^p_{rs} = c^p_{sr}$ = 1E12 и на 6, а если иначе, то $c^p_{l_1 l_m} = c^p_{l_m l_1}$ = 1E12;

5.6) если $l_2 = r$, то $c^p_{l_2 k} = c^p_{kl_2} = 1E12,\ k = \overline{1, n}$, а если $l_{m-1}=r$, то $c^p_{l_{m-1}k} = c^p_{kl_{m-1}} = 1E12,\ k = \overline{1, n}$;

6) возврат на 5.1.

7) контур перемещений замыкается путем введения еще одного перехода коммивояжера ($m=m+1$ и $l_m= l_1$). При этом $m=n+1$.

Общая стоимость замкнутого контура переходов коммивояжера

$$C = \sum_{k=1}^{n} c_{l_k l_{k+1}} .$$

С целью упрощения алгоритм не исключает повторную замену стоимостей переходов на бесконечно большое значение (принято 1E12).

Пример.

Задана транспортная сеть, схема которой приведена на рисунке 3.30, и стоимости переходов в таблице 3.23. Необходимо решить задачу коммивояжера методом ближайшего соседа.

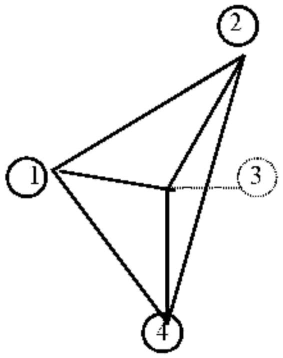


Рисунок 3.30 – Схема транспортной сети

Таблица 3.23 – Стоимости переходов

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

Решение.

1) создается рабочий массив $c^p_{ij} = \begin{cases} c_{ij}, \text{если } i \neq j \\ 1E12, \text{если } i = j \end{cases}$, $i = \overline{1, n}$, $j = \overline{1, n}$ (таблица 3.24);

2) задается пустым текущее множество переходов коммивояжера L (число переходов $m = 0$).

3) находится переход коммивояжера максимальной стоимости из массива c^p_{ij} как $\max_{ij} c^p_{ij} = c_{2\text{-}4} (i \neq j)$;

4) изменяется m на $m=m+1=0+1=1$ и один из пунктов, например пункт 2 вводим во множество L ($l_m=l_1=2$);

Таблица 3.24 – Рабочий массив стоимостей переходов (исходный)

I	J			
	1	2	3	4
1	1E12	7	5	6
2	7	1E12	6	8
3	5	6	1E12	4
4	6	8	4	1E12

5) составляется путь переходов коммивояжера (последняя цифра подпункта – номер итерации):

5.1.1) находится множество М стоимостей переходов, соединенных с пунктом 2, т.е. рассматриваются при данной итерации переходы c^p_{2j};

5.2.1)находится во множестве М переход с минимальной стоимостью как $\min\limits_j M = \min\limits_j \{c_{2,1}; c_{2,3}; c_{2,4}\} = c_{2\text{-}3}$=6. Так как $c_{2\text{-}3} \neq$1E12, то на 5.3.1;

5.3.1) изменяется m на m=m+1=1+1=2;

5.4.1) текущее множество перемещений коммивояжера L дополняется переходом 2-3. Поскольку l_1=r, то $l_m=l_2=l_1$=2 и l_1=s=3. Имеем L={l_1=3 и l_2=2};

5.5.1) поскольку m= 2, то $c^p_{2\text{-}3} = c^p_{3\text{-}2}$ = 1E12 и на 6.

Все замены приведены в таблице 3.25 рабочего массива – 1-е изменение;

Таблица 3.25 – Рабочий массив стоимостей переходов (1-е изменение)

I	J			
	1	2	3	4
1	1E12	7	5	6
2	7	1E12	1E12	8
3	5	1E12	1E12	4
4	6	8	4	1E12

6.1) на 5.1.2;

5.1.2) находится множество М стоимостей переходов, соединенных с пунктами l_1 и l_m ,т.е. рассматриваются звенья c^p_{3j} и c^p_{2j};

5.2.2) находится переход минимальной стоимости из массива М как $\min\limits_j M = c_{3\text{-}4}$=4. Так как $c_{3\text{-}4} \neq$1E12, то на 5.3.2;

5.3.2) изменяется m на m=m+1=2+1=3;

5.4.2) текущее множество перемещений коммивояжера L дополняется переходом 3-4. Поскольку l_1=r=3, то $l_m=l_3=l_2$=2, $l_{m-1}=l_2=l_1$=3 и l_1=s=4. Имеем L={l_1=4, l_2=3, l_3=2};

5.5.2) поскольку m>2, то $c^p_{l_1 l_3} = c^p_{l_3 l_1}$ =1E12, т.е. $c^p_{4\text{-}2} = c^p_{2\text{-}4}$ = 1E12;

5.6.2) поскольку l_2 = r =3 , то $c^p_{3k} = c^p_{k3}$ = 1E12, $k = \overline{1, n}$ (в третьей строке и 3-м столбце проставляются 1E12). Все замены приведены в таблице 3.26 рабочего массива – 2-е изменение;

Таблица 3.26 – Рабочий массив стоимостей переходов (2-е изменение)

I	j			
	1	2	3	4
1	1E12	7	1E12	6
2	7	1E12	1E12	1E12
3	1E12	1E12	1E12	1E12
4	6	1E12	1E12	1E12

6.2) на 5.1.3.

5.1.3) находится множество М стоимостей переходов, соединенных с пунктами l_1 и l_m (т.е. рассматриваются звенья c^p_{4j} и c^p_{2j});

5.2.3) находится переход минимальной стоимости из массива М как $\min\limits_j M = c_{4\text{-}1}$=6. Так как $c_{4\text{-}1} \neq$1E12, то на 5.3.3;

5.3.3) изменяется m на m=m+1=3+1=4;

5.4.3) текущее множество перемещений коммивояжера L дополняется переходом 4-1. Поскольку

l_1=r, то $l_m=l_4=l_3=2$, $l_{m-1}=l_3=l_2=3$, $l_{m-2}=l_2=l_1=4$ и l_1=s=1. Имеем L={l_1=1, l_2=4 , l_3=3 и l_4=2};

5.5.3) поскольку m>2, то $c^p_{1\text{-}2} = c^p_{2\text{-}1} =$ 1E12;

5.6.3) поскольку l_2= r =4 , то $c^p_{4k} = c^p_{k4}$= 1E12, $k = \overline{1, n}$.

Все замены приведены в таблице 3.27 рабочего массива – 3-е изменение;

Таблица 3.27 – Рабочий массив стоимостей переходов (3-е изменение)

I	j			
	1	2	3	4
1	1E12	1E12	1E12	1E12
2	1E12	1E12	1E12	1E12
3	1E12	1E12	1E12	1E12
4	1E12	1E12	1E12	1E12

6.3) на 5.1.4.

5.1.4) находится множество М стоимостей переходов, соединенных с пунктами l_1 и l_m (т.е. рассматриваются переходы c^p_{1j} и c^p_{2j});

5.2.4) находится переход минимальной стоимости из массива М как $\min\limits_j M$ = 1E12 (любое звено). Так как любой элемент массива равен 1E12, то на 7;

7) контур перемещений замыкаем путем введения еще одного перехода (m=m+1=4+1=5 и $l_5 = l_1$=1. Тогда контур перемещений коммивояжера 1-4-3-2-1, так как L={l_1=1, l_2=4, l_3=3, l_4=2 и l_5=1}. Любой из пунктов может быть началом переходов данной последовательности. При этом общая стоимость замкнутого контура переходов коммивояжера С не изменится

$$C = \sum_{k=1}^{n} c_{l_k l_{k+1}} = c_{1\text{-}4} + c_{4\text{-}3} + c_{3\text{-}2} + c_{2\text{-}1} = 6+4+6+7 = 23 \text{ ед.}$$

Задача о коммивояжере на основе алгоритма метода сумм решается в следующем порядке:

1) принимается один из пунктов за начальный (базовый);

2) считается, что все другие пункты входят в путь перемещений;
3) пункты поочередно включаются в путь перемещений по алгоритму метода сумм.

Задача о коммивояжере на основе алгоритма метода Кларка-Райта, применяемого для маршрутизации перевозок мелких партий ресурса, решается с учетом следующих условий:
1) исходный пункт – один из пунктов, принадлежащий звену с наибольшей стоимостью;
2) единичные объемы ресурса по пунктам;
3) допускаемое число промежуточных пунктов заезда не менее n-1;
4) один вид транспортного средства по вместимости и эта вместимость не менее n-1;
5) объединение отдельных маршрутов с нулевыми выигрышами до полной их взаимной увязки в один маршрут.

Пример.

Решить ранее поставленную задачу на основе метода Кларка-Райта. Допускаемое число промежуточных пунктов принимаем не менее 3, вместимость транспортного средства – не менее трех.

Решение.

По алгоритму метода Кларка-Райта выполняем следующее:

1) в качестве начального пункта принимаем пункт 2, как принадлежащий звену с максимальной длиной (A=2) и строим систему маятниковых маршрутов, на каждом из которых предполагается обслуживать один пункт. Для каждого такого маршрута назначаем транспортное средство заданной вместимости (таблица 3.28).

Таблица 3.28 – Система первоначальных маршрутов

Шифр маршрута i	Схема A-i-A	Объем ресурса	Число промежуточных пунктов n_i	Вместимость трансп.средства q_i
1	2-1-2	1	1	3
3	2-3-2	1	1	3
4	2-4-2	1	1	3

2) рассчитываются выигрыши для всех возможных вариантов попарного объединения маршрутов, образованных согласно пункту 1.

Выигрыши рассчитываются по формуле

$\Delta c_{ij} = c_{Ai} + c_{Aj} - c_{ij}$,

где c_{ij} – величина сокращения пробега транспортного средства при объединении маршрутов A-i-A и A-j-A ;

c_{Ai}, c_{Aj} – стоимость перемещения от начального пункта A соответственно до пунктов i и j;

c_{ij} – расстояние между пунктами i и j, ед.

Возможные варианты и выигрыши приведены в таблице 3.29.

Таблица 3.29 – Расчет выигрышей от попарного объединения маршрутов

Маршрут i	Маршрут j	Выигрыш Δc_{ij}	Примечание
1	3	7+6-5=8	-1 (3.4.2)
1	4	7+8-6=9	-1 (п.3.4.2 и 3.5.2)
3	4	6+8-4=10	-1 (п.3.5.1)

3) последовательно производится объединение маршрутов следующим образом:

3.1.1) находим максимальный выигрыш от попарного объединения исходных маршрутов. Это два маршрута r =3 и s=4. Поскольку максимальный выигрыш >0, то переход на п. 3.2.1);

3.2.1) оцениваем возможность объединения маршрутов с учетом наличия транспортных средств необходимой вместимости и выполнения других заданных ограничений. Для возможного нового маршрута определяем общий объем перевозимого ресурса как сумму ресурсов объединяемых маршрутов $Qr_{(s)}=Qr+Qs=Q_3+Q_4=1+1=2$ и число пунктов заезда на объединенном маршруте $n_{r(s)}=n_r+n_s=n_3+n_4=1+1=2$. В нашем случае полученные параметры не превышают допускаемых и объединение возможно.

3.3.1) Формируем новый объединенный маршрут, состоящий из двух объединяемых по пунктам с найденным максимальным выигрышем. Полученный маршрут имеет вид А-3-4-А;

3.4.1)

- маршруты с шифрами 3 и 4 аннулируются ($Q_3=0$, $Q_4=0$);
- формируется шифр маршрута, определяемый номерами крайних пунктов (пункты 3 и 4);
- назначается объем перевозок $Q_{3(4)}=2$ и число промежуточных пунктов заезда $n_{3(4)}=2$;
- назначается транспортное средство, удовлетворяющее условию $q_{3(4)}=\min\limits_i q_i \ (\text{для } q_i \geq Q_{3(4)})=3$;
- поскольку $n_{r(s)}=2$ то далее;

поскольку не выполняется $n_r>1$ и $n_s>1$, то на 3.5.1);

3.5.1) реальное значение выигрыша $c_{3\text{-}4}$ заменяется отрицательным ($c_{3\text{-}4}=-1$);

3.6.1) переход на 3.1.2);

3.1.2) находим максимальный выигрыш от попарного объединения исходных маршрутов. Это два маршрута $r=1$ и $s=4$. Поскольку выигрыш не отрицательный – то решение не закончено;

3.2.2) оцениваем возможность объединения маршрутов с учетом наличия транспортных средств необходимой вместимости и выполнения других заданных ограничений. Для возможного нового маршрута определяем общий объем перевозимого ресурса как сумму ресурсов объединяемых маршрутов $Q_{r(s)}=Q_r+Q_s,=Q_1+Q_4=1+2=3$ и число пунктов заезда на объединенном маршруте $n_{r(s)}=n_r+n_s=n_1+n_4=1+2=3$. Полученные параметры не превышают допускаемых и объединение возможно.

3.3.2) Формируем новый объединенный маршрут, состоящий из двух объединяемых по пунктам с найденным максимальным выигрышем. Полученный маршрут имеет вид А-3-4-1-А;

3.4.2)

- маршруты с шифрами 3(4) и 1 аннулируются ($Q_{3(4)}=0$, $Q_1=0$);
- формируется шифр маршрута, определяемый номерами крайних пунктов (пункты 3 и 1);
- назначается объем перевозок $Q_{3(1)}=3$ и число промежуточных пунктов заезда $n_{3(1)}=3$;
- назначается транспортное средство, удовлетворяющее условию $q_{3(1)}=\min\limits_i q_i \ (\text{для } q_i \geq Q_{3(1)})=3$;
- поскольку $n_{r(s)}>2$, то на -1 заменяется выигрыши между пунктами маршрута 1 и 3 ($\Delta c_{1,3}=-1$);
- поскольку выполняется $n_r>1$, то $\Delta c_{4i}=\Delta c_{i4}-1$, $i=\overline{1,m}$;

3.5.2) реальное значение выигрыша $c_{1,4}$ заменяется отрицательным ($c_{1,4}=-1$);

4) переход на 3.1.3);

3.1.3) находим максимальный выигрыш от попарного объединения исходных маршрутов. Это два маршрута $r=1$ и $s=3$. При этом максимальный выигрыш отрицательный и поэтому решение закончено.

В результате получен следующий путь коммивояжера: 2-3-4-1-2 (длина пути 23 ед.), который совпал с обратным путем пути 1-4-3-2-1, полученным методом ближайшего соседа (длина пути 23 ед.).

Метод Кларка-Райта как и метод "ближайшего соседа" не гарантирует получение оптимального решения. Такое решение обеспечивает метод ветвей и границ.

Метод ветвей и границ – один из методов решения различных задач комбинаторного типа. Для решения задачи о коммивояжере может применяться алгоритм Литтла.

Алгоритм простейшей реализации метода ветвей и границ следующий:

1) принимается один из пунктов за начальный пункт ветвления, например, один из пунктов, принадлежащих переходу с наибольшей стоимостью (длиной). Стоимость (длина) ветвления у начального пункта принимается равной нулю;

2) из пункта с минимальной стоимостью ветвления (минимальной текущей оценкой границы ветвления) производятся ветвления (включение переходов), не приводящие к преждевременному зацикливанию (в ветви отсутствуют пункты с одинаковыми номерами кроме последнего n-го шага ветвления по каждой ветви) и рассчитываются стоимости ветвления у вновь включенных в ветви пунктов; каждая ветвь на n-м шаге замыкается на начальный пункт. Стоимость ветвления у вновь включенных пунктов рассчитывается по формуле $Z_{ji}=Z_{j,i-1} + c_i$, где Z_{ji} и $Z_{j,i-1}$– соответственно оценка стоимости j-й ветви на шаге i и i-1;c_i – стоимость перехода, включаемого в ветвь на i-м шаге;

3) находится минимальное значение из всех рассчитанных стоимостей дерева ветвления. Если какая-то ветвь имеет число переходов (звеньев) n и минимальное значение стоимости ветвления, то оптимальное решение получено, а иначе необходимо продолжать ветвление (переход на п. 2).

Одна из ветвей с минимальным значением стоимости ветвления у конечного пункта и включающая все n пунктов, дает оптимальное решение.

Пример.

Решить ранее поставленную задачу методом ветвей и границ.

Решение.

В качестве начального пункта принимаем пункт 2, как принадлежащий переходу с максимальной стоимостью.

Порядок ветвления и оценка стоимостей ветвей показаны на рисунке 3.31 (в треугольнике даны номера пунктов, в квадрате текущие оценки ветвления и возле линий стоимости переходов).

Получены оптимальные решения (2-1-3-4-2 или 2-4-3-1-2) с Z=23, которые дают оптимальный вариант перемещений коммивояжера для рассматриваемого примера.

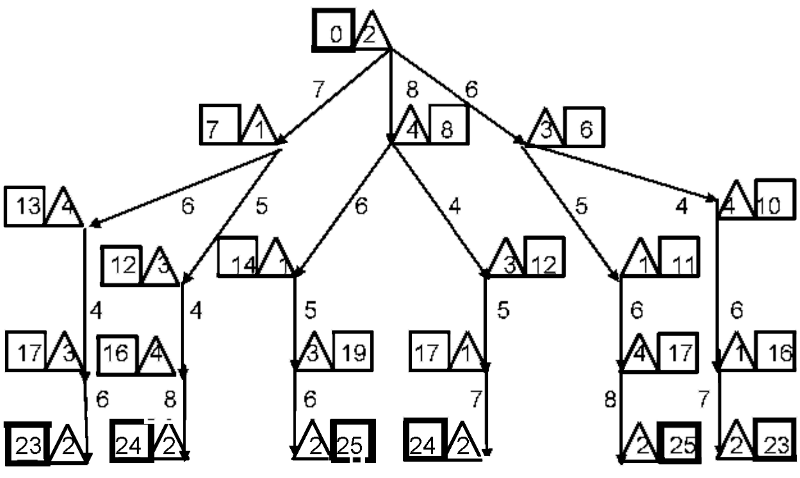


Рисунок 3.31 – Пример решения задачи о коммивояжере методом ветвей и границ

3.10. Задачи упорядочения и согласования

Задачи упорядочения – это задачи определения оптимальной последовательности событий, а задачи согласования рассматривают сетевое планирование и управление.

Основа решения первых – теория расписаний, вторых – теория графов.

Предметом сетевого планирования и управления (СПУ) является разработка и оптимизация сетевых графиков.

Сетевой график – это ориентированный граф, дуги которого имеют одну или несколько числовых характеристик. Дугами изображают работы, а вершинами – события.

Работа – это процесс, сопровождающийся затратами времени и ресурсов. Если имеются затраты только времени – это так называемые фиктивные работы (естественная сушка и т.п.).

Событие – это итог процесса или его части. Оно совершается, когда закончены все предшествующие ему работы.

Путь – это любая непрерывная последовательность работ от исходного события до завершающего (т.е. до конечной цели).

Длина пути определяется суммой продолжительности образующих его работ.

Рассмотрим сетевой график с одним исходным и одним завершающим событием, известными продолжительностями t_{ij} работ ij, где i – событие, соответствующее началу работы и j – соответственно окончанию. Общее число событий m, число работ n.

Ниже приведен пример сетевого графика (рисунок 3.32) и длительности работ (таблица 3.30).

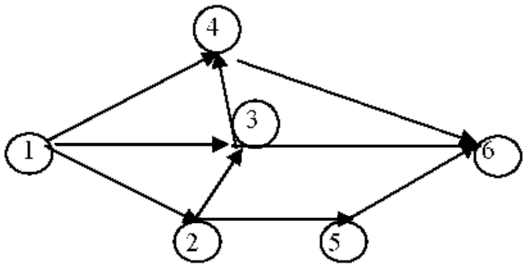


Рисунок 3.32– Пример схемы сетевого графика

Таблица 3.30 – Длительность работ

i	j	t_{ij}
1	2	6
1	3	5
1	4	7
2	3	3
2	5	5
3	4	4
3	6	10
4	6	11
5	6	5

Расчеты для сетевого графика включают отыскание следующих основных параметров:

– ранний и поздний сроки начала работ $T_{р.нij}$ и $T_{п.нij}$;
– ранний и поздний сроки окончания работ $T_{р.оij}$ и $T_{п\ оij}$;
– ранний и поздний сроки наступления событий $T_{рi}$ и $T_{пi}$;
– полный и свободный резервы времени каждой работы $R_{пij}$ и $R_{сij}$;
– резерв времени событий R_i;
– критическое время сетевого графика $t_{кр}$ и перечень работ, образующих критический путь;
– полный резерв R_l времени путей l, альтернативных критическому.

На основании этих характеристик определяется перечень работ, образующих критический путь – путь максимальной длины от начального до завершающего события, имеющий критическое время $t_{кр}$.

Для расчета необходимо произвести правильную нумерацию событий, т.е. для любой работы ij номер предшествующего события i должен быть меньше номера последующего события j.

Алгоритм вычислений

1) Присваивается исходному событию начальный, например, нулевой момент времени раннего срока свершения $T_{pi=1}$=0;

2) Последовательно для каждого события $j = \overline{2, m}$ рассчитываются:

$$T_{p.oij} = T_{pi} + t_{ij};$$
$$T_{pj} = \max_{i \in B_j} T_{p.oij},$$

где B_j – множество событий i, соединенных работами с j- м событием.

3) Критическое время, представляющее минимальный период времени в течение которого может быть выполнен весь процесс сетевого графика, $t_{кр} = T_{pm}$.

4) Определяется критический путь. Это путь, продолжительность которого равна критическому времени, и который включает работы, определяющие критическое время.

5) Поздний срок свершения завершающего события принимается равным критическому $t_{кр}$ или заданному директивному времени $t_{дир}$ ($t_{дир} \geq t_{кр}$):

$$T_{пm} = t_{кр} \text{ или } T_{пm} = t_{дир}.$$

6) Последовательно для каждого пункта $i = \overline{m-1, 1, -1}$ рассчитываются:

$$T_{п.нij} = T_{пj} - t_{ij};$$
$$T_{пi} = \min_{j \in A_j} T_{п.нij},$$

где A_i – множество событий j, соединенных работами с i-м событием.

В результате находятся $T_{пi}$, $i = \overline{1, m}$.

7) Рассчитываются резервы времени событий

$$R_i = T_{пj} - T_{pi}.$$

8) Рассчитываются полные резервы времени работ – максимальное время на которое можно отсрочить или увеличить продолжительность работы ij, не изменяя установленного позднего срока наступления завершающего события

$$R_{пij} = T_{пj} - T_{p.oij} = T_{пj} - T_{pi} - t_{ij}.$$

9) Рассчитываются свободные резервы времени работ – максимальное время на которое можно отсрочить или увеличить продолжительность работы ij, при условии, что все события будут выполнены в свои ранние сроки

$$R_{cij} = T_{pj} - T_{p.oij} = T_{pj} - T_{pi} - t_{ij}.$$

Величина свободного резерва меньше или равна величине полного резерва ($R_{cij} \leq R_{пij}$).

10) Полный резерв времени интересующих альтернативных путей определяется как разность между длиной критического пути и длиной любого другого полного пути t_l

$$R_l = t_{кр} - t_l.$$

Пример.

Рассчитать параметры приведенного выше сетевого графика. Директивное время принять равным критическому пути.

Решение.

1) $T_{i=1}=0$;

2.1) j=2

$T_{р.о1,2} = T_{р1} + t_{1,2} = 0+6=6$

$T_{р2} = \max\limits_{i \in B_j} T_{р.оij} = \max\limits_{i \in B_j}\{6\} = 6$;

2.2) *j*=3

$T_{р.о1,3} = T_{р1} + t_{1,3} = 0+5=5$

$T_{р.о2,3} = T_{р2} + t_{2,3} = 6+3=9$

$T_{р3} = \max\limits_{i \in B_j} T_{р.оij} = \max\limits_{i \in B_j}\{5;9\} = 9$;

2.3) *j*=4

$t_{р.о1,4} = T_{р1} + t_{1,4} = 0+7=7$

$t_{р.о3,4} = T_{р3} + t_{3,4} = 9+4=13$

$T_{р4} = \max\limits_{i \in B_j} T_{р.оij} = \max\limits_{i \in B_j}\{7;13\} = 13$;

2.4) j=5

$t_{р.о2,5} = T_{р2} + t_{2,5} = 6 + 5 = 11$

$T_{р5} = \max\limits_{i \in B_j} T_{р.оij} = \max\limits_{i \in B_j}\{11\} = 11$;

2.5) *j*=6

$T_{р.о3,6} = T_{р3} + t_{3,6} = 9 + 10 = 19$

$T_{р.о4,6} = T_{р4} + t_{4,6} = 13 + 11 = 24$

$T_{р.о5,6} = T_{р5} + t_{5,6} = 11 + 5 = 16$

$T_{р6} = \max\limits_{i \in B_j} T_{р.оij} = \max\limits_{i \in B_j}\{19;24;16\} = 24$.

Результаты расчетов сведены в таблице 3.31.

3) $t_{кр} = T_{рm} = 24$.

4) Критический путь включает работы, определяющие критическое время (работы 1-2, 2-3, 3-4, 4-6).

5) $T_{пm} = t_{кр} = 24$ (m=6).

Таблица 3.31 – Расчет ранних, поздних сроков свершения и резервов времени событий

Событие i	$T_{рi}$	$T_{пi}$	R_i
1	0	0	0
2	6	6	0
3	9	9	0
4	13	13	0
5	11	19	8
6	24	24	0

6.1) *i*=5

$T_{п.н5,6} = T_{п6} - t_{5,6} = 24-5=19$

$T_{п5} = \min\limits_{j \in A_i} T_{п.нij} = \min\limits_{j \in A_i}\{19\} = 19$;

6.2) *i*=4

$T_{п.н4,5} = T_{п6} - t_{4,6} = 24- 11 =13$

$T_{п4} = \min\limits_{j \in A_i} T_{п.нij} = \min\limits_{j \in A_i}\{13\} = 13$;

6.3) i=3

$T_{п.н3,4} = T_{п4} - t_{3,4} = 13 - 4 = 9$

$T_{п.н3,6} = T_{п6} - t_{3,6} = 24 - 10 = 14$

$T_{п3} = \min\limits_{j \in A_i} T_{п.нij} = \min\limits_{j \in A_i}\{9;14\} = 9$;

6.4) i=2

$T_{п.н2,3} = T_{п3} - t_{2,3} = 9 - 3 = 6$

$T_{п.н2,5} = T_{п5} - t_{2,5} = 19 - 5 = 14$

$T_{п2} = \min\limits_{j \in A_i} T_{п.нij} = \min\limits_{j \in A_i}\{6;14\} = 6$;

6.5) i=1

$T_{п.н1,2} = T_{п2} - t_{1,2} = 6 - 6 = 0$

$T_{п.н1,3} = T_{п3} - t_{1,3} = 9 - 5 = 4$

$T_{п.н1,4} = T_{п4} - t_{1,4} = 13 - 7 = 6$

$T_{п1} = \min\limits_{j \in A_i} T_{п.нij} = \min\limits_{j \in A_i}\{0;4;6\} = 0$.

Результаты расчетов занесены в таблицу.

7) Расчет резервов времени событий $R_i = T_{пj} - T_{рi}$ произведен в таблице 3.32.

8) и 9) Расчет полных и свободных резервов времени работ приведен в нижеприведенной таблице.

10) Одним из альтернативных путей является путь 1-2-5-6 длиной 16 ед. Полный резерв этого пути равен разности между длиной критического пути и его длиной (24-16=8) и равен 8 ед.

Текст учебной программы на Бейсике для расчета основных параметров сетевого графика приведен в приложении 10.

Таблица 3.32 – Расчет резервов работ

i	j	t_{ij}	$T_{рi}$	$T_{рj}$	$T_{пj}$	$R_{пij}$	$R_{сij}$
1	2	6	0	6	6	0	0
1	3	5	0	9	9	4	4
1	4	7	0	13	13	6	6
2	3	3	6	9	9	0	0
2	5	5	6	11	19	8	0
3	4	4	9	13	13	0	0
3	6	10	9	24	24	5	5
4	6	11	13	24	24	0	0
5	6	5	11	24	24	8	8

Для задач СПУ с вероятностным временем выполнения операций дается вероятностная оценка параметров.

Оптимизация и рационализация сетевых графиков производится путем сокращения продолжительности критического пути за счет изменения топологии и детализации работ. Уменьшение длительности критического пути может быть достигнуто также сокращением длительности работ за счет изменения технологии, применения более высокопроизводительной техники, переброски ресурсов с некритического пути на критический.

Метод изменения топологии сетевого графика состоит в возможном изменении последовательности работ, в том числе параллельном их проведении.

Метод детализации применяют в случае, если отдельные работы можно разделить на элементы с последующим изменением их топологии.

3.11. Состязательные задачи

Состязательные задачи (игры) могут быть:

по вариантности стратегий – с "чистой" (применяется одна из возможных стратегий) или смешанной (если применяется несколько стратегий);

по числу применяемых стратегий – на конечные и бесконечные;

по количественному результату – с нулевой или ненулевой суммой (разностью);

по характеру взаимоотношений игроков – некооперативные (антагонистические) и кооперативные (коалиционные);

по числу сторон – двух или многих игроков;

по характеру протекания – непрерывные и дискретные;

по количеству информации у сторон – с полной, с вероятностной или с отсутствием, в т.ч. игры с природой, когда партнера заменяет среда, безразличная к действиям игрока.

Простейшей и наиболее разработанной является теория "матричных" игр двух сторон с нулевой суммой.

Исходные данные задаются в виде матрицы выигрышей c_{ij} (таблица 3.33).

Таблица 3.33 – Пример матрицы выигрышей

Стратегии стороны A_i	Стратегии стороны B_j				Наименьший Выигрыш
	B1	B2	B3	B4	
A1	11	15	13	10	10
A2	12	14	11	16	11
A3	9	11	12	15	9
Наибольший Проигрыш	12	15	13	16	–

В этом примере сторона А имеет три возможные стратегии А1, А2, А3, а В – четыре (В1, В2, В3, В4). Сторона А не знает, как поступит сторона В, однако действуя наиболее целесообразно, она должна выбирать стратегию А2, которая гарантирует ей наибольший (11) из трех наименьших выигрышей (10,11,9).

Принято называть, что сторона руководствуется принципом максиминного выигрыша:

$$a = \max_i (\min_j c_{ij}) .$$

Определяемая таким образом величина (а) называется нижней ценой игры, максиминным выигрышем или максимином. Для приведенного примера а=11.

Если рассуждать аналогично, то сторона В должна выбирать стратегию В1, которая гарантирует ей наименьший (12) из четырех возможных наибольших проигрышей, т.е. она руководствуется принципом максиминного проигрыша:

$$b = \min_j (\max_i c_{ij}) .$$

Величина (b) называется верхней ценой игры, или минимаксом. Для приведенного примера b=12.

Справедливо, что b≥a.

Такая стратегия сторон называется принципом минимакса или принципом осторожности.

Если а = в, то такая точка называется седловой.

Рациональные правила поведения сторон:

1. Если известна стратегия стороны В, то сторона А должна выбрать стратегию A_i, которая дает максимальный выигрыш.

2. Если стратегия стороны В неизвестна, то сторона А должна воспользоваться своей максиминной стратегией.

3. Если стратегия стороны В неизвестна, но игра имеет седловую точку, то наиболее выгодно для стороны А не отклоняться от седловой точки.

Если матрица игры не имеет седловой точки, то оказывается, что для определения успеха необходимо выбирать стратегии сторон А и В с определенными вероятностями (частотами) при многократной игре. Такие стратегии называются смешанными.

Решение игровых задач в смешанных стратегиях осуществляется по итеративным алгоритмам Брауна или Неймана или сводится к задаче линейного программирования.

В основе алгоритма Брауна (фиктивной игры) лежит предположение, что игра играется много раз, а стороны выбирают свои стратегии, руководствуясь опытом ранее сыгранных партий.

Если считать, что выполнено k итераций и в результате получены оценки смешанных стратегий $A_c = (A_1, A_2, ..., A_k)$ стороны А и $B_c = (B_1, B_2, ..., B_k)$ стороны В, то на очередной итерации стороной А выбирается такая "чистая" стратегия, которая обеспечит ей максимум в предположении, что сторона В применит смешанную стратегию B_c. Аналогично сторона В применяет "чистую" стратегию, которая дает минимум выигрыша стороне А, если последней будет использована смешанная стратегия A_c.

Для парной игры обозначим вероятность применения стратегии A_i через p_i, а B_j через q_j (A_i – стратегия стороны А, B_j – стратегия стороны В).

Сумма вероятностей всех стратегий для каждой из сторон равна 1. Тогда

$$\sum_{i=1}^{m} p_i = 1 ; \sum_{j=1}^{n} q_j = 1 .$$

Средний выигрыш стороны А

$$Z = c_{1,1}p_1q_1 + c_{1,2}p_1q_2 + ... = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}p_iq_j ,$$

где m и n – соответственно число различных стратегий сторон А и В.

Для поиска оптимума необходимо взять частные производные и приравнять их к нулю:

$$\frac{\partial Z}{\partial p_i} = 0,\ i = \overline{1, m} ;$$
$$\frac{\partial Z}{\partial p_j} = 0,\ i = \overline{1, n} .$$

Решение может быть получено реализацией итерационного процесса путем многократного имитационного моделирования игры. Например, следующей стратегией стороны А для приведенного примера является стратегия А2, дающую максимум при стратегии противоположной стороны В1, а сторона В применит стратегию В3, которая дает минимум выигрыша стороной А при ее предыдущей стратегии А2. При третьей итерации сторона А должна применить стратегию А1, дающую максимум выигрыша (11+13=24) при предыдущих стратегиях (В1,В3) противоположной стороны, а стороне В необходимо применить стратегию B3, которая дает минимум выигрыша стороной A при ее предыдущих стратегиях A2 и A2 (11+11=22). Аналогично итерационный процесс моделирования проводят до равенства значений цен игры сторон с заданной точностью. Пример игры для пяти итераций приведен в таблице 3.34.

Таблица 3.34 – Пример моделирования игры двух сторон

Номер итера-ции k	Выбранная стратегия		Накопленные результаты стороны A при стратегиях стороны B				Накопленные результаты стороны B при стратегиях стороны A		
	Сторона A	Сторона B	B1	B2	B3	B4	A1	A2	A3
1	A2	B1	12	14	11	16	11	12	9
2	A2	B3	24	28	22	32	24	23	21
3	A1	B3	35	43	35	42	37	34	33
4	A1	B1	46	58	48	52	48	46	42
5	A1	B1	57	73	61	62	59	58	51

Для рассматриваемого примера при k=5 средняя цена игры стороны A равна Z_A=11.4 (57/5) и стороны B – Z_B=11.8 (59/5). При k=300 p_1=0.3284, p_2=0.6716, p_3=0, q_1=0.6668, q_2=0, q_3=0.3332, q_4=0, Z_A=11.657, Z_B=11.667.

При решении задачи на основе итерационного алгоритма может быть использована его компьютерная реализация. Пример учебной программы приведен в приложении 11.

ЗАКЛЮЧЕНИЕ

На современном этапе развития прикладной математики и вычислительной техники большинство математических моделей может быть реализовано с применением имеющихся компьютерных пакетов прикладных программ. Часть таких пакетов указана в разделе 1.5.

В основу поиска оптимального решения в компьютерных пакетах прикладных программ закладываются как классические методы оптимизации, так и новые подходы, например, генетические алгоритмы [18]. Укрупненная схема отыскания оптимального решения с применением генетического алгоритма может быть следующей:

1) создание начальной популяции, удовлетворяющей ограничениям, и вычисление целевой функции;
2) выполнение операции селекции;
3) выполнение операции скрещивания;
4) выполнение операции мутации;
5) оценка текущего состояния решения задачи исходя из целевой функции и выполнения ограничений. Если условие достижения цели не выполнено, то на п. 2, а иначе формирование оптимального решения.

Оптимизация на основе генетических алгоритмов может произведена, например, на основе применения соответствующей процедуры прикладного пакета MATLAB.

Математические модели могут быть представлены также в виде нейронных сетей, в частности для целей прогнозирования параметров функционирования систем.

Для обработки информации и принятия решений можно широко использовать такую универсальную компьютерную программу как Excel. Она позволяет путем применения различных функций вычислять параметры распределения случайных величин, находить регрессионные зависимости, визуализировать результаты в виде различного вида диаграмм. Применение надстройки Excel позволяет решать оптимизационные задачи, в частности задачи линейного программирования. Средства поиска решения Microsoft Excel используют алгоритм нелинейной оптимизации Generalized Reduced Gradient (GRG2) и алгоритм симплексного метода и метода «branch-and-bound» для решения линейных и целочисленных задач с ограничениями. В составе Microsoft Excel в папке Office\Samples находится книга с примерами (Solvsamp.xls) использования процедуры поиска решения (Solver.xls). Для применения любого из шести примеров: «Структура производства», «Транспортная задача», «График занятости», «Управление капиталом», «Портфель ценных бумаг» и «Проектирование цепи», необходимо открыть книгу, перейдите к нужному листу и выбрать команду **Поиск решения** в меню **Сервис**. В примерах уже подобраны целевая и влияющие ячейки, а также ограничения. Недостаком широкого применения пакета Excel для решения оптимизационных задач является сложность передачи данных, полученных в других компьютерных программах.

В практике транспортной деятельности у специалиста возникают задачи, для решения которых требуется разработать алгоритмы и пользовательские компьютерные программы. Поэтому специалист по организации и выполнению перевозок должен владеть основами алгоритмизации и программирования задач принятия решений по транспортной деятельности, а также навыками применения необходимых компьютерных пакетов прикладных программ.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

ОСНОВНАЯ ЛИТЕРАТУРА

1. Банди, Б. Методы оптимизации. Вводный курс: Пер. с англ. / Б. Банди. – М.: Радио и связь, 1988. – 128 с.
2. Банди, Б. Основы линейного программирования: Пер. с англ. / Б. Банди. –М.: Радио и связь, 1989. – 176 с.
3. Герасимович, А.И. Теория вероятностей и математическая статистика / А.И. Герасимович, Я.И. Матвеева. – Мн.: Выш.шк., 1978. – 301с.
4. Геронимус, Б.Л. Экономико-математические методы в планировании на автомобильном транспорте / Б.Л. Геронимус, Л.В. Царфин. – М.: Транспорт, 1988. – 192 с.
5. Зайченко, Ю.П. Исследование операций / Ю.П. Зайченко. – Киев: Вища шк.,1988. – 549 с.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Балашевич, В.А. Алгоритмизация математических методов планирования и управления / В.А. Балашевич. – Минск: Выш.шк.,1978. – 144с.
2. Ванчукевич, В.Ф. Грузовые автомобильные перевозки / В.Ф. Ванчукевич, В.Н. Седюкевич, В.С. Холупов. – Минск: Выш.шк.,1989. – 272 с.
3. Вентцель, Е.С. Теория вероятностей и ее инженерные приложения / Е.С. Вентцель, Л.А. Овчаров. – М.: Наука, 1988. – 480 с.
4. Гмурман, В.Е. Теория вероятностей и математическая статистика: учебное пособие для вузов / В.Е. Гмурман. – М.: Высш.шк., 2003. – 479 с.
5. Гринчишин, Я.Т. Алгоритмы и программы на Бейсике / Я.Т. Гринчишин, В.И. Ефимов, А.Н. Ломакович. – М.: Просвещение, 1988. – 160 с.
6. Дайитбеков, Д.М. Программное обеспечение статистической обработки данных / Д.М. Дайитбеков, О.В. Калмыкова, А.И. Черепанов. – М.: Финансы и статистика, 1984. – 192 с.
7. Дьяконов, В.П. Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ / В.П. Дьяконов. – М.: Наука,1987. – 240с.
8. Задания и методические указания к контрольным работам по дисциплине «Математические модели в расчетах на ЭВМ» для студентов специальностей 24.01 и 24.04 / В.Н. Седюкевич, Д.В. Рожанский. – Минск: БГПА, 1992. – 38 с.
9. Исследование операций: В 2-х томах / Под ред. Дж. Моудера, С. Элмаграби. – М.: Мир, 1981. Т.1 –712 с., Т. 2 – 677 с.
10. Кожин, А.П. Математические методы в планировании и управлении грузовыми автомобильными перевозками / А.П. Кожин – М.: Высш.шк., 1979. – 304 с.
11. Кудрявцев, Е.М. Исследование операций в задачах, алгоритмах и программах / Е.М. Кудрявцев. – М.: Радио и связь, 1984. – 184 с.
12. Кузнецов, А.В. Руководство к решению задач по математическому программированию / А.В. Кузнецов, Н.И. Холод, Л.С. Костевич. – Минск: Выш.шк., 2001. – 448 с.
13. Лабораторные работы по дисциплине «Математические модели в расчетах на ЭВМ» для студентов спец. 24.01- «Организация перевозок и управление на транспорте» и 24.04- «Организация дорожного движения» / В.Н. Седюкевич, Д.В. Рожанский. – Минск: БГПА, 1993. – 76 с.
14. Сакович, В.А. Исследование операций (детерминированные методы и модели): Справочное пособие / В.А. Сакович. – Минск: Выш. шк., 1985. – 256 с.
15. Советов, Б.Я. Моделирование систем / Б.Я.Советов, С.А.Яковлев. – М.: Высш.шк.,1985. –271 с.
16. Таха, Х. Введение в исследование операций. Пер.с англКн.1 и Кн. 2. / Х. Таха. – М.: Мир,1985. – 479 с. и 496 с.
17. Харин, Ю.С. Основы имитационного и статистического моделирования / Ю.С. Харин, В.И. Малюгин, В.П. Кирлица. – Минск: ДизайнПро, 1997. – 288 с.
18. Гладков, Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик; под ред. В.М. Курейчика. – М.: ФИЗМАТЛИТ, 2006. – 320 с.

ПРИЛОЖЕНИЕ 1
КОМПЬЮТЕРНАЯ ПРОГРАММА ПРИНЯТИЯ РЕШЕНИЙ В УСЛОВИЯХ РИСКА И НЕОПРЕДЕЛЕННОСТИ

```
10 CLS:PRINT"ПРИНЯТИЕ РЕШЕНИЙ В УСЛОВИЯХ РИСКА И НЕОПРЕДЕЛЕННОСТИ"
20 PRINT"ВВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ"
30 INPUT"ЧИСЛО ВОЗМОЖНЫХ ЗНАЧЕНИЙ УПРАВЛЯЕМОГО ПАРАМЕТРА";IK
40 INPUT"ЧИСЛО РАССМАТРИВАЕМЫХ СОСТОЯНИЙ СРЕДЫ";IR
45 DIM X(IK),U(IR),V(IK,IR),VS(IK,IR),P(IR),EP(IK)
50 PRINT"ЗНАЧЕНИЯ УПРАВЛЯЕМОГО ПАРАМЕТРА": FOR I=1 TO IK:PRINT ""I"-ГО":INPUT X(I):NEXT I
60 PRINT"ЗНАЧЕНИЯ ПАРАМЕТРА СОСТОЯНИЯ СРЕДЫ"
65 FOR I=1 TO IR:PRINT ""I" -ГО":INPUT U(I):NEXT I
70 PRINT"ВЕЛИЧИНЫ ЭФФЕКТА V(...,...) ПРИ УКАЗАННЫХ ЗНАЧЕНИЯХ"
75 PRINT"УПРАВЛЯЕМОГО ПАРАМЕТРА И СОСТОЯНИЯ СРЕДЫ"
80 FOR I=1 TO IK:FOR J=1 TO IR: PRINT"V("X(I)","U(J)")=":INPUT V(I,J):NEXT J:NEXT I
90 PRINT:PRINT"ПРИЗНАК УСЛОВИЙ: 1 - В УСЛОВИЯХ РИСКА, 2 - В УСЛОВИЯХ"
95 INPUT"НЕОПРЕДЕЛЕННОСТИ, 3 - ВЫХОД ИЗ ПРОГРАММЫ ";IN
100 IF IN=2 THEN 200
105 IF IN=3 THEN 400
107 REM ****** ПРИНЯТИЕ РЕШЕНИЯ В УСЛОВИЯХ РИСКА
110 PRINT"ВЕРОЯТНОСТИ СОСТОЯНИЯ СРЕДЫ P(...)"
115 FOR I=1 TO IR:PRINT "P("U(I)")=":INPUT P(I):NEXT I
120 ERO=-1E+12:ER=0
130 FOR I=1 TO IK:ER=0
135 FOR J=1 TO IR:ER=ER+V(I,J)*P(J):NEXT J
140 IF ER>ERO THEN ERO=ER:IRO=I
145 NEXT I
150 PRINT"ОПТИМАЛЬНЫЙ ВАРИАНТ ПРИ ЗНАЧЕНИИ УПРАВЛЯЕМОГО "
155 PRINT"ПАРАМЕТРА "USING"#####.###";X(IRO);:PRINT ". ОЖИДАЕМЫЙ ЭФФЕКТ" USING"######.###";ERO
160 GOTO 90
170 REM ****** ПРИНЯТИЕ РЕШЕНИЯ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ
200 PRINT" КРИТЕРИЙ │ОПТ.ЗНАЧ.УПРАВЛ.ПАРАМЕТРА│ЭФФЕКТ": PRINT
210 VW=-1E+06:FOR I=1 TO IK
212 EP(I)=V(I,1):FOR J=1 TO IR-1
215 IF EP(I)>V(I,J+1) THEN EP(I)=V(I,J+1)
220 NEXT J
225 IF VW<EP(I) THEN VW=EP(I):IW=I
230 NEXT I
240 PRINT" ВАЛЬДА";:PRINT TAB(25)USING"#####.###";X(IW),VW
245 INPUT"ВВЕДИТЕ КОЭФФИЦИЕНТ ДОВЕРИЯ ДЛЯ КРИТЕРИЯ ГУРВИЦА ";UG
250 VG=-1E+06:FOR I=1 TO IK
265 GI=V(I,1):FOR J=1 TO IR-1
270 IF GI>V(I,J+1) THEN GI=V(I,J+1)
280 NEXT J
285 GA=V(I,1):FOR J=1 TO IR-1
290 IF GA<V(I,J+1) THEN GA=V(I,J+1)
295 NEXT J
300 EP(I)=UG*GA+(1-UG)*GI
305 IF VG<EP(I) THEN VG=EP(I):IW=I
310 NEXT I
315 PRINT" ГУРВИЦА(К=" USING"#.##";UG;:PRINT")";: PRINT TAB(25)USING"#####.###";X(IW),VG
320 PL=-1E+06:FOR I=1 TO IK:EP(I)=0
325 FOR J=1 TO IR:EP(I)=EP(I)+V(I,J)/IR:NEXT J
330 IF PL<EP(I) THEN PL=EP(I):IW=I
335 NEXT I
340 PRINT" ЛАПЛАСА ";:PRINT TAB(25)USING"#####.###"; X(IW),PL
352 FOR J=1 TO IR:ER=-1E+06
353 FOR I=1 TO IK:IF ER<V(I,J) THEN ER=V(I,J)
354 NEXT I
355 FOR I=1 TO IK:VS(I,J)=V(I,J)-ER:NEXT I
356 NEXT J
360 VW=-1E+06:FOR I=1 TO IK
365 EP(I)=VS(I,1):FOR J=1 TO IR-1
370 IF EP(I)>VS(I,J+1) THEN EP(I)=VS(I,J+1)
375 NEXT J
380 IF VW<EP(I) THEN VW=EP(I):IW=I
385 NEXT I
390 PRINT" СЭВИДЖА ";:PRINT tab(25)USING"#####.###";X(IW),VW
400 END
```

ПРИЛОЖЕНИЕ 2
КОМПЬЮТЕРНАЯ ПРОГРАММА ИССЛЕДОВАНИЯ РАСПРЕДЕЛЕНИЯ СЛУЧАЙНЫХ ВЕЛИЧИН

```
COLOR 7,0:CLS:LOCATE 2,60,0:COLOR 25,5,15:PRINT"ВКЛЮЧИТЕ ПРИНТЕР"
2 COLOR 7,7,15:LOCATE 4:COLOR 0,7,15
3 PRINT TAB(27)" ЗАДАЧА      R N D";:PRINT SPC(37):PRINT TAB(26)" ";
4 COLOR 15,0,15:PRINT"КАФЕРА";:COLOR 23,5,15:PRINT"*БНТУ*";
5 COLOR 15,0,15:PRINT" ОАПДД ";:COLOR 7,7,15:PRINT SPC(35):COLOR 1,14
6 PRINT TAB(20)"ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕНИЯ СЛУЧАЙНЫХ";
7 PRINT TAB(80)" ";:PRINT TAB(33)"ВЕЛИЧИН";
8 PRINT TAB(80)" ":DEFINT I-N
19 PRINT:COLOR 7,0,7
35 COLOR 1,6,14
  IMAX=3
  LOCATE 5,33,0:COLOR 7,5,15:PRINT"*БНТУ*";
  I=1
  LOCATE 8,30,0:COLOR 25,4:PRINT"ВВЕДИТЕ РЕЖИМ"
50    J=1
     IF J=I THEN COLOR 4,11 ELSE COLOR 1,11
     LOCATE 9,25,0:PRINT " ╔══════════════════════════════╗ "
     LOCATE 10,25,0:PRINT"║ Д И С К Р Е Т Н О Е          ║"
     LOCATE 11,25,0:PRINT"║   *РАСПРЕДЕЛЕНИЕ*            ║"
     LOCATE 12,25,0:PRINT"╚══════════════════════════════╝ "
     J=2
      IF J=I THEN COLOR 4,11 ELSE COLOR 1,11
      LOCATE 13,25,0:PRINT"╔══════════════════════════════╗ "
      LOCATE 14,25,0:PRINT"║ Н Е П Р Е Р Ы В Н О Е        ║"
      LOCATE 15,25,0:PRINT"║   *РАСПРЕДЕЛЕНИЕ*            ║"
      LOCATE 16,25,0:PRINT"╚══════════════════════════════╝ "
       J=3
       IF J=I THEN COLOR 4,11 ELSE COLOR 1,11
       LOCATE 21,25,0:PRINT"╔══════════════════════════════╗ "
       LOCATE 22,25,0:PRINT"║    **  ВЫХОД   **            ║"
       LOCATE 23,25,0:PRINT"╚══════════════════════════════╝ "
95 AM$=INKEY$
  IF LEN(AM$)=0 THEN 95
  AM1$=RIGHT$(AM$,1)
  IF ASC(AM1$)=80 THEN I=I+1:GOTO 100
  IF ASC(AM1$)=72 THEN I=I-1:GOTO 100
  IF ASC(AM1$)=13 GOTO 110
 GOTO 95
100 IF I>IMAX THEN I=1
   IF I<1 THEN I=IMAX
 GOTO 50
110 COLOR 1,11
   IF I=1 THEN SHELL "RNDD.EXE":GOTO 1
   IF I=2 THEN SHELL"RNDN.EXE":GOTO 1
120 END
```

```
REM МОДУЛЬ ИССЛЕДОВАНИЯ НЕПРЕРЫВНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН
2 COLOR 7,7,15:CLS:LOCATE 10:PRINT:COLOR 0,7,15
3 PRINT TAB(27)"З А Д А Ч А   R  N  D  N":PRINT TAB(26)" ";
4 COLOR 23,0,15:PRINT"КАФЕДРА";:COLOR 23,5,15:PRINT" *БНТУ*";
5 COLOR 15,0,15:PRINT" О А П Д Д":PRINT:COLOR 1,14
6 PRINT TAB(21)"ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕНИЯ НЕПРЕРЫВНОЙ";
7 PRINT TAB(80)" ";:PRINT TAB(27)"СЛУЧАЙНОЙ ВЕЛИЧИНЫ";
8 PRINT TAB(80)" ":DEFINT I-N
9 COLOR 7,10,15
12 PRINT:COLOR 7,0,7
20 DIM A(1500),AA(1500)
22 OPEN"O",#3,"RNDNR.REZ"
25 COLOR 0,7
30 PRINT "ВВЕДЕНЫ ЛИ ИСХОДНЫЕ ДАННЫЕ"
35 COLOR 15,0
40 PRINT "ДА- 1, НЕТ- 0":INPUT IR1
50 IF IR1=0 THEN 200
55 INPUT"СПЕЦИФИКАЦИЯ ФАЙЛА С ИСХОДНЫМИ ДАННЫМИ";IMF$
60 OPEN"I",#1,IMF$
```

```
65 INPUT #1,SI$
70 INPUT #1,NO
80 FOR I=1 TO NO:INPUT #1,A(I):NEXT I
90 CLOSE #1
100 CLS:PRINT" № П/П    ЧИСЛО"
105 FOR I=1 TO NO:PRINT TAB(((I-1) MOD 4)*16+1)" | "I;USING"#####.####";A(I);
107 IF I MOD 4=0 THEN PRINT
110 NEXT I
115 COLOR 23,0
120 PRINT "ТРЕБУЕТСЯ ЛИ КОРРЕКТИРОВАТЬ ИСХОДНЫЕ ЗНАЧЕНИЯ"
130 PRINT "СЛУЧАЙНОЙ ВЕЛИЧИНЫ. ДА- 1,НЕТ- 0":INPUT IR0
135 COLOR 15,0
140 IF IR1=0 AND IR0=0 THEN 230
150 IF IR0=0 THEN 270
160 PRINT "ПОРЯДКОВЫЙ НОМЕР(СЧИТАЯ С ЕДИНИЦЫ) ЧИСЛА,ПОДЛЕЖАЩЕГО"
170 PRINT "ИЗМЕНЕНИЮ":INPUT IS
175 PRINT "ЗАМЕНЯТЬ ЧИСЛО "A(IS)" ?":INPUT IRR:IF IRR=0 THEN 120
180 IF IS<1 OR IS>NO THEN PRINT "ВЫ ДОПУСТИЛИ ОШИБКУ":GOTO 160
190 PRINT "НОВОЕ ЧИСЛО":INPUT A(IS):IR1=0:GOTO 120
200 CLS:PRINT "ВВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ":INPUT "НАИМЕНОВАНИЕ ДАННЫХ";SI$
205 INPUT "РАЗМЕР ВЫБОРКИ";NO
210 PRINT "ПОСЛЕДОВАТЕЛЬНО ЧИСЛА ВЫБОРКИ"
220 FOR I=1 TO NO:PRINT I"-E":INPUT A(I):NEXT I:GOTO 100
230 IF IR1=1 THEN 232
231 INPUT"ЗАДАЙТЕ ИМЯ ФАЙЛА ДЛЯ ИСХОДНЫХ ДАННЫХ";IMF$
232 OPEN "O",#1,IMF$
235 WRITE #1,SI$
240 WRITE #1,NO
250 FOR I=1 TO NO:WRITE #1,A(I):NEXT I
260 CLOSE #1
270 DIM UB(3),FR(50),ST(50),X(50),P(50),PT(50),P1(50),IN(7)
280 DIM P9(50),XX(50),A1(500),PM(7),PS(7),Z$(7),Z1$(7),Z2$(7)
285 PRINT #3, TAB(10)"ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕНИЯ СЛУЧАЙНОЙ";
286 PRINT #3," ВЕЛИЧИНЫ":PRINT #3, TAB(10)"("SI$")"
290 Z$(1)="НОРМАЛЬНОЕ":Z1$(1)="M":Z2$(1)="SIGMA"
300 Z$(2)="ЛОГНОРМАЛЬНОЕ":Z1$(2)="LNU":Z2$(2)="SIGMA^2"
310 Z$(3)="РЕЛЕЯ":Z1$(3)="SIGMA":Z2$(3)=""
320 Z$(4)="ЭКСПОНЕНЦИАЛЬНОЕ":Z1$(4)="LAMBDA":Z2$(4)=""
330 Z$(5)="ЭРЛАНГА":Z1$(5)="LAMBDA":Z2$(5)="K"
340 Z$(6)="ВЕЙБУЛЛА":Z1$(6)="B":Z2$(6)="LAMBDA"
342 Z$(7)="РАВНОМЕРНОЕ":Z1$(7)="A":Z2$(7)="B"
345 Z7$="   РАСПРЕДЕЛЕНИЕ НЕ ПОДХОДИТ"
346 Z8$=" РАСПРЕДЕЛЕНИЕ "
347 F1$="-#.##^^^^":F2$="-#.##^^^^"
350 PRINT #3, :PRINT #3,  TAB(5)"ИСХОДНЫЕ ДАННЫЕ"
355 FOR I=1 TO NO:AA(I)=A(I):NEXT I:NF=NO:GOSUB 360:GOTO 410
360 N4=0
370 FOR I=1 TO NF
375 N4=N4+1:IF N4>8 THEN N4=1:PRINT #3,
380 PRINT #3,USING F1$;AA(I);
390 NEXT I
405 PRINT #3,:PRINT #3,:RETURN
410 ID=1
420 ID=2*ID
430 IF ID<=NO THEN 420
440 ID=INT((ID-1)/2!)
450 IF ID=0 THEN 580
460 FOR I1=1 TO NO-ID:J=I1
470 L=J+ID
480 IF A(L)>=A(J) THEN 510
490 XR=A(J):A(J)=A(L):A(L)=XR:J=J-ID
500 IF J>0 THEN 470
510 NEXT I1
520 GOTO 440
580 PRINT #3, "    НАИМЕНЬШЕЕ ВЫБОРОЧНОЕ ЗНАЧЕНИЕ  =";
585 PRINT #3, USING F1$;A(1)
590 PRINT #3, "    НАИБОЛЬШЕЕ ВЫБОРОЧНОЕ ЗНАЧЕНИЕ  =";
592 PRINT #3, USING F1$;A(NO)
```

```
595 A(1)=A(1)+1.E-36:A(NO)=A(NO)+1.E-36
596 UB(1)=A(1):UB(3)=A(NO)
597 FOR I=2 TO NO:IF A(I)=0 THEN A(I)=1.E-36:NEXT I
600 N8=CINT(1.44*LOG(NO)):N9=0
610 IF N8<4 THEN N8=4
620 PRINT "   *РЕКОМЕНДУЕМОЕ ЧИСЛО ИНТЕРВАЛОВ  "N8
630 GOTO 700
640 IF IR9=1 THEN 1090
650 PRINT"ТРЕБУЕТСЯ ПРОДОЛЖИТЬ РАСЧЕТЫ С ЭТИМИ ДАННЫМИ";
651 INPUT"(ДА- 1,НЕТ- 0)";N5
660 IF N5=0 THEN 3070
670 PRINT"*ТРЕБУЕТСЯ ИЗМЕНЯТЬ ЧИСЛО ИНТЕРВАЛОВ ИЛИ"
680 INPUT "*СМЕЩЕНИЕ(ДА- 1,НЕТ- 0)";N7
690 IF N7=0 THEN 1160
700 INPUT " * ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ";N
710 UB(2)=N+2
720 INPUT " *ВВЕДИТЕ ЗНАЧЕНИЕ СМЕЩЕНИЯ";XM
730 GOSUB 2070
740 IF N9<>0 THEN 800
750 N9=1:V=ST(3)/ST(2)
760 PRINT #3, "    МАТЕМАТИЧЕСКОЕ ОЖИДАНИЕ        =";
765 PRINT #3, USING F1$;ST(2)
770 PRINT #3, "    СРЕДНЕКВАДРАТИЧЕСКОЕ ОТКЛОНЕНИЕ =";
775 PRINT #3, USING F1$;ST(3)
780 PRINT #3, "    КОЭФФИЦИЕНТ ВАРИАЦИИ          =";
785 PRINT #3, USING F1$;V
790 PRINT #3, "     -------*****--------           "
800 PRINT #3, "     ЧИСЛО ИНТЕРВАЛОВ    ="N
810 PRINT #3, "     ЗНАЧЕНИЕ СМЕЩЕНИЯ   =";
815 PRINT #3, USING F1$;XM
820 PT(1)=PT(1)*100
840 FOR I=1 TO N
850 P9(I+1)=PT(I+1):PT(I+1)=P9(I+1)*100:NEXT I
860 R=(UB(3)-UB(1))/(UB(2)-2)
870 FOR I=1 TO N:XI=I:X(I)=UB(1)+R*XI-R/2:NEXT I
880 PRINT #3, :PRINT #3, "       СЕРЕДИНЫ ИНТЕРВАЛОВ"
890 FOR I=1 TO N:AA(I)=X(I):NEXT I:NF=N:GOSUB 360
920 PRINT #3, "     ВЕКТОР ЭМПИРИЧЕСКИХ ЧАСТОТ "
930 FOR I=1 TO N:AA(I)=FR(I+1):NEXT I:NF=N:GOSUB 360
950 PRINT #3, "     ВЕКТОР ЭМПИРИЧЕСКИХ ЧАСТОСТЕЙ"
960 FOR I=1 TO N:AA(I)=P9(I+1):NEXT I:NF=N:GOSUB 360
980 PRINT #3, TAB(15)"** X **":PRINT #3,
990 PRINT "ТРЕБУЕТСЯ ЛИ ПОИСК ЗАКОНА РАСПРЕДЕЛЕНИЯ ПО МАКСИМУМУ"
1000 INPUT "ЗНАЧЕНИЯ КРИТЕРИЯ РОМАНОВСКОГО(ДА- 1,НЕТ- 0)";IR9
1010 IF IR9=0 THEN 1160
1020 PRINT#3," ┌──────────────┬───────────────────┬──────────────┬──────────────┐"
1025 PRINT#3," │              │                   │              │              │"
1030 PRINT#3," │ РАСПРЕ-      │  ПАРАМЕТРЫ        │ЧИСЛО         │ КРИТЕРИЙ     │"
1040 PRINT#3," │ ДЕЛЕНИЕ      │                   │ СТЕПЕНЕЙ     │  РОМАНОВ-    │"
1050 PRINT#3," │              │                   │ СВОБОДЫ      │  СКОГО       │"
1060 PRINT#3," └──────────────┴───────────────────┴──────────────┴──────────────┘"
1065 PRINT #3,
1070 MI=0:RM=1E+10
1080 MI=MI+1:ON MI GOTO 1290,1500,1590,1670,1750,1870,2061
1090 IF RK<RM THEN RM=RK:JM=MI
1100 PRINT #3, Z$(MI);TAB(18)Z1$(MI);"=";:PRINT #3, USING F1$;PM(MI);
1110 IF IN(MI)=1 THEN PRINT #3, " "Z2$(MI);"=";:PRINT #3, USING F1$;PS(MI);
1120 PRINT #3, TAB(50)NU;:PRINT #3, TAB(58)USING F1$;RK
1130 IF MI<7 THEN 1080
1140 N3=JM:IR9=0
1150 PRINT #3, :PRINT #3, :PRINT #3, "НАИЛУЧШЕЕ ПО КРИТЕРИЮ РОМАНОВСКОГО - ";:GOTO 1250
1160 PRINT "  *  ВВЕДИТЕ НОМЕР РАСПРЕДЕЛЕНИЯ;"
1170 PRINT
1180 PRINT "      1 - "Z$(1)
1190 PRINT "      2 - "Z$(2)
1200 PRINT "      3 - "Z$(3)
1210 PRINT "      4 - "Z$(4)
1220 PRINT "      5 - "Z$(5)
```

```
1230 PRINT "       6 - "Z$(6)
1235 PRINT "       7 - "Z$(7)
1240 INPUT N3
1250 ON N3 GOTO 1260,1500,1590,1670,1750,1870,2061
1260 PRINT #3, :PRINT #3, TAB(10)Z$(1);Z8$
1270 PRINT #3, :PRINT #3, TAB(5)Z1$(1)"=";:PRINT #3, USING F1$;ST(2)
1280 PRINT #3, TAB(5)Z2$(1)"=";:PRINT #3, USING F1$;ST(3)
1290 IN(1)=1:PM(1)=ST(2):PS(1)=ST(3):XW=(XX(1)-ST(2))/ST(3):GOSUB 2820
1300 FOR I=1 TO N:FO=FW
1310 XW=(XX(I+1)-ST(2))/ST(3):GOSUB 2820:P(I)=FW-FO:NEXT I
1320 NU=N-3
1330 IF IR9=1 THEN 1480
1340 PRINT #3, :PRINT #3, "  ВЕКТОР ТЕОРЕТИЧЕСКИХ ЧАСТОСТЕЙ"
1350 FOR I=1 TO N:AA(I)=P(I):NEXT I:NF=N:GOSUB 360
1360 INPUT"ТРЕБУЮТСЯ ЛИ ГРАФИКИ РАСПРЕДЕЛЕНИЯ(ДА- 1,НЕТ- 0)";IR2
1370 IF IR2=0 THEN 1480
1380 PRINT #3,
1382 PRINT #3," ┌──────────────────────────────────────────────────────────────────────────>";
1385 PRINT #3,
1390 FOR I=1 TO N:P4=INT(P(I)*80)+1:P5=INT(PT(I+1)*80/100)+1
1400 IF P4<P5 THEN 1430
1410 IF P4>P5 THEN 1440
1411 PRINT #3,"│";
1412 FOR II=2 TO P4:PRINT #3,"░";:NEXT II:PRINT #3,
1413 PRINT #3,"│";
1414 FOR II=2 TO P4-1:PRINT #3,"░";:NEXT II
1420 PRINT #3,TAB(P4)"*"
1421 PRINT #3,"│";
1422 FOR II=2 TO P4:PRINT #3,"░";:NEXT II:PRINT #3,
1428 GOTO 1450
1430  PRINT #3,"│";
1431 FOR II=2 TO P5:PRINT #3,"░";:NEXT II:PRINT #3,
1432 IF P4<2 THEN 1434
1433 PRINT #3,"│";:FOR II=2 TO P4-1:PRINT #3,"░";:NEXT II
1434 PRINT #3,TAB(P4)"*";
1435 FOR II=P4+1 TO P5:PRINT #3,"░";:NEXT II
1437 PRINT #3,:PRINT #3,"│";
1438 FOR II=2 TO P5:PRINT #3,"░";:NEXT II:PRINT #3,
1439 GOTO 1450
1440  PRINT #3,"│";
1441 FOR II=2 TO P5:PRINT #3,"░";:NEXT II:PRINT #3,
1442 PRINT #3,"│";
1443 FOR II=2 TO P5:PRINT #3,"░";:NEXT II
1445 PRINT #3,TAB(P4)"*"
1446 PRINT #3,"│";
1447 FOR II=2 TO P5:PRINT #3,"░";:NEXT II:PRINT #3,
1450 NEXT I
1455 PRINT #3,"│"
1456 PRINT #3,"V"
1460 PRINT #3,"* -ТЕОРЕТИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ"
1480 GOSUB 2250
1490 GOTO 640
1500 IN(2)=1:SI=LOG((ST(3)/(ST(2)-XM))^2+1):SL=SQR(SI)
1510 YL=LOG(ST(2)-XM)-SI/2:IF IR9=1 THEN 1550
1520 PRINT #3,:PRINT #3, TAB(10)Z$(2);Z8$
1530 PRINT #3,:PRINT #3, TAB(5)Z1$(2)"=";USING F1$;YL
1540 PRINT #3, TAB(5)Z2$(2)"=";:PRINT #3, USING F1$;SI
1550 PM(2)=YL:PS(2)=SI:XW=(LOG(XX(1)-XM)-YL)/SL:GOSUB 2820
1560 FOR I=1 TO N:FO=FW
1570 XW=(LOG(XX(I+1)-XM)-YL)/SL:GOSUB 2820:P(I)=FW-FO:NEXT I
1580 GOTO 1320
1590 IN(3)=0:SG=(ST(2)-XM)/1.2533:IF IR9=1 THEN 1620
1600 PRINT #3,:PRINT #3, TAB(10)Z8$;Z$(3)
1610 PRINT #3,:PRINT #3, TAB(5)Z1$(3)"=";:PRINT #3, USING F1$;SG
1620 PM(3)=SG:XW=(XX(1)-XM):GOSUB 2890
1630 FOR I=1 TO N:FO=FW
1640 XW=(XX(I+1)-XM):GOSUB 2890:P(I)=FW-FO:NEXT I
1650 NU=N-2
```

```
1660 GOTO 1330
1670 PL=1/(ST(2)-XM):IN(4)=0:IX=1:PM(4)=PL:IF IR9=1 THEN 1700
1680 PRINT #3,:PRINT #3, TAB(10) Z$(4);Z8$
1690 PRINT #3,:PRINT #3, TAB(5)Z1$(4)"=";:PRINT #3, USING F1$;PL
1700 XW=XX(1)-XM:GOSUB 2920
1710 FOR I=1 TO N:FO=FW
1720 XW=XX(I+1)-XM:GOSUB 2920:P(I)=FW-FO:NEXT I
1730 IF IX=2 THEN 1320
1740 NU=N-2:GOTO 1330
1750 IF((ST(2)-XM)/ST(3))^2<85 THEN 1755
1752 PRINT #3, Z$(5),Z7$:IF IR9=0 THEN 1160 ELSE 1080
1755 K=CINT(((ST(2)-XM)/ST(3))^2)
1760 IF K=0 THEN K=1
1770 EL=K/(ST(2)-XM):IN(5)=1:PM(5)=EL:PS(5)=K:IF IR9=1 THEN 1810
1780 PRINT #3,:PRINT #3, TAB(10)Z8$;Z$(5)
1790 PRINT #3,:PRINT #3, TAB(5)"LAMBDA=";:PRINT #3, USING F1$;EL
1800 PRINT #3, TAB(5)"K="K
1810 KK=K-1
1820 IF KK=O THEN IX=2:PL=EL:GOTO 1700
1830 XW=XX(1)-XM:GOSUB 2950
1840 FOR I=1 TO N:FO=FW
1850 XW=XX(I+1)-XM:GOSUB 2950:P(I)=FW-FO:NEXT I
1860 GOTO 1320
1870 B=.3:E=.001:H=.4:GOSUB 1930
1875 IF LOG(((ST(2)-XM)/G3)^B)<85 THEN 1880
1877 PRINT #3, Z$(6),Z7$:IF IR9=0 THEN 1160 ELSE 1140
1880 F1=F:B=B+H:GOSUB 1930
1890 IF ABS(F)<ABS(F1) THEN 1875
1900 H=-H/4:IF ABS(H)>E/4 THEN 1875
1910 B=B+H/4:Z=1/B+1:GOSUB 1970
1920 WL=((ST(2)-XM)/G3)^B
1926 IF IR9=0 THEN 2000 ELSE 2030
1930 Z=2/B+1:GOSUB 1970
1940 G1=G3:Z=1/B+1:GOSUB 1970
1950 G2=G3:F=G1/G2/G2-1-(ST(3)/(ST(2)-XM))^2
1960 RETURN
1970 G3=EXP(-Z)*Z^Z*(1+1/12/Z+1/(288*Z^2)-.7/(288*Z^3))
1980 G3=SQR(6.28319/Z)*G3
1990 RETURN
2000 PRINT #3,:PRINT #3, TAB(10)Z8$;Z$(6)
2010 PRINT #3, :PRINT #3, "        B=";:PRINT #3, USING F1$;B
2020 PRINT #3, "     LAMBDA=";:PRINT #3, USING F1$;WL
2030 IN(6)=1:PM(6)=B:PS(6)=WL:XW=XX(1)-XM:GOSUB 3010
2040 FOR I=1 TO N:FO=FW
2050 XW=XX(I+1)-XM:GOSUB 3010:P(I)=FW-FO:NEXT I
2060 GOTO 1320
2061 IN(7)=1:A=ST(2)-SQR(3)*(ST(3)):B=ST(2)+SQR(3)*(ST(3)):IF IR9=1 THEN 2065
2062 PRINT #3,:PRINT #3, TAB(10)Z$(7);Z8$
2063 PRINT #3,:PRINT #3, TAB(5)Z1$(7)"=";:PRINT #3, USING F1$;A
2064 PRINT #3, TAB(5)Z2$(7)"=";:PRINT #3, USING F1$;B
2065 PM(7)=A:PS(7)=B
2066 FOR I=1 TO N
2067 P(I)=1/N:NEXT I
2069 GOTO 1320
2070 N=UB(2)-2:R=(UB(3)-UB(1))/(UB(2)-2)
2080 FOR I=1 TO N:XI=I:XX(I)=UB(1)+R*(XI-1)
2090 FR(I+1)=0!:NEXT I
2100 XX(N+1)=UB(3)
2110 II=1:J=0:A(NO+1)=1E+30:VN=NO
2120 J=J+1
2130 I=II
2140 IF A(I)>XX(J+1) THEN 2170
2150 FR(J+1)=FR(J+1)+1
2160 I=I+1:GOTO 2140
2170 II=I:IF II>NO THEN 2180 ELSE 2120
2180 FOR I=1 TO N:PT(I+1)=FR(I+1)/VN:NEXT I
2190 SS=0:ST(2)=0:I=1
2200 IF A(I)<XX(1) OR A(I)>XX(N+1) THEN 2220
```

```
2210 ST(2)=ST(2)+A(I)/VN:SS=SS+A(I)^2/VN
2220 I=I+1:IF I<= NO THEN 2200
2230 ST(3)=SQR(VN/(VN-1)*(SS-ST(2)^2))
2240 RETURN
2250 FOR I=1 TO N:P1(I)=P(I):P9(I+1)=PT(I+1):NEXT I
2260 N1=N:N11=NU:NN=N-N11+1:SM=P1(1)*NO:SQ=0
2270 I=2:L1=0
2280 IF N-L1<=NN THEN 2330
2290 IF SM>=5 THEN 2330
2300 SM=SM+P1(I)*NO
2310 I=I+1:L1=L1+1
2320 GOTO 2280
2330 SM=P1(N)*NO
2340 I=0:L2=0
2350 IF N-L1-L2<=NN GOTO 2380
2360 IF SM>=5 GOTO 2380
2370 L2=L2+1:I=I+1:SM=SM+P1(N-I)*NO:GOTO 2350
2380 IF L1=0 THEN 2440
2390 FOR I=1 TO L1:PT(2)=PT(2)+PT(I+2)
2400 P(1)=P(1)+P(I+1):NEXT I
2410 N=N-L1
2420 FOR I=2 TO N:PT(I+1)=PT(I+1+L1)
2430 P(I)=P(I+L1):NEXT I
2440 IF L2=0 THEN 2480
2445 N=N-L2
2450 FOR I=1 TO L2:PT(N+1)=PT(N+1)+PT(N+1+I)
2460 P(N)=P(N)+P(N+I):NEXT I
2480 FOR I=1 TO N
2485 IF P(I)<1E-33 THEN SQ=9.999999E+33:GOTO 2495
2490 SQ=SQ+((P(I)-PT(I+1)/100)^2)/P(I)
2495 NEXT I
2500 SQ=SQ*NO
2510 NU=N11-L2-L1
2520 RK=(SQ-NU)/SQR(2*NU)
2530 IF IR9=1 THEN 2600
2540 PRINT #3, :PRINT #3, "    ВЕКТОР ТЕОРЕТИЧЕСКИХ ЧАСТОСТЕЙ С"
2550 PRINT #3, "    ОБЬЕДИНЕНИЕМ ИНТЕРВАЛОВ НА КОНЦАХ"
2560 FOR I=1 TO N:AA(I)=P(I):NEXT I:NF=N:GOSUB 360
2570 PRINT #3, :PRINT #3, "     ЗНАЧЕНИЕ ХИ-КВАДРАТ      =";:
2575 PRINT #3, USING F1$;SQ
2580 PRINT #3, "     ЧИСЛО СТЕПЕНЕЙ СВОБОДЫ         ="NU
2590 PRINT #3, "     ЗНАЧЕНИЕ КРИТЕРИЯ РОМАНОВСКОГО =";
2595 PRINT #3, USING F2$;RK
2600 N=N1
2610 FOR I=1 TO N:PT(I+1)=P9(I+1)
2620 P(I)=P1(I):NEXT I
2630 IF IR9=1 THEN 3060
2640 INPUT "ТРЕБУЕТСЯ ЛИ РАСЧЕТ КРИТЕРИЯ МИЗЕСА(ДА- 1,НЕТ- 0)";IR3
2650 IF IR3=0 THEN 3060
2660 SW=0:I2=0:A1(0)=1E-38
2670 I2=I2+1
2680 IF I2>NO THEN 3040
2690 ON N3 GOSUB 2740,2760,2780,2780,2780,2780
2700 IF A1(I2)=A1(I2-1) THEN GOSUB 2800:GOTO 2670
2710 XW=A1(I2)
2720 ON N3 GOSUB 2820,2820,2890,2920,2950,3010
2730 GOSUB 2800:GOTO 2670
2740 FOR I=1 TO NO:A1(I)=(A(I)-ST(2))/ST(3):NEXT I
2750 RETURN
2760 FOR I=1 TO NO:A1(I)=(LOG(A(I)-XM)-YL)/SL:NEXT I
2770 RETURN
2780 FOR I=1 TO NO:A1(I)=A(I)-XM:NEXT I
2790 RETURN
2800 IF FW=>1 THEN FW=.999999
2805 IF FW<=0 THEN FW=1E-36
2808 SW=SW+(2*I2-1)*LOG(FW)+(2*NO-2*I2+1)*LOG(1-FW)
2810 RETURN
2820 C1=.049867347#:C2=.021141006#:C3=3.277626E-03
```

```
2830 C4=3.8004E-05:C5=4.8891E-05:C6=5.383E-06
2840 IF XW>0 THEN 2860
2850 XW=ABS(XW):GOSUB 2860:FW=1!-FW:GOTO 2880
2860 FW=1+C1*XW+C2*XW^2+C3*XW^3+C4*XW^4+C5*XW^5+C6*XW^6
2870 IF FW>230. THEN FW=1. ELSE FW=1!-FW^(-16!)/2.
2880 RETURN
2890 IF ((XW/SG)^2/2)>85 THEN FW=1!:GOTO 2910
2900 FW=1!-EXP(-((XW/SG)^2/2)):GOTO 2910
2910 RETURN
2920 IF PL*XW>85! THEN FW=1!:GOTO 2940
2930 FW=1!-EXP(-PL*XW):GOTO 2940
2940 RETURN
2950 IF EL*XW>80! THEN FW=1!:GOTO 3000
2960 VL=1!:VE=1!
2970 FOR M=1 TO KK
2980 VL=VL*EL*XW/M:VE=VE+VL:NEXT M
2990 FW=1!-EXP(-EL*XW)*VE
3000 RETURN
3010 IF XM^B/WL>85! THEN FW=1!:GOTO 3030
3020 FW=1!-EXP(-XW^B/WL):GOTO 3030
3030 RETURN
3040 W2=-NO-1/NO*SW
3050 PRINT #3, "     ЗНАЧЕНИЕ КРИТЕРИЯ МИЗЕСА       =";
3055 PRINT #3, USING F1$;W2
3060 RETURN
3070 INPUT"ТРЕБУЕТСЯ ПРОДОЛЖИТЬ РАСЧЕТЫ С ДРУГИМИ ДАННЫМИ";IR1
3080 IF IR1<>1 THEN 3095
3085 ERASE A,AA,UB,FR,ST,X,P,PT,P1,IN,P9,XX,A1,PM,PS,Z$,Z1$,Z2$
3095 COLOR 25,5,15:PRINT "   РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ RNDN.REZ"
3096 PRINT "   ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПОВТОРНОГО РАСЧЕТА В ФАЙЛЕ "IMF$
3097 BBB$=INKEY$
3098 IF LEN(BBB$)=0 THEN 3097
3099 RUN"RND.EXE"
3100 END

REM МОДУЛЬ ИССЛЕДОВАНИЯ ДИКРЕТНЫХ СЛУЧАЙНЫХ ВЕЛИЧИН
2 COLOR 7,7,15:CLS:LOCATE 10:PRINT:COLOR 0,7,15
3 PRINT TAB(27)"З А Д А Ч А   R  N  D  D":PRINT TAB(26)" ";
4 COLOR 15,0,15:PRINT"КАФЕДРА";:COLOR 23,5,15:PRINT" *БНТУ*";
5 COLOR 15,0,15:PRINT" О А П Д Д":PRINT:COLOR 1,14
6 PRINT TAB(21)"ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕНИЯ ДИСКРЕТНОЙ";
7 PRINT TAB(80)" ";:PRINT TAB(27)"СЛУЧАЙНОЙ ВЕЛИЧИНЫ";
8 PRINT TAB(80)" ":DEFINT I-N
9 COLOR 7,10,15: PRINT:COLOR 7,0,7
20 DIM JA(100),NA(100)
25 COLOR 0,7:OPEN"O",#3,"RNDDR.REZ"
30 PRINT "ВВЕДЕНЫ ЛИ ИСХОДНЫЕ ДАННЫЕ"
35 COLOR 15,0
40 PRINT "ДА- 1, НЕТ- 0":INPUT IR1
50 IF IR1=0 THEN 200
55 INPUT"СПЕЦИФИКАЦИЯ ФАЙЛА С ИСХОДНЫМИ ДАННЫМИ  ",IMF$
60 OPEN"I",#1,IMF$
65 INPUT #1,SI$
70 INPUT #1,NO
80 FOR I=1 TO NO:INPUT #1,JA(I),NA(I):NEXT I
90 CLOSE #1
100 CLS:PRINT"ПОРЯДКОВ. N  ВЕЛИЧИНА  ЧИСЛО ЗНАЧЕНИЙ"
110 FOR I=1 TO NO:PRINT  I,JA(I),NA(I):NEXT I
115 COLOR 23,0
120 PRINT "ТРЕБУЕТСЯ ЛИ КОРРЕКТИРОВАТЬ ИСХОДНЫЕ ДАННЫЕ"
130 PRINT "  ДА- 1,НЕТ- 0":INPUT IR0
135 COLOR 15,0
140 IF IR1=0 AND IR0=0 THEN 230
150 IF IR0=0 THEN 270
160 PRINT "ПОРЯДКОВЫЙ НОМЕР(СЧИТАЯ С ЕДИНИЦЫ) ЧИСЕЛ,ПОДЛЕЖАЩИХ"
170 PRINT "ИЗМЕНЕНИЮ":INPUT IS
175 PRINT "ЗАМЕНЯТЬ ЧИСЛА "JA(IS)," ",NA(IS):INPUT IRR:IF IRR=0 THEN 120
180 IF IS<1 OR IS>NO THEN PRINT "ВЫ ДОПУСТИЛИ ОШИБКУ":BEEP:GOTO 160
190 PRINT "НОВОЕ ЧИСЛО":INPUT JA(IS),NA(IS):IR1=0:GOTO 120
```

```
200 CLS: PRINT "ВВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ":INPUT "НАИМЕНОВАНИЕ ДАННЫХ";SI$
205 INPUT "ЧИСЛО ГРУПП ЗНАЧЕНИЙ СЛУЧАЙНОЙ ВЕЛИЧИНЫ";NO
210 PRINT "ПОСЛЕДОВАТЕЛЬНО ЧЕРЕЗ ЗАПЯТУЮ ЗНАЧЕНИЕ СЛУЧАЙНОЙ ВЕЛИЧИНЫ"
215 PRINT "И ЧИСЛО НАБЛЮДЕНИЙ"
220 FOR I=1 TO NO:PRINT I"-Я ПАРА":INPUT JA(I),NA(I):NEXT I:GOTO 100
230 IF IR1=1 THEN 232
231 INPUT"ЗАДАЙТЕ СПЕЦИФИКАЦИЮ ФАЙЛА ДЛЯ ИСХОДНЫХ ДАННЫХ";IMF$
232 OPEN "O",#1,IMF$
235 WRITE #1,SI$
240 WRITE #1,NO
250 FOR I=1 TO NO:WRITE #1,JA(I),NA(I):NEXT I
260 CLOSE #1
270 DIM IUB(3),ST(3),P(100),P8(100),P0(100),BJ(100)
280 DIM P9(100),JR(100),PM(4),PS(4),Z$(4),Z1$(4)
281 NNS=0:SNN=0.:NNA=0:FOR I=1 TO NO:NNS=NNS+JA(I)*NA(I)
283 NNA=NNA+NA(I):SNN=SNN+JA(I)^2*NA(I):NEXT I
285 PRINT #3,  TAB(10)"ИССЛЕДОВАНИЕ РАСПРЕДЕЛЕНИЯ ДИСКРЕТНОЙ СЛУЧАЙНОЙ ВЕЛИЧИНЫ"
286 PRINT #3,  TAB(10)"("SI$")"
290 Z$(1)="БИНОМИАЛЬНОЕ":Z1$(1)="PB"
300 Z$(2)="ПУАССОНА":Z1$(2)="AP"
310 Z$(3)="ГЕОМЕТРИЧЕСКОЕ(ФАРРИ) (0 ...N)":Z1$(3)="PG"
320 Z$(4)="ГЕОМЕТРИЧЕСКОЕ(ФАРРИ) (1 ...N)":Z1$(4)="PG"
345 Z7$="    РАСПРЕДЕЛЕНИЕ НЕ ПОДХОДИТ"
346 Z8$=" РАСПРЕДЕЛЕНИЕ "
347 F$=".###^^^^":F1$="-.###^^^^"
350 PRINT #3, :PRINT #3,  TAB(5)"ИСХОДНЫЕ ДАННЫЕ"
360 N4=INT(NO/8!):N5=NO-N4*8
370 FOR J=1 TO N4
380 FOR I=8*J-7 TO 8*J:PRINT #3, TAB(9*I-72*J+64) JA(I);:NEXT I
385 FOR I=8*J-7 TO 8*J:PRINT #3, TAB(9*I-72*J+64) NA(I);:NEXT I
390 PRINT #3,:NEXT J
400 FOR I=8*N4+1 TO NO:PRINT #3, TAB(9*I-72*N4-8) JA(I);:NEXT I
405 FOR I=8*N4+1 TO NO:PRINT #3, TAB(9*I-72*N4-8) NA(I);:NEXT I
406 PRINT #3,
410 ID=1
420 ID=2*ID
430 IF ID<=NO THEN 420
440 ID=INT((ID-1)/2!)
450 IF ID=0 THEN 530
460 FOR I1=1 TO NO-ID:J=I1
470 L=J+ID
480 IF JA(L)>=JA(J) THEN 510
490 IX=JA(J):IX1=NA(J):JA(J)=JA(L):NA(J)=NA(L):JA(L)=IX:NA(L)=IX1:J=J-ID
500 IF J>0 THEN 470
510 NEXT I1
520 GOTO 440
530 IUB(1)=JA(1):IUB(3)=JA(NO):N=IUB(3)-IUB(1)+1
535 JR(1)=NA(1)
540 II=0:FOR I=2 TO NO
542 IF JA(I-1)+1=JA(I) THEN JR(I+II)=NA(I):GOTO 560
545 JR(I+II)=0:II=II+1:JR(I+II)=NA(I)
560 NEXT I
570 FOR I=1 TO N:JA(I)=IUB(1)-1+I:NA(I)=JR(I):NEXT I
580 PRINT #3, "    НАИМЕНЬШЕЕ ВЫБОРОЧНОЕ ЗНАЧЕНИЕ =";
585 PRINT #3, JA(1)
590 PRINT #3, "    НАИБОЛЬШЕЕ ВЫБОРОЧНОЕ ЗНАЧЕНИЕ =";
592 PRINT #3, JA(NO)
630 GOTO 742
640 INPUT"ПРОДОЛЖИТЬ РАСЧЕТЫ С ЭТИМИ ДАННЫМИ(ДА- 1,НЕТ- 0)";N5
660 IF N5=0 THEN 3070 ELSE 980
742 ST(2)=NNS/NNA:ST(3)=SQR((SNN-NNA*ST(2)^2)/(NNA-1))
750 N9=1:V=ST(3)/ST(2)
760 PRINT #3, "     МАТЕМАТИЧЕСКОЕ ОЖИДАНИЕ        =";
765 PRINT #3, USING F$;ST(2)
770 PRINT #3, "     СРЕДНЕКВАДРАТИЧЕСКОЕ ОТКЛОНЕНИЕ =";
775 PRINT #3, USING F$;ST(3)
780 PRINT #3, "     КОЭФФИЦИЕНТ ВАРИАЦИИ           =";
785 PRINT #3, USING F$;V: PRINT #3, "     -------*****--------            "
```

```
880 PRINT #3, :PRINT #3, "       ЗНАЧЕНИЯ ПЕРЕМЕННОЙ"
885 NF=N:FOR I= 1 TO N:BJ(I)=JA(I):NEXT I:GOSUB 890:GOTO 950
890 N4=INT(NF/8):N5=NF-N4*8:FOR J=1 TO N4
900 FOR I=8*J-7 TO 8*J:PRINT #3, TAB(9*I-72*J+64)USING F$;BJ(I);:NEXT I
910 PRINT #3,:NEXT J
920 FOR I=8*N4+1 TO NF:PRINT #3, TAB(9*I-72*N4-8)USING F$;BJ(I);:NEXT I
930 PRINT #3,:PRINT #3,:RETURN
950 PRINT #3, "      ВЕКТОР ЭМПИРИЧЕСКИХ ЧАСТОСТЕЙ"
960 FOR I=1 TO N:P9(I)=NA(I)/NNA:BJ(I)=P9(I):NEXT I:NF=N:GOSUB 890
980 PRINT #3, TAB(15)"** X **"
1160 PRINT "  *  ВВЕДИТЕ НОМЕР РАСПРЕДЕЛЕНИЯ"
1170 PRINT: PRINT "       1 - "Z$(1): PRINT "       2 - "Z$(2): PRINT "       3 - "Z$(3): PRINT "       4 - "Z$(4)
1240 INPUT N3
1250 ON N3 GOTO 1260,1500,1590,1680
1260 NW=NNA'INPUT" ВВЕДИТЕ ЧИСЛО ИСПЫТАНИЙ (РАЗМЕР ВЫБОРКИ)";NW
1265 PRINT #3,:PRINT #3, TAB(10)Z$(1),Z8$
1270 PB=ST(2)/NW:PRINT #3, :PRINT #3, TAB(5)Z1$(1)"=";:PRINT #3, USING F$;PB
1300 FOR I=1 TO N
1310 K=JA(I):GOSUB 2820:P(I)=PT:NEXT I
1320 NU=N-2
1340 PRINT #3, :PRINT #3, "  ВЕКТОР ТЕОРЕТИЧЕСКИХ ЧАСТОСТЕЙ":PRINT #3,
1350 FOR I=1 TO N:BJ(I)=P(I):NEXT I:NF=N:GOSUB 890
1360 INPUT"ТРЕБУЮТСЯ ЛИ ГРАФИКИ РАСПРЕДЕЛЕНИЯ(ДА- 1,НЕТ- 0)";IR2
1370 IF IR2=0 THEN 1480
1380 PRINT #3,:PRINT #3,:FOR I=2 TO 50:PRINT #3, TAB(I)"-";:NEXT I:PRINT #3,
1390 FOR I=1 TO N:P4=INT(P(I)*72):P5=INT(P9(I)*72)
1400 IF P4<P5 THEN 1430
1410 IF P4>P5 THEN 1440
1420 PRINT #3,"I";TAB(P4)"$":GOTO 1450
1430 PRINT #3,"I";TAB(P4)"+";TAB(P5)"#":GOTO 1450
1440 PRINT #3,"I";TAB(P5)"#";TAB(P4)"+"
1450 PRINT #3,"I":NEXT I:PRINT #3,
1460 PRINT #3,"+ -ТЕОРЕТ.РАСПРЕДЕЛЕНИЕ, # -ЭМПИРИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ,";
1470 PRINT #3," $ -ПРИ СОВПАДЕНИИ"
1480 GOSUB 2250
1490 GOTO 640
1500 PRINT #3,:PRINT #3, TAB(10)Z8$,Z$(2)
1530 AP=ST(2):PRINT #3,:PRINT #3, TAB(5)Z1$(2)"=";USING F$;AP
1560 FOR I=1 TO N
1570 K=JA(I):GOSUB 2850:P(I)=PT:NEXT I
1580 GOTO 1320
1590 PRINT #3,:PRINT #3, TAB(10)Z8$,Z$(3)
1610 PG=1/ST(2):PRINT #3,:PRINT #3, TAB(5)Z1$(3)"=";USING F$;PG
1630 FOR I=1 TO N
1640 K=JA(I):GOSUB 2890:P(I)=PT:NEXT I
1660 GOTO 1320
1680 PRINT #3,:PRINT #3, TAB(10)Z8$,Z$(4)
1690 PG=1/ST(2):PRINT #3,:PRINT #3, TAB(5)Z1$(3)"=";USING F$;PG
1710 FOR I=1 TO N
1720 K=JA(I):GOSUB 2920:P(I)=PT:NEXT I
1740 GOTO 1320
2250 FOR I=1 TO N:P8(I)=P9(I):PO(I)=P(I):NEXT I
2260 N1=N:N11=NU:NN=N-N11+1:SM=P9(1)*NNA:SQ=0.
2270 I=2:L1=0
2280 IF N-L1<=NN THEN 2330
2290 IF SM>=5 THEN 2330
2300 SM=SM+P9(I)*NNA
2310 I=I+1:L1=L1+1
2320 GOTO 2280
2330 SM=P9(N)*NNA
2340 I=0:L2=0
2350 IF N-L1-L2<=NN GOTO 2380
2360 IF SM>=5 GOTO 2380
2370 L2=L2+1:I=I+1:SM=SM+P9(N-I)*NNA:GOTO 2350
2380 IF L1=0 THEN 2440
2390 FOR I=1 TO L1:P8(1)=P8(1)+P8(I+1)
2400 PO(1)=PO(1)+PO(I+1):NEXT I
2410 N=N-L1
```

```
2420 FOR I=2 TO N:P8(I)=P8(I+L1)
2430 PO(I)=PO(I+L1):NEXT I
2440 IF L2=0 THEN 2480
2445 N=N-L2
2450 FOR I=1 TO L2:P8(N)=P8(N)+P8(N+I)
2460 PO(N)=PO(N)+PO(N+I):NEXT I
2480 FOR I=1 TO N
2485 IF PO(I)<1E-36 THEN SQ=1.E+36:GOTO 2495
2490 SQ=SQ+((PO(I)-P8(I))^2)/PO(I)
2495 NEXT I
2500 SQ=SQ*NNA
2510 NU=N11-L2-L1
2520 RK=(SQ-NU)/SQR(2*NU)
2540 PRINT #3, :PRINT #3, :PRINT #3, "    ВЕКТОР ТЕОРЕТИЧЕСКИХ ЧАСТОСТЕЙ С"
2550 PRINT #3, "     ОБЬЕДИНЕНИЕМ ИНТЕРВАЛОВ НА КОНЦАХ"
2560 FOR I=1 TO N:BJ(I)=PO(I):NEXT I:NF=N:PRINT #3,:GOSUB 890
2570 PRINT #3, :PRINT #3, :PRINT #3, "           ЗНАЧЕНИЕ ХИ-КВАДРАТ      =";:
2575 PRINT #3, USING F$;SQ
2580 PRINT #3, "     ЧИСЛО СТЕПЕНЕЙ СВОБОДЫ         ="NU
2590 PRINT #3, "     ЗНАЧЕНИЕ КРИТЕРИЯ РОМАНОВСКОГО =";
2595 PRINT #3, USING F1$;RK
2600 N=N1
2640 INPUT "ТРЕБУЕТСЯ ЛИ РАСЧЕТ КРИТЕРИЯ МИЗЕСА(ДА- 1,НЕТ- 0)";IR3
2650 IF IR3=0 THEN 3060
2660 SW=0.:I2=0:FT=0.:FO=0.
2670 I2=I2+1
2680 IF I2>N THEN 3040
2700 FT=FT+P(I2):IF FT<=0 THEN FT=1.E-36
2705 IF FT>=1. THEN FT=.999999
2800 FO=FO+P9(I2):IF FO<=0. THEN FO=1.E-36
2805 IF FO>=1. THEN FO=.999999
2806 SW=SW+(FO-FT)^2/(FT*(1-FT))
2810 GOTO 2670
2820 PI=1.:FOR J=1 TO K:PI=PI*(NW+1-J)/J:NEXT J:PT=PI*PB^K*(1.-PB)^(NW-K)
2830 RETURN
2850 FOR J=0 TO K:IF J=0 THEN PI=1. ELSE PI=PI*J
2885 NEXT J
2860 PT=AP^K/PI*EXP(-AP)
2880 RETURN
2890 PT=PG*(1.-PG)^K
2910 RETURN
2920 PT=PG/(1.-PG)*(1.-PG)^K
2940 RETURN
3040 'W2=-NO-1/NO*SW
3050 PRINT #3, "       ЗНАЧЕНИЕ КРИТЕРИЯ МИЗЕСА     =";
3055 PRINT #3, USING F$;SW
3060 RETURN
3070 PRINT"ПРОДОЛЖИТЬ РАСЧЕТЫ С ДРУГИМИ ДАННЫМИ(ДА- 1,НЕТ- 0"
3075 INPUT N5
3080 IF N5<>1 THEN 3095
3085 ERASE JA,NA,IUB,ST,P,P8,PO,BJ,P9,JR,PM,PS,S$,Z1$
3090 GOTO 19
3095 COLOR 25,5,15:PRINT "   РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ RNDD.REZ"
3096 PRINT "   ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПОВТОРНОГО РАСЧЕТА В ФАЙЛЕ "IMF$
3097 BBB$=INKEY$
3098 IF LEN(BBB$)=0 THEN 3097
3099 RUN"RND.EXE"
3100 END
```

ПРИЛОЖЕНИЕ 3
КОМПЬЮТЕРНАЯ ПРОГРАММА ОДНОФАКТОРНОГО КОРРЕЛЯЦИОННО-РЕГРЕССИОННОГО АНАЛИЗА

```
5 CLS
8 PRINT TAB(24)"З А Д А Ч А    P R O G N"
9 PRINT TAB(21)"К А Ф Е Д Р А";:PRINT" *БНТУ*";
10 PRINT" О А П Д Д "
17 PRINT TAB(10)"ОДНОФАКТОРНЫЙ КОРРЕЛЯЦИОННО-РЕГРЕССИОННЫЙ АНАЛИЗ ТЭП"
20 DEFINT I-N
25 F1$="+#.###^^^^"
35 GOTO 50
40 N=N1-1:DIM Q(255),Y(255),T3(255),X(255),F(11),E(11),R(11)
45 RETURN
50 PRINT"ВВЕДИТЕ ЗНАЧЕНИЕ ФАКТОРА ДЛЯ РАСЧЕТНОЙ ТОЧКИ";: INPUT T4
70 PRINT"ВВЕДЕНЫ ЛИ ИСХОДНЫЕ ДАННЫЕ. ДА- 1,НЕТ- 0":INPUT R5
80 IF R5=0 THEN PRINT"ВВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ":GOTO 300
90 OPEN"I",#2,"PROGN.DAT"
95 INPUT #2,FS$
97 INPUT #2,N1:GOSUB 40
100 FOR I=0 TO N
110 INPUT #2,T3(I),Q(I)
120 NEXT I
130 CLOSE #2
140 CLS:PRINT"   ИСХОДНЫЕ ДАННЫЕ":PRINT"ПОРЯДКОВ. N   ФАКТОР       ТЭП"
142 FOR I=1 TO N+1:PRINT I,T3(I-1),Q(I-1):NEXT I
149 PRINT"ТРЕБУЕТСЯ ЛИ КОРРЕКТИРОВАТЬ ИСХОДНЫЕ ДАННЫЕ. ДА- 1,НЕТ- 0"
150 INPUT R6
160 IF R5=0 AND R6=0 THEN 340
170 IF R6=0 THEN 371
180 PRINT"КОРРЕКТИРОВАТЬ ЗНАЧЕНИЯ ФАКТОРА? ДА- 1,НЕТ- 0"
190 INPUT R8:IF R8=0 THEN 250
200 PRINT"ПОРЯДКОВЫЙ НОМЕР(СЧИТАЯ С ЕДИНИЦЫ) ЗНАЧЕНИЯ ФАКТОРА,"
210 PRINT"ПОДЛЕЩАЩЕГО ИЗМЕНЕНИЮ":INPUT IS
220 IF IS<1 OR IS>N+1 THEN PRINT"ВЫ ДОПУСТИЛИ ОШИБКУ":GOTO 210
230 IS=IS-1:PRINT"НОВОЕ ЗНАЧЕНИЕ ФАКТОРА":INPUT T3(IS)
240 GOTO 180
250 PRINT"КОРРЕКТИРОВАТЬ ЗНАЧЕНИЯ ТЭП ? ДА- 1,НЕТ- 0":INPUT R9
260 IF R9=0 THEN R5=0:GOTO 140
270 PRINT"ПОРЯДКОВЫЙ НОМЕР(СЧИТАЯ С ЕДИНИЦЫ) ЗНАЧЕНИЯ ТЭП":INPUT IS
280 IF IS<1 OR IS>N+1 THEN PRINT"ВЫ ДОПУСТИЛИ ОШИБКУ":GOTO 270
290 IS=IS-1:PRINT"НОВОЕ ЗНАЧЕНИЕ ТЭП":INPUT Q(IS):GOTO 250
300 CLS:INPUT"НАЗВАНИЕ ПРОГНОЗИРУЕМОГО ТЭП";FS$
302 INPUT"ЧИСЛО ЗНАЧЕНИЙ ФАКТОРА";N1:GOSUB 40
305 PRINT"ПОСЛЕДОВАТЕЛЬНО ЗНАЧЕНИЯ ФАКТОРА"
310 FOR I=0 TO N:INPUT T3(I):NEXT I
320 PRINT"ПОСЛЕДОВАТЕЛЬНО ЗНАЧЕНИЯ ТЭП"
330 FOR I=0 TO N:INPUT Q(I):NEXT I:GOTO 140
340 OPEN"O",#3,"PROGN.DAT"
345 WRITE #3,FS$
347 WRITE #3,N1
350 FOR I=0 TO N
360 WRITE #3,T3(I),Q(I):NEXT I
370 CLOSE #3
371 FOR I=0 TO N-1:K=I
372 J=K+1:IF T3(K)<=T3(J) THEN 378
373 TX=T3(K):QX=Q(K):T3(K)=T3(J):Q(K)=Q(J):T3(J)=TX:Q(J)=QX:K=K-1
374 IF K>=0 THEN 372
378 NEXT I
380 D$="ВЫРАВНИВАЮЩАЯ ЗАВИСИМОСТЬ"
392 N5=N+1:T3(N5)=T4
396 DIM Q1(255),T(255)
400 FOR I=0 TO N5:T(I)=T3(I):NEXT I
412 OPEN"O",#1,"PROGN.REZ"
415 PRINT #1, TAB(10)"ОДНОФАКТОРНЫЙ АНАЛИЗ ТЭП"
416 PRINT #1, TAB(5)"("FS$")"
420 MM=-1:PRINT"ТРЕБУЕТСЯ ЛИ НАЙТИ НАИБОЛЕЕ ПОДХОДЯЩУЮ МОДЕЛЬ"
430 PRINT"ЗАВИСИМОСТИ ?  НЕТ - 0, ТРЕБУЕТСЯ ПО КРИТЕРИЮ"
440 PRINT"ФИШЕРА - 1, ПО СРЕДНЕЙ ЛИНЕЙНОЙ ОШИБКЕ"
```

```
450 PRINT"АППРОКСИМАЦИИ - 2":INPUT R7:IF R7=0 THEN R8=1:GOTO 460
455 PRINT"РАССМАТРИВАТЬ ЛИ ФУНКЦИИ,ИМЕЮЩИЕ ЭКСТРЕМУМ.ДА- 1,НЕТ- 0"
456 INPUT R8
460 RRR=1
465 IF R7<>0 THEN M=0:MO=-1:FF=0:EE=100:GOTO 530
470 PRINT"ВВЕДИТЕ НОМЕР МАТЕМАТИЧЕСКОЙ МОДЕЛИ M"
480 PRINT"МАТЕМАТИЧЕСКИМ МОДЕЛЯМ ПРИСВОЕНЫ СЛЕДУЮЩИЕ НОМЕРА"
490 PRINT"0- T/(A*T+B)  1- A*T+B  2- A*T^2+B*T+C  3- A*T^K+B"
500 PRINT"4- 1/(A*T+B)  5- A*LN T+B*T+C  6- A/T+B  7- B*A^T"
510 PRINT"8- 1/(A*EXP(-T)+B)  9- B*T^A  10- A*LN T+B";
520 PRINT" 11- B*A^(1/T)":INPUT M:GOTO 550
530 PRINT #1,:PRINT #1, D$;"          КРИТЕРИЙ ФИШЕРА   СРЕДНЯЯ ЛИНЕЙНАЯ"
540 PRINT #1, TAB(43)"  F      ОШИБКА АППРОКСИМАЦИИ E":PRINT #1,
550 ON M+1 GOTO 560,630,700,880,960,1030,1220,1290,1360,1430,1500,1570
560 FOR I=0 TO N:Y(I)=1/Q(I):X(I)=1/T(I):NEXT I
570 GOSUB 2110
580 B=A1:A=B1
590 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
600 PRINT #1," Q=T/("USING F1$;A;:PRINT #1,"*T";:PRINT #1, USING F1$;B;
605 PRINT #1,")";
610 FOR I=0 TO N5:Q1(I)=T(I)/(A*T(I)+B):NEXT I
620 GOTO 1640
630 FOR I=0 TO N:Y(I)=Q(I):X(I)=T(I):NEXT I
640 GOSUB 2110
650 A=A1:B=B1
660 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
670 PRINT #1," Q="USING F1$;A;:PRINT #1,"*T";:PRINT #1, USING F1$;B;
680 FOR I=0 TO N5:Q1(I)=A*T(I)+B:NEXT I
690 GOTO 1640
700 S1=0:S2=0:S3=0:S4=0:S5=0:S6=0
710 FOR I=0 TO N:S1=S1+T(I):NEXT I
720 T1=S1/N1
730 FOR I=0 TO N
740 S2=S2+Q(I):S3=S3+(T(I)-T1)^2
750 S4=S4+(T(I)-T1)^4
760 S5=S5+Q(I)*(T(I)-T1)
770 S6=S6+Q(I)*(T(I)-T1)^2
780 NEXT I
790 B1=S5/S3
800 A=(S6-S2*S3/N1)/(S4-S3^2/N1)
810 C1=(S2-A*S3)/N1
820 B=B1-2*A*T1
830 C=C1+A*T1^2-B1*T1
840 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
850 PRINT #1," Q="USING F1$;A;:PRINT #1,"*T^2"USING F1$;B;
855 PRINT #1,"*T"USING F1$;C;
860 FOR I=0 TO N5:Q1(I)=A*T(I)^2+B*T(I)+C:NEXT I
870 GOTO 1640
880 PRINT:INPUT"ПОКАЗАТЕЛЬ СТЕПЕНИ В СТЕПЕННОЙ ЗАВИСИМОСТИ T";PK
890 FOR I=0 TO N:Y(I)=Q(I):X(I)=T(I)^PK:NEXT I
900 GOSUB 2110
910 B=B1:A=A1
920 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
930 PRINT #1," Q="USING F1$;A;:PRINT #1,"*T^"USING F1$;PK;
935 PRINT #1, " "USING F1$;B;
940 FOR I=0 TO N5:Q1(I)=A*T(I)^PK+B:NEXT I
950 GOTO 1640
960 FOR I=0 TO N:Y(I)=1/Q(I):X(I)=T(I):NEXT I
970 GOSUB 2110
980 B=B1:A=A1
990 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1000 PRINT #1," Q= 1/("USING F1$;A;:PRINT #1,"*T"USING F1$;B;
1005 PRINT #1,")";
1010 FOR I=0 TO N5:Q1(I)=1/(A*T(I)+B):NEXT I
1020 GOTO 1640
1030 S1=0:S2=0:S3=0:S4=0:S5=0:S6=0
1040  S7=0:S8=0
1050 FOR I=0 TO N:X(I)=LOG(T(I)):S1=S1+X(I):S2=S2+T(I):NEXT I
```

```
1060 X1=S1/N1:T1=S2/N1
1070 FOR I=0 TO N
1080 S3=S3+Q(I):S4=S4+(X(I)-X1)^2
1090 S5=S5+(T(I)-T1)^2
1100 S6=S6+Q(I)*(X(I)-X1)
1110 S7=S7+Q(I)*(T(I)-T1)
1120 S8=S8+(X(I)-X1)*(T(I)-T1)
1130 NEXT I
1140 C1=S3/N1
1150 B=(S7-S6*S8/S4)/(S5-S8^2/S4)
1160 A=(S6-B*S8)/S4
1170 C=C1-A*X1-B*T1
1180 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1190 PRINT #1," Q="USING F1$;A;:PRINT #1,"*LN T"USING F1$;B;:PRINT #1,"*T";:
1195 PRINT #1, USING F1$;C;
1200 FOR I=0 TO N5:Q1(I)=A*LOG(T(I))+B*T(I)+C:NEXT I
1210 GOTO 1640
1220 FOR I=0 TO N:Y(I)=Q(I):X(I)=1/T(I):NEXT I
1230 GOSUB 2110
1240 B=B1:LET A=A1
1250 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1260 PRINT #1," Q="USING F1$;A;:PRINT #1, "/T"USING F1$;B;
1270 FOR I=0 TO N5:Q1(I)=A/T(I)+B:NEXT I
1280 GOTO 1640
1290 FOR I=0 TO N:Y(I)=LOG(Q(I)):X(I)=T(I):NEXT I
1300 GOSUB 2110
1310 A=EXP(A1):B=EXP(B1)
1320 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1330 PRINT #1," Q="USING F1$;B;:PRINT #1,"*"USING F1$;A;:PRINT #1,"^T";
1340 FOR I=0 TO N5:Q1(I)=B*A^T(I):NEXT I
1350 GOTO 1640
1360 FOR I=0 TO N:Y(I)=1/Q(I):X(I)=EXP(-T(I)):NEXT I
1370 GOSUB 2110
1380 A=A1:B=B1
1390 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1400 PRINT #1," Q=1/("USING F1$;A;:PRINT #1,"*EXP(-T)"USING F1$;B;
1405 PRINT #1,")";
1410 FOR I=0 TO N5:Q1(I)=1/(A*EXP(-T(I))+B):NEXT I
1420 GOTO 1640
1430 FOR I=0 TO N:Y(I)=LOG(Q(I)):X(I)=LOG(T(I)):NEXT I
1440 GOSUB 2110
1450 A=A1:B=EXP(B1)
1460 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1470 PRINT #1," Q="USING F1$;B;:PRINT #1,"*T^"USING F1$;A;
1480 FOR I=0 TO N5:Q1(I)=B*T(I)^A:NEXT I
1490 GOTO 1640
1500 FOR I=0 TO N:Y(I)=Q(I):X(I)=LOG(T(I)):NEXT I
1510 GOSUB 2110
1520 A=A1:B=B1
1530 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1540 PRINT #1," Q="USING F1$;A;:PRINT #1,"*LN T"USING F1$; B;
1550 FOR I=0 TO N5:Q1(I)=A*LOG(T(I))+B:NEXT I
1560 GOTO 1640
1570 FOR I=0 TO N:Y(I)=LOG(Q(I)):X(I)=1/T(I):NEXT I
1580 GOSUB 2110
1590 A=EXP(A1):B=EXP(B1)
1600 IF R7=0 THEN PRINT #1,:PRINT #1, D$;
1610 PRINT #1," Q="USING F1$;B;:PRINT #1,"*"USING F1$;A;
1615 PRINT #1,"^(1/T)";
1620 FOR I=0 TO N5:Q1(I)=B*A^(1/T(I)):NEXT I
1630 GOTO 1640
1640 S=0:C2=0:C3=0:C4=0
1650 FOR I=0 TO N:C2=C2+Q(I):NEXT I
1660 Q5=C2/(N+1)
1665 IIR=0
1680 FOR I=0 TO N
1685 IF Q1(I)=0 THEN IIR=1:GOTO 1710
1690 S=S+ABS((Q(I)-Q1(I))/Q1(I))
```

```
1700 C3=C3+(Q(I)-Q5)^2:C4=C4+(Q(I)-Q1(I))^2
1710 NEXT I
1715 IF IIR=0 AND C4<C3 THEN 1720
1717 PRINT"ЗАВИСИМОСТЬ N% "M" НЕ ПОДХОДИТ":E(M)=1:F(M)=0:R(M)=0:GOTO 1730
1720 R=SQR(1-C4/C3):IF C4=0 THEN F(M)=1E20 ELSE F(M)=(C3-C4)*(N-1)/C4:E(M)=S/N1
1730 IF MM=13 THEN 1810
1740 IF R7=0 THEN 1840
1750 IF R7=2 THEN 1770
1760 IF FF<F(M) THEN FF=F(M):MO=M:GOTO 1780
1765 GOTO 1780
1770 IF EE>E(M) THEN EE=E(M):MO=M
1780 PRINT #1, TAB(40) USING F1$;F(M);TAB(60) E(M)
1790 IF M=11 THEN M=MO:MM=13:R8=R7:R7=0:PRINT #1,:GOTO 550
1800 M=M+1:IF R8=0 AND M=2 OR M=5 THEN M=M+1
1805 GOTO 550
1810 PRINT #1,:PRINT #1,"(НАИБОЛЕЕ ЗНАЧИМАЯ ПО";
1820 IF R8=1 THEN PRINT #1," КРИТЕРИЮ ФИШЕРА)":PRINT #1,:GOTO 1840
1830 PRINT #1," СРЕДНЕЙ ЛИНЕЙНОЙ":PRINT #1,"ОШИБКЕ АППРОКСИМАЦИИ)":PRINT #1,
1840 PRINT #1,:PRINT #1,"   T","    Q","    Q1"
1850 FOR I=0 TO N:PRINT #1, TAB(1)USING F1$;T(I)TAB(16)Q(I)TAB(32)Q1(I)
1855 NEXT I
1860 PRINT #1,:PRINT #1,"КОЭФФИЦИЕНТ КОРРЕЛЯЦИИ R="USING F1$;R
1870 PRINT #1,"КРИТЕРИЙ ФИШЕРА F="USING F1$;F(M)
1880 PRINT #1,"КОЭФФИЦИЕНТ ЛИНЕЙНОЙ ОШИБКИ АППРОКСИМАЦИИ E="USING F1$;E(M)
1890 PRINT #1,"ЗНАЧЕНИЕ ПОКАЗАТЕЛЯ В РАСЧЕТНОЙ ТОЧКЕ="USING F1$;T4;
1900 PRINT #1," -- "USING F1$;Q1(N5):PRINT #1,:PRINT #1,"ТОЧЕЧНАЯ ДИАГРАММА"
1910 FOR I=1 TO 56:PRINT #1, TAB(I)"-";:NEXT I:PRINT #1, ">";:PRINT #1, TAB(58)"Q"
1920 Q7=Q1(0)
1930 FOR I=1 TO N5
1940 IF Q1(I)>Q7 THEN Q7=Q1(I):GOTO 1950
1950 NEXT I
1960 A3=55/Q7
1970 FOR I=0 TO N:PRINT #1,"I":PRINT #1,"I";
1980 Q2=CINT(Q1(I)*A3):Q3=CINT(Q(I)*A3)
1990 IF Q3>Q2 THEN PRINT #1, TAB(Q2)"*";TAB(Q3)"+":GOTO 2020
2000 IF Q3=Q2 THEN PRINT #1, TAB(Q2)"#":GOTO 2020
2010 PRINT #1, TAB(Q3)"+";TAB(Q2)"*"
2020 NEXT I
2030 PRINT #1,"I";:PRINT #1,TAB(Cint(Q1(N5)*A3))"&"
PRINT #1,"v":PRINT #1,"T":PRINT #1,
2050 PRINT #1," + ЭМПИРИЧЕСКИЕ, * ТЕОРЕТИЧЕСКИЕ, # ПРИ СОВПАДЕНИИ, & ДЛЯ РАСЧЕТНОЙ ТОЧКИ"
2070 PRINT #1,:PRINT #1, TAB(20)".....***....."
2080 PRINT"ТРЕБУЕТСЯ ЛИ ПРОДОЛЖИТЬ РАСЧЕТЫ ? ДА- 1, НЕТ- 0"
2090 INPUT K5:IF K5=1 THEN 420
2100 PRINT:GOTO 2180
2110 S1=0:S2=0:S3=0:S4=0
2120 FOR I=0 TO N:S1=S1+X(I):NEXT I
2130 X1=S1/N1
2140 FOR I=0 TO N:S2=S2+Y(I):S3=S3+(X(I)-X1)^2
2150 S4=S4+(X(I)-X1)*Y(I):NEXT I
2155 IF S3=0 THEN 2170
2160 A1=S4/S3:B1=S2/N1-X1*A1
2170 RETURN
2180 CLOSE #1
2185 COLOR 25,5,15:PRINT "   РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ PROGN.REZ"
2187 PRINT "   ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПОВТОРНОГО РАСЧЕТА В ФАЙЛЕ PROGN.DAT"
2188 BBB$=INKEY$
2190 IF LEN(BBB$)=0 THEN 2188
2200 END
```

ПРИЛОЖЕНИЕ 4
КОМПЬЮТЕРНАЯ ПРОГРАММА ПРОВЕДЕНИЯ МНОГОФАКТОРНОГО КОРРЕЛЯЦИОННО-РЕГРЕССИОННОГО АНАЛИЗА

```
1 COLOR 7,0:CLS:LOCATE 2,60,0:COLOR 25,5,15
2 COLOR 7,7,15:LOCATE 4:COLOR 0,7,15
3 PRINT TAB(27)"ЗАДАЧА     KORREG";:PRINT SPC(37):PRINT TAB(26)" ";
4 COLOR 15,0,15:PRINT"КАФЕРА";:COLOR 23,5,15:PRINT"*БНТУ*";
5 COLOR 15,0,15:PRINT" ОАПДД ";:COLOR 7,7,15:PRINT SPC(35):COLOR 1,14
6 PRINT TAB(22)"КОРРЕЛЯЦИОННО-РЕГРЕССИОННЫЙ";
7 PRINT TAB(80)" ";:PRINT TAB(31)"А Н А Л И З";
8 PRINT TAB(80)" ":DEFINT I-N
19 PRINT:COLOR 7,0,7
OPEN"O",#1,"KORREG.REZ"
130 H0$=" ВЫ ДОПУСТИЛИ ОШИБКУ":H1$=" ЗНАЧЕНИЯ ":H4$="ВВЕДИТЕ"
140 H5$=" ФАКТОРА":H6$="ОПЫТА":H7$="КОРРЕКТИРОВАТЬ"
150 H8$=" ИСХОДНЫЕ ДАННЫЕ":F$="+#.##^^^^"
160 H9$=" ЧЕРЕЗ ЗАПЯТУЮ ПОРЯДКОВЫЕ НОМЕРА "
170 P4$="НОВАЯ ВЕЛИЧИНА "
180 PRINT"ВВЕДЕНЫ ЛИ "H8$".ДА- 1,НЕТ- 0":INPUT M0:CLS
190 IF M0=0 THEN 390
195 INPUT"ЗАДАЙТЕ СПЕЦИФИКАЦИЮ ФАЙЛА ДАННЫХ";FAI$
200 OPEN"I",#2,FAI$
210 INPUT #2,K,NN
220 GOSUB 620
230 FOR I=0 TO K:FOR J=O TO N:INPUT #2,X0(J,I):NEXT J:NEXT I
240 CLOSE #2
250 CLS:PRINT TAB(10)H8$
260 PRINT "    Y";:FOR I=1 TO K:PRINT TAB(10*I+4)"X("I")";:NEXT I
270 PRINT:FOR J=0 TO N:PRINT J+1;:FOR I=0 TO K
280 PRINT TAB(10*I+5)USING F$;X0(J,I);:NEXT I:PRINT
282 IF J+1 MOD 20=0 THEN INPUT AAA:CLS
285 NEXT J
288 PRINT
290 COLOR 0,10,2
300 PRINT H7$"  "H8$". ДА- 1,НЕТ- 0";:COLOR 7,0,7:INPUT M8
310 IF M0=0 AND M8=0 THEN 500
320 IF M8=0 THEN 560
330 M0=0
340 PRINT H4$;H9$;H6$;" И"H5$
350 PRINT " ( ФУНКЦИЮ СЧИТАТЬ НУЛЕВЫМ ФАКТОРОМ )"
360 INPUT JS,IS
370 IF JS<1 OR IS<0 OR IS>K OR JS>NN THEN PRINT H0$:BEEP:GOTO 340
380 PRINT P4$:INPUT X0(IS,JS)
385 GOTO 290
390 CLS:PRINT H4$;H8$
400 INPUT"ЧИСЛО ФАКТОРОВ";K
410 INPUT"ЧИСЛО ОПЫТОВ";NN
420 GOSUB 620
430 PRINT" ПОСЛЕДОВАТЕЛЬНО ЧИСЛО И ВВОД"
440 FOR I=1 TO K
450 PRINT H4$;H1$;I" -ГО"H5$
460 FOR J=0 TO N:INPUT X0(J,I):NEXT J:NEXT I
470 PRINT" ФУНКЦИИ"
480 FOR J=0 TO N:INPUT X0(J,0):NEXT J:PRINT
490 GOTO 250
500 INPUT"ЗАДАЙТЕ СПЕЦИФИКАЦИЮ ФАЙЛА ДЛЯ ДАННЫХ";FAI$
505 OPEN"O",#2,FAI$
510 WRITE #2,K,N+1
520 FOR I=0 TO K
530 FOR J=0 TO N:WRITE #2,X0(J,I)
540 NEXT J:NEXT I
550 CLOSE #2
560 REM
570 PRINT "УРАВНЕНИЕ ИМЕЕТ АДДИТИВНЫЙ ВИД- ВВЕДИТЕ 1"
580 PRINT "ИЛИ МУЛЬТИПЛИКАТИВНЫЙ ВИД- ВВЕДИТЕ 0":INPUT M8
590 PRINT "УРАВНЕНИЕ ИМЕЕТ СВОБОДНЫЙ ЧЛЕН ? ДА- 1,НЕТ- 0"
600 INPUT M7:M7=1-M7
610 GOTO 650
```

```
620 N=NN-1:K4=K*2+1:K5=K+4:K6=K+1:K7=K+3
630 DIM X0(N,K5),X1(K,K4),X2(K,N),X3(K,K4),X5(N)
640 RETURN
650 C=.05
660 FOR J=0 TO N:X0(J,K6)=X0(J,0):NEXT J
665 IF M7=1 THEN 680
670 FOR J=0 TO N:X0(J,0)=1:NEXT J
680 S7=0:S8=0
690 FOR I=O TO N:S7=S7+X0(I,K6)^2:S8=S8+X0(I,K6):NEXT I
700 S9=S7-S8*S8/(N+1)
710 PRINT #1, TAB(5)"РЕЗУЛЬТАТЫ";
720 PRINT #1," КОРРЕЛЯЦИОННО-РЕГРЕССИОННОГО АНАЛИЗА"
780 IF M8=1 THEN 810
790 FOR J=1 TO K:FOR I=0 TO N:X0(I,J)=LOG(X0(I,J)):NEXT I:NEXT J
800 FOR I=0 TO N:X5(I)=X0(I,K6):X0(I,K6)=LOG(X0(I,K6)):NEXT I
810 FOR I=M7 TO K:FOR J=0 TO N:X2(I,J)=X0(J,I):NEXT J:NEXT I
820 GOSUB 840
830 GOTO 880
840 FOR I=M7 TO K:FOR J=M7 TO K:A=0
850 FOR L=0 TO N:A=A+X2(I,L)*X0(L,J):NEXT L
860 X1(I,J)=A:NEXT J:NEXT I
870 RETURN
880 FOR I=M7 TO K:FOR J=M7 TO K
890 IF I<>J THEN X1(I,J+K6)=0:GOTO 910
900 X1(I,J+K6)=1
910 NEXT J
920 NEXT I
930 K1=M7
940 IF K=M7 THEN X1(1,K+2)=1/A:GOTO 1270
950 GOSUB 1830
960 K2=K1:IF K2>=K-1 THEN 980
970 K1=K1+1:GOTO 950
980 B3=1:FOR I=M7 TO K:B3=B3*X1(I,I):NEXT I
990 IF B3=0 THEN PRINT #1, "ОПРЕДЕЛИТЕЛЬ РАВЕН НУЛЮ ":GOTO 1950
1000 FOR I=K6 TO K4
1010 X1(K,I)=X1(K,I)/X1(K,K):NEXT I
1020 K2=K-1
1030 FOR I=M7 TO K2:FOR J=K6 TO K4
1040 X1(I,J)=X1(I,J)-X1(I,K2+1)*X1(K2+1,J)
1050 NEXT J:NEXT I
1060 K2=K2-1:IF K2>=M7 THEN 1030
1070 GOSUB 840
1080 FOR I=M7 TO K:FOR J=M7 TO K
1090 IF I<>J THEN X3(I,J)=0:GOTO 1110
1100 X3(I,J)=1
1110 NEXT J
1120 NEXT I
1130 FOR I=M7 TO K:FOR J=M7 TO K:A=0
1140 FOR L=M7 TO K:A=A+X1(I,L)*X1(L,J+K6):NEXT L
1150 X3(I,J)=2*X3(I,J)-A:NEXT J:NEXT I
1160 FOR I=M7 TO K:FOR J=M7 TO K:A=0
1170 FOR L=M7 TO K:A=A+X1(I,L+K6)*X3(L,J):NEXT L
1180 X3(I,J+K6)=A:NEXT J:NEXT I
1190 FOR I=M7 TO K:FOR J=M7 TO K
1200 X1(I,J+K6)=X3(I,J+K6):NEXT J:NEXT I
1210 A1=0:FOR I=M7 TO K:FOR J=M7 TO K
1220 A1=A1+ABS(X3(I,J)):NEXT J:NEXT I
1230 A2=0:FOR I=M7 TO K:A2=A2+ABS(X3(I,I)):NEXT I
1240 IF A1-A2<C GOTO 1270
1250 PRINT "СИСТЕМА СЛАБО ОБУСЛОВЛЕНА. &A=";A1-A2
1260 GOTO 1080
1270 FOR I=M7 TO K:A=0
1280 FOR L=0 TO N:A=A+X2(I,L)*X0(L,K6):NEXT L
1290 X0(I,K7)=A:NEXT I
1300 FOR I=M7 TO K:A=0
1310 FOR L=M7 TO K:A=A+X1(I,L+K6)*X0(L,K7):NEXT L
1320 X0(I,K5)=A:NEXT I
1330 FOR I=0 TO N:A=0
```

```
1340 FOR L=M7 TO K:A=A+X0(I,L)*X0(L,K5):NEXT L
1350 X0(I,K7)=A:NEXT I
1360 A3=0:A6=0:A7=0:A8=0
1370 FOR I=0 TO N
1380 A6=A6+(X0(I,K6)-X0(I,K7))^2
1390 IF M8=1 THEN 1440
1400 X0(I,K7)=EXP(X0(I,K7))
1410 A8=A8+(X0(I,K7)-X5(I))^2
1420 A3=A3+ABS((X0(I,K7)-X5(I))/X0(I,K7))
1430 GOTO 1450
1440 A3=A3+ABS((X0(I,K6)-X0(I,K7))/X0(I,K7))
1450 NEXT I
1460 IF M8=0 THEN S2=A8:GOTO 1480
1470 S2=A6
1480 E=A3/(N+1)
1490 PRINT #1,:PRINT #1, TAB(5)"УРАВНЕНИЕ РЕГРЕССИИ Y = ";
1500 IF M8=1 AND M7=0 THEN 1530
1510 IF M8=1 THEN 1580
1515 IF M7=1 THEN 1550
1520 PRINT #1, USING F$;EXP(X0(0,K5));:PRINT #1,"*":GOTO 1550
1530 PRINT #1, USING F$;X0(0,K5);:PRINT #1, " + ":GOTO 1580
1540 PRINT #1,
1550 FOR I=1 TO K:PRINT #1, "*X("I")^";
1560 PRINT #1, USING F$;X0(I,K5);:NEXT I
1570 GOTO 1600
1580 FOR I=1 TO K:PRINT #1, " ";:PRINT #1, USING F$;X0(I,K5);
1590 PRINT #1," X("I")";:NEXT I
1600 PRINT #1,
1602 PRINT #1,:PRINT #1, TAB(5)"ЗНАЧЕНИЯ ФАКТОРОВ И ЗАВИСИМОЙ ПЕРЕМЕННОЙ"
1604 PRINT #1," I";
1606 FOR J=1 TO K:PRINT #1,"    X("J")";:NEXT J
1608 PRINT #1,"    Y ЭКСПЕР.  Y ТЕОРЕТ.":PRINT #1,
1610 FOR I=0 TO N:PRINT #1, I+1;
1612 FOR J=1 TO K:PRINT #1, TAB(10*J-4) USING F$;X0(I,J);:NEXT J
1620 IF M8=0 THEN 1660
1630 PRINT #1,TAB (10*K+6)USING F$;X0(I,K6),
1645 PRINT #1,TAB(10*(K+1)+6)USING F$;X0(I,K7)
1650 GOTO 1680
1660 PRINT #1, TAB(10*K+6)USING F$;X5(I),
1675 PRINT #1, TAB(10*(K+1)+6)USING F$;X0(I,K7)
1680 NEXT I
1685 PRINT #1, :PRINT #1, "S^2 ОСТАТОЧНАЯ =";:PRINT #1, USING F$;S2
1690 PRINT #1, "S^2 ПОЛНАЯ=";:PRINT #1, USING F$;S9
1700 IF S9>S2 GOTO 1720
1710 PRINT"ПРИНЯТЫЙ ВИД УРАВНЕНИЯ РЕГРЕССИИ НЕ ПОДХОДИТ":GOTO 1950
1720 R=SQR((S9-S2)/S9)
1730 PRINT #1, "КОЭФФИЦИЕНТ МНОЖЕСТВЕННОЙ КОРРЕЛЯЦИИ R=";
1735 PRINT #1, USING F$;R
1740 PRINT #1, "КОЭФФИЦИЕНТ ЛИНЕЙНОЙ ОШИБКИ";
1750 PRINT #1," АППРОКСИМАЦИИ E=";:PRINT #1, USING F$;E
1760 PRINT #1, "КРИТЕРИЙ ФИШЕРА F=";
1765 PRINT #1, USING F$;(S9-S2)*(N-K)/(S2*K)
1770 PRINT #1, "КРИТЕРИЙ СТЬЮДЕНТА T ДЛЯ ПАРАМЕТРОВ УРАВНЕНИЯ"
1780 S=A6/(N-K)
1790 FOR I=M7 TO K
1800 T=ABS(X0(I,K5)/SQR(S*X1(I,I+K6)))
1810 PRINT #1, "T("I")=";:PRINT #1, USING F$;T:NEXT I
1820 PRINT #1,:PRINT #1, ".....***.....":PRINT #1,:GOTO 1950
1830 I1=K1
1840 IF X1(I1,K1)<>0 THEN K2=I1:GOTO 1870
1850 I1=I1+1:IF I1>K THEN 1940
1860 GOTO 1840
1870 FOR J1=K1 TO K4:B1=X1(K1,J1):X1(K1,J1)=X1(K2,J1)
1880 X1(K2,J1)=B1:NEXT J1
1890 FOR J2=K1+1 TO K4
1900 X1(K1,J2)=X1(K1,J2)/X1(K1,K1):NEXT J2
1910 FOR I3=K1+1 TO K:FOR J3=K1+1 TO K4
1920 X1(I3,J3)=X1(I3,J3)-X1(K1,J3)*X1(I3,K1)
```

```
1930 NEXT J3:NEXT I3
1940 RETURN
1950 COLOR 7,0:CLS:CLOSE #1
1952 COLOR 25,5,15:PRINT "    РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ KORREG.REZ"
1954 PRINT "    ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПОВТОРНОГО РАСЧЕТА В ФАЙЛЕ  "FAI$
1956 BBB$=INKEY$
1958 IF LEN(BBB$)=0 THEN 1956
1960 END
```

ПРИЛОЖЕНИЕ 5
КОМПЬЮТЕРНАЯ ПРОГРАММА ВЫРАВНИВАНИЯ ДИНАМИЧЕСКОГО РЯДА МНОГОЧЛЕНОМ РЯДА ФУРЬЕ

```
1 COLOR 7,0:CLS:LOCATE 2,60,0:COLOR 25,5,15
2 COLOR 7,7,15:LOCATE 4:COLOR 0,7,15
3 PRINT TAB(27)" ЗАДАЧА   F U R Y E";:PRINT SPC(37):PRINT TAB(26)" ";
4 COLOR 15,0,15:PRINT"КАФЕРА";:COLOR 23,5,15:PRINT"*БНТУ*";
5 COLOR 15,0,15:PRINT" ОАПДД ";:COLOR 7,7,15:PRINT SPC(35):COLOR 1,14
6 PRINT TAB(20)"ВЫРАВНИВАНИЕ   ДИНАМИЧЕСКОГО   РЯДА";
7 PRINT TAB(80)" ";:PRINT TAB(33)"МНОГОЧЛЕНОМ ФУРЬЕ";
8 PRINT TAB(80)" ":DEFINT I-N
19 PRINT:COLOR 7,0,7
25 H2$=" ВЫ ДОПУСТИЛИ ОШИБКУ":F3$="+#.##^^^^":F4$="+#.####^^^^"
27 DIM YS(8640),TS(8640),YT(8640),SAK(8640),SAP(8640)
28 P2=6.2832:M4=0
30 H0$=" ИСХОДНЫЕ ДАННЫЕ":H1$=" ДА - 1, НЕТ - 0"
40 H2$=" ВЫ ДОПУСТИЛИ ОШИБКУ"
45 DIM A(100),B(100),IK(100)
50 PRINT"ВВЕДЕНЫ ЛИ";H0$;H1$:INPUT M0:IF M0=0 THEN 110
60 OPEN"I",#1,"FURXE.DAT"
65 INPUT #1,H3$
70 INPUT #1,N
80 FOR I=1 TO N:INPUT #1,TS(I),YS(I):NEXT I
90 CLOSE #1
100 GOTO 150
110 PRINT"ВВЕДИТЕ ";H0$
115 PRINT"НАИМЕНОВАНИЕ ИСХОДНЫХ ДАННЫХ ";:INPUT H3$
116 'OPEN "I",#3,"CON":INPUT #3,H3$:CLOSE #3
120 PRINT"ЧИСЛО РАСЧЕТНЫХ ТОЧЕК  ";:INPUT N
130 PRINT"ЭМПИРИЧЕСКИЕ ЗНАЧЕНИЯ АРГУМЕНТА, ФУНКЦИИ В РАСЧЕТНЫХ ТОЧКАХ"
140 FOR I=1 TO N:PRINT I"-Я ПАРА ";:INPUT TS(I),YS(I):NEXT I
150 PRINT"ЗНАЧЕНИЯ  ФУНКЦИИ":PRINT "I              YS(I)"
160 FOR I=1 TO N:PRINT I,YS(I):NEXT I
170 PRINT "ТРЕБУЕТСЯ ЛИ КОРРЕКТИРОВАТЬ ЗНАЧЕНИЯ ФУНКЦИИ"
180 PRINT  H1$:INPUT M3:IF M3=1 THEN 250
190 IF M0=1 AND M4=0 THEN 280
200 OPEN"O",#1,"FURXE.DAT"
210 WRITE #1,H3$
215 WRITE #1,N
220 FOR I=1 TO N:WRITE #1,TS(I),YS(I):NEXT I
230 CLOSE #1
240 GOTO 280
250 PRINT"НОМЕР РАСЧЕТНОЙ ТОЧКИ,ДЛЯ КОТОРОЙ ТРЕБУЕТСЯ КОРРЕКТИРОВКА"
260 INPUT IS:M4=1
270 PRINT"ВВЕДИТЕ НОВОЕ ЗНАЧЕНИЕ АРГУМЕНТА, ФУНКЦИИ":INPUT TS(IS),YS(IS):GOTO 170
280 OPEN"O",#3,"FURXE.REZ"
285 SY=0:FOR I=1 TO N:SY=SY+YS(I):NEXT I
290 A0=SY/N:S1=0:FOR I=1 TO N:S1=S1+(YS(I)-A0)^2:NEXT I
300 MM=0:F1=0:E1=1000
320 PRINT"ТРЕБУЕТСЯ НАЙТИ НАИБОЛЕЕ ПОДХОДЯЩУЮ ЗАВИСИМОСТЬ ПО МИНИМУМУ"
330 PRINT"СРЕДНЕЙ ЛИНЕЙНОЙ ОШИБКИ АППРОКСИМАЦИИ - ВВЕДИТЕ 1 ИЛИ"
340 PRINT"ПО МАКСИМУМУ КРИТЕРИЯ ФИШЕРА - ВВЕДИТЕ 2":INPUT M0
341 PRINT "A0=" USING F3$;A0
342 PRINT" K          A(K)        B(K)            E             F"
343 IUUU=INT(N/2):IF N/2-INT(N/2)>.4 THEN IUUU=IUUU+1
344 TT=TS(N)-TS(1)
345 FOR KII=1 TO IUUU
347 M=MM+1:GOSUB 400
350 IF M0=2 THEN 380
360 IF E2>=E1 THEN 395
370 E1=E2:GOTO 392
380 IF F2<=F1 THEN 395
390 F1=F2
392 MM=MM+1:IK(MM)=KII:A(MM)=AK:B(MM)=BK:FOR I=1 TO N:SAK(I)=SAP(I):NEXT I
395 NEXT KII
397 GOTO 550
400 REM
430 K=KII
```

```
435 SAT=0:SBT=0:FOR I=1 TO N:SAT=SAT+YS(I)*COS(P2*K*I/N)
440 SBT=SBT+YS(I)*SIN(P2*K*I/N):NEXT I
450 AK=SAT/N*2:BK=SBT/N*2:PRINT K;
455 PRINT TAB(16) USING F3$;AK;
456 PRINT TAB(30) USING F3$;BK,
470 S2=0:SV=0:FOR I=1 TO N
485 SAP(I)=SAK(I)+AK*COS(P2*K*I/N)+BK*SIN(P2*K*I/N)
490 YT(I)=SAP(I)+A0:IF YT(I)=0 THEN YT(I)=.000001
495 SV=SV+ABS((YT(I)-YS(I))/YT(I))':PRINT I;
496' PRINT TAB(16) USING F3$;YS(I);:PRINT TAB(32) USING F3$;YT(I)
500 S2=S2+(YS(I)-YT(I))^2:NEXT I
502 IF S2=0 THEN S2=0.000001
507 E2=SV/N
510 PRINT TAB(45) USING F3$;E2,
520 F2=(S1-S2)*(N-M-1)/S2/M
522 PRINT TAB(60)USING F3$;F2
525 INPUT AAA
540 RETURN
550 REM
552 PRINT #3,"    РЕЗУЛЬТАТЫ АППРОКСИМАЦИИ ДАННЫХ РЯДОМ ФУРЬЕ"
555 PRINT #3, TAB(5)"(";H3$;")"
560 PRINT #3,:PRINT #3,"ВИД МНОГОЧЛЕНА ОПРЕДЕЛЕН ПО ";
570 IF M0=2 THEN PRINT #3,"МАКСИМУМУ КРИТЕРИЯ ФИШЕРА":GOTO 590
580 PRINT #3,"МИНИМУМУ ЛИНЕЙН.ОШИБКИ АППРОКСИМАЦИИ"
590 PRINT #3,"A0=";:PRINT #3, USING F4$;A0
595 SMAX=0
600 PRINT #3, " K              A(K)         B(K)"
610 FOR K=1 TO MM:PRINT #3,  IK(K);
615 PRINT #3, TAB(16) USING F4$;A(K);
616 PRINT #3, TAB(32) USING F4$;B(K):NEXT K
620 PRINT #3,:PRINT #3," I              YS(I)          YT(I)"
625 SV=0:S2=0
630 FOR I=1 TO N:PRINT #3, I;
 YT(I)=SAK(I)+A0:IF YT(I)=0 THEN YT(I)=.000001
632 IF YS(I)>SMAX THEN SMAX=YS(I)
633 IF YT(I)>SMAX THEN SMAX=YT(I)
635 PRINT #3, TAB(16) USING F4$;YS(I);:PRINT #3, TAB(32) USING F4$;YT(I)
636 SV=SV+ABS((YS(I)-YT(I))/YT(I)):S2=S2+(YS(I)-YT(I))^2:NEXT I
637 IF S2=0 THEN S2=0.000001
638 E2=SV/N
639 F2=(S1-S2)*(N-MM-1)/S2/MM
640 PRINT #3,"СРЕДНЯЯ ЛИНЕЙНАЯ ОШИБКА АППРОКСИМАЦИИ ";:PRINT #3, USING F3$;E2
642 PRINT #3,"КРИТЕРИЙ ФИШЕРА ";:PRINT #3, USING F3$;F2
1360 INPUT"ТРЕБУЮТСЯ ЛИ ПОСТРОЕНИЕ ДИАГРАММЫ (ДА- 1,НЕТ- 0)";IR2
1370 IF IR2=0 THEN 1645
1371 L=0
1372 IF 10^(L-6)>SMAX THEN KKK=CINT(SMAX/10^(L-7))*10^(L-7):GOTO 1375
1373 L=L+1
1374 GOTO 1372
1375 SMAX=56./KKK
1380 PRINT #3,
1381 FOR I=1 TO 5: PRINT #3,TAB(I*11+6)I*KKK/5;:NEXT I
1382     PRINT #3,
1383 PRINT #3,"     +--------------------------------------------------------->";
1385 PRINT #3,
1390 FOR I=1 TO N:P4=CINT(YT(I)*SMAX):P5=CINT(YS(I)*SMAX)
1400 IF P4<P5 THEN 1432
1410 IF P4>P5 THEN 1442
1413 PRINT #3,USING"#####.";TS(I);:PRINT #3,"|";
1414 FOR II=2 TO P4-1:PRINT #3,"-";:NEXT II
1420 PRINT #3,TAB(P4+6)"-"
1428 GOTO 1450
1432 IF P4<2 THEN 1435
1433 PRINT #3,USING"#####.";TS(I);:PRINT #3,"|";
1434 FOR II=2 TO P4-1:PRINT #3,"-";:NEXT II
1435 PRINT #3,"*";
1436 FOR II=P4+1 TO P5:PRINT #3,"-";:NEXT II:PRINT #3,
1439 GOTO 1450
```

```
1442 PRINT #3,USING"#####.";TS(I);:PRINT #3,"│";
1443 FOR II=2 TO P5:PRINT #3,"-";:NEXT II
1445 PRINT #3,TAB(P4+6)"*"
1450 NEXT I
1455 PRINT #3,"       │"
1456 PRINT #3,"       V"
1460 PRINT #3,"* ТЕОРЕТИЧЕСКИЕ ЗНАЧЕНИЯ, - ЭКСПЕРИМЕНТАЛЬНЫЕ ЗНАЧЕНИЯ И ПРИ СОВПАДЕНИИ"
1480 PRINT #3,"                    С ТЕОРЕТИЧЕСКИМИ)"
1485 CLOSE #3
1645 PRINT"ТРЕБУЕТСЯ ЛИ ПРОДОЛЖИТЬ РАСЧЕТЫ ?":INPUT M5
1646 IF M5=1 THEN 170
1647 COLOR 25,5,15:PRINT "   РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ FURYE.REZ"
1648 PRINT "   ИСХОДНЫЕ ДАННЫЕ ДЛЯ ПОВТОРНОГО РАСЧЕТА В ФАЙЛЕ FURYE.DAT"
1649 BBB$=INKEY$
1650 IF LEN(BBB$)=0 THEN 1649
1660 END
```

ПРИЛОЖЕНИЕ 6
КОМПЬЮТЕРНАЯ ПРОГРАММА РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ СИМПЛЕКС-МЕТОДОМ

```
'Программа решения задачи линейного программирования симплекс-методом
 color 7, 7, 15: cls : locate 10: print : color 0, 7, 15:defint i-n
 print tab(28); "Решение задачи линейного программирования симплекс-методом"
 print tab(27); " ";
 color 15, 0, 15: print "Кафедра"; : color 23, 5, 15: print "*БНТУ*";
 color 15, 0, 15: print " О А П Д Д": print : color 1, 14
delay 1
   print : color 7, 0, 7: cls
FM$="#####.###"
 print"Введите исходные данные"
 PRINT "Вид экстремума, max - ввести 1, min - ввести 0":INPUT IMX
IF IMX=1 THEN IMX=-1 ELSE IMX=1
INPUT "Число уравнений в системе ограничений типа >=";IGC
INPUT "Число уравнений в системе ограничений типа =";IEC
INPUT "Число уравнений в системе ограничений типа <= ";LC
 INPUT "Число неизвестных";N
 MM=IGC+IEC:M=MM+LC:MK=IGC+LC:N1=MK+N
IP=N1+MM:M1=M+1:M2=M+2:N0=N1
DIM A(M2,IP),IBS(M),V(M2),NB(IP),SL(IP),XR(M),ZXP(M)
FOR I=1 TO IGC:FOR J=1 TO N
PRINT "Коэффициент при "J"-м неизвестном в "I"-м уравнении типа ?a(";
PRINT I","J")":INPUT A(I,J)
NEXT J
PRINT "Свободный член этого уравнения b ("I")":INPUT A(I,0)
NEXT I
FOR I=IGC+1 TO MM:FOR J=1 TO N
PRINT "Коэффициент при "J"-м неизвестном в "I-IGC"-м уравнении типа = a(";
PRINT I-IGC","J")":INPUT A(I,J)
NEXT J
PRINT "Свободный член этого уравнения b ("I-IGC")":INPUT A(I,0)
NEXT I
FOR I=MM+1 TO M:FOR J=1 TO N
PRINT "Коэффициент при "J"-м неизвестном в "I-MM"-м уравнении типа ? a(";
PRINT I-MM","J")":INPUT A(I,J)
NEXT J
PRINT "Свободный член этого уравнения b ("I-MM")":INPUT A(I,0)
NEXT I
 FOR J=1 TO N:PRINT "Коэффициент целевой функции при переменной "J")"
 INPUT A(M1,J)
     NEXT J
 OPEN"O",#1,"SIMPL.REZ"
 PRINT #1, TAB(4)"РЕШЕНИЕ ОБЩЕЙ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ"
 PRINT #1, TAB(13)"СИМПЛЕКС-МЕТОДОМ"
 PRINT #1,: PRINT #1, TAB(4)"Исходные данные":PRINT #1,
 PRINT #1, TAB(3)"Уравнения системы ограничений"
FOR  I=1 TO M
 FOR J=1 TO N
IF J>1 AND A(I,J)>=0 THEN PRINT  #1, " +";
PRINT #1, USING FM$;A(I,J);:PRINT #1,"x("j")";
NEXT J
IF I<=IGC THEN PRINT #1,">=";:GOTO 55
IF I>IGC AND I<=MM THEN PRINT #1,"=";:GOTO 55
 PRINT #1,"<=";
55 PRINT #1, USING FM$;A(I,0)
NEXT I
  PRINT #1, :PRINT #1, TAB(3)"Целевая функция"
FOR J=1 TO N
IF J>1 AND A(M1,J)>=0 THEN PRINT #1," +";
PRINT #1, USING FM$;A(M1,J);:PRINT #1,"x("j")";:PRINT #1,"  ";
NEXT J
IF IMX=-1 THEN PRINT #1,"= max" else PRINT #1,"= min"
GOSUB SIMPL
goto 7000
SIMPL:
 FOR J=1 TO N:A(M1,J)=IMX*A(M1,j)
```

```
      NEXT J
   IF IGC=0 THEN 150
FOR I=1 TO IGC
 A(I,N+I)=-1: A(I,N1+I)=1
IBS(I)=N1+I
 NEXT I
150 IF IEC=0 THEN 200
 FOR I=IGC+1 TO MM
A(I,N1+I)=1:IBS(I)=N1+I
NEXT I
200 IF LC=0 THEN 240
 FOR I=MM+1 TO M
A(I,N+I-IEC)=1:IBS(I)=N+I-IEC
NEXT I
240 IF MM=0 THEN 400
   L=1:N0=IP
FOR I=1 TO MM:FOR J=0 TO N1
A(M2,J)=A(M2,J)-A(I,J)
 NEXT J
NEXT I
    ML=M1+L
FOR I=1 TO M:NB(IBS(I))=1:NEXT I
400 ZERO=1E-3
500   SMIN=-ZERO:IS=0:PV=0:ML=M1+L
FOR J=1 TO N0
IF NB(J)=1 THEN 550
IF A(ML,J)>=SMIN THEN 550
SMIN=A(ML,J):IS=J
550 NEXT J
IF IS=0 THEN 1900
BMIN=1E12:IR=0
FOR I=1 TO M
IF A(I,IS)<=ZERO THEN 810
RT=A(I,0)/A(I,IS)
IF RT>=BMIN THEN 810
IR=I:BMIN=A(I,0)/A(I,IS)
 810 NEXT I
IF IR=0 THEN 1800
PV=A(IR,IS)
FOR J=0 TO N0
A(IR,J)=A(IR,J)/PV
NEXT J
FOR I=1 TO ML: V(I)=A(I,IS):NEXT I
FOR I=1 TO ML
 IF I=IR THEN 1120
FOR J=0 TO N0
A(I,J)=A(I,J)-V(I)*A(IR,J)
 NEXT J
1120 NEXT I
NB(IBS(IR))=0:NB(IS)=1:IBS(IR)=IS
K=K+1
 GOTO 500
1800 PRINT "ПЕРЕМЕННАЯ "S" НЕ ИМЕЕТ ОГРАНИЧЕНИЙ":GOTO 4000
 1900 IF L=0 THEN 2000
  IF ABS(A(ML,0))>=ZERO THEN 1960
L=0:N0=N1
GOTO 500
1960 PRINT "ОГРАНИЧЕНИЯ НЕ ИМЕЮТ ДОПУСТИМОГО РЕШЕНИЯ"
 GOTO 4000
2000 PRINT #1,: PRINT #1, "  Оптимальное решение"
FOR J=1 TO N
K=0
FOR I=1 TO M
IF J<>IBS(I) THEN 2005
PRINT #1,"    ";"x("J")=";using FM$;A(I,0)
K=K+1
2005 NEXT I
IF K=0 THEN PRINT #1,"    ";"x("J")=";using FM$;0
```

```
  NEXT J
Z=-IMX*A(M1,0):PRINT  #1, " Zopt= ";using FM$;Z
4000 RETURN
7000 PRINT #1, :PRINT #1, TAB(5)".....***....."
CLOSE #1
 COLOR 25,5,15:PRINT "РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ  SIMPL.REZ"
7806 BBB$=INKEY$
 IF LEN(BBB$)=0 THEN 7806
 END
```

ПРИЛОЖЕНИЕ 7
КОМПЬЮТЕРНАЯ ПРОГРАММА ОТЫСКАНИЯ КРАТЧАЙШИХ РАССТОЯНИЙ МЕЖДУ ПУНКТАМИ ТРАНСПОРТНОЙ СЕТИ

```
2 COLOR 7,7,15:CLS:LOCATE 10:PRINT:COLOR 0,7,15
3 PRINT TAB(21)"З А Д А Ч А   K R T R S T":PRINT TAB(20)" ";
4 COLOR 15,0,15:PRINT"КАФЕДРА";:COLOR 23,0,15:PRINT" *БНТУ*";
5 COLOR 15,0,15:PRINT" О А П Д Д":PRINT
6 PRINT"ПРОГРАММА РАСЧЕТА КРАТЧАЙШИХ РАССТОЯНИЙ"
8 PRINT"МЕЖДУ ПУНКТАМИ ТРАНСПОРТНОЙ СЕТИ":COLOR 7,7,15
19 PRINT:COLOR 7,0,7:DEFINT I-N
30 DIM C(50,50),IP(50,50),AG$(50)
40 H0$="   ИСХОДНЫЕ ДАННЫЕ":H1$=" ДА - 1, НЕТ - 0"
50 H2$="КОРРЕКТИРОВАТЬ ЛИ ":H3$="ВЫ ДОПУСТИЛИ ОШИБКУ"
60 PRINT "ВВЕДЕНЫ ЛИ "H0$;H1$:INPUT M0:IF M0=0 THEN 160
70 OPEN"I",#1,"KRTRST.DAT"
80 INPUT #1,N,MM
90 FOR I=1 TO N:INPUT #1,AG$(I)
100 FOR J=1 TO N:IF MM=0 AND J>I-1 THEN 130
110 IF I=J THEN 130
120 INPUT #1,C(I,J)
130 NEXT J:NEXT I
140 CLOSE #1
150 CLS: PRINT TAB(5)H0$:GOSUB 460:GOTO 280
160 CLS:PRINT"ВВЕДИТЕ ";H0$:INPUT "ЧИСЛО ПУНКТОВ(ВЕРШИН)";N
170 PRINT"ПОСЛЕДОВАТЕЛЬНО СОКРАЩЕННЫЕ НАИМЕН.(КОДЫ) ПУНКТОВ"
180 FOR I=1 TO N:PRINT I"-ГО":INPUT AG$(I):NEXT I
190 INPUT"ИМЕЮТСЯ ЛИ ДОРОГИ С ОДНОСТОРОННИМ ДВИЖЕНИЕМ. ДА- 1,НЕТ-0";MM
200 PRINT"ВВЕДИТЕ ДЛИНЫ ЗВЕНЬЕВ ТРАНСПОРТНОЙ СЕТИ. ЕСЛИ ЗВЕНО "
210 PRINT"ОТСУСТВУЕТ, ТО ВВОДИТЕ( 0 )"
220 FOR I=1 TO N:FOR J=1 TO N:IF MM=0 AND J>I-1 THEN 260
230 IF I=J THEN 260
240 PRINT "C("AG$(I)"."AG$(J)")":INPUT C(I,J)
250 IF C(I,J)=0 THEN C(I,J)=1E+20 ELSE P(I,J)=0
260 NEXT J:NEXT I
270 CLS:GOSUB 460
280 PRINT H2$;H0$:PRINT H1$;:INPUT M1:IF M1=0 THEN 350
290 PRINT"ЧЕРЕЗ ЗАПЯТУЮ ПОРЯДКОВЫЕ НОМЕРА(ИНДЕКСЫ) ПУНКТОВ,"
300 MR=1:PRINT "МЕЖДУ КОТОРЫМИ ТРЕБУЕТСЯ СКОРРЕКТИРОВАТЬ РАССТОЯНИЕ"
310 INPUT IS,JS
320 IF IS<1 OR JS>N OR JS<1 OR JS>N THEN PRINT H3$:GOTO 290
330 IF MM=0 AND JS>IS-1 THEN PRINT H3$:GOTO 290
340 PRINT"НОВОЕ ЗНАЧЕНИЕ РАССТОЯНИЯ":INPUT C(IS,JS):GOTO 280
350 IF M0=1 AND MR=0 THEN 560
360 OPEN "O",#1,"KRTRST.DAT"
370 WRITE #1,N,MM
380 FOR I=1 TO N:WRITE #1,AG$(I)
390 FOR J=1 TO N:IF MM=0 AND J>I-1 THEN 410
400 WRITE #1,C(I,J)
410 NEXT J:NEXT I
420 CLOSE #1
430 IF MR=0 THEN 560
440 PRINT "СКОРРЕКТИРОВАННЫЕ";H0$
450 GOSUB 460:GOTO 560
460 PRINT H0$:PRINT:PRINT"   ДЛИНА ЗВЕНЬЕВ ТРАНСПОРТНОЙ СЕТИ"
470 PRINT "    J";:FOR J=1 TO N:PRINT TAB(6*J+6)AG$(J);:NEXT J:PRINT
480 FOR J=1 TO N:PRINT TAB(6*J+4)"*"J;:NEXT J:PRINT
490 PRINT " I"
500 FOR I=1 TO N:PRINT AG$(I);"*";I;
510 FOR J=1 TO N
520 IF C(I,J)>1E+19 THEN PRINT TAB(6*J+6)" - ";:GOTO 540
525 IF MM=0 AND J>I-1 THEN 540
530 PRINT TAB(6*J+6) C(I,J);
540 NEXT J:PRINT
550 NEXT I:PRINT :PRINT :RETURN
560 IF MM=1 THEN 580
570 FOR I=1 TO N:FOR J=I+1 TO N:C(I,J)=C(J,I):NEXT J:NEXT I
580 FOR I=1 TO N:PRINT"ИДЕТ "I"-Я ИТЕРАЦИЯ"
585 FOR J=1 TO N:IF C(I,J)>1E+19 THEN 640
```

```
590 FOR K=1 TO N
600 IF C(I,K)>1E+19 THEN 630
610 S=C(J,I)+C(I,K)
620 IF S<C(J,K) THEN C(J,K)=S:IP(J,K)=I
630 NEXT K
640 NEXT J:NEXT I
750 OPEN "O",#1,"KRTRST.REZ"
755 WRITE #1,N,MM
760 FOR I=1 TO N
765 WRITE #1,AG$(I)
770 FOR J=1 TO N
775 WRITE #1,C(I,J),IP(I,J)
780 NEXT J
785 NEXT I
790 CLOSE #1
795 OPEN "O",#1,"KRTRSTP.REZ"
800 PRINT #1," КРАТЧАЙШИЕ РАССТОЯНИЯ МЕЖДУ ПУНКТАМИ"
805 FOR I=1 TO N
810 PRINT #1,"От "AG$(I)" до"
815 FOR J=1 TO N
820 PRINT #1,AG$(J),USING"######.###";C(I,J),
825 IF IP(I,J)=0 THEN PRINT #1, "" else PRINT #1," через "AG$(IP(I,J))
830 NEXT J
835 NEXT I
840 CLOSE #1
850 COLOR 25,5,15:PRINT "   РЕЗУЛЬТАТЫ РАСЧЕТА ДЛЯ ИСПОЛЬЗОВАНИЯ В ДРУГИХ ПРОГРАММАХ В ФАЙЛЕ
KRTRST.REZ"
860 PRINT "   РЕЗУЛЬТАТЫ РАСЧЕТА ДЛЯ ВЫВОДА НА ПЕЧАТЬ В ФАЙЛЕ KRTRST.PRN"
865 PRINT "   ДАННЫЕ ДЛЯ ПОВТОРНОГО РАСЧЕТА В ФАЙЛЕ KRTRST.DAT"
870 BBB$=INKEY$
880 IF LEN(BBB$)=0 THEN 870
910 END
```

ПРИЛОЖЕНИЕ 8
КОМПЬЮТЕРНАЯ ПРОГРАММА РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

```
'МОДУЛЬ ГОЛОВНОЙ ПРОГРАММЫ
1 COLOR 7,0:CLS:LOCATE 2,60,0:COLOR 25,5,15
2 COLOR 7,7,15:LOCATE 3:COLOR 0,7,15
3 PRINT TAB(27)"ЗАДАЧА    T R A N";:PRINT SPC(37):PRINT TAB(26)" ";
4 COLOR 15,0,15:PRINT"КАФЕДРА";:COLOR 23,5,15:PRINT"*БНТУ*";
5 COLOR 15,0,15:PRINT" ОАПДД ";:COLOR 7,7,15:PRINT SPC(34):COLOR 1,14
6 PRINT TAB(22)"ОПТИМИЗАЦИЯ ПОСТАВОК РЕСУРСА (ТЗЛП)";
8 PRINT TAB(80)" ":DEFINT I-N
19 PRINT:COLOR 7,0,7:CLS
35 COLOR 1,6,14
  IMAX=3
  I=1
  LOCATE 8,30,0:COLOR 25,4:PRINT"ВВЕДИТЕ РЕЖИМ"
50    J=1
    IF J=I THEN COLOR 4,11 ELSE COLOR 1,11
    LOCATE 9,25,0:PRINT " ╔══════════════════════════════════╗ "
    LOCATE 10,25,0:PRINT"║ ВВОД И КОРРЕКТИРОВКА              ║"
    LOCATE 11,25,0:PRINT"║    ИСХОДНЫХ ДАННЫХ                ║"
    LOCATE 12,25,0:PRINT"╚══════════════════════════════════╝ "
    J=2
     IF J=I THEN COLOR 4,11 ELSE COLOR 1,11
     LOCATE 13,25,0:PRINT"╔══════════════════════════════════╗ "
     LOCATE 14,25,0:PRINT"║  ВЫПОЛНЕНИЕ РАСЧЕТОВ              ║"
     LOCATE 15,25,0:PRINT"║                                   ║"
     LOCATE 16,25,0:PRINT"╚══════════════════════════════════╝ "
      J=3
      IF J=I THEN COLOR 4,11 ELSE COLOR 1,11
      LOCATE 21,25,0:PRINT"╔══════════════════════════════════╗ "
      LOCATE 22,25,0:PRINT"║     ** ВЫХОД   **                 ║"
      LOCATE 23,25,0:PRINT"╚══════════════════════════════════╝ "
95 AM$=INKEY$
  IF LEN(AM$)=0 THEN 95
  AM1$=RIGHT$(AM$,1)
  IF ASC(AM1$)=80 THEN I=I+1:GOTO 100
  IF ASC(AM1$)=72 THEN I=I-1:GOTO 100
  IF ASC(AM1$)=13 GOTO 110
 GOTO 95
100 IF I>IMAX THEN I=1
   IF I<1 THEN I=IMAX
 GOTO 50
110 COLOR 1,11
   IF I=1 THEN SHELL "TRANW.EXE":GOTO 1
   IF I=2 THEN SHELL"TRANR.EXE":GOTO 1
120 END

rem МОДУЛЬ ПРОГРАММЫ ВВОДА ДАННЫХ
2 COLOR 7, 7, 15: CLS : LOCATE 10: PRINT : COLOR 0, 7, 15: DEFINT I-N
3 PRINT TAB(28); "З А Д А Ч А   T  R  A  N": PRINT TAB(27); " ";
4 COLOR 15, 0, 15: PRINT "КАФЕДРА"; : COLOR 23, 5, 15: PRINT "*БНТУ*";
5 COLOR 15, 0, 15: PRINT " О А П Д Д": PRINT : COLOR 1, 14
6 PRINT TAB(24); "ЗАДАЧА ОПТИМИЗАЦИИ  ПОСТАВОК РЕСУРСА (ТЗЛП)";
8 PRINT TAB(80); " "; : PRINT TAB(20); "*** ВВОД, ВЫВОД И КОРРЕКТИРОВКА ";
9 PRINT "ДАННЫХ ***"; : PRINT TAB(80);" "; : COLOR 7, 10, 15
DELAY 3
19 PRINT : COLOR 7, 0, 7: CLS : GOTO 100
70 DIM A(M + 1), B(N + 1), C(M + 1, N + 1), AG$(M), BG$(N)
80 IF M3 = 1 THEN DIM D(M + 1, N + 1)
90 RETURN
100 FM$ = "####.#":IRAST=0
105 H0$ = "ВЫ ДОПУСТИЛИ ОШИБКУ": H1$ = "ЗАПАСЫ У ПОСТАВЩИКОВ"
110 H2$ = "ПОТРЕБНОСТИ У ПОТРЕБИТЕЛЕЙ"
120 H3$ = "СТОИМОСТИ(РАССТОЯНИЯ,ВРЕМЯ) ПЕРЕВОЗКИ ЕДИНИЦЫ ГРУЗА"
130 H4$ = "ПОСТ.       ПОТРЕБИТЕЛИ"
140 H5$ = "ОГРАНИЧЕНИЯ НА РАЗМЕР КОРРЕСПОНДЕНЦИЙ"
150 H6$ = "МЕЖДУ ПОСТАВЩИКАМИ И ПОТРЕБИТЕЛЯМИ": H7$ = "КОРРЕКТИРОВАТЬ"
```

```
160 H8$ = "ИСХОДНЫЕ ДАННЫЕ"
170 H9$ = "ЧЕРЕЗ ЗАПЯТУЮ ПОРЯДКОВЫЕ НОМЕРА|(ИНДЕКСЫ) ПОСТАВЩИКА И"
180 P0$ = "ПОТРЕБИТЕЛЯ ДЛЯ КОТОРЫХ ТРЕБУЕТСЯ КОРРЕКТИРОВКА"
190 P1$ = "ПОСЛЕДОВАТЕЛЬНО СОКРАЩЕН.НАИМЕНОВАНИЯ(КОДЫ)"
200 P2$ = "ПОРЯДКОВЫЙ НОМЕР (ИНДЕКС) ПУНКТА (СЧИТАЯ С ЕДИНИЦЫ) ДЛЯ"
210 P3$ = "КОТОРОГО ТРЕБУЕТСЯ КОРРЕКТИРОВКА"
220 P4$ = "НОВАЯ ВЕЛИЧИНА ": P5$ = " БУДЕТ КОРРЕКТИРОВАТЬСЯ "
225 P6$ = " КОРРЕКТИРОВАТЬ ? ДА - 1, НЕТ - 0"
226 P7$ = "< УКАЖИТЕ СПЕЦИФИКАЦИЮ ФАЙЛА ИСХОДНЫХ ДАННЫХ"
227 P8$ = " ЧИСЛО ПОСТАВЩИКОВ  ": P9$ = " ЧИСЛО ПОТРЕБИТЕЛЕЙ  "
228 S0$ = " НАЛИЧИЕ ОГРАНИЧЕНИЙ НА РАЗМЕР КОРРЕСПОНДЕНЦИЙ(ДА- 1,НЕТ- 0) "
230 PRINT "ВВЕДЕНЫ ЛИ ПОЛНЫЕ "; H8$; ".ДА- 1,НЕТ- 0"; : INPUT M0
232 IF M0 = 0 THEN 820
233 PRINT : PRINT P7$; : INPUT FAILW$
235 OPEN "I", #1, FAILW$
237 INPUT #1, M, N, M3
240 GOSUB 70
244 FOR I = 1 TO M: INPUT #1, AG$(I), A(I): NEXT I
248 FOR J = 1 TO N: INPUT #1, BG$(J), B(J): NEXT J
252 FOR I = 1 TO M: FOR J = 1 TO N: INPUT #1, C(I, J)
253 NEXT J: NEXT I
254 IF M3 = 0 THEN 260
258 FOR I = 1 TO M: FOR J = 1 TO N: INPUT #1, D(I, J)
259 NEXT J: NEXT I
260 CLOSE #1
340 MN = M: NN = N
350 PRINT TAB(10); H8$: PRINT TAB(5); H1$
360 PRINT " I  "; : FOR I = 1 TO MN: PRINT TAB(6 * I); AG$(I); : NEXT I: PRINT
370 PRINT "Q(I)"; : FOR I = 1 TO MN: PRINT TAB(6 * I); USING FM$; A(I); : NEXT I: PRINT
380 PRINT TAB(5); H2$
390 PRINT " J  "; : FOR J = 1 TO NN: PRINT TAB(6 * J); BG$(J); : NEXT J: PRINT
400 PRINT "Q(J)"; : FOR J = 1 TO NN: PRINT TAB(6 * J); USING FM$; B(J); : NEXT J: PRINT
410 PRINT TAB(5); H3$: PRINT
420 PRINT H4$
430 FOR J = 1 TO NN: PRINT TAB(6 * J); BG$(J); : NEXT J: PRINT
440 FOR I = 1 TO MN: PRINT AG$(I); : FOR J = 1 TO NN: PRINT TAB(6 * J); USING FM$; C(I, J);
450 NEXT J: PRINT : NEXT I: PRINT
460 IF M3 = 0 THEN 520
470 PRINT TAB(5); H5$: PRINT
480 PRINT H4$
490 FOR J = 1 TO NN: PRINT TAB(6 * J); BG$(J); : NEXT J: PRINT
500 FOR I = 1 TO MN: PRINT AG$(I); : FOR J = 1 TO NN: PRINT TAB(6 * J); USING FM$; D(I, J);
510 NEXT J: PRINT : NEXT I: PRINT
520 COLOR 0, 7: PRINT H7$; "  "; H8$; ". ДА- 1,НЕТ- 0"; : COLOR 7, 0
530 INPUT M8
540 IF M0 = 0 AND M8 = 0 THEN 1020
550 IF M8 = 0 THEN 1100
560 M0 = 0
570 PRINT H7$; "  "; H1$; ".ДА- 1,НЕТ- 0"; : INPUT M5
580 IF M5 = 0 THEN 620
590 PRINT P2$: PRINT P3$; " ЗАПАСА"; : INPUT NS
600 IF NS < 1 OR NS > M THEN PRINT H0$: BEEP: GOTO 590
605 PRINT P5$; USING FM$; "QA("; AG$(NS); ")"; P6$; : INPUT ME
606 IF ME = 0 THEN 570
610 PRINT P4$; " ЗАПАСА"; : INPUT A(NS): GOTO 570
620 PRINT H7$; "  "; H2$; ".ДА- 1,НЕТ- 0"; : INPUT M6
630 IF M6 = 0 THEN 670
640 PRINT P2$: PRINT P3$; " ПОТРЕБНОСТИ"; : INPUT JS
650 IF JS < 1 OR JS > M THEN PRINT H0$: BEEP: GOTO 640
655 PRINT P5$; USING FM$; "QB("; BG$(JS); ")"; P6$; : INPUT ME
656 IF ME = 0 THEN 620
660 PRINT P4$; " ПОТРЕБНОСТИ"; : INPUT B(JS): GOTO 620
670 PRINT H7$; "  "; H3$
680 PRINT "C(I,J).ДА- 1,НЕТ- 0";
690 INPUT M7: IF M7 = 0 THEN 750
700 PRINT H9$: PRINT P0$; " C(I,J)";
710 INPUT NS, JS
720 IF NS < 1 OR JS < 1 OR NS > M OR JS > N THEN PRINT H0$: BEEP: GOTO 700
```

```
725 PRINT P5$; USING FM$; "C("; AG$(NS); ","; BG$(JS); ")"; P6$; : INPUT ME
726 IF ME = 0 THEN 670
730 PRINT P4$; " СТОИМОСТИ(РАСТОЯНИЯ,ВРЕМЕНИ)"; : INPUT C(NS, JS)
740 GOTO 670
750 IF M3 = 0 THEN 350
760 PRINT H7$; " ОГРАНИЧЕНИЯ D(I,J).ДА- 1,НЕТ- 0";
770 INPUT M2: IF M2 = 0 THEN 350
780 PRINT H9$: PRINT P0$; " D(I,J)";
790 INPUT NS, JS
800 IF NS < 1 OR JS < 1 OR NS > M OR JS > N THEN PRINT H0$: BEEP: GOTO 780
805 PRINT P5$; USING FM$; "D("; AG$(NS); ","; BG$(JS); ")"; P6$; : INPUT ME
806 IF ME = 0 THEN 760
810 PRINT P4$; " ОГРАНИЧЕНИЯ"; : INPUT D(NS, JS): GOTO 760
820 PRINT "ИМЕЕТСЯ ФАЙЛ С КРАТЧ.РАССТОЯНИМИ.НЕТ - N ,ДА- Y"; : AAP$ = INPUT$(1)
   IF AAP$ = "N" OR AAP$ = "N" OR AAP$ = "T" OR AAP$ = "T" THEN 825
  rem PRINT "ВВЕДИТЕ СПЕЦИФИКАЦИЮ ФАЙЛА С КРАТЧАЙШИМИ РАССТОЯНИМИ";
  rem INPUT FAAA$
   IRAST=1
  PRINT S0$; : INPUT M3
   OPEN "I", #1, "KRTRST.REZ": INPUT #1, N, AAA: M = N
   GOSUB 70
 FOR I = 1 TO N
   INPUT #1, AG$(I): BG$(I) = AG$(I)
  FOR J = 1 TO N
  INPUT #1, C(I, J), AAA
  NEXT J
  NEXT I
  CLOSE #1
    GOTO 890
825 PRINT:PRINT P8$; : INPUT M
830 PRINT P9$; : INPUT N
835 PRINT S0$; : INPUT M3
840 GOSUB 70
850 PRINT P1$; " ПОСТАВЩИКОВ"
860 FOR I = 1 TO M: PRINT I; "-ГО"; : INPUT AG$(I): NEXT I
870 PRINT P1$; " ПОТРЕБИТЕЛЕЙ"
880 FOR J = 1 TO N: PRINT J; "-ГО"; : INPUT BG$(J): NEXT J
890 PRINT H1$
900 FOR I = 1 TO M: PRINT AG$(I); : INPUT A(I): NEXT I
910 PRINT H2$
920 FOR J = 1 TO N: PRINT BG$(J); : INPUT B(J): NEXT J
   IF IRAST=1 THEN 960
930 PRINT H3$: PRINT H6$
940 FOR I = 1 TO M: FOR J = 1 TO N: PRINT "C("; AG$(I); ","; BG$(J); ")";
950 INPUT C(I, J): NEXT J: NEXT I
960 IF M3 = 0 THEN 1010
970 PRINT H5$: PRINT H6$
980 FOR I = 1 TO M
985 FOR J = 1 TO N: PRINT "D("; AG$(I); ","; BG$(J); ")";
988 IF A(I) = 0 OR B(J) = 0 THEN D(1, J) = 0: GOTO 990
990 INPUT D(I, J)
1000 NEXT J
1005 NEXT I
1010 GOTO 340
1020 PRINT : PRINT P7$; : INPUT FAILW$
1022 OPEN "O", #1, FAILW$
1025 WRITE #1, M, N, M3
1028 FOR I = 1 TO M: WRITE #1, AG$(I), A(I): NEXT I
1030 FOR J = 1 TO N: WRITE #1, BG$(J), B(J): NEXT J
1035 FOR I = 1 TO M: FOR J = 1 TO N: WRITE #1, C(I, J)
1040 NEXT J: NEXT I
1045 IF M3 = 0 THEN 1070
1050 FOR I = 1 TO MN: FOR J = 1 TO NN: WRITE #1, D(I, J)
1060 NEXT J: NEXT I
1070 CLOSE #1
1100 PRINT : INPUT " ВЫВОДИТЬ  ИСХОДНЫЕ ДАННЫЕ В ФАЙЛ ДЛЯ ПЕЧАТИ.ДА - 1,НЕТ - 0 "; INDI
1105 IF INDI = 0 THEN 1275
1110 OPEN"O",#4,"DANTRZAD.PRN"
```

```
1111 PRINT #4, TAB(5); H8$: PRINT #4,
1114 PRINT #4, TAB(5); P8$; M
1115 PRINT #4, TAB(5); P9$; N
1116 PRINT #4, TAB(5); S0$; M3
1118 PRINT #4, TAB(5); H1$
1120 PRINT #4, " I  "; : FOR I = 1 TO MN: PRINT #4, TAB(6 * I); AG$(I); : NEXT I: PRINT #4,
1130 PRINT #4, : PRINT #4, "Q(I)"; : FOR I = 1 TO MN: PRINT #4, TAB(6 * I); USING FM$; A(I); : NEXT I
1140 PRINT #4, : PRINT #4, TAB(5); H2$
1150 PRINT #4, " J "; : FOR J = 1 TO NN: PRINT #4, TAB(6 * J); BG$(J); : NEXT J: PRINT #4,
1160 PRINT #4, : PRINT #4, "Q(J)"; : FOR J = 1 TO NN: PRINT #4, USING FM$; B(J); : NEXT J
1170 PRINT #4, : PRINT #4, : PRINT #4, TAB(5); H3$: PRINT #4,
1180 PRINT #4, H4$
1190 FOR J = 1 TO NN: PRINT #4, TAB(6 * J); BG$(J); : NEXT J: PRINT #4, : PRINT #4,
1200 FOR I = 1 TO MN: PRINT #4, AG$(I); : FOR J = 1 TO NN: PRINT #4, USING FM$; C(I, J);
1210 NEXT J: PRINT #4, : NEXT I: PRINT #4, : PRINT #4,
1220 IF M3 = 0 THEN 1275
1230 PRINT #4, TAB(5); H5$: PRINT #4,
1240 PRINT #4, H4$
1250 FOR J = 1 TO NN: PRINT #4, TAB(6 * J); BG$(J); : NEXT J: PRINT #4, : PRINT #4,
1260 FOR I = 1 TO MN: PRINT #4, AG$(I); : FOR J = 1 TO NN: PRINT #4, USING FM$; D(I, J);
1270 NEXT J: PRINT #4, : NEXT I: PRINT #4,
1272 CLOSE #4
1275 COLOR 25,5,15:PRINT "   ИСХОДНЫЕ ДАННЫЕ ДЛЯ РАСЧЕТА В ФАЙЛЕ  "FAILW$
1276 PRINT "   ИСХОДНЫЕ ДАННЫЕ ДЛЯ ВЫВОДА НА ПЕЧАТЬ В ФАЙЛЕ DANTRZAD.PRN"
1277 BBB$=INKEY$
1278 IF LEN(BBB$)=0 THEN 1277
1280 END

rem МОДУЛЬ ПРОГРАММЫ ВЫПОЛНЕНИЯ РАСЧЕТОВ
2 COLOR 7,7,15:CLS:LOCATE 10:PRINT:COLOR 0,7,15:DEFINT I-N
3 PRINT TAB(28)"З А Д А Ч А   T  R  A  N ":PRINT TAB(27)" ";
4 COLOR 15,0,15:PRINT"КАФЕДРА";:COLOR 23,5,15:PRINT"*БНТУ*";
5 COLOR 15,0,15:PRINT" О А П Д Д":PRINT:COLOR 1,14
6 PRINT TAB(24)"ЗАДАЧА ОПТИМИЗАЦИИ ПОСТАВОК РЕСУРСА";
8 PRINT TAB(80)" ";:PRINT TAB(20)"***ВЫПОЛНЕНИЕ РАСЧЕТОВ И ВЫВОД РЕЗУЛЬТАТОВ***";
9 PRINT TAB(80)" ";:COLOR 7,10,15
19 PRINT:COLOR 7,0,7
60 GOTO 226
70 DIM A(M+1),B(N+1),C(M+1,N+1),AG$(M),BG$(N)
80 IF M3=1 THEN DIM D(M+1,N+1)
90 RETURN
226 P7$="< УКАЖИТЕ СПЕЦИФИКАЦИЮ ФАЙЛА ИСХОДНЫХ ДАННЫХ"
228 PRINT:PRINT P7$:INPUT FAILW$
250 OPEN"I",#1,FAILW$
260 INPUT #1,M,N,M3
270 GOSUB 70
280 FOR I=1 TO M:INPUT #1,AG$(I),A(I):NEXT I
290 FOR J=1 TO N:INPUT #1,BG$(J),B(J):NEXT J
300 FOR I=1 TO M:FOR J=1 TO N:INPUT #1,C(I,J):NEXT J:NEXT I
310 IF M3=0 THEN 330
320 FOR I=1 TO M:FOR J=1 TO N:INPUT #1,D(I,J):NEXT J:NEXT I
330 CLOSE #1
340 MN=M:NN=N
1280 REM
1290 REM ПРОВЕРКА БАЛАНСА
1300 AZ=0:FOR I=1 TO M:AZ=AZ+A(I):NEXT I
1310 BZ=0:FOR J=1 TO N:BZ=BZ+B(J):NEXT J
1320 IF AZ=BZ THEN 1380
1330 IF AZ>BZ THEN 1360
1340 M=M+1:A(M)=BZ-AZ:FOR J=1 TO N:C(M,J)=0:IF M3=1 THEN D(M,J)=1E+10
1350 NEXT J:GOTO 1380
1360 N=N+1:B(N)=AZ-BZ:FOR I=1 TO M:C(I,N)=0:IF M3=1 THEN D(I,N)=1E+10
1370 NEXT I
1380 DIM DYNAMIC DA(M),DB(N),IR(M),IC(N),JR(M),JC(N)
1390 DIM DYNAMIC U(M),V(N),IU(M),IV(N),IT(M+N),JT(M+N)
1400 DIM DYNAMIC X(M,N),IX(M,N),MM(M,N)
1410 IF M3=1 THEN DIM DYNAMIC CP(M,N),DK(M,N)
1420 IF M3=0 THEN 1440
```

```
1430 FOR I=1 TO M:FOR J=1 TO N:CP(I,J)=C(I,J):NEXT J:NEXT I
1440 FOR I=1 TO M:DA(I)=A(I):IR(I)=0:JR(I)=0:NEXT I
1450 FOR  J=1 TO N:DB(J)=B(J):IC(J)=0:JC(J)=0:NEXT J
1460 J0=0:JT=0:MR=0
1470 REM     ПОИСК МИНИМАЛЬНОЙ СТОИМОСТИ
1480 II=0:JJ=0:Y=1E+12
1490 FOR I=1 TO M
1500 IF IR(I)=1 THEN 1560
1510 FOR J=1 TO N
1520 IF IC(J)=1 THEN 1550
1530 IF C(I,J)>Y THEN 1550
1540 Y=C(I,J):II=I:JJ=J
1550 NEXT J
1560 NEXT I
1570 REM    ЗАНЕСЕНИЕ КОРРЕСПОНДЕНЦИИ В II И JJ
1580 IF DA(II)<=DB(JJ) THEN 1610
1590 X(II,JJ)=DB(JJ):IX(II,JJ)=1:DA(II)=DA(II)-DB(JJ):DB(JJ)=0
1600 IC(JJ)=1:J0=J0+1:JT=JT+1:GOTO 1640
1610 IF DA(II)=DB(JJ) AND MR=M-1 THEN 1590
1620 X(II,JJ)=DA(II):IX(II,JJ)=1
1630 DB(JJ)=DB(JJ)-DA(II):DA(II)=0:IR(II)=1:J0=J0+1:MR=MR+1
1640 JR(II)=JR(II)+1:JC(JJ)=JC(JJ)+1
1650 IF J0<M+N-1 THEN 1480
1660 MR=MR+1
1670 REM   РАСЧЕТ ПОТЕНЦИАЛОВ СТРОК И СТОЛБЦОВ
1680 FOR I=1 TO M:IU(I)=0:U(I)=0:NEXT I
1690 FOR J=1 TO N:IV(J)=0:V(J)=0:NEXT J
1700 KT=0:L=0
1710 FOR I=1 TO M
1720 IF JR(I)<KT THEN 1740
1730 KT=JR(I):L=I
1740 NEXT I
1750 U(L)=0:IU(L)=1:J0=1
1760 FOR J=1 TO N
1770 IF IX(L,J)=0 THEN 1790
1780 V(J)=C(L,J):IV(J)=1:J0=J0+1
1790 NEXT J
1800 FOR I=1 TO M:FOR J=1 TO N
1810 IF IX(I,J)=0 THEN 1880
1820 IF IU(I)=0 AND IV(J)=0 THEN 1880
1830 IF IU(I)=1 AND IV(J)=1 THEN 1880
1840 IF IU(I)=0 AND IV(J)=1 THEN 1870
1850 V(J)=C(I,J)-U(I):IV(J)=1:J0=J0+1
1860 GOTO 1880
1870 U(I)=C(I,J)-V(J):IU(I)=1:J0=J0+1
1880 NEXT J:NEXT I
1890 IF J0<>M+N THEN 1800
1900 REM   ПОИСК МИНИМАЛЬНОГО ОЦЕНОЧНОГО ПАРАМЕТРА
1910 DL=0:K=0:L=0
1920 FOR I=1 TO M:FOR J=1 TO N
1930 IF IX(I,J)=1 THEN 1960
1940 DT=C(I,J)-U(I)-V(J):IF DT>=DL THEN 1960
1950 DL=DT:K=I:L=J
1960 NEXT J:NEXT I
1970 IF ABS(DL)<=1E-06 THEN 2490
1980 REM
1990 PRINT:GOSUB 2660
2000 REM   ПОСТРОЕНИЕ КОНТУРА ДЛЯ ПЕРЕХОДА К НОВОМУ БАЗИСУ
2010 FOR I=1 TO M:IU(I)=0:NEXT I
2020 FOR J=1 TO N:IV(J)=0:NEXT J
2030 FOR I=1 TO M+N:IT(I)=0:JT(I)=0:NEXT I
2040 FOR I=1 TO M:FOR J=1 TO N
2050 MM(I,J)=0
2060 NEXT J:NEXT I
2070 KT=1:IP=0:IT(KT)=K:JT(KT)=L:MM(K,L)=1:IU(K)=1
2080 PRINT KT,K;L
2090 LR=O:LC=0:II=IT(KT):JJ=0
2100 FOR J=1 TO N
```

```
2110 IF LC=1 OR IX(II,J)=0 OR IV(J)=1 OR MM(II,J)=1 THEN 2150
2120 IF JC(J)=1 AND J=L THEN 2140
2130 IF JC(J)=1 THEN IP=1:GOTO 2150
2140 LC=1:JJ=J:IV(J)=1:J=N
2150 NEXT J
2160 IF JJ<>0 THEN 2190
2170 IF IP>0 THEN IP=0
2180 KT=KT-1:GOTO 2210
2190 KT=KT+1:IT(KT)=II:JT(KT)=JJ:MM(II,JJ)=1
2200 IF JT(KT)=L AND KT>2 THEN 2330
2210 LR=0:LC=0:II=0:JJ=JT(KT)
2220 FOR I=1 TO M
2230 IF LR=1 OR IX(I,JJ)=0 OR IU(I)=1 OR MM(I,JJ)=1 THEN 2260
2240 IF JR(I)=1 AND IP=0 THEN IP=1:GOTO 2260
2250 LR=1:II=I:IU(I)=1:I=M
2260 NEXT I
2270 IF II<>0 THEN 2300
2280 IF IP>0 THEN IP=0
2290 KT=KT-1:GOTO 2090
2300 KT=KT+1:IP=0:IT(KT)=II:JT(KT)=JJ:MM(II,JJ)=1
2310 GOTO 2090
2320 REM    ПОЛУЧЕНИЕ НОВОГО БАЗИСА
2330 W=1E+10:LL=0:KK=0
2340 FOR I=2 TO KT STEP 2
2350 IF X(IT(I),JT(I))>=W THEN 2370
2360 W=X(IT(I),JT(I)):KK=IT(I):LL=JT(I)
2370 NEXT I:WW=-W
2380 FOR I=1 TO KT
2390 WW=WW*(-1)
2400 X(IT(I),JT(I))=X(IT(I),JT(I))+WW
2410 NEXT I
2420 IX(K,L)=1:IX(KK,LL)=0
2430 JR(K)=JR(K)+1:JR(KK)=JR(KK)-1
2440 JC(L)=JC(L)+1:JC(LL)=JC(LL)-1
2450 REM
2460 'PRINT"DL="DL;"W="W;"KT="KT;"KK="KK;"LL="LL
2470 PRINT"ПРЕОБРАЗОВАНИЕ ЗАКОНЧЕНО УСПЕШНО"
2480 GOTO 1680
2490 IF M3=0 THEN 2620
2500 M4=0:FOR I=1 TO M:FOR J=1 TO N
2510 IF C(I,J)>1E+09 THEN 2550
2520 IF X(I,J)<=D(I,J) THEN 2550
2530 DK(I,J)=D(I,J):A(I)=A(I)-D(I,J):B(J)=B(J)-D(I,J):M4=1
2540 C(I,J)=1E+10
2550 NEXT J:NEXT I
2560 IF M4=0 THEN 2580
2570 FOR I= 1 TO M:FOR J=1 TO N:X(I,J)=0:IX(I,J)=0:NEXT J:NEXT I:GOTO 1440
2580 FOR I= 1 TO M:FOR J=1 TO N:IF C(I,J)>1E+09 THEN X(I,J)=DK(I,J)
2590 NEXT J:NEXT I
2600 IF M3=0 THEN 2620
2610 FOR I=1 TO MN:FOR J=1 TO NN:C(I,J)=CP(I,J):NEXT J:NEXT I
2620 PRINT:PRINT"ОКОНЧАТЕЛЬНОЕ РЕШЕНИЕ"
2630 GOSUB 2660
2635 PRINT:PRINT" ПОТРЕБУЕТСЯ ВЫВОД НА ПРИНТЕР. ДА - 1,НЕТ - 0)"
2636 INPUT INDI:IF INDI=0 THEN 2860
2640 PRINT:PRINT" < ЗАДАЙТЕ СПЕЦИФИКАЦИЮ ФАЙЛА РЕЗУЛЬТАТОВ РАСЧЕТА"
2650 INPUT FAILR$:OPEN "O",#1,FAILR$:GOTO 2750
2660 CC=0
2670 IF DL=0 THEN PRINT" I           J","XIJ","CIJ","СТОИМОСТЬ"
2680 FOR I=1 TO MN:FOR J=1 TO NN
2690 IF X(I,J)=0 THEN 2720
2700 PP=C(I,J)*X(I,J):CC=CC+PP
2710 IF KL=0 THEN PRINT AG$(I);TAB(6)BG$(J);TAB(15) USING "#####.#";X(I,J),C(I,J),PP
2720 NEXT J:NEXT I
2730 PRINT"ЦЕЛЕВАЯ ФУНКЦИЯ(ОБЩИЕ ЗАТРАТЫ)   ";CC
2740 RETURN
2750 PRINT #1,:PRINT #1, TAB(5)"ОПТИМАЛЬНЫЙ ПЛАН ПОСТАВОК":PRINT #1,
2770 PRINT #1, " ПУНКТЫ       РАЗМЕР     СТОИМОСТЬ  СТОИМОСТЬ"
```

```
2780 PRINT #1, "            ПОСТАВКИ              ПОСТАВКИ"
2790 PRINT #1, " I   J          XIJ         CIJ"
2800 FOR I=1 TO MN:FOR J=1 TO NN
2810 IF X(I,J)=0 THEN 2840
2820 PP=C(I,J)*X(I,J)
2830 PRINT #1, AG$(I);TAB(6)BG$(J);TAB(11) USING "#######.###";X(I,J),C(I,J),PP
2840 NEXT J:NEXT I
2850 PRINT #1,"ЦЕЛЕВАЯ ФУНКЦИЯ(ОБЩИЕ ЗАТРАТЫ)   ";CC
2855 CLOSE #1
2856 COLOR 25,5,15:PRINT "РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ  "FAILR$
2857 BBB$=INKEY$
2858 IF LEN(BBB$)=0 THEN 2857
2860 END
```

ПРИЛОЖЕНИЕ 9
КОМПЬЮТЕРНАЯ ПРОГРАММА РАЗРАБОТКИ СБОРОЧНО-РАЗВОЗОЧНЫХ МАРШРУТОВ НА ОСНОВЕ МЕТОДА КЛАРКА-РАЙТА

```
DEFINT I-N
 DEFINT D
 COLOR 7,0:CLS:LOCATE 2,60,0:COLOR 25,5,15:PRINT"         "'ВКЛЮЧИТЕ ПРИНТЕР"
 COLOR 7,7,15:LOCATE 4:COLOR 0,7,15
 PRINT TAB(27)"ЗАДАЧА   K R T R T";:PRINT SPC(37):PRINT TAB(26)" ";
 COLOR 15,0,15:PRINT"КАФЕДРА";:COLOR 23,5,15:PRINT"*БНТУ*";
 COLOR 15,0,15:PRINT" ОАПДД ";:COLOR 7,7,15:PRINT SPC(35):COLOR 1,14
 PRINT TAB(22)"РАЗРАБОТКА СБОРОЧНО-РАЗВОЗОЧНЫХ МАРШРУТОВ";
 PRINT TAB(80)" ";:PRINT TAB(25)"НА ОСНОВЕ МЕТОДА КЛАРКА-РАЙТА";
 PRINT TAB(80)" ": TT=TIMER
 PRINT:COLOR 7,0,7
 GOTO 7
6 DIM DYNAMIC P$(120),Q(120)
DIM DYNAMIC DL(120,120)
DIM DYNAMIC DL1(120,120)
DIM DYNAMIC IB(120,120)
DIM IB1(120),DLP(120):RETURN
7 OPEN"O",#2,"KLRRAIT.REZ"
  F1$="####.#"
 PRINT #2, TAB(5)"РАЗРАБОТКА РАЗВОЗОЧНЫХ(СБОРОЧНЫХ) МАРШРУТОВ ПО МЕТОДУ КЛАРКА-
РАЙТА":PRINT #2,
 PRINT"ВВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ":PRINT
 PRINT"ВВЕДЕНЫ ЛИ КРАТЧАЙШИЕ РАССТОЯНИЯ. ДА -1, НЕТ -0":INPUT AAP
  IF AAP=0 THEN 20
  rem PRINT"ВВЕДИТЕ СПЕЦИФИКАЦИЮ ФАЙЛА С КРАТЧАЙШИМИ РАССТОЯНИМИ";
  rem INPUT FAAA$
CLS:COLOR 7,4:LOCATE 20,5,0:PRINT"* Ж Д И Т Е *":COLOR 7,0
    OPEN"I",#1,"KRTRST.REZ":INPUT #1,N1,AAA:N=N1-1
    GOSUB 6
 FOR I=0 TO N
  INPUT #1,P$(I)
FOR J=0 TO N
  INPUT #1,DL(I,J),AAA
  NEXT J
  NEXT I
  CLOSE #1
  LOCATE 20,5,0:PRINT"             "
 INPUT"НАИМЕНОВАНИЕ (КОД) ИСХОДНОГО (БАЗОВОГО) ПУНКТА";PP$
  FOR I=0 TO N:IF P$(I)=PP$ THEN II=I
   NEXT I
   FOR J=0 TO N:DLP(J)=DL(0,J):DL(0,J)=DL(II,J):DL(II,J)=DLP(J):NEXT J
FOR I=0 TO N:DLP(I)=DL(I,0):DL(I,0)=DL(I,II):DL(I,II)=DLP(I):NEXT I
   PPP$=P$(0):P$(0)=P$(II):P$(II)=PPP$:GOTO 30
 P$(0)=PP$
20 INPUT"ЧИСЛО ПУНКТОВ ЗАВОЗА(ВЫВОЗА) ГРУЗА N ";N:
   GOSUB 6
  INPUT"НАИМЕНОВАНИЕ (КОД) ИСХОДНОГО (БАЗОВОГО) ПУНКТА";P$(0)
 PRINT"НАИМЕНОВАНИЕ (КОД) ПРОМЕЖУТОЧНОГО ПУНКТА"
 FOR I=1 TO N:PRINT I"-ГО":INPUT P$(I):NEXT I
30 INPUT"ЧИСЛО ТИПОВ АВТОМОБИЛЕЙ K ";K:K0=K
   DIM Q1(K),IA1(K)
 INPUT"ПРЕДЕЛЬНО ДОПУСТИМОЕ ЧИСЛО ПУНКТОВ ЗАЕЗДА NP ";NP
 INPUT"ДОПУСКАЕМЫЙ ПЕРЕГРУЗ АВТОМОБИЛЯ E1,% ";E1
 PRINT" ОБЬЕМЫ ПЕРЕВОЗОК ПО ПУНКТАМ"
   K8=1
 FOR I=1 TO N:PRINT"Q("P$(I)")":INPUT Q(I)
   IB(I,K8)=I
IB1(I)=1   'IF Q(I)=0 THEN IB(I,K8)=I:IB1(I)=0
    NEXT I
IF AAP=1 THEN 170
 PRINT"РАССТОЯНИЯ(СТОИМОСТИ) ПЕРЕВОЗОК МЕЖДУ ПУНКТАМИ"
 FOR I=1 TO N:FOR J=0 TO I-1
 PRINT"DL("P$(I)","P$(J)")":INPUT DL(I,J):NEXT J:NEXT I
 FOR I=1 TO N:FOR J=0 TO I-1:DL(J,I)=DL(I,J):NEXT J:NEXT I
170 PRINT"РЯД ИМЕЮЩИХСЯ АВТОМОБИЛЕЙ ПО МЕРЕ УВЕЛИЧЕНИЯ"
```

```
 PRINT"ИХ ГРУЗОПОДЬЕМНОСТИ"
 FOR K3=1 TO K:PRINT"Q1("K3")":INPUT Q1(K3):NEXT K3
CLS:COLOR 7,4:LOCATE 20,5,0:PRINT"* Ж Д И Т Е *":COLOR 7,0
 PRINT #2, TAB(10)"ИСХОДНЫЕ ДАННЫЕ"
 PRINT #2, TAB(5)"ОБЬЕМЫ ЗАВОЗА(ВЫВОЗА) ГРУЗА ПО ПУНКТАМ":PRINT #2,
     NN=0:N7=1
45 IF N<=NN+40 THEN N9=N-NN:GOSUB 50:GOTO 55
N9=40:GOSUB 50:NN=NN+40:N7=N7+40:GOTO 45
    REM
 50 PRINT #2,"КОД";:FOR I=1 TO N9:PRINT #2,TAB(6*I+7)P$(I+NN);:NEXT I
 PRINT #2,:PRINT #2,"ПУНКТА"
 PRINT #2,"ОБЬЕМ ";:FOR I=1 TO N9
   PRINT #2, TAB(6*I+4)USING F1$;Q(I+NN);:NEXT I
 PRINT #2,:PRINT #2,"ГРУЗА":PRINT #2,:RETURN
   REM
 55 PRINT #2, TAB(5)"РАССТОЯНИЯ(СТОИМОСТИ) ПЕРЕВОЗОК МЕЖДУ"
 PRINT #2, TAB(12)"ПУНКТАМИ I И J"
      NN=0:N7=1
145 IF N<=NN+40 THEN N9=N-NN:GOSUB 150:GOTO 155
N9=40:GOSUB 150:NN=NN+40:N7=N7+40:GOTO 145
150 PRINT #2,"  J     ";P$(II);
  FOR J=1 TO N9-1:PRINT #2, TAB(6*J+9) P$(J+NN);:NEXT J
 PRINT #2,
 PRINT #2," I"
 FOR I=N7 TO N:PRINT #2, P$(I);
 FOR J=0 TO N9
  IF NN+J>I-1 THEN 152
  PRINT #2, TAB(6*J+6)USING F1$;DL(I,J+NN);:NEXT J
 152 PRINT #2,
   NEXT I
     RETURN
     REM
155 PRINT #2, TAB(5)"ПРЕДЕЛЬНО ДОПУСКАЕМОЕ ЧИСЛО ПУНКТОВ ЗАЕЗДА-"NP
 PRINT #2, TAB(5)"ДОПУСКАЕМЫЙ ПЕРЕГРУЗ АВТОМОБИЛЯ- "USING F1$;E1;
PRINT #2,"%"
 PRINT #2,:PRINT #2, TAB(5)"ЗАДАННЫЙ РЯД ГРУЗОПОДЬЕМНОСТЕЙ АВТОМОБИЛЕЙ"
 FOR K3=1 TO K:PRINT #2, TAB(K3*6+5)USING F1$; Q1(K3);:NEXT K3
 PRINT #2,:PRINT #2,:PRINT #2, TAB(10)"РЕЗУЛЬТАТЫ РАСЧЕТОВ"
 PRINT #2,:PRINT #2,
 FOR I=2 TO N:FOR J=1 TO I-1
DL1(I,J)=DL(I,0)+DL(J,0)-DL(I,J)
 NEXT J:NEXT I
 PRINT #2, TAB(5)"ВЫИГРЫШИ ПРИ ОБЬЕДИНЕНИИ МАЯТНИКОВЫХ МАРШРУТОВ"
 PRINT #2, TAB(5)P$(0) "- I- "P$(0)" И "P$(0)"- J- "P$(0):PRINT #2,
    REM
    NN=0:N7=2
245 IF N<=NN+40 THEN N9=N-NN:GOSUB 250:GOTO 255
N9=40:GOSUB 250:NN=NN+40:N7=N7+40:GOTO 245
250 PRINT #2,"     J";
   FOR J=1 TO N9-1:PRINT #2, TAB(6*J+7) P$(J+NN);:NEXT J
 PRINT #2,
 PRINT #2,"  I"
 FOR I=N7 TO N:PRINT #2, P$(I);
 FOR J=1 TO N9-1
  IF NN+J>I-1 THEN 252
  PRINT #2, TAB(6*J+4)USING F1$; DL1(I,J+NN);:NEXT J
252 PRINT #2,
     NEXT I:PRINT #2,
   RETURN
       REM
255  FOR I=1 TO N
     IF Q(I)>0.00001 THEN 256
      FOR J=1 TO N: DL1(I,J)=-1: DL1(J,I)=-1: NEXT J
256 NEXT I
E1=1+E1/100
480 DL3=0
 FOR I=2 TO N
 FOR J=1 TO I-1
```

```
 IF DL3<DL1(I,J) THEN DL3=DL1(I,J):J6=J:J7=I
540 NEXT J
550 NEXT I
IF DL3=<0 THEN 680
DL1(J7,J6)=-1:DL1(J6,J7)=-1:IB2=IB1(J6)+IB1(J7)
 IF IB2>NP THEN  480
   QP=Q(J6)+Q(J7):IF QP>Q1(K0)*E1 THEN 480
  Q(J6)=0:Q(J7)=0: IB4=IB1(J6):IB3=IB(J6,IB4):IB5=IB(J7,IB1(J7))
 FOR I=1 TO IB4:IB(IB3,I)=IB(J6,IB4+1-I):NEXT I
 FOR I=IB4+1 TO IB2:IB(IB3,I)=IB(J7,I-IB4):NEXT I
 FOR I=1 TO IB2:IB(IB5,I)=IB(IB3,IB2+1-I):NEXT I
 IF IB1(J6)<2 THEN 551
  FOR I=1 TO N:DL1(J6,I)=-1:DL1(I,J6)=-1:NEXT I
551 IF IB1(J7)<2 THEN 552
   FOR I=1 TO N:DL1(J7,I)=-1:DL1(I,J7)=-1:NEXT I
552 IB1(IB3)=IB2:IB1(IB5)=IB2:Q(IB3)=QP:Q(IB5)=QP
  DL1(IB3,IB5)=-1:DL1(IB5,IB3)=-1
 GOTO 480
680 PRINT #2, TAB(5)"МАРШРУТЫ ПЕРЕВОЗОК ГРУЗОВ":PRINT #2,:I2=0
 FOR I=1 TO N:IF Q(I)=0 THEN 860
 I2=I2+1: DSLL=0
 IB6=IB1(I):IB7=IB(I,IB6)
 PRINT #2, TAB(5)"МАРШРУТ N"I2" "P$(0);
 IB6=IB1(I):IB7=IB(I,IB6)
 FOR I1=1 TO IB6:PRINT #2,"-"P$(IB(I,I1));:NEXT I1
      DSLL=DSLL+DL(0,IB(I,1)):DSLL=DSLL+DL(0,IB(I,IB6))
    FOR I1=1 TO IB6-1:DSLL=DSLL+DL(IB(I,I1),IB(I,I1+1)):NEXT I1
   PRINT #2,"- "P$(0):IF IB6=1 THEN 790
  PRINT #2, TAB(17)" ИЛИ  "P$(0);
 FOR I1=1 TO IB6:PRINT #2,"-"P$(IB(IB7,I1));:NEXT I1
   PRINT #2,"- "P$(0)
790 PRINT #2, TAB(5)"ОБЪЕМ ПЕРЕВОЗОК"USING F1$;Q(I)
 PRINT #2, TAB(5)"ТРЕБУЕТСЯ АВТОМОБИЛЬ ГРУЗОПОДЪЕМНОСТЬЮ  -";
 K3=0
820 K3=K3+1
IF E1*Q1(K3)<Q(I) THEN 820
  PRINT #2, USING F1$;Q1(K3)
 Q(I)=0:Q(IB7)=0:IA1(K3)=IA1(K3)+1
   PRINT #2, TAB(5)"ДЛИНА МАРШРУТА - "USING F1$;DSLL:PRINT #2,
860 NEXT I
 PRINT #2,"ТРЕБУЕМОЕ ОБЩЕЕ ЧИСЛО АВТОМОБИЛЕЙ ПО ГРУЗОПОДЬЕМНОСТЯМ"
 PRINT #2, TAB(5)"ГРУЗОПОДЬЕМН.  ЧИСЛО":PRINT #2,
 FOR K3=1 TO K:PRINT #2, TAB(10) USING F1$;Q1(K3);TAB(20) IA1(K3):NEXT K3
 PRINT #2,:PRINT #2, TAB(5)".....***....."
 CLOSE #2
 COLOR 25,5,15:PRINT "РЕЗУЛЬТАТЫ РАСЧЕТА В ФАЙЛЕ  KLRRAIT.REZ"
870 BBB$=INKEY$
  IF LEN(BBB$)=0 THEN 870
 END
```

ПРИЛОЖЕНИЕ 10
КОМПЬЮТЕРНАЯ ПРОГРАММА РАСЧЕТА ПАРАМЕТРОВ СЕТЕВОГО ГРАФИКА

```
5 CLS:PRINT"РАСЧЕТ ПАРАМЕТРОВ СЕТЕВОГО ГРАФИКА"
 6 FM$="#########.#":FM1$="#######.#":DEFINT I-N
 7 PRINT"ВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ":INPUT" ЧИСЛО СОБЫТИЙ = ";M
 8 DIM T(M,M)'МАССИВ ДЛИТЕЛЬНОСТИ РАБОТ
 9 DIM TR(M),TP(M)'МАССИВЫ РАННИХ И ПОЗДНИХ СРОКОВ СВЕРШ.СОБЫТ.
 15 FOR I=1 TO M-1
 16 CLS:LOCATE 5,35
 17 COLOR 5,2:PRINT"ЕСЛИ РАБОТЫ НЕТ, ТО ВВОДИТЕ НОЛЬ"
 18 COLOR 7,0,0:LOCATE 8,1,0
 19 FOR J=I+1 TO M
 20 PRINT"ДЛИТ.РАБОТ T("I","J")";:INPUT T(I,J):IF T(I,J)=0 THEN T(I,J)=1E10
 25 NEXT J:NEXT I
 30 CLS
 32 FOR I=1 TO M:TR(I)=0:TP(I)=1E6:NEXT I
 35 'РАСЧЕТ РАННИХ СРОКОВ СВЕРШЕНИЯ СОБЫТИЙ
 40 FOR I=1 TO M-1:FOR J=I+1 TO M
 45 IF T(I,J)>100000! THEN 50
 47 TT=TR(I)+T(I,J):IF TT>TR(J) THEN TR(J)=TT
 50 NEXT J:NEXT I
 55 'РАСЧЕТ ПОЗДНИХ СРОКОВ СВЕРШЕНИЯ СОБЫТИЙ
 60 TP(M)=TR(M):TKR=TP(M)
 70 FOR J=M TO 2 STEP -1:FOR I=J-1 TO 1 STEP -1
 75 IF T(I,J)>100000! THEN 80
 77 TT=TP(J)-T(I,J):IF TT<TP(I) THEN TP(I)=TT
 80 NEXT I:NEXT J
 85 PRINT"     РАСЧЕТ ПАРАМЕТРОВ СЕТЕВОГО ГРАФИКА"
 90 PRINT "№ СОБЫ- РАНН.СРОК  ПОЗДН.СРОК  РЕЗЕРВ"
 94 PRINT "  ТИЯ   СВЕРШЕНИЯ  СВЕРШЕНИЯ   ВРЕМЕНИ"
 100 FOR I=1 TO M:PRINT I;:
 102 PRINT USING FM$;TR(I),TP(I),TP(I)-TR(I):NEXT I
 105 PRINT"КРИТИЧЕСКОЕ ВРЕМЯ ГРАФИКА-   " USING FM$;TKR
 110 'РАСЧЕТ ПОЛНЫХ И СВОБОДНЫХ РЕЗЕРВОВ РАБОТ
 112 PRINT"      Р Е З Е Р В Ы  Р А Б О Т"
 113 PRINT"НАЧАЛЬН.   КОНЕЧН.   ДЛИТ.   ПОЛНЫЙ   СВОБОДН."
 114 PRINT"СОБЫТ.      СОБЫТ.   РАБОТЫ   РЕЗЕРВ   РЕЗЕРВ
 115 PRINT"РАБОТЫ      РАБОТЫ             РАБОТЫ   РАБОТЫ
 116 FOR I=1 TO M-1:FOR J=I+1 TO M
 120 IF T(I,J)>100000! THEN 150
 125 PRINT I,J;:
 130 PRINT  USING FM1$;T(I,J),TP(J)-TR(I)-T(I,J),TR(J)-TR(I)-T(I,J)
 150 NEXT J:NEXT I
 INPUT OST
 200 END
```

ПРИЛОЖЕНИЕ 11
КОМПЬЮТЕРНАЯ ПРОГРАММА РЕШЕНИЯ ИГРОВОЙ ЗАДАЧИ ДВУХ СТОРОН

```
CLS
 PRINT "МОДЕЛИРОВАНИЕ ИГРОВОЙ ЗАДАЧИ ДВУХ СТОРОН"
   PRINT"ВВЕДИТЕ ИСХОДНЫЕ ДАННЫЕ"
   FM1$="######.##":FM2$="#.####"
 INPUT"ЧИСЛО СТРАТЕГИЙ СТОРОНЫ А";M
 INPUT"ЧИСЛО СТРАТЕГИЙ СТОРОНЫ В";N
INPUT"ТОЧНОСТЬ РЕШЕНИЯ Е";E
IF M>N THEN MN=M ELSE MN=N
 DIM C(M,N),CP(MN),CN(M),CM(N),IA(10000),IB(10000),SB(M),CA(N),A(M),B(N)
    'ВВОД ДАННЫХ
   PRINT"ЦЕНУ ИГРЫ ДЛЯ СТРАТЕГИЙ ИГРОКОВ"
 FOR I=1 TO M :FOR J=1 TO N
      PRINT" C(A("I"),B("J"))=";
 INPUT C(I,J)
     NEXT J:NEXT I
      JI=0
      JI=JI+1
 'НАЧАЛЬНОЕ РЕШЕНИЕ
   MNP=N:IZN=1
FOR I=1 TO M
 FOR J=1 TO N: CP(J)=C(I,J)
 NEXT J
 GOSUB 1000:CN(I)=CP(IPP)
 NEXT I
  MNP=M:IZN=-1
  FOR J=1 TO N
   FOR I=1 TO M: CP(I)=C(I,J)
 NEXT I
 GOSUB 1000:CM(J)=CP(IPP)
 NEXT J
      FOR I=1 TO M
     CP(I)=CN(I)
 NEXT I
  MNP=M
  IZN=-1
 GOSUB 1000:IA(JI)=IPP
      FOR J=1 TO N
     CP(J)=CM(J)
 NEXT J
   MNP=N
  IZN=+1
 GOSUB 1000:IB(JI)=IPP
    IF ABS(CM(IB(JI))-CN(IA(JI)))>0 OR JI>10000 THEN 800
     PRINT "ИМЕЕТСЯ СЕДЛОВАЯ ТОЧКА С ЦЕНОЙ "CM(IB(JI)):GOTO 3000
'РАСЧЕТ НАКОПЛЕННЫХ РЕЗУЛЬТАТОВ СТОРОНЫ А
800 PRINT" РЕШЕНИЕ ПРИ" JI"-Й ИТЕРАЦИИ  A("IA(JI)") И B("IB(JI)")":PRINT "SA   ";
  FOR J=1 TO N
     SA(J)=SA(J)+C(IA(JI),J):PRINT USING FM1$;SA(J),
 NEXT J
 PRINT:PRINT "SB  ";
'РАСЧЕТ НАКОПЛЕННЫХ РЕЗУЛЬТАТОВ СТОРОНЫ В
     FOR I=1 TO M
     SB(I)=SB(I)+C(I,IB(JI)): PRINT USING FM1$;SB(I),
 NEXT I
    PRINT
 'РАСЧЕТ ПОСЛЕДУЮЩИХ ИТЕРАЦИЙ
900 JI=JI+1:JP=JI-1
    FOR J=1 TO N
     CP(J)=SA(J)
```

```
NEXT J
 MNP=N
  IZN=+1
GOSUB 1000:IB(JI)=IPP
    FOR I=1 TO M
    CP(I)=SB(I)
NEXT I
 MNP=M
   IZN=-1
GOSUB 1000:IA(JI)=IPP
 IF ABS(SA(IB(JI))/JP-SB(IA(JI))/JP)<E THEN 2000
   GOTO 800
1000 REM
   IPP=1
   FOR IP1=2 TO MNP
IF CP(IP1)*IZN<CP(IPP)*IZN THEN IPP=IP1
 NEXT IP1
    RETURN
'РАСЧЕТ ВЕРОЯТНОСТЕЙ СТРАТЕГИЙ
2000 FOR I=1 TO M
    A(I)=0
    FOR J=1 TO JP
     IF I=IA(J) THEN A(I)=A(I)+1
 NEXT J
NEXT I
      FOR I=1 TO N
    B(I)=0
    FOR J=1 TO JP
     IF I=IB(J) THEN B(I)=B(I)+1
 NEXT J
  NEXT I
    PRINT "СРЕДНЯЯ ЦЕНА ИГРЫ СТОРОНЫ А ="USING FM1$;SA(IB(JI))/JP
  PRINT "СРЕДНЯЯ ЦЕНА ИГРЫ СТОРОНЫ В ="USING FM1$;SB(IA(JI))/JP
    PRINT "ЧИСЛО ИТЕРАЦИЙ ="JP
    PRINT "ВЕРОЯТНОСТИ СТРАТЕГИЙ"
        FOR I=1 TO M
    PRINT "A("I")=",USING FM2$;A(I)/JP
  NEXT I
      FOR I=1 TO N
    PRINT "B("I")=",USING FM2$;B(I)/JP
  NEXT I
  PRINT" SA И SB - СООТВЕТСТВЕННО НАКОПЛЕННЫЕ РЕЗУЛЬТАТЫ СТОРОН А И В"
INPUT OST
3000 END
```