

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение

высшего образования

«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт химических технологий и инжиниринга
ФГБОУ ВО УГНТУ в г. Sterлитамаке

Кафедра информатики, математики и физики

**МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ЗАДАЧ
ЭЛЕКТРОТЕХНИКИ, ТЕОРИИ НАДЕЖНОСТИ И
КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ**

Учебно-методическое пособие
по выполнению самостоятельной и практической работы

Стерлитамак
2024

Составитель: Рахман П.А., к.т.н., доцент каф. ИМФ

Рецензент: Быковский Н.А., к.т.н., доцент каф. АТИС

Учебно-методическое пособие утверждено на заседании кафедры «Информатика, математика и физика» 22.04.2024, протокол № 5.

Учебно-методическое пособие по выполнению самостоятельной и практической работы по дисциплине «Математические основы технических систем».

Предназначено для студентов, обучающихся по направлению 27.04.04 – «Управление в технических системах».

Содержание

Введение	2
1. Математические модели задач электротехники	3
1.1. Моделирование электрических схем во временной области.....	3
1.2. Моделирование электрических схем в частотной области.....	4
1.3. Примеры математического моделирования электрических схем	5
2. Математическое моделирование задач теории надежности	18
2.1. Исходная цепь Маркова в модели надежности технических систем.....	18
2.2. Преобразованная цепь Маркова для расчета среднего времени наработки до первого отказа технических систем.....	22
2.3. Редуцированная цепь Маркова для расчета вероятности безотказной работы технических систем	25
2.4. Расчет нестационарного коэффициента оперативной готовности.....	27
3. Математические модели задач комбинаторной оптимизации	29
3.1. Задача псевдодвулевой оптимизации о рюкзаке	29
3.1.1. Описание модели задачи о рюкзаке и метода ее решения.....	29
3.1.2. Пример решения задачи о рюкзаке методом локального поиска	31
3.1.3. Анализ верхней границы объема перебора	33
3.1.4. Анализ нижней границы качества решений.....	35
3.1.5. Экспериментальное исследование объема перебора и качества решений.....	37
3.2. Специализированная задача распределения вычислительных ресурсов	38
3.2.1. Математическая модель задачи распределения вычислительных ресурсов.....	40
3.2.2. Декомпозиция задачи распределения вычислительных ресурсов	44
3.2.3. Пример решения задачи распределения вычислительных ресурсов	48
Список использованных источников	50

Введение

В настоящее время мир заполнен огромным множеством различных технических систем, используемых как в повседневной жизни человека, так и в производственных и бизнес процессах предприятий. Технические системы различаются по назначению, функциям, структуре, составу и параметрам входящих в них элементов, алгоритмам функционирования, а также особенностями их взаимодействия с «окружающим миром», и в том числе с другими системами. Системы обладают рядом технических характеристик, как основных, например: производительность, быстродействие и мощность, так и качественных, например: надежность, эффективность и точность.

Для анализа существующих технических систем и их технических характеристик, и проектирования новых систем по заданным требованиям к характеристикам применяют математическое моделирование технических систем. Математическое моделирование дает возможность численно оценить те или иные технические характеристики существующих и проектируемых систем с учетом их структуры, состава и параметров входящих в систему элементов, особенностей функционирования и взаимодействия с «окружающим миром».

Для математического моделирования технических систем применяется множество различных разделов математики:

- Линейные алгебраические уравнения и системы алгебраических уравнений.
- Дифференциальные уравнения и системы дифференциальных уравнений.
- Интегральные преобразования (преобразование Лапласа).
- Теория вероятностей и математическая статистика.
- Теория множеств и комбинаторика.
- Булева алгебра.
- Теория графов.
- Цепи Маркова.
- Вычислительная математика.
- Методы дискретной оптимизации.

Выбор математического аппарата для математического моделирования конкретной технической системы определяется целями и задачами моделирования, а также с учетом структуры, функций и других особенностей технической системы.

В рамках данного учебно-методического пособия рассматриваются типовые математические модели задач электротехники, теории надежности и комбинаторной оптимизации, которые широко используются для анализа и проектирования программно-аппаратных систем, в том числе, систем управления, вычислительных систем и сетей, а также их отдельных элементов.

1. Математические модели задач электротехники

1.1. Моделирование электрических схем во временной области

Одной из типовых задач электротехники является анализ электрических цепей с одним входом и выходом (рис. 1.1), который базируется на выводе математической модели, связывающей функцию входного напряжения $x(t)$ с функцией выходного напряжения $y(t)$, а также выводе функций выходного напряжения для заданных функций входного напряжения.

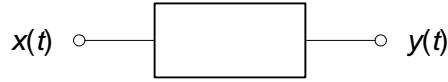


Рис. 1.1. Обобщенная модель электрической схемы с одним входом и выходом.

В общем случае электрическая схема может содержать большое количество как линейных, так и нелинейных элементов, внутренние источники постоянных и переменных токов и напряжений, и, соответственно, математическая модель, связывающая функции входного и выходного напряжения оказывается достаточно сложной.

В случае же электрических схем, содержащих только линейные элементы (резисторы, катушки индуктивности и конденсаторы, усилители токов и напряжений с постоянными коэффициентами усиления), связь между функциями входного и выходного напряжений может быть представлена дифференциальным уравнением следующего вида:

$$\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + \dots + a_{n-1} \frac{dy(t)}{dt} + a_n y(t) = \frac{d^m x(t)}{dt^m} + b_1 \frac{d^{m-1} x(t)}{dt^{m-1}} + \dots + b_{m-1} \frac{dx(t)}{dt} + b_m x(t). \quad (1.1)$$

Где, $a_1 \dots a_n, b_1 \dots b_m$ – некоторые действительные постоянные коэффициенты.

Особо отметим, что порядок $m \geq 0$ правой части уравнения может быть как меньше или равен порядка $n \geq 0$ левой части уравнения, так и больше него.

Очевидно, что данное уравнение можно преобразовать по Лапласу, используя теорему о дифференцировании оригинала с учетом начальных условий для функций входного и выходного напряжений и их производных. При решении практических задач в большинстве случаев начальные условия либо равны нулю, либо ненулевые условия для функций входного и выходного напряжений и их производных взаимно компенсируются, и в итоге преобразование Лапласа дает связь между изображениями функций входного и выходного напряжения в виде формулы, не содержащей какие-либо начальные условия:

$$Y(p) = \frac{p^m + b_1 p^{m-1} + \dots + b_{m-1} p + b_m}{p^n + a_1 p^{n-1} + \dots + a_{n-1} p + a_n} X(p). \quad (1.2)$$

Обозначим полученную дробь как передаточную функцию $W(p)$ комплексного аргумента p , и тогда имеем:

$$Y(p) = W(p)X(p). \quad (1.3)$$

Таким образом, если для электрической схемы вывести передаточную функцию $W(p)$, то тогда для заданной функции входного напряжения $x(t)$, для которой известно изображение $X(p)$, по формуле (1.3) можно получить изображение выходного напряжения $Y(p)$, а затем также найти оригинал функции выходного напряжения $y(t)$.

Теперь особо отметим, что в электротехнике большое значение имеет, так называемая переходная функция $h(t)$, которая получается на выходе схемы, при подаче на вход схемы единичной функции (функции Хевисайда): $\eta(t) = \begin{cases} 1, & \text{при } t \geq 0; \\ 0, & \text{при } t < 0. \end{cases}$ Тогда, учитывая, что

$\eta(t) \rightarrow 1/p$, и обозначив изображение функции $h(t)$ как $H(p)$, по формуле (53) имеем:

$$H(p) = W(p)/p. \quad (1.4)$$

Наконец, подставляя формулу (1.4) в формулу (1.3) получаем:

$$Y(p) = p H(p) X(p). \quad (1.5)$$

Тогда, выполняя обратное преобразование Лапласа изображения $Y(p)$ мы можем получить следующие два интеграла Дюамеля, связывающие функции входного и выходного напряжений при помощи переходной функции $h(t)$:

$$y(t) = x(0)h(t) + \int_0^t \frac{dx(\tau)}{d\tau} h(t-\tau) d\tau. \quad (1.6)$$

$$y(t) = x(t)h(0) + \int_0^t x(\tau) \frac{dh(t-\tau)}{d\tau} d\tau. \quad (1.7)$$

Таким образом, если электрической схемы вывести переходную функцию $h(t)$, то тогда для любой функции входного напряжения $x(t)$ по формулам (1.6) или (1.7) можно также найти функцию выходного напряжения $y(t)$.

1.2. Моделирование электрических схем в частотной области

Другой типовой задачей электротехники является анализ электрических цепей с одним входом и выходом в частотной области, который позволяет проанализировать влияние схемы, оказываемое на амплитуду и фазу гармонических входных сигналов вида $\sin(\omega t + \varphi)$.

При частотном анализе схема рассматривается как «комплексный преобразователь» (рис. 1.2) с передаточной функцией $W(p)$.

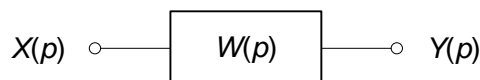


Рис. 1.2. Обобщенная модель электрической схемы как комплексного преобразователя.

Для анализа электрических схем в частотной области, используют связь между односторонним преобразованием Фурье и преобразованием Лапласа. По определению, одностороннее преобразование Фурье для функции $w(t)$ вычисляется как:

$$W(i\omega) = \int_0^{\infty} w(t) e^{-i\omega t} dt. \quad (1.8)$$

Тогда, учитывая, что изображение по Лапласу вычисляется как $W(p) = \int_0^{\infty} w(t) e^{-pt} dt$,

мы можем получить простую связь между комплексной передаточной функцией $W(p)$ и, так называемой, частотной передаточной функцией:

$$W(i\omega) = \lim_{p \rightarrow i\omega} W(p). \quad (1.9)$$

Далее, частотную передаточную функцию $W(i\omega)$ в соответствии с теорией функций комплексного переменного можно представить как:

$$W(i\omega) = \operatorname{Re}(W(i\omega)) + i \cdot \operatorname{Im}(W(i\omega)). \quad (1.10)$$

Где, $\operatorname{Re}(W(i\omega))$ – действительная часть функции $W(i\omega)$, а $\operatorname{Im}(W(i\omega))$ – мнимая часть.

Тогда, можно найти амплитуду $A(\omega)$ и фазу $\Psi(\omega)$ частотной передаточной функции $W(i\omega)$ по следующим формулам теории функций комплексного переменного:

$$A(\omega) = \sqrt{\operatorname{Re}(W(i\omega))^2 + \operatorname{Im}(W(i\omega))^2}. \quad (1.11)$$

$$\Psi(\omega) = \arctg\left(\frac{\operatorname{Im}(W(i\omega))}{\operatorname{Re}(W(i\omega))}\right). \quad (1.12)$$

В электротехнике, функция $A(\omega)$ известна как амплитудно-частотная характеристика, а функция $\Psi(\omega)$ как фазово-частотная характеристика.

1.3. Примеры математического моделирования электрических схем

Пример 1. Задана электрическая схема фильтра низких частот первого порядка, состоящего из резистора R и конденсатора C , соединенных последовательно (рис. 1.3).

Начальное напряжение на конденсаторе $u_C(0) = 0$.

Найти функцию выходного напряжения $y(t)$ фильтра для следующих функций входного напряжения $x(t)$:

1) Единичная функция (Хевисайда): $\eta(t) = \begin{cases} 1, & \text{при } t \geq 0; \\ 0, & \text{при } t < 0. \end{cases}$

2) Прямоугольные колебания с периодом T , заданные в виде ряда при помощи функции Хевисайда: $\sum_{k=0}^{\infty} (\eta(t - kT) - \eta(t - kT - T/2)) = \begin{cases} 1, & \text{при } kT \leq t < kT + T/2; \\ 0, & \text{при } kT + T/2 \leq t < (k+1)T. \end{cases}$

3) Синусоидальные колебания с заданной частотой и фазой: $\sin(\omega t + \varphi)$.

Найти амплитудно-частотную $A(\omega)$ и фазово-частотную характеристики $\Psi(\omega)$.

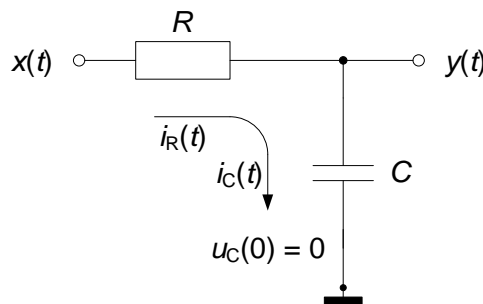


Рис. 1.3. Электрическая схема фильтра низких частот первого порядка.

Анализ во временной области. Заметим, что поскольку конденсатор C подключен параллельно выходу фильтра, то выходное напряжение фильтра равно напряжению на конденсаторе: $y(t) = u_C(t)$. Тогда, поскольку начальное напряжение на конденсаторе равно нулю, то начальное значение выходного напряжения также равно нулю:

$$y(0) = 0.$$

Также заметим, что в силу последовательного соединения через резистор R и конденсатор C течет один и тот же ток:

$$i_R(t) = i_C(t).$$

Из электротехники известно, что токи и напряжения на резисторе и конденсаторе связаны между собой следующими соотношениями:

$$u_R(t) = i_R(t)R; \quad i_C(t) = C \frac{du_C(t)}{dt}.$$

Также заметим, что входное напряжение $x(t)$ складывается из напряжений на резисторе и конденсаторе:

$$x(t) = u_R(t) + u_C(t).$$

Тогда, учитывая приведенные выше соотношения токов и напряжений на резисторе и конденсаторе, и равенство токов $i_R(t) = i_C(t)$, получаем:

$$x(t) = i_R(t)R + u_C(t) = i_C(t)R + u_C(t) = RC \frac{du_C(t)}{dt} + u_C(t).$$

Тогда, учитывая что $y(t) = u_C(t)$, получаем следующее дифференциальное уравнение, с начальным условием, связывающее выходное напряжение с входным напряжением:

$$\frac{dy(t)}{dt} + \frac{1}{RC} y(t) = \frac{1}{RC} x(t); \quad y(0) = 0.$$

Обозначим коэффициент затухания фильтра как: $\alpha = \frac{1}{RC}$.

Тогда, дифференциальное уравнение принимает следующий вид:

$$\frac{dy(t)}{dt} + \alpha y(t) = \alpha x(t); \quad y(0) = 0.$$

Выполним преобразование Лапласа дифференциального уравнения с учетом нулевого начального условия $y(0) = 0$, обозначив изображения функций $x(t)$ и $y(t)$ как $X(p)$ и $Y(p)$, а также используя теорему о дифференцировании оригинала. Тогда, получаем уравнение:

$$pY(p) + \alpha Y(p) = \alpha X(p).$$

Отсюда, окончательно получаем формулу, связывающую изображения функций входного и выходного напряжений:

$$Y(p) = \frac{\alpha}{p + \alpha} X(p).$$

1) Пусть в качестве функции входного напряжения задана единичная функция (Хевисайда) $x(t) = \eta(t)$. Тогда, ее изображение выглядит как:

$$X(p) = \frac{1}{p}.$$

Соответственно, изображение функции выходного напряжения выглядит как:

$$Y(p) = \frac{\alpha}{p(p + \alpha)} = \frac{1}{p} - \frac{1}{p + \alpha}.$$

Учитывая преобразования Лапласа $\frac{1}{p} \rightarrow 1$ и $\frac{1}{p + \alpha} \rightarrow e^{-\alpha t}$, получаем оригинал функции выходного напряжения (рис. 1.4):

$$y(t) = 1 - e^{-\alpha t}.$$

2) Пусть в качестве функции входного напряжения задана функция прямоугольных колебаний с периодом T , заданная в виде ряда: $\sum_{k=0}^{\infty} (\eta(t - kT) - \eta(t - kT - T/2))$.

Тогда, с учетом преобразования $\eta(t) \leftarrow 1/p$ и теоремы о запаздывании оригинала $f(t - \tau) \leftarrow e^{-p\tau} F(p)$ ее изображение выглядит как:

$$X(p) = \sum_{k=0}^{\infty} \left(\frac{1}{p} e^{-pkT} - \frac{1}{p} e^{-p(kT + T/2)} \right).$$

Соответственно, изображение функции выходного напряжения выглядит как:

$$Y(p) = \sum_{k=0}^{\infty} \left(\frac{\alpha}{p(p + \alpha)} e^{-pkT} - \frac{\alpha}{p(p + \alpha)} e^{-p(kT + T/2)} \right) = \sum_{k=0}^{\infty} \left(\left(\frac{1}{p} - \frac{1}{p + \alpha} \right) e^{-pkT} - \left(\frac{1}{p} - \frac{1}{p + \alpha} \right) e^{-p(kT + T/2)} \right).$$

Тогда, учитывая, $\frac{1}{p} \rightarrow \eta(t)$ и $\frac{1}{p + \alpha} \rightarrow e^{-\alpha t} \eta(t)$, и теорему о запаздывании оригинала, получаем оригинал функции выходного напряжения (рис. 1.5):

$$y(t) = \sum_{k=0}^{\infty} (\eta(t - kT)(1 - e^{-\alpha(t - kT)}) - \eta(t - kT - T/2)(1 - e^{-\alpha(t - kT - T/2)})).$$

3) Пусть в качестве функции входного напряжения задана функция $x(t) = \sin(\omega t + \varphi)$. Тогда, учитывая, что $\sin(\omega t + \varphi) = \sin \omega t \cos \varphi + \cos \omega t \sin \varphi$, ее изображение выглядит как:

$$X(p) = \frac{\omega}{p^2 + \omega^2} \cos \varphi + \frac{p}{p^2 + \omega^2} \sin \varphi.$$

Тогда, изображение функции выходного напряжения выглядит как:

$$Y(p) = \frac{\alpha\omega}{(p+\alpha)(p^2+\omega^2)} \cos \varphi + \frac{\alpha p}{(p+\alpha)(p^2+\omega^2)} \sin \varphi.$$

Разложим полученные дроби следующим образом:

$$\begin{aligned} Y(p) &= \frac{\alpha\omega}{(p+\alpha)(p^2+\omega^2)} \cos \varphi + \frac{\alpha p}{(p+\alpha)(p^2+\omega^2)} \sin \varphi = \\ &= \frac{\alpha\omega}{\alpha^2+\omega^2} \left(\frac{1}{p+\alpha} - \frac{p-\alpha}{p^2+\omega^2} \right) \cos \varphi + \frac{\alpha^2}{\alpha^2+\omega^2} \left(\frac{p+\omega^2/\alpha}{p^2+\omega^2} - \frac{1}{p+\alpha} \right) \sin \varphi = \\ &= \frac{\alpha\omega}{\alpha^2+\omega^2} \left(\frac{1}{p+\alpha} - \frac{p}{p^2+\omega^2} + \frac{\alpha}{\omega} \cdot \frac{\omega}{p^2+\omega^2} \right) \cos \varphi + \frac{\alpha^2}{\alpha^2+\omega^2} \left(\frac{p}{p^2+\omega^2} + \frac{\omega}{\alpha} \cdot \frac{\omega}{p^2+\omega^2} - \frac{1}{p+\alpha} \right) \sin \varphi. \end{aligned}$$

Учитывая преобразования Лапласа $\frac{1}{p+\alpha} \rightarrow e^{-\alpha t}$, $\frac{p}{p^2+\omega^2} \rightarrow \cos \omega t$ и $\frac{\omega}{p^2+\omega^2} \rightarrow \sin \omega t$, получаем оригинал функции выходного напряжения:

$$y(t) = \frac{\alpha\omega}{\alpha^2+\omega^2} \left(e^{-\alpha t} - \cos \omega t + \frac{\alpha}{\omega} \sin \omega t \right) \cos \varphi + \frac{\alpha^2}{\alpha^2+\omega^2} \left(\cos \omega t + \frac{\omega}{\alpha} \sin \omega t - e^{-\alpha t} \right) \sin \varphi.$$

Представим полученную функцию в следующем виде:

$$y(t) = \frac{\alpha^2}{\alpha^2+\omega^2} \left(\sin \omega t \left(\cos \varphi + \frac{\omega}{\alpha} \sin \varphi \right) + \cos \omega t \left(\sin \varphi - \frac{\omega}{\alpha} \cos \varphi \right) - e^{-\alpha t} \left(\sin \varphi - \frac{\omega}{\alpha} \cos \varphi \right) \right).$$

Теперь, воспользуемся следующими тригонометрическими тождествами:

$$\cos \varphi + \frac{\omega}{\alpha} \sin \varphi = \frac{\sqrt{\alpha^2+\omega^2}}{\alpha} \cos \left(\varphi - \arctg \left(\frac{\omega}{\alpha} \right) \right); \quad \sin \varphi - \frac{\omega}{\alpha} \cos \varphi = \frac{\sqrt{\alpha^2+\omega^2}}{\alpha} \sin \left(\varphi - \arctg \left(\frac{\omega}{\alpha} \right) \right).$$

Тогда, имеем следующее:

$$y(t) = \frac{\alpha}{\sqrt{\alpha^2+\omega^2}} \left(\sin \omega t \cos \left(\varphi - \arctg \left(\frac{\omega}{\alpha} \right) \right) + \cos \omega t \sin \left(\varphi - \arctg \left(\frac{\omega}{\alpha} \right) \right) - e^{-\alpha t} \sin \left(\varphi - \arctg \left(\frac{\omega}{\alpha} \right) \right) \right).$$

Далее, учитывая формулу $\sin \omega t \cos \gamma + \cos \omega t \sin \gamma = \sin(\omega t + \gamma)$, получаем:

$$y(t) = \frac{\alpha}{\sqrt{\alpha^2+\omega^2}} \left(\sin \left(\omega t + \varphi - \arctg \left(\frac{\omega}{\alpha} \right) \right) - e^{-\alpha t} \sin \left(\varphi - \arctg \left(\frac{\omega}{\alpha} \right) \right) \right).$$

Введем следующие обозначения:

$$A(\omega) = \frac{\alpha}{\sqrt{\alpha^2+\omega^2}}; \quad \Psi(\omega) = -\arctg \left(\frac{\omega}{\alpha} \right).$$

Тогда окончательно получаем функцию выходного напряжения (рис. 1.6):

$$y(t) = A(\omega)(\sin(\omega t + \varphi + \Psi(\omega)) - e^{-\alpha t} \sin(\varphi + \Psi(\omega))).$$

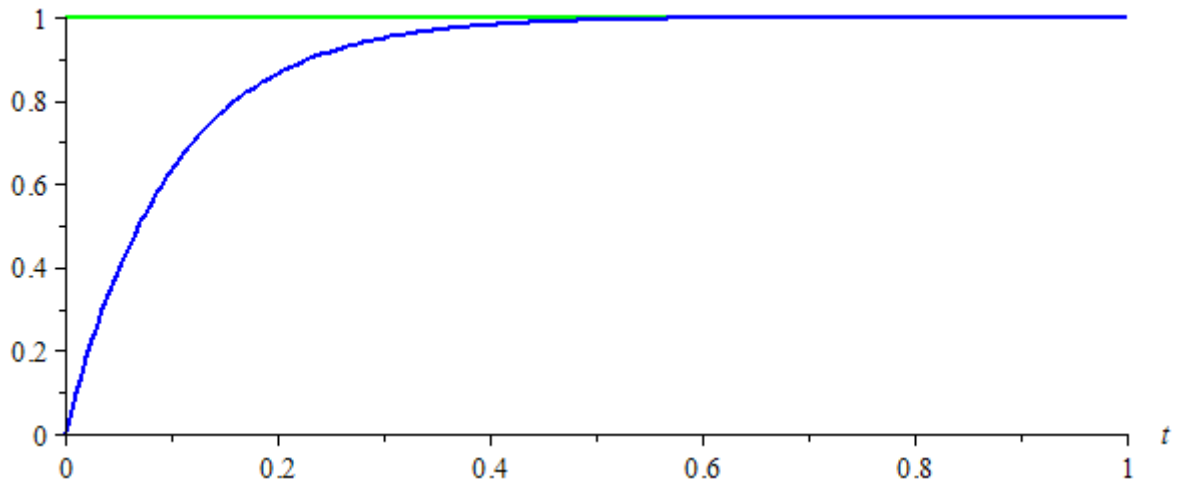


Рис. 1.4. Графики функций входных и выходных напряжений фильтра при $\alpha = 10$ и единичной функции на входе фильтра.

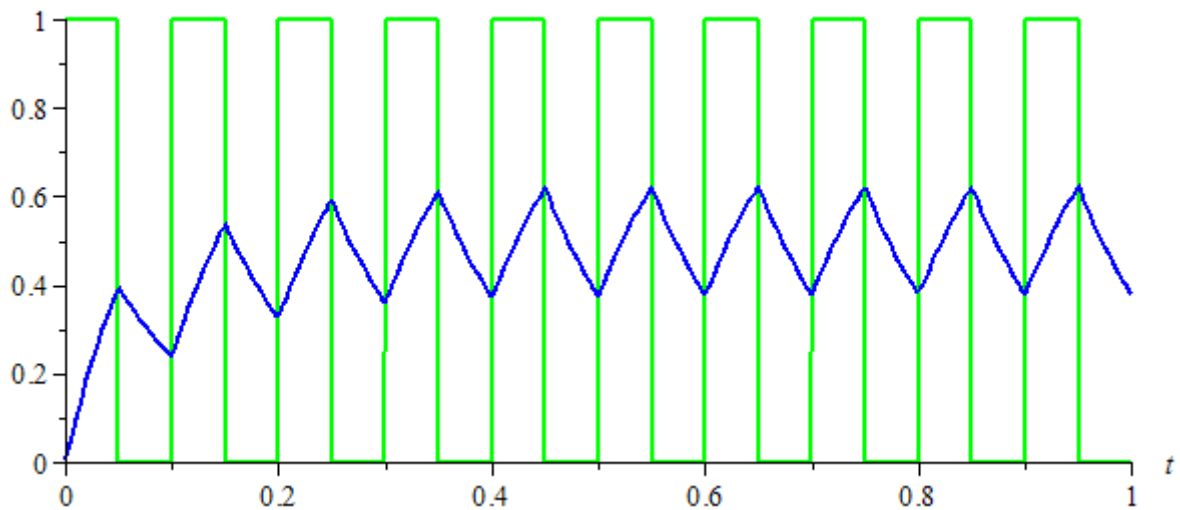


Рис. 1.5. Графики функций входных и выходных напряжений фильтра при $\alpha = 10$ и функции прямоугольных колебаний с периодом $T = 0,1$ на входе фильтра.

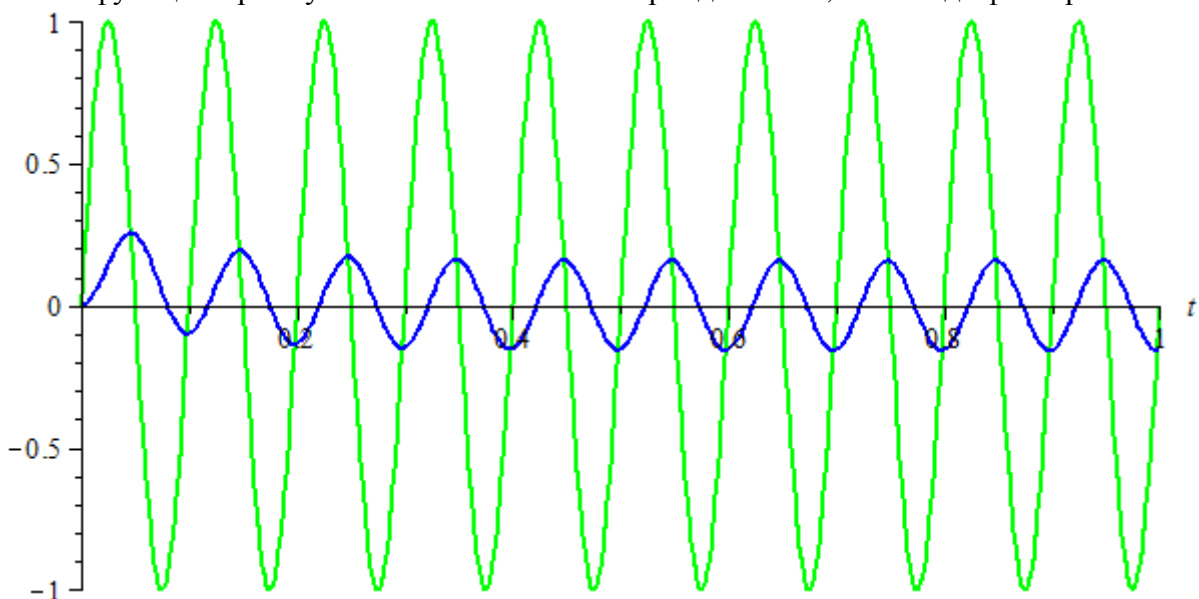


Рис. 1.6. Графики функций входных и выходных напряжений фильтра при $\alpha = 10$ и синусоидальной функции с круговой частотой $\omega = 2\pi \cdot 10$ и фазой $\varphi = 0$ на входе фильтра.

Анализ в частотной области. Выше при анализе фильтра мы вывели следующую формулу, связывающую изображения функций входного и выходного напряжений:

$$Y(p) = \frac{\alpha}{p + \alpha} X(p).$$

Соответственно, комплексная передаточная функция: $W(p) = \frac{\alpha}{p + \alpha}$.

Вычислим теперь частотную передаточную функцию, и преобразуем ее так, чтобы можно было отдельно выделить ее действительную и мнимую части. Тогда, получаем:

$$W(i\omega) = \frac{\alpha}{i\omega + \alpha} \cdot \frac{\alpha - i\omega}{\alpha - i\omega} = \frac{\alpha^2}{\alpha^2 + \omega^2} - i \frac{\alpha\omega}{\alpha^2 + \omega^2}.$$

Тогда, амплитудно-частотная характеристика (рис. 1.7) фильтра:

$$A(\omega) = \frac{\alpha}{\alpha^2 + \omega^2} \sqrt{\alpha^2 + \omega^2} = \frac{\alpha}{\sqrt{\alpha^2 + \omega^2}}.$$

Соответственно, фазово-частотная характеристика (рис. 1.8) фильтра:

$$\Psi(\omega) = -\operatorname{arctg}\left(\frac{\omega}{\alpha}\right).$$

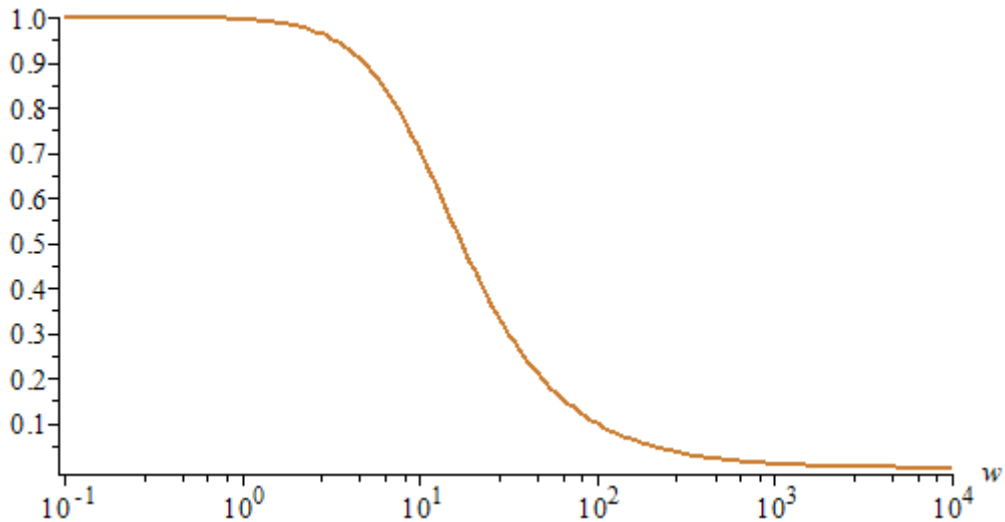


Рис. 1.7. График амплитудно-частотной характеристики фильтра при $\alpha = 10$.

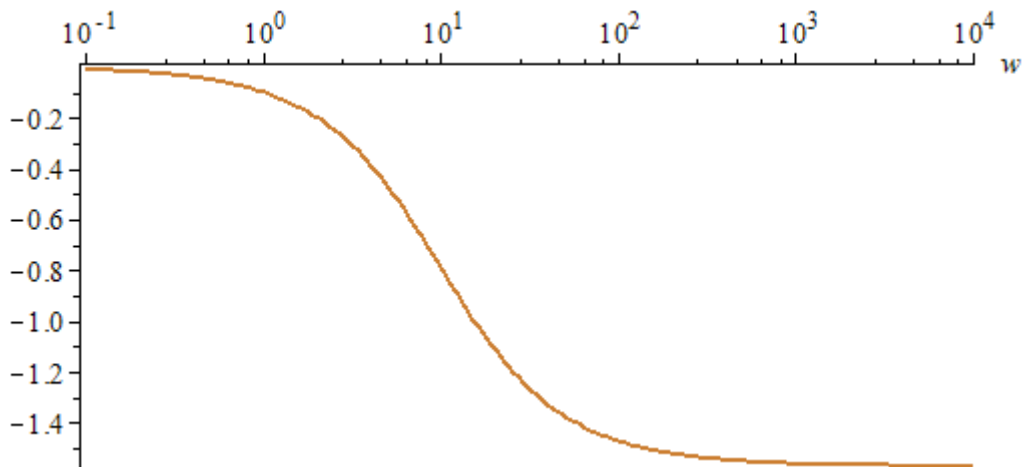


Рис. 1.8. График фазово-частотной характеристики фильтра при $\alpha = 10$.

Пример 2. Задана электрическая схема полосового фильтра второго порядка, состоящего из резистора R , катушки индуктивности L и конденсатора C (рис. 1.9).

Начальное напряжение на конденсаторе $u_C(0) = 0$, начальный ток катушки индуктивности $i_L(0) = 0$.

Найти функцию выходного напряжения $y(t)$ фильтра для следующих функций входного напряжения $x(t)$:

1) Единичная функция (Хевисайда): $\eta(t) = \begin{cases} 1, & \text{при } t \geq 0; \\ 0, & \text{при } t < 0. \end{cases}$

2) Прямоугольные колебания с периодом T , заданные в виде ряда при помощи функции Хевисайда: $\sum_{k=0}^{\infty} (\eta(t - kT) - \eta(t - kT - T/2)) = \begin{cases} 1, & \text{при } kT \leq t < kT + T/2; \\ 0, & \text{при } kT + T/2 \leq t < (k+1)T. \end{cases}$

3) Синусоидальные колебания с заданной частотой и фазой: $\sin(\omega t + \varphi)$.

Найти амплитудно-частотную $A(\omega)$ и фазово-частотную характеристики $\Psi(\omega)$.

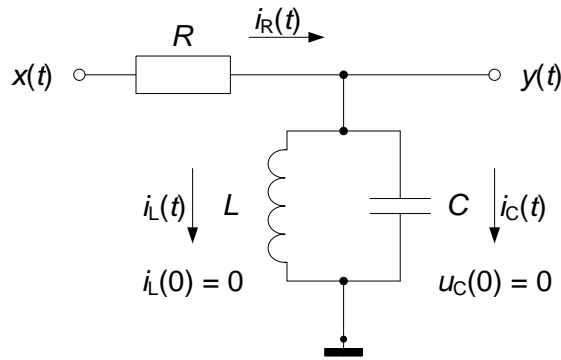


Рис. 1.9. Электрическая схема полосового фильтра второго порядка.

Анализ во временной области. Заметим, что поскольку конденсатор C подключен параллельно выходу фильтра, то выходное напряжение фильтра равно напряжению на конденсаторе: $y(t) = u_C(t)$. Тогда, поскольку начальное напряжение на конденсаторе равно нулю, то начальное значение выходного напряжения также равно нулю:

$$y(0) = 0.$$

Также заметим, что ток, протекающий через резистор R , равен сумме токов, протекающих через катушку индуктивности L и конденсатор C :

$$i_R(t) = i_C(t) + i_L(t).$$

Из электротехники известно, что токи и напряжения на резисторе, конденсаторе и катушке индуктивности связаны между собой следующими соотношениями:

$$u_R(t) = i_R(t)R; \quad i_C(t) = C \frac{du_C(t)}{dt}; \quad u_L(t) = L \frac{di_L(t)}{dt}.$$

Тогда, учитывая эти соотношения, и то, что $i_L(0) = 0$, $u_C(0) = 0$ и $y(t) = u_C(t)$, получаем, что начальное значение производной выходного напряжения составляет:

$$\frac{dy(0)}{dt} = \frac{du_C(0)}{dt} = \frac{1}{C} i_C(0) = \frac{1}{C} (i_R(0) - i_L(0)) = \frac{u_R(0)}{RC} = \frac{x(0) - u_C(0)}{RC} = \frac{x(0)}{RC}.$$

Теперь также заметим, что входное напряжение $x(t)$ складывается из напряжений на резисторе и конденсаторе, и кроме того, напряжения на конденсаторе и на катушке индуктивности совпадают и равны выходному напряжению:

$$x(t) = u_R(t) + u_C(t); \quad u_C(t) = u_L(t) = y(t).$$

Тогда, учитывая приведенные выше соотношения токов и напряжений на резисторе, катушке индуктивности и конденсаторе, а также $i_R(t) = i_L(t) + i_C(t)$, получаем:

$$x(t) = (i_C(t) + i_L(t))R + u_C(t) = \left(C \frac{du_C(t)}{dt} + i_L(t)\right)R + u_C(t).$$

Продифференцируем обе части уравнения, и учтем, что $di_L(t)/dt = u_L(t)/L$ и $u_L(t) = u_C(t)$, и тогда получим:

$$\frac{dx(t)}{dt} = R \left(C \frac{d^2 u_C(t)}{dt^2} + \frac{di_L(t)}{dt} \right) + \frac{du_C(t)}{dt} = R \left(C \frac{d^2 u_C(t)}{dt^2} + \frac{1}{L} u_C(t) \right) + \frac{du_C(t)}{dt}.$$

Тогда, учитывая что $y(t) = u_C(t)$, получаем следующее дифференциальное уравнение с начальными условиями, связывающее выходное напряжение с входным:

$$\frac{d^2 y(t)}{dt^2} + \frac{1}{RC} \frac{dy(t)}{dt} + \frac{1}{LC} y(t) = \frac{1}{RC} \frac{dx(t)}{dt}; \quad y(0) = 0; \quad \frac{dy(0)}{dt} = \frac{1}{RC} x(0).$$

Введем следующие обозначения:

- Коэффициент затухания фильтра: $\beta = \frac{1}{2RC}$.
- Собственная круговая частота свободных колебаний: $\omega_0 = \frac{1}{\sqrt{LC}}$.

Тогда, дифференциальное уравнение принимает следующий вид:

$$\frac{d^2 y(t)}{dt^2} + 2\beta \frac{dy(t)}{dt} + \omega_0^2 y(t) = 2\beta \frac{dx(t)}{dt}; \quad y(0) = 0; \quad \frac{dy(0)}{dt} = 2\beta x(0).$$

Выполним преобразование Лапласа дифференциального уравнения с учетом нулевых начальных условий, обозначив изображения функций $x(t)$ и $y(t)$ как $X(p)$ и $Y(p)$, а также используя теорему о дифференцировании оригинала. Тогда, получаем уравнение:

$$p^2 Y(p) - py(0) - \frac{dy(0)}{dt} + 2\beta(pY(p) - y(0)) + \omega_0^2 Y(p) = 2\beta(pX(p) - x(0)).$$

Учитывая, начальные условия $y(0) = 0$ и $dy(0)/dt = 2\beta x(0)$, окончательно получаем формулу, связывающую изображения функций входного и выходного напряжений:

$$Y(p) = \frac{2\beta p}{p^2 + 2\beta p + \omega_0^2} X(p).$$

1) Пусть в качестве функции входного напряжения задана единичная функция (Хевисайда) $x(t) = \eta(t)$. Тогда, ее изображение выглядит как:

$$X(p) = \frac{1}{p}.$$

Соответственно, изображение функции выходного напряжения выглядит как:

$$Y(p) = \frac{2\beta}{p^2 + 2\beta p + \omega_0^2}.$$

Чтобы упростить нахождение оригинала $y(t)$ – функции выходного напряжения, преобразуем знаменатель $p^2 + 2\beta p + \omega_0^2 = (p + \beta)^2 + \omega_0^2 - \beta^2$, и получим:

$$Y(p) = \frac{2\beta}{(p + \beta)^2 + \omega_0^2 - \beta^2}.$$

Теперь рассмотрим три возможных случая параметров фильтра, когда собственная частота ω_0 больше, равна или меньше коэффициента затухания β .

Случай А. Собственная частота фильтра больше коэффициента затухания: $\omega_0 > \beta$.

В этом случае $\omega_0^2 - \beta^2 > 0$, и записываем изображение $Y(p)$ следующим образом:

$$Y(p) = \frac{2\beta}{\sqrt{\omega_0^2 - \beta^2}} \cdot \frac{\sqrt{\omega_0^2 - \beta^2}}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2}.$$

Тогда, учитывая преобразование $\frac{\sqrt{\omega_0^2 - \beta^2}}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} \rightarrow e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2})$, получаем оригинал функции выходного напряжения фильтра (рис. 1.10):

$$y(t) = \frac{2\beta}{\sqrt{\omega_0^2 - \beta^2}} e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2}).$$

Случай Б. Собственная частота фильтра равна коэффициенту затухания: $\omega_0 = \beta$.

В этом случае $\omega_0^2 - \beta^2 = 0$, и имеем следующее изображение $Y(p)$:

$$Y(p) = \frac{2\beta}{(p + \beta)^2}.$$

Тогда, учитывая преобразование $\frac{1}{(p + \beta)^2} \rightarrow te^{-\beta t}$, получаем оригинал функции выходного напряжения фильтра (рис. 1.11):

$$y(t) = 2\beta te^{-\beta t}.$$

Случай В. Собственная частота фильтра меньше коэффициента затухания: $\omega_0 < \beta$.

В этом случае $\omega_0^2 - \beta^2 < 0$, и записываем изображение $Y(p)$ следующим образом:

$$Y(p) = \frac{2\beta}{\sqrt{\beta^2 - \omega_0^2}} \cdot \frac{\sqrt{\beta^2 - \omega_0^2}}{(p + \beta)^2 - (\sqrt{\beta^2 - \omega_0^2})^2}.$$

Тогда, учитывая преобразование $\frac{\sqrt{\beta^2 - \omega_0^2}}{(p + \beta)^2 - (\sqrt{\beta^2 - \omega_0^2})^2} \rightarrow e^{-\beta t} \text{sh}(t\sqrt{\beta^2 - \omega_0^2})$, получаем оригинал функции выходного напряжения фильтра (рис. 1.12):

$$y(t) = \frac{2\beta}{\sqrt{\beta^2 - \omega_0^2}} e^{-\beta t} \text{sh}(t\sqrt{\beta^2 - \omega_0^2}).$$

2) Пусть в качестве функции входного напряжения задана функция прямоугольных колебаний с периодом T , заданная в виде ряда: $\sum_{k=0}^{\infty} (\eta(t - kT) - \eta(t - kT - T/2))$.

Тогда, с учетом преобразования $\eta(t) \leftarrow 1/p$ и теоремы о запаздывании оригинала $f(t - \tau) \leftarrow e^{-p\tau} F(p)$ ее изображение выглядит как:

$$X(p) = \sum_{k=0}^{\infty} \left(\frac{1}{p} e^{-pkT} - \frac{1}{p} e^{-p(kT + T/2)} \right).$$

Соответственно, изображение функции выходного напряжения выглядит как:

$$Y(p) = \sum_{k=0}^{\infty} \left(\frac{2\beta}{(p + \beta)^2 + \omega_0^2 - \beta^2} e^{-pkT} - \frac{2\beta}{(p + \beta)^2 + \omega_0^2 - \beta^2} e^{-p(kT + T/2)} \right).$$

Ограничимся рассмотрением случая, когда собственная частота фильтра больше коэффициента затухания: $\omega_0 > \beta$.

Тогда, учитывая преобразование $\frac{\sqrt{\omega_0^2 - \beta^2}}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} \rightarrow e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2})$ и теорему о запаздывании оригинала, получаем оригинал функции выходного напряжения (рис. 1.13):

$$y(t) = \frac{2\beta}{\sqrt{\omega_0^2 - \beta^2}} \sum_{k=0}^{\infty} \left(\eta(t - kT) e^{-\beta(t - kT)} \sin((t - kT)\sqrt{\omega_0^2 - \beta^2}) - \right. \\ \left. - \eta(t - kT - T/2) e^{-\beta(t - kT - T/2)} \sin((t - kT - T/2)\sqrt{\omega_0^2 - \beta^2}) \right).$$

3) Пусть в качестве функции входного напряжения задана функция $x(t) = \sin(\omega t + \varphi)$. Тогда, учитывая, что $\sin(\omega t + \varphi) = \sin \omega t \cos \varphi + \cos \omega t \sin \varphi$, ее изображение выглядит как:

$$X(p) = \frac{\omega}{p^2 + \omega^2} \cos \varphi + \frac{p}{p^2 + \omega^2} \sin \varphi.$$

Тогда, изображение функции выходного напряжения выглядит как:

$$Y(p) = \frac{2\beta\omega p}{(p^2 + \omega^2)(p^2 + 2\beta p + \omega_0^2)} \cos \varphi + \frac{2\beta p^2}{(p^2 + \omega^2)(p^2 + 2\beta p + \omega_0^2)} \sin \varphi.$$

Выполним следующее разложение дробей:

$$Y(p) = \frac{2\beta\omega(\omega_0^2 - \omega^2)}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\frac{1}{p^2 + \omega^2} \left(p + \frac{2\beta\omega^2}{\omega_0^2 - \omega^2} \right) - \frac{1}{p^2 + 2\beta p + \omega_0^2} \left(p + \frac{2\beta\omega_0^2}{\omega_0^2 - \omega^2} \right) \right) \cos \varphi +$$

$$+ \frac{4\beta^2\omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\frac{1}{p^2 + \omega^2} \left(p - \frac{\omega_0^2 - \omega^2}{2\beta} \right) - \frac{1}{p^2 + 2\beta p + \omega_0^2} \left(p - \frac{\omega_0^2(\omega_0^2 - \omega^2)}{2\beta\omega^2} \right) \right) \sin \varphi.$$

Далее, ограничимся рассмотрением случая, когда собственная частота фильтра больше коэффициента затухания: $\omega_0 > \beta$, и представим $Y(p)$ в следующем виде:

$$Y(p) = \frac{2\beta\omega(\omega_0^2 - \omega^2)}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\frac{p}{p^2 + \omega^2} + \left(\frac{2\beta\omega}{\omega_0^2 - \omega^2} \right) \frac{\omega}{p^2 + \omega^2} - \frac{p + \beta}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} - \right. \\ \left. - \left(\frac{\omega_0^2 + \omega^2}{\omega_0^2 - \omega^2} \frac{\beta}{\sqrt{\omega_0^2 - \beta^2}} \right) \frac{\sqrt{\omega_0^2 - \beta^2}}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} \right) \cos \varphi +$$

$$+ \frac{4\beta^2\omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\frac{p}{p^2 + \omega^2} - \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right) \frac{\omega}{p^2 + \omega^2} - \frac{p + \beta}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} + \right. \\ \left. + \left(\frac{2\beta^2\omega^2 + \omega_0^2(\omega_0^2 - \omega^2)}{2\beta^2\omega^2} \frac{\beta}{\sqrt{\omega_0^2 - \beta^2}} \right) \frac{\sqrt{\omega_0^2 - \beta^2}}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} \right) \sin \varphi.$$

Учтем преобразования $\frac{\omega}{p^2 + \omega^2} \rightarrow \sin(\omega t)$, $\frac{\sqrt{\omega_0^2 - \beta^2}}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} \rightarrow e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2})$,

$\frac{p}{p^2 + \omega^2} \rightarrow \cos(\omega t)$ и $\frac{p + \beta}{(p + \beta)^2 + (\sqrt{\omega_0^2 - \beta^2})^2} \rightarrow e^{-\beta t} \cos(t\sqrt{\omega_0^2 - \beta^2})$, и тогда, наконец, получаем

оригинал функции выходного напряжения фильтра:

$$y(t) = \frac{2\beta\omega(\omega_0^2 - \omega^2)}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\cos(\omega t) + \left(\frac{2\beta\omega}{\omega_0^2 - \omega^2} \right) \sin(\omega t) - e^{-\beta t} \cos(t\sqrt{\omega_0^2 - \beta^2}) - \right. \\ \left. - \left(\frac{\omega_0^2 + \omega^2}{\omega_0^2 - \omega^2} \frac{\beta}{\sqrt{\omega_0^2 - \beta^2}} \right) e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2}) \right) \cos \varphi +$$

$$+ \frac{4\beta^2\omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\cos(\omega t) - \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right) \sin(\omega t) - e^{-\beta t} \cos(t\sqrt{\omega_0^2 - \beta^2}) + \right. \\ \left. + \left(\frac{2\beta^2\omega^2 + \omega_0^2(\omega_0^2 - \omega^2)}{2\beta^2\omega^2} \frac{\beta}{\sqrt{\omega_0^2 - \beta^2}} \right) e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2}) \right) \sin \varphi.$$

Выполним ряд преобразований и представим функцию $y(t)$ в следующем виде:

$$\begin{aligned}
 y(t) = & \frac{4\beta^2\omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\left(\sin(\omega t) + \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right) \cos(\omega t) \right) \cos \varphi \right) + \\
 & + \frac{4\beta^2\omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\left(\cos(\omega t) - \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right) \sin(\omega t) \right) \sin \varphi \right) - \\
 & - \frac{4\beta^2\omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} e^{-\beta t} \cos(t\sqrt{\omega_0^2 - \beta^2}) \left(\sin \varphi + \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right) \cos \varphi \right) - \\
 & - \frac{\beta\omega(\omega_0^2 + \omega^2)}{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2} \left(\frac{2\beta}{\sqrt{\omega_0^2 - \beta^2}} \right) e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2}) \left(\cos \varphi - \left(\frac{2\beta^2\omega^2 + \omega_0^2(\omega_0^2 - \omega^2)}{\beta\omega(\omega_0^2 + \omega^2)} \right) \sin \varphi \right).
 \end{aligned}$$

Теперь, воспользуемся следующими тригонометрическими тождествами:

$$\sin \gamma + \frac{b}{a} \cos \gamma = \frac{\sqrt{a^2 + b^2}}{a} \sin \left(\gamma + \arctg \left(\frac{b}{a} \right) \right);$$

$$\cos \gamma - \frac{b}{a} \sin \gamma = \frac{\sqrt{a^2 + b^2}}{a} \cos \left(\gamma + \arctg \left(\frac{b}{a} \right) \right);$$

$$\sin \theta \cos \varphi + \cos \theta \sin \varphi = \sin(\theta + \varphi).$$

Тогда, получаем следующее:

$$\begin{aligned}
 y(t) = & \frac{2\beta\omega}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}} \sin \left(\omega t + \varphi + \arctg \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right) \right) - \\
 & - \frac{2\beta\omega}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}} e^{-\beta t} \cos(t\sqrt{\omega_0^2 - \beta^2}) \sin \left(\varphi + \arctg \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right) \right) - \\
 & - \frac{\sqrt{\omega_0^4 + \beta^2\omega^2}}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}} \left(\frac{2\beta}{\sqrt{\omega_0^2 - \beta^2}} \right) e^{-\beta t} \sin(t\sqrt{\omega_0^2 - \beta^2}) \cos \left(\varphi + \arctg \left(\frac{2\beta^2\omega^2 + \omega_0^2(\omega_0^2 - \omega^2)}{\beta\omega(\omega_0^2 + \omega^2)} \right) \right).
 \end{aligned}$$

Введем следующие обозначения:

$$A(\omega) = \frac{2\beta\omega}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}}; \quad \Psi(\omega) = \arctg \left(\frac{\omega_0^2 - \omega^2}{2\beta\omega} \right);$$

$$E(\omega) = \frac{\sqrt{\omega_0^4 + \beta^2\omega^2}}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}}; \quad \Theta(\omega) = \arctg \left(\frac{2\beta^2\omega^2 + \omega_0^2(\omega_0^2 - \omega^2)}{\beta\omega(\omega_0^2 + \omega^2)} \right).$$

Тогда, окончательно получаем функцию выходного напряжения (рис. 1.14):

$$\begin{aligned}
 y(t) = & A(\omega) \sin(\omega t + \varphi + \Psi(\omega)) - \\
 & - e^{-\beta t} \left(A(\omega) \cos(t\sqrt{\omega_0^2 - \beta^2}) \sin(\varphi + \Psi(\omega)) + E(\omega) \left(\frac{2\beta}{\sqrt{\omega_0^2 - \beta^2}} \right) \sin(t\sqrt{\omega_0^2 - \beta^2}) \cos(\varphi + \Theta(\omega)) \right).
 \end{aligned}$$

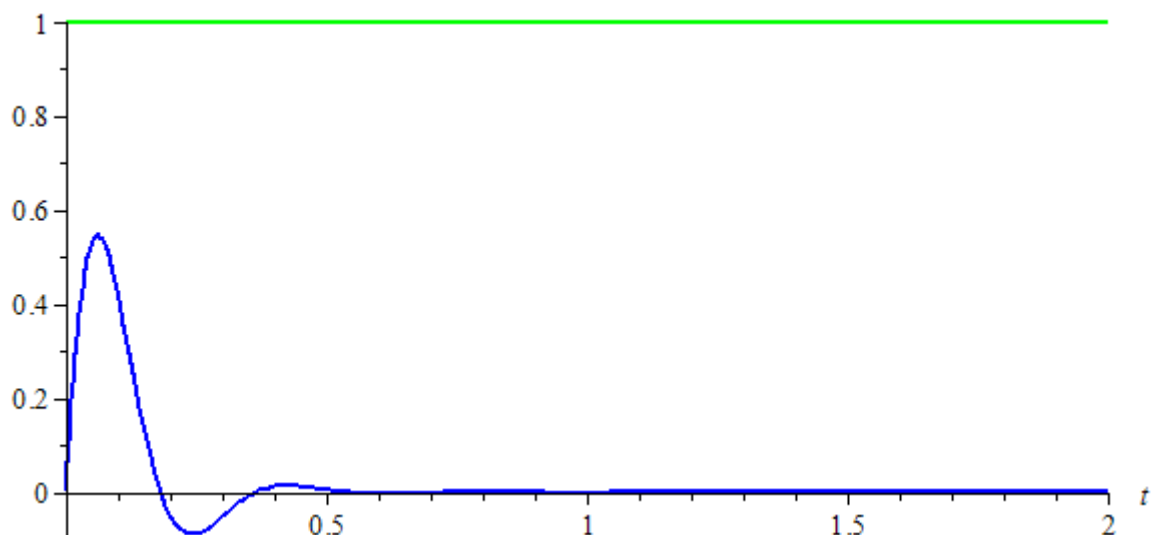


Рис. 1.10. Графики функций входных и выходных напряжений фильтра при параметрах фильтра $\beta = 10$ и $\omega_0 = 20$ и единичной функции на входе фильтра.

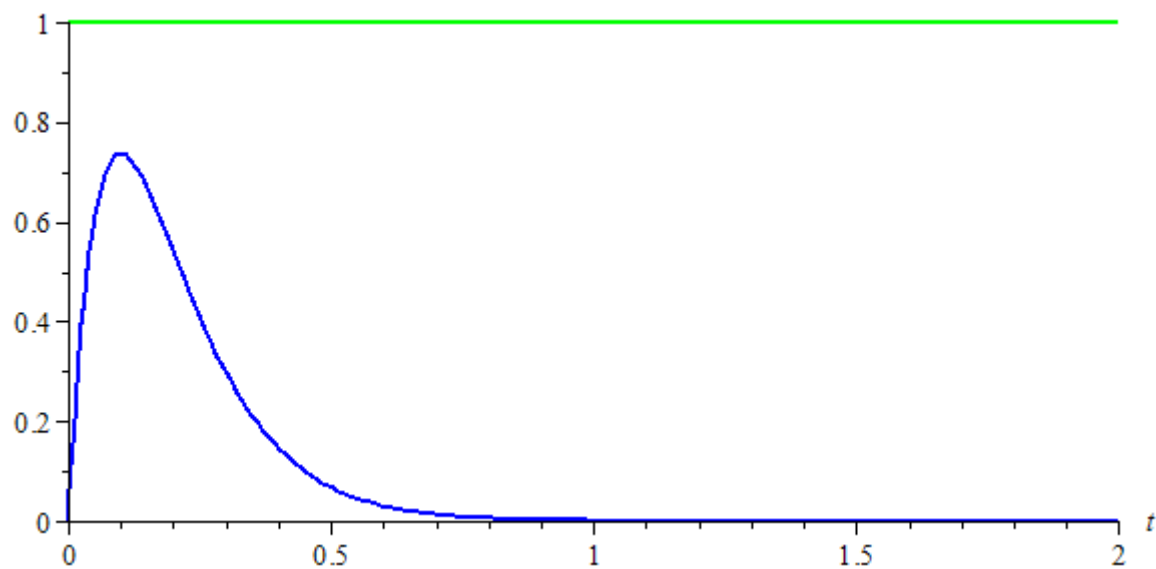


Рис. 1.11. Графики функций входных и выходных напряжений фильтра при параметрах фильтра $\beta = 10$ и $\omega_0 = 10$ и единичной функции на входе фильтра.

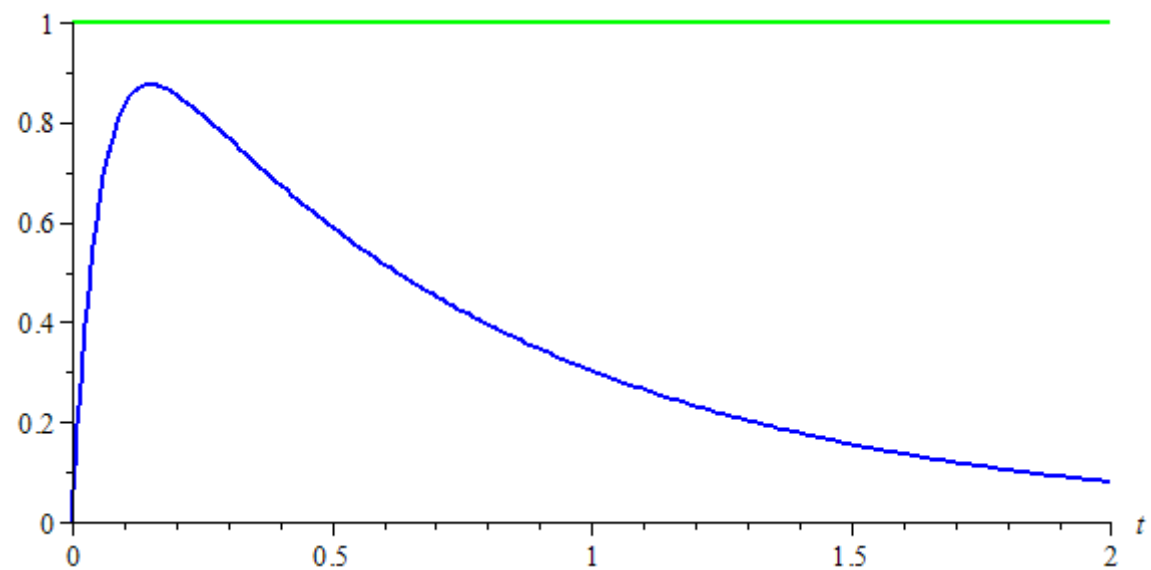


Рис. 1.12. Графики функций входных и выходных напряжений фильтра при параметрах фильтра $\beta = 10$ и $\omega_0 = 5$ и единичной функции на входе фильтра.

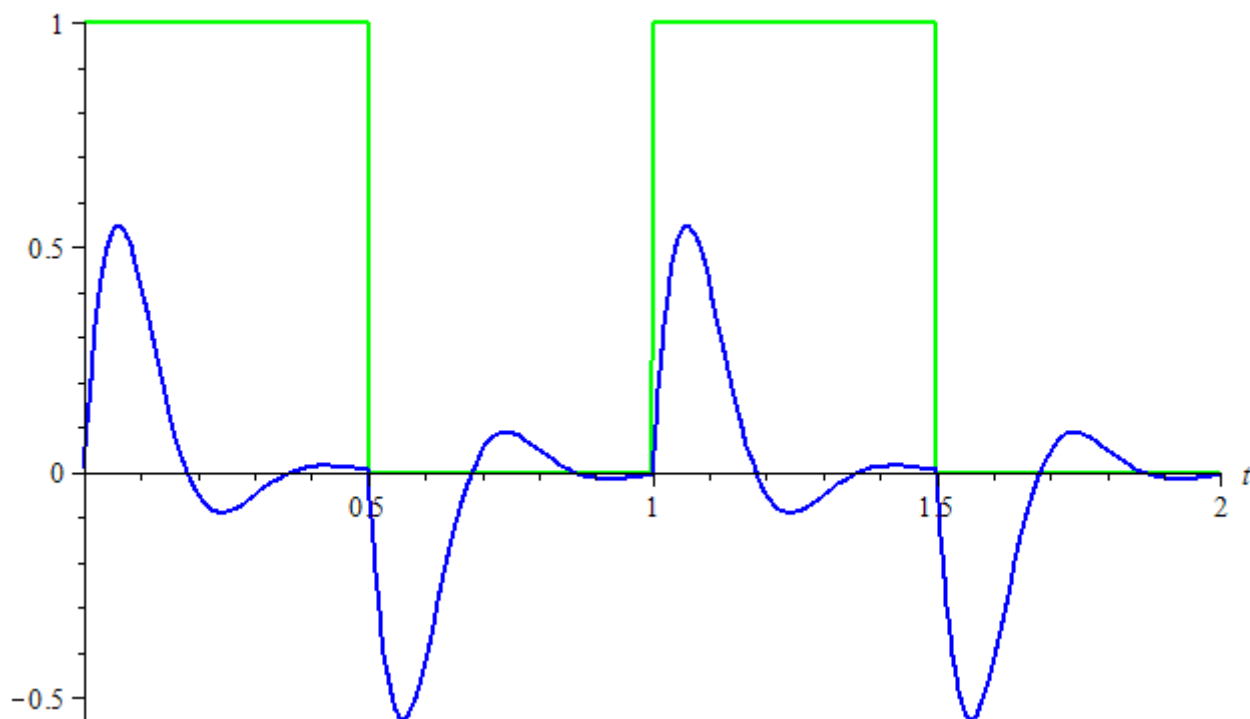


Рис. 1.13. Графики функций входных и выходных напряжений фильтра при параметрах фильтра $\beta = 10$ и $\omega_0 = 20$ и функции прямоугольных колебаний с периодом $T = 1$ на входе фильтра.

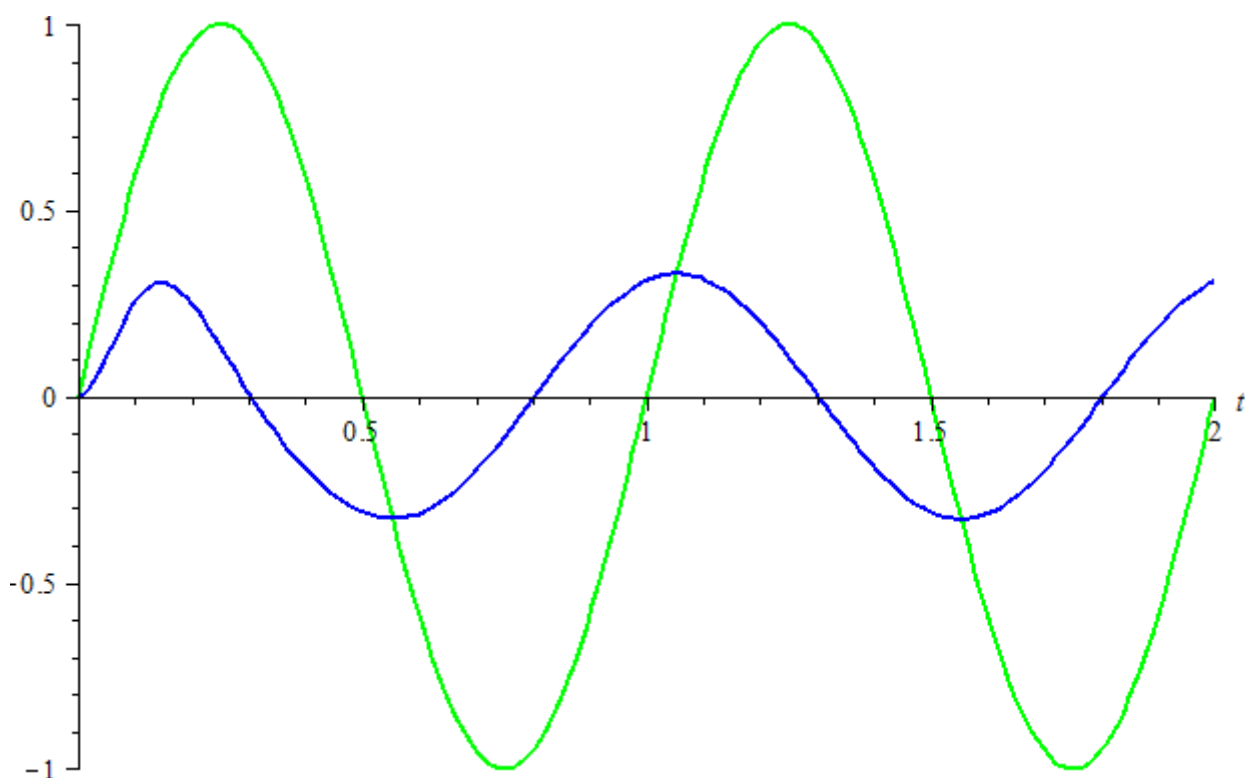


Рис. 1.14. Графики функций входных и выходных напряжений при параметрах фильтра $\beta = 10$ и $\omega_0 = 20$ и синусоидальной функции с круговой частотой $\omega = 2\pi \cdot 1$ и фазой $\varphi = 0$ на входе фильтра.

Анализ в частотной области. Выше при анализе фильтра мы вывели следующую формулу, связывающую изображения функций входного и выходного напряжений:

$$Y(p) = \frac{2\beta p}{p^2 + 2\beta p + \omega_0^2} X(p).$$

Соответственно, комплексная передаточная функция: $W(p) = \frac{2\beta p}{p^2 + 2\beta p + \omega_0^2}$.

Вычислим теперь частотную передаточную функцию, и преобразуем ее так, чтобы можно было отдельно выделить ее действительную и мнимую части. Тогда, получаем:

$$W(i\omega) = \frac{2\beta i\omega}{-\omega^2 + 2\beta i\omega + \omega_0^2} \cdot \frac{(\omega_0^2 - \omega^2) - 2\beta i\omega}{(\omega_0^2 - \omega^2) - 2\beta i\omega} = \frac{4\beta^2 \omega^2}{(\omega_0^2 - \omega^2)^2 + 4\beta^2 \omega^2} + i \frac{2\beta \omega (\omega_0^2 - \omega^2)}{(\omega_0^2 - \omega^2)^2 + 4\beta^2 \omega^2}.$$

Тогда, амплитудно-частотная характеристика (рис. 1.15) фильтра:

$$A(\omega) = \frac{2\beta \omega}{(\omega_0^2 - \omega^2)^2 + 4\beta^2 \omega^2} \sqrt{4\beta^2 \omega^2 + (\omega_0^2 - \omega^2)^2} = \frac{2\beta \omega}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2 \omega^2}}.$$

Соответственно, фазово-частотная характеристика (рис. 1.16) фильтра:

$$\Psi(\omega) = \arctg\left(\frac{\omega_0^2 - \omega^2}{2\beta \omega}\right).$$

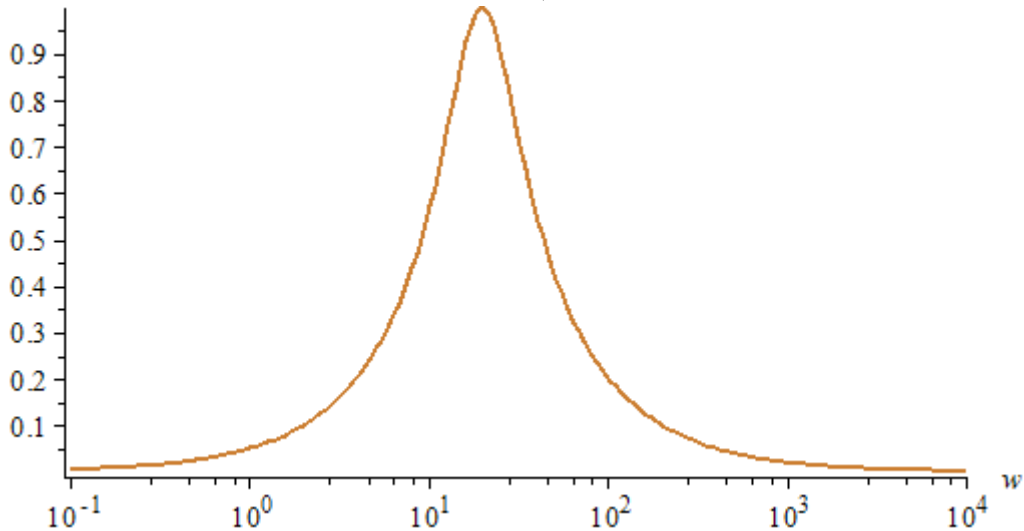


Рис. 1.15. График амплитудно-частотной характеристики фильтра при $\beta = 10$ и $\omega_0 = 20$.

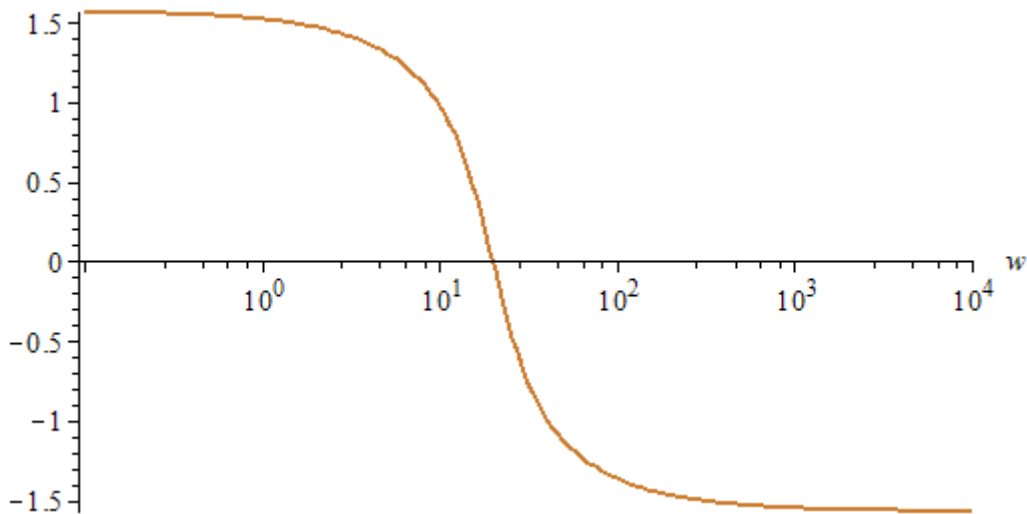


Рис. 1.16. График фазово-частотной характеристики фильтра при $\beta = 10$ и $\omega_0 = 20$.

2. Математическое моделирование задач теории надежности

2.1. Исходная цепь Маркова в модели надежности технических систем

В общем случае цепь Маркова с непрерывным временем (рис. 2.1) в моделях надежности технических систем содержит множество состояний E , состоящее из подмножества работоспособных состояний E_+ и подмножества неработоспособных состояний E_- . В свою очередь, подмножество E_+ содержит подмножество H_+ граничных работоспособных состояний, имеющих прямые переходы в неработоспособные состояния. Аналогично, подмножество E_- содержит подмножество H_- граничных неработоспособных состояний, имеющих прямые переходы в работоспособные состояния.

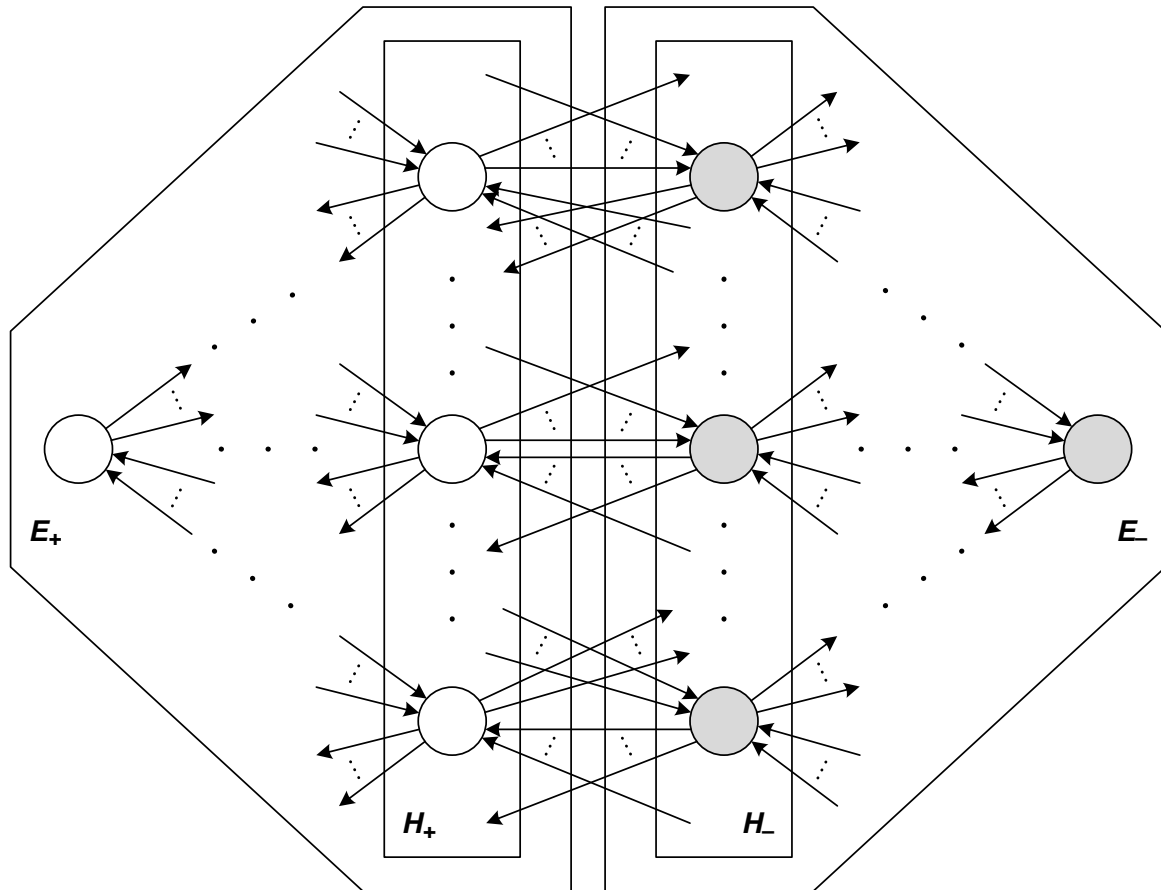


Рис. 2.1. Исходная цепь Маркова в модели надежности системы.

У каждого состояния $i \in E$ (рис. 2.2) могут быть как входящие переходы с интенсивностями γ_{ri} из подмножества состояний $\{r \in R_i\}$, имеющих переходы в i -е состояние, так и исходящие переходы с интенсивностями γ_{is} в подмножество состояний $\{s \in S_i\}$, имеющих переходы из i -го состояния.

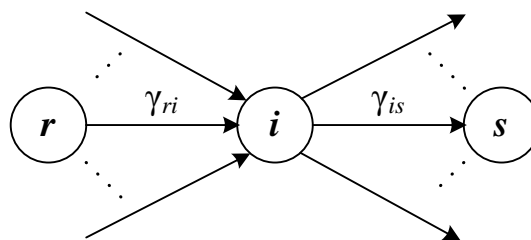


Рис. 2.2. Входящие и исходящие переходы для состояния i .

Математически модель цепи Маркова с непрерывным временем представляет собой систему дифференциальных уравнений Колмогорова-Чепмена для каждого состояния $i \in E$, решив которую можно найти функции вероятностей $P_i(t)$ всех состояний системы, при заданных начальных условиях $P_i(0)$ для всех состояний и, учитывая, что сумма вероятностей всех состояний системы всегда равна 1:

$$\begin{cases} \sum_{i \in E} P_i(t) = 1; \\ \forall i \in E: \frac{dP_i(t)}{dt} = \sum_{r \in R_i} (\gamma_{ri} P_r(t)) - P_i(t) \sum_{s \in S_i} \gamma_{is}. \end{cases} \quad (2.1)$$

Решив систему дифференциальных уравнений при некоторых заданных начальных условиях: обычно одно из работоспособных состояний k^* выбирается начальным и для него $P_{k^*}(0) = 1$, а для остальных состояний $P_i(0) = 0$, и, получив функции вероятностей $P_i(t)$, можно вывести функцию нестационарного коэффициента готовности системы как сумму функций вероятностей всех работоспособных состояний:

$$K(t) = \sum_{i \in E_+} P_i(t). \quad (2.2)$$

В стационарном случае $t \rightarrow \infty$, производные функций вероятностей стремятся к нулю $dP_i(t)/dt \rightarrow 0$, и система превращается в систему линейных алгебраических уравнений, решив которую можно найти, стационарные вероятности P_i всех состояний системы:

$$\begin{cases} \sum_{i \in E} P_i = 1; \\ \forall i \in E: \sum_{r \in R_i} (\gamma_{ri} P_r) - P_i \sum_{s \in S_i} \gamma_{is} = 0. \end{cases} \quad (2.3)$$

Решив систему линейных алгебраических уравнений и получив вероятности P_i , можно рассчитать стационарный коэффициент готовности системы как сумму стационарных вероятностей всех работоспособных состояний системы:

$$K_s = \sum_{i \in E_+} P_i. \quad (2.4)$$

Среднее время наработки на отказ системы T_F рассчитывается по следующей топологической формуле:

$$T_F = \sum_{i \in E_+} P_i / \left(\sum_{i \in H_+} P_i \sum_{j \in E_-} \gamma_{ij} \right). \quad (2.5)$$

Среднее время наработки на отказ системы вычисляется как отношение суммы стационарных вероятностей всех работоспособных состояний к взвешенной сумме стационарных вероятностей граничных работоспособных состояний. В качестве весов для каждого из граничных работоспособных состояний выступают суммы интенсивностей переходов из граничного состояния во все неработоспособные состояния.

Наконец, также можно рассчитать среднее время восстановления по топологической формуле, аналогичной предыдущей, с той лишь разницей, что в числителе формулы используется сумма стационарных вероятностей всех неработоспособных состояний:

$$T_R = \sum_{i \in E_-} P_i / \left(\sum_{i \in H_+} P_i \sum_{j \in E_-} \gamma_{ij} \right). \quad (2.6)$$

Особо отметим, что стационарный коэффициент готовности связан со средним временем наработки на отказ и средним временем восстановления простым тождеством:

$$K_s = \frac{T_F}{T_F + T_R}. \quad (2.7)$$

Пример. Имеется вычислительная система, состоящая из трех функционально одинаковых, но различных по параметрам надежности вычислительных узлов.

На узлах выполняются одинаковые расчеты, затем итоговый результат выбирается по мажоритарному принципу. Соответственно, система считается работоспособной, если хотя бы два из трех узлов исправны и выдают корректный результат.

Узлы независимы по отказам и по восстановлению.

Интенсивности отказов узлов: $\lambda_1 = 1/8760 \text{ час}^{-1}$, $\lambda_2 = 2/8760 \text{ час}^{-1}$ и $\lambda_3 = 3/8760 \text{ час}^{-1}$.

Интенсивности восстановлений узлов: $\mu_1 = 1/24 \text{ час}^{-1}$, $\mu_2 = 3/24 \text{ час}^{-1}$ и $\mu_3 = 2/24 \text{ час}^{-1}$.

Ниже на рис. 2.3 представлена цепь Маркова, описывающая модель надежности рассматриваемой вычислительной системы:

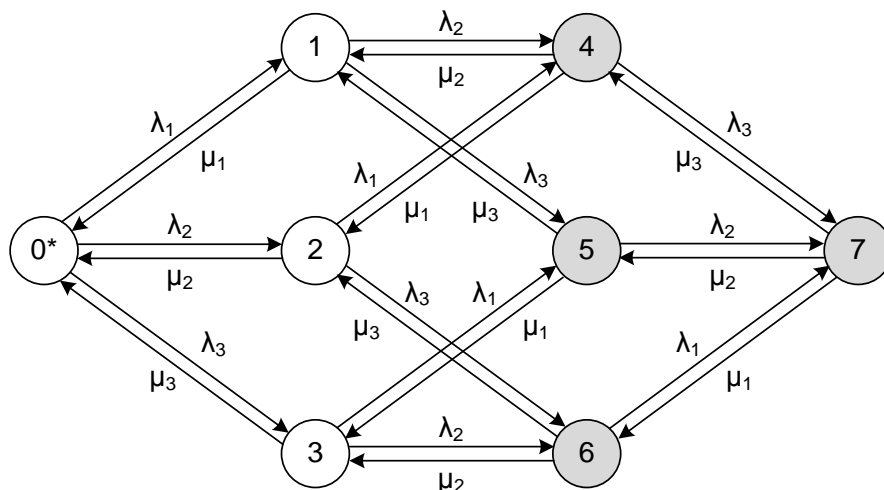


Рис. 2.3. Цепь Маркова для мажоритарной вычислительной системы с тремя узлами.

Состояние 0 – все узлы работоспособны. Состояние 1 – узел 1 неработоспособен, 2 – узел 2 неработоспособен, 3 – узел 3 неработоспособен. Состояние 4 – узлы 1 и 2 неработоспособны, 5 – узлы 1 и 3 неработоспособны, 6 – узлы 2 и 3 неработоспособны. Состояние 7 – все узлы неработоспособны. Состояния 0, 1, 2 и 3 системы считаются работоспособными, при которых все три или два из трех узла работоспособны.

Начальное состояние системы – 0 (все узлы работоспособны).

Рассчитаем значение нестационарного коэффициента готовности при $t = 100$ часов.

Для этого составляем систему дифференциальных уравнений Колмогорова-Чепмена для вышеприведенной цепи Маркова, с учетом того, что состояние 0 является начальным:

$$\left\{ \begin{array}{l} P_0(0) = 1; P_1(0) = P_2(0) = P_3(0) = P_4(0) = P_5(0) = P_6(0) = P_7(0) = 0; \\ P_0(t) + P_1(t) + P_2(t) + P_3(t) + P_4(t) + P_5(t) + P_6(t) + P_7(t) = 1; \\ dP_0(t)/dt = -(\lambda_1 + \lambda_2 + \lambda_3)P_0(t) + \mu_1 P_1(t) + \mu_2 P_2(t) + \mu_3 P_3(t); \\ dP_1(t)/dt = \lambda_1 P_0(t) - (\mu_1 + \lambda_2 + \lambda_3)P_1(t) + \mu_2 P_4(t) + \mu_3 P_5(t); \\ dP_2(t)/dt = \lambda_2 P_0(t) - (\mu_2 + \lambda_1 + \lambda_3)P_2(t) + \mu_1 P_4(t) + \mu_3 P_6(t); \\ dP_3(t)/dt = \lambda_3 P_0(t) - (\mu_3 + \lambda_1 + \lambda_2)P_3(t) + \mu_1 P_5(t) + \mu_2 P_6(t); \\ dP_4(t)/dt = \lambda_2 P_1(t) + \lambda_1 P_2(t) - (\lambda_3 + \mu_1 + \mu_2)P_4(t) + \mu_3 P_7(t); \\ dP_5(t)/dt = \lambda_3 P_1(t) + \lambda_1 P_3(t) - (\lambda_2 + \mu_1 + \mu_3)P_5(t) + \mu_2 P_7(t); \\ dP_6(t)/dt = \lambda_3 P_2(t) + \lambda_2 P_3(t) - (\lambda_1 + \mu_2 + \mu_3)P_6(t) + \mu_1 P_7(t); \\ dP_7(t)/dt = \lambda_3 P_4(t) + \lambda_2 P_5(t) + \lambda_1 P_6(t) - (\mu_1 + \mu_2 + \mu_3)P_7(t). \end{array} \right. \quad (2.8)$$

Соответственно, функция для нестационарного коэффициента готовности выводится как сумма функций вероятностей всех работоспособных состояний системы:

$$K(t) = P_0(t) + P_1(t) + P_2(t) + P_3(t). \quad (2.9)$$

Решая систему дифференциальных уравнений численным методом на ЭВМ для заданных в примере значений интенсивностей отказов и восстановлений узлов, получаем значение нестационарного коэффициента готовности при $t = 100$ часов:

$$K(t) \approx 0,9999766674.$$

Теперь рассчитаем стационарный коэффициент готовности, среднее время наработки на отказ и среднее время восстановления рассматриваемой системы. Имеем следующую стационарную систему уравнений Колмогорова-Чепмена для цепи Маркова:

$$\begin{cases} P_0 + P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 = 1; \\ 0 = -(\lambda_1 + \lambda_2 + \lambda_3)P_0 + \mu_1 P_1 + \mu_2 P_2 + \mu_3 P_3; \\ 0 = \lambda_1 P_0 - (\mu_1 + \lambda_2 + \lambda_3)P_1 + \mu_2 P_4 + \mu_3 P_5; \\ 0 = \lambda_2 P_0 - (\mu_2 + \lambda_1 + \lambda_3)P_2 + \mu_1 P_4 + \mu_3 P_6; \\ 0 = \lambda_3 P_0 - (\mu_3 + \lambda_1 + \lambda_2)P_3 + \mu_1 P_5 + \mu_2 P_6; \\ 0 = \lambda_2 P_1 + \lambda_1 P_2 - (\lambda_3 + \mu_1 + \mu_2)P_4 + \mu_3 P_7; \\ 0 = \lambda_3 P_1 + \lambda_1 P_3 - (\lambda_2 + \mu_1 + \mu_3)P_5 + \mu_2 P_7; \\ 0 = \lambda_3 P_2 + \lambda_2 P_3 - (\lambda_1 + \mu_2 + \mu_3)P_6 + \mu_1 P_7; \\ 0 = \lambda_3 P_4 + \lambda_2 P_5 + \lambda_1 P_6 - (\mu_1 + \mu_2 + \mu_3)P_7. \end{cases} \quad (2.10)$$

Решая систему уравнений, выводим формулы для стационарных вероятностей всех состояний системы:

$$\begin{cases} P_0 = \frac{1}{D}; \quad P_1 = \frac{\lambda_1/\mu_1}{D}; \quad P_2 = \frac{\lambda_2/\mu_2}{D}; \quad P_3 = \frac{\lambda_3/\mu_3}{D}; \\ P_4 = \frac{(\lambda_1/\mu_1)(\lambda_2/\mu_2)}{D}; \quad P_5 = \frac{(\lambda_1/\mu_1)(\lambda_3/\mu_3)}{D}; \quad P_6 = \frac{(\lambda_2/\mu_2)(\lambda_3/\mu_3)}{D}; \\ P_7 = \frac{(\lambda_1/\mu_1)(\lambda_2/\mu_2)(\lambda_3/\mu_3)}{D}; \quad D = \left(1 + \frac{\lambda_1}{\mu_1}\right) \left(1 + \frac{\lambda_2}{\mu_2}\right) \left(1 + \frac{\lambda_3}{\mu_3}\right). \end{cases} \quad (2.11)$$

Тогда, учитывая, что в состояниях 0-3 система работоспособна, имеем формулу для стационарного коэффициента готовности:

$$K_S = P_0 + P_1 + P_2 + P_3 = \left(1 + \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} + \frac{\lambda_3}{\mu_3}\right) / \left(\left(1 + \frac{\lambda_1}{\mu_1}\right) \left(1 + \frac{\lambda_2}{\mu_2}\right) \left(1 + \frac{\lambda_3}{\mu_3}\right)\right). \quad (2.12)$$

Далее, согласно топологическому методу и учитывая, что состояния 0, 1, 2 и 3 считаются работоспособными, а также учитывая интенсивности переходов из граничных работоспособных состояний 1, 2 и 3 в неработоспособные состояния, среднее время наработки на отказ может вычислено по следующей формуле:

$$T_F = \frac{P_0 + P_1 + P_2 + P_3}{(\lambda_2 + \lambda_3)P_1 + (\lambda_1 + \lambda_3)P_2 + (\lambda_1 + \lambda_2)P_3} = \frac{1 + \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} + \frac{\lambda_3}{\mu_3}}{\frac{\lambda_1(\lambda_2 + \lambda_3)}{\mu_1} + \frac{\lambda_2(\lambda_1 + \lambda_3)}{\mu_2} + \frac{\lambda_3(\lambda_1 + \lambda_2)}{\mu_3}}. \quad (2.13)$$

Аналогично, согласно топологическому методу и учитывая, что состояния 4, 5, 6 и 7 считаются неработоспособными, а также учитывая интенсивности переходов из граничных работоспособных состояний 1, 2 и 3 в неработоспособные состояния, среднее время наработки на восстановление может вычислено по следующей формуле:

$$T_R = \frac{P_4 + P_5 + P_6 + P_7}{(\lambda_2 + \lambda_3)P_1 + (\lambda_1 + \lambda_3)P_2 + (\lambda_1 + \lambda_2)P_3} = \frac{1 + \frac{\mu_1}{\lambda_1} + \frac{\mu_2}{\lambda_2} + \frac{\mu_3}{\lambda_3}}{\frac{\mu_1(\mu_2 + \mu_3)}{\lambda_1} + \frac{\mu_2(\mu_1 + \mu_3)}{\lambda_2} + \frac{\mu_3(\mu_1 + \mu_2)}{\lambda_3}}. \quad (2.14)$$

Подставляя заданные в примере значения интенсивностей отказов и восстановлений узлов, получаем значения показателей надежности:

1) Стационарный коэффициент готовности системы:

$$K_S \approx 0,9999764153.$$

2) Среднее время наработки на отказ системы:

$$T_F = 265080 \text{ часов.}$$

3) Среднее время восстановления системы:

$$T_R \approx 6,25 \text{ часов.}$$

2.2. Преобразованная цепь Маркова для расчета среднего времени наработки до первого отказа технических систем

Для расчета среднего времени наработки до первого отказа необходимо применить дополнительное специальное преобразование к исходной цепи Маркова (рис. 2.1), но так чтобы среднее время наработки на отказ преобразованной цепи совпадало со средним временем наработки до первого отказа исходной цепи Маркова. Тогда, очевидно, среднее время наработки до первого отказа можно будет легко вычислить топологическим методом по преобразованной цепи. Преобразование сводится к следующим действиям:

- Удаление всех исходящих переходов из всех неработоспособных состояний подмножества E_- в любые другие состояния.
- Удаление неработоспособных состояний, не имеющих переходы из работоспособных состояний, поскольку в результате предыдущей операции у них не остается ни входящих, ни исходящих переходов.
- Объединение всех оставшихся тупиковых неработоспособных состояний в одно агрегированное неработоспособное состояние F. При этом для каждого граничного работоспособного состояния все его переходы в неработоспособные состояния объединяются в один агрегированный переход в неработоспособное состояние F с суммированием соответствующих интенсивностей.
- Добавление фиктивного перехода с некоторой ненулевой интенсивностью δ из агрегированного состояния F в начальное работоспособное состояние k^* .

В результате преобразования получившаяся цепь Маркова (рис. 2.4) будет состоять из подмножества работоспособных состояний E_+ исходной цепи Маркова и одного агрегированного неработоспособного состояния F.

Переходы между работоспособными состояниями системы и их интенсивности остаются такими же, как и в исходной цепи Маркова, а для каждого граничного работоспособного состояния подмножества H_+ его переходы в неработоспособные состояния подмножества E_- заменяются агрегированным переходом в неработоспособное состояние F с суммированием соответствующих интенсивностей:

$$\forall i \in H_+ : \gamma_{iF} = \sum_{j \in E_-} \gamma_{ij}. \quad (2.15)$$

Далее, добавляется фиктивный переход из состояния F в начальное работоспособное состояние k^* с ненулевой интенсивностью δ :

$$\gamma_{F,k^*} = \delta. \quad (2.16)$$

Далее, для преобразованной цепи Маркова можно составить стационарную систему уравнений Колмогорова-Чепмена:

$$\begin{cases} \sum_{i \in E_+} \tilde{P}_i + \tilde{P}_F = 1; \\ \forall i \in E_+ \cup \{F\} : \sum_{r \in R_i} (\gamma_{ri} \tilde{P}_r) - \tilde{P}_i \sum_{s \in S_i} \gamma_{is} = 0. \end{cases} \quad (2.17)$$

Наконец, используя стационарные вероятности P_i , можно рассчитать среднее время наработки на отказ преобразованной цепи, которое совпадает также и со средней наработкой до первого отказа, по следующей формуле топологического метода:

$$T_{FF} = \sum_{i \in E_+} \tilde{P}_i / \sum_{i \in H_+} \tilde{P}_i \gamma_{iF}. \quad (2.18)$$

Равенство среднего времени наработки до отказа среднему времени наработки на отказ в преобразованной цепи гарантируется тем, что благодаря фиктивному переходу система после первого отказа (равно как и последующих отказов) всегда возвращается в начальное состояние k^* и в среднем на отказ наработывает столько же времени, сколько при наработке до первого отказа.

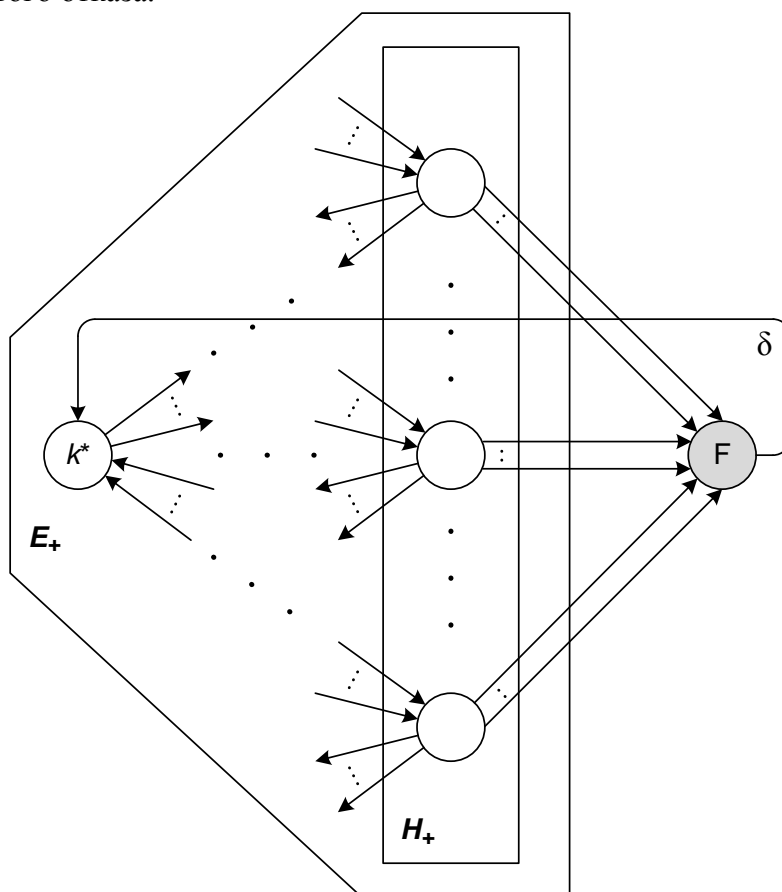


Рис. 2.4. Преобразованная цепь Маркова в обобщенной модели надежности системы для расчета среднего времени наработки до первого отказа.

Пример. Рассмотрим приведенную выше мажоритарную вычислительную систему, состоящую из трех вычислительных узлов, и вычислим для нее среднюю наработку до первого отказа при заданном начальном состоянии $k^* = 0$.

Для этого сначала выполняем преобразование исходной цепи Маркова мажоритарной вычислительной системы (рис. 2.3):

- 1) Удаляем все исходящие переходы из неработоспособных состояний 4-7.
- 2) Поскольку у состояния 7 после данной операции не остается ни входящих, ни исходящих переходов, то мы его также удаляем.
- 3) Объединяем все неработоспособные состояния 4-6 в одно агрегированное состояние F. При этом для каждого граничного работоспособного состояния 1-3 переходы из них в неработоспособные состояния 4-6 объединяем в агрегированные переходы с суммированием соответствующих интенсивностей и заводим их в состояние F.
- 4) Добавляем фиктивный переход с интенсивностью δ из агрегированного неработоспособного состояния F в начальное работоспособное состояние 0.

В результате получаем следующую преобразованную цепь Маркова (рис. 2.5) для рассматриваемой мажоритарной вычислительной системы:

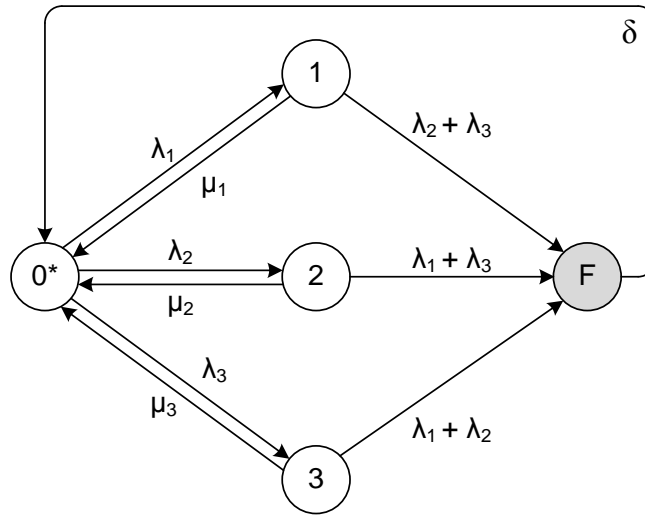


Рис. 2.5. Преобразованная цепь Маркова для мажоритарной вычислительной системы.

Далее, составляем стационарную систему уравнений Колмогорова-Чепмена для преобразованной цепи Маркова:

$$\begin{cases} \tilde{P}_0 + \tilde{P}_1 + \tilde{P}_2 + \tilde{P}_3 + \tilde{P}_F = 1; \\ 0 = -(\lambda_1 + \lambda_2 + \lambda_3)\tilde{P}_0 + \mu_1\tilde{P}_1 + \mu_2\tilde{P}_2 + \mu_3\tilde{P}_3 + \delta\tilde{P}_F; \\ 0 = \lambda_1\tilde{P}_0 - (\mu_1 + \lambda_2 + \lambda_3)\tilde{P}_1; \\ 0 = \lambda_2\tilde{P}_0 - (\mu_2 + \lambda_1 + \lambda_3)\tilde{P}_2; \\ 0 = \lambda_3\tilde{P}_0 - (\mu_3 + \lambda_1 + \lambda_2)\tilde{P}_3; \\ 0 = (\lambda_2 + \lambda_3)\tilde{P}_1 + (\lambda_1 + \lambda_3)\tilde{P}_2 + (\lambda_1 + \lambda_2)\tilde{P}_3 - \delta\tilde{P}_F. \end{cases} \quad (2.19)$$

Согласно топологическому методу и учитывая, что состояния 0-3 считаются работоспособными, а также интенсивности переходов из граничных работоспособных состояний 1-3 в агрегированное состояние F, среднее время наработки до первого отказа системы вычисляется через стационарные вероятности работоспособных состояний:

$$T_{FF} = \frac{\tilde{P}_0 + \tilde{P}_1 + \tilde{P}_2 + \tilde{P}_3}{(\lambda_2 + \lambda_3)\tilde{P}_1 + (\lambda_1 + \lambda_3)\tilde{P}_2 + (\lambda_1 + \lambda_2)\tilde{P}_3}. \quad (2.20)$$

Теперь выразим из системы уравнений вероятности состояний 1, 2 и 3 через вероятность состояния 0:

$$\begin{cases} \tilde{P}_1 = \frac{\lambda_1}{\mu_1 + \lambda_2 + \lambda_3} \tilde{P}_0; & \tilde{P}_2 = \frac{\lambda_2}{\mu_2 + \lambda_1 + \lambda_3} \tilde{P}_0; & \tilde{P}_3 = \frac{\lambda_3}{\mu_3 + \lambda_1 + \lambda_2} \tilde{P}_0. \end{cases} \quad (2.21)$$

Подставляя полученные выражения в вышеприведенную формулу для расчета T_{FF} , выводим итоговую формулу для расчета среднего времени наработки до первого отказа рассматриваемой мажоритарной вычислительной системы:

$$T_{FF} = \frac{1 + \frac{\lambda_1}{\mu_1 + \lambda_2 + \lambda_3} + \frac{\lambda_2}{\mu_2 + \lambda_1 + \lambda_3} + \frac{\lambda_3}{\mu_3 + \lambda_1 + \lambda_2}}{\frac{\lambda_1(\lambda_2 + \lambda_3)}{\mu_1 + \lambda_2 + \lambda_3} + \frac{\lambda_2(\lambda_1 + \lambda_3)}{\mu_2 + \lambda_1 + \lambda_3} + \frac{\lambda_3(\lambda_1 + \lambda_2)}{\mu_3 + \lambda_1 + \lambda_2}}. \quad (2.22)$$

Наконец, подставляя заданные в примере значения интенсивностей отказов и восстановлений узлов, получаем среднюю наработку до первого отказа:

$$T_{FF} \approx 267165 \text{ часов.}$$

2.3. Редуцированная цепь Маркова для расчета вероятности безотказной работы технических систем

Для расчета вероятности безотказной работы (ВБР) в модели надежности на базе цепи Маркова, согласно теории надежности необходимо специальным образом редуцировать исходную цепь Маркова (рис. 2.1), а затем решить для нее систему дифференциальных уравнений при заданном начальном состоянии. Начальное состояние k^* системы должно быть из подмножества работоспособных состояний E_+ .

Редуцирование исходной цепи Маркова сводится к следующему:

- Удаляются все исходящие переходы из всех неработоспособных состояний подмножества E_- в любые другие состояния, и тем самым они делаются «тупиковыми».
- Удаляются неработоспособные состояния, которые не имеют переходов из работоспособных состояний, поскольку в результате предыдущей операции у них не остается ни входящих, ни исходящих переходов.
- Объединяются все оставшиеся тупиковые неработоспособные состояния в одно агрегированное неработоспособное состояние F . При этом для каждого граничного работоспособного состояния все его переходы в неработоспособные состояния объединяются в один агрегированный переход в неработоспособное состояние F с суммированием соответствующих интенсивностей.

Таким образом, в результате редуцирования цепи Маркова (рис. 2.6) остаются только работоспособные состояния и агрегированное неработоспособное состояние F , имеющее прямые переходы из граничных работоспособных состояний.

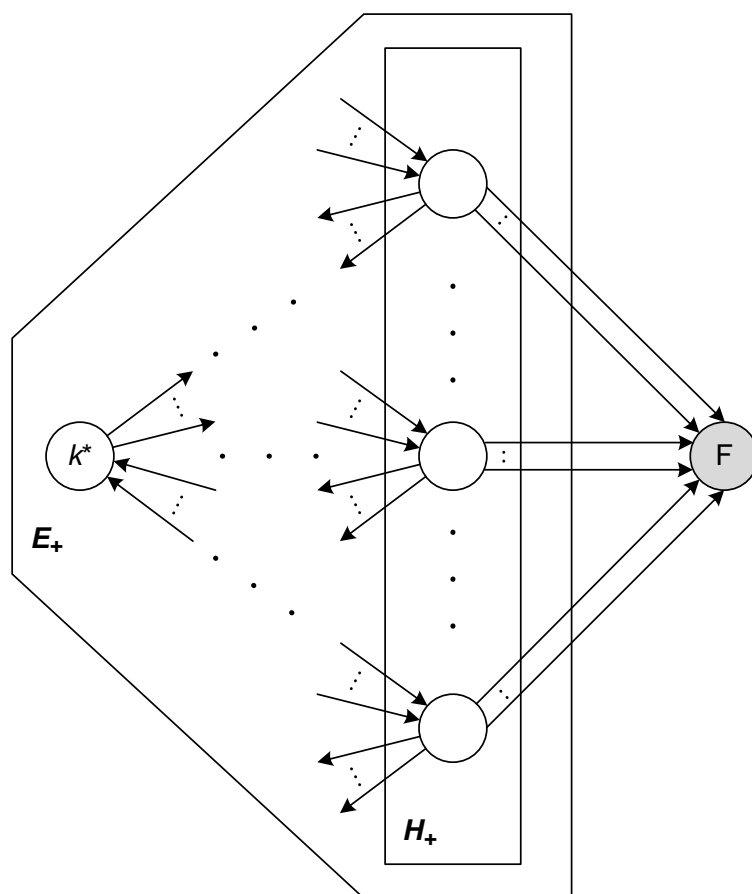


Рис. 2.6. Редуцированная цепь Маркова в обобщенной модели надежности системы для расчета вероятности безотказной работы системы.

Далее, учитывая то, что в редуцированной цепи Маркова работоспособные состояния не имеют входящих переходов из неработоспособных состояний, и, в уравнениях Колмогорова-Чепмена для работоспособных состояний вероятности неработоспособных состояний не участвуют, то система дифференциальных уравнений составляется только для работоспособных состояний. В качестве начальных условий для состояния k^* вероятность задается равной 1, а для всех остальных состояний вероятность задается равной 0.

$$\begin{cases} \hat{P}_{k^*}(0) = 1; \quad \forall i \in E_+ \setminus \{k^*\} : \hat{P}_i(0) = 0; \\ \forall i \in E_+ : \frac{d\hat{P}_i(t)}{dt} = \sum_{r \in R_i} (\gamma_{ri} \hat{P}_r(t)) - \hat{P}_i(t) \sum_{s \in S_i} \gamma_{is}. \end{cases} \quad (2.23)$$

Решив систему уравнений, получаем функции вероятностей всех работоспособных состояний для редуцированной цепи Маркова при заданном начальном состоянии k^* .

Тогда, функция вероятности безотказной работы $P_{U,k^*}(\Delta t)$ системы в течение заданного интервала времени Δt при заданном начальном состоянии k^* выводится как сумма функций вероятностей работоспособных состояний редуцированной цепи Маркова:

$$P_{U,k^*}(\Delta t) = \sum_{i \in E_+} \hat{P}_i(\Delta t). \quad (2.24)$$

Пример. Рассмотрим приведенную выше мажоритарную вычислительную систему, состоящую из трех вычислительных узлов, и вычислим для нее вероятность безотказной работы в течение интервала времени $\Delta t = 100$ часов при начальном состоянии $k^* = 0$.

В исходной цепи Маркова (рис. 2.3) удаляем все исходящие переходы из неработоспособных состояний 4-7. Далее, поскольку у состояния 7 после данной операции не остается ни входящих, ни исходящих переходов, то мы его также удаляем. Наконец, объединяем все неработоспособные состояния 4-6 в одно агрегированное состояние F. При этом для каждого граничного работоспособного состояния 1-3 переходы из них в неработоспособные состояния 4-6 объединяем в агрегированные переходы с суммированием соответствующих интенсивностей и заводим их в состояние F.

В результате получаем следующую редуцированную цепь Маркова (рис. 2.7):

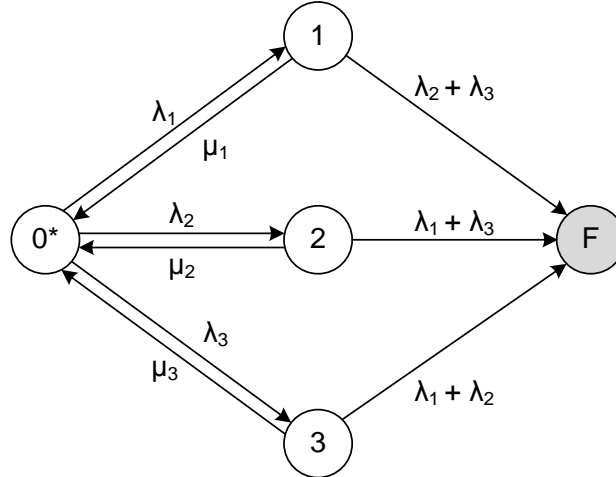


Рис. 2.7. Редуцированная цепь Маркова для мажоритарной вычислительной системы.

Соответственно, система уравнений Колмогорова-Чепмена для всех работоспособных состояний редуцированной цепи Маркова и заданного начального состояния 0:

$$\begin{cases} \hat{P}_0(0) = 1; \quad \hat{P}_1(0) = 0; \quad \hat{P}_2(0) = 0; \quad \hat{P}_3(0) = 0; \\ d\hat{P}_0(t)/dt = -(\lambda_1 + \lambda_2 + \lambda_3)\hat{P}_0(t) + \mu_1\hat{P}_1(t) + \mu_2\hat{P}_2(t) + \mu_3\hat{P}_3(t); \\ d\hat{P}_1(t)/dt = \lambda_1\hat{P}_0(t) - (\mu_1 + \lambda_2 + \lambda_3)\hat{P}_1(t); \\ d\hat{P}_2(t)/dt = \lambda_2\hat{P}_0(t) - (\mu_2 + \lambda_1 + \lambda_3)\hat{P}_2(t); \\ d\hat{P}_3(t)/dt = \lambda_3\hat{P}_0(t) - (\mu_3 + \lambda_1 + \lambda_2)\hat{P}_3(t). \end{cases} \quad (2.25)$$

Решая систему уравнений численным методом на ЭВМ для заданных в примере интенсивностей отказов и восстановлений узлов, вычисляем вероятность безотказной работы системы в течение интервала времени $\Delta t = 100$ часов:

$$P_{U,0}(\Delta t) = \bar{P}_0(\Delta t) + \bar{P}_1(\Delta t) + \bar{P}_2(\Delta t) + \bar{P}_3(\Delta t) \approx 0,99968413.$$

2.4. Расчет нестационарного коэффициента оперативной готовности

Расчет коэффициента оперативной готовности системы, описываемой моделью надежности на базе цепи Маркова, является наиболее трудоемкой задачей теории надежности. Для этого требуется решение системы дифференциальных уравнений Колмогорова-Чепмена для исходной цепи Маркова при заданном начальном состоянии, а также решение систем дифференциальных уравнений для редуцированной цепи Маркова для каждого работоспособного состояния, выбираемого в качестве начального состояния.

На первом этапе рассматривается исходная цепь Маркова (рис. 2.1), выбирается одно конкретное начальное состояние k^* из числа работоспособных состояний, затем составляется и решается система дифференциальных уравнений Колмогорова-Чепмена:

$$\begin{cases} P_{k^*}(0) = 1; \quad \forall i \in E \setminus \{k^*\}: P_i(0) = 0; \\ \forall i \in E: \frac{dP_i(t)}{dt} = \sum_{r \in R_i} (\gamma_{ri} P_r(t)) - P_i(t) \sum_{s \in S_i} \gamma_{is}; \quad \sum_{i \in E} P_i(t) = 1. \end{cases} \quad (2.26)$$

В результате решения выводятся функции вероятностей $P_i(t)$ для всех состояний исходной цепи Маркова, в том числе для всех работоспособных состояний.

На втором этапе рассматривается редуцированная цепь Маркова (рис. 2.6) и для нее составляется и несколько раз решается система дифференциальных уравнений, содержащая только работоспособные состояния, для различных случаев начальных условий. В каждом случае в качестве начального состояния выбирается конкретное работоспособное состояние $k \in E_+$, и, соответственно, начальное значение вероятности для него задается равным 1, а для остальных состояний начальные значения их вероятностей задаются равными 0.

$$\begin{cases} \bar{P}_k(0) = 1; \quad \forall i \in E_+ \setminus \{k\}: \bar{P}_i(0) = 0; \\ \forall i \in E_+: \frac{d\bar{P}_i(t)}{dt} = \sum_{r \in R_i} (\gamma_{ri} \bar{P}_r(t)) - \bar{P}_i(t) \sum_{s \in S_i} \gamma_{is}. \end{cases} \quad (2.27)$$

После решения системы дифференциальных уравнений для конкретного начального работоспособного состояния $k \in E_+$ выводится соответствующая функция безотказной работы системы в течение заданного интервала времени Δt для рассматриваемого начального работоспособного состояния, как сумма функций вероятностей работоспособных состояний, выведенных в результате решения системы дифференциальных уравнений:

$$P_{U,k}(\Delta t) = \sum_{i \in E_+} \bar{P}_i(\Delta t). \quad (2.28)$$

Тогда, функция коэффициента оперативной готовности, отражающей вероятность того, что система будет находиться в одном из работоспособных состояний в момент времени t и, начиная с этого момента безотказно проработает в течение заданного интервала времени Δt , выводится как сумма произведений соответствующих функций вероятностей работоспособных состояний $k \in E_+$, выведенных из исходной цепи Маркова, на функции вероятности безотказной работы, выведенных из редуцированной цепи Маркова:

$$R(t, \Delta t) = \sum_{k \in E_+} P_k(t) P_{U,k}(\Delta t). \quad (2.29)$$

Пример. Рассмотрим приведенную выше мажоритарную вычислительную систему, состоящую из трех вычислительных узлов, и вычислим для нее коэффициент оперативной готовности в течение заданного интервала $\Delta t = 100$ часов, начиная с заданного момента времени $t = 100$ часов при начальном состоянии $k^* = 0$.

На первом этапе составляем систему дифференциальных уравнений Колмогорова-Чепмена для исходной цепи Маркова рассматриваемой вычислительной системы (рис. 2.3) для расчета вероятностей работоспособных состояний при начальном состоянии 0:

$$\left\{ \begin{array}{l} P_0(0) = 1; P_1(0) = P_2(0) = P_3(0) = P_4(0) = P_5(0) = P_6(0) = P_7(0) = 0; \\ P_0(t) + P_1(t) + P_2(t) + P_3(t) + P_4(t) + P_5(t) + P_6(t) + P_7(t) = 1; \\ dP_0(t)/dt = -(\lambda_1 + \lambda_2 + \lambda_3)P_0(t) + \mu_1 P_1(t) + \mu_2 P_2(t) + \mu_3 P_3(t); \\ dP_1(t)/dt = \lambda_1 P_0(t) - (\mu_1 + \lambda_2 + \lambda_3)P_1(t) + \mu_2 P_4(t) + \mu_3 P_5(t); \\ dP_2(t)/dt = \lambda_2 P_0(t) - (\mu_2 + \lambda_1 + \lambda_3)P_2(t) + \mu_1 P_4(t) + \mu_3 P_6(t); \\ dP_3(t)/dt = \lambda_3 P_0(t) - (\mu_3 + \lambda_1 + \lambda_2)P_3(t) + \mu_1 P_5(t) + \mu_2 P_6(t); \\ dP_4(t)/dt = \lambda_2 P_1(t) + \lambda_1 P_2(t) - (\lambda_3 + \mu_1 + \mu_2)P_4(t) + \mu_3 P_7(t); \\ dP_5(t)/dt = \lambda_3 P_1(t) + \lambda_1 P_3(t) - (\lambda_2 + \mu_1 + \mu_3)P_5(t) + \mu_2 P_7(t); \\ dP_6(t)/dt = \lambda_3 P_2(t) + \lambda_2 P_3(t) - (\lambda_1 + \mu_2 + \mu_3)P_6(t) + \mu_1 P_7(t); \\ dP_7(t)/dt = \lambda_3 P_4(t) + \lambda_2 P_5(t) + \lambda_1 P_6(t) - (\mu_1 + \mu_2 + \mu_3)P_7(t). \end{array} \right. \quad (2.30)$$

В результате решения системы уравнений численным методом для заданного момента времени $t = 100$ часов и при заданных в примере интенсивностях отказов и восстановлений узлов и получаем значения вероятностей работоспособных состояний:

$$\begin{aligned} P_0(t) &\approx 0,9914180076; & P_1(t) &\approx 0,0026744734; \\ P_2(t) &\approx 0,0018108038; & P_3(t) &\approx 0,0040733819. \end{aligned}$$

На втором этапе составляем систему дифференциальных уравнений Колмогорова-Чепмена для редуцированной цепи Маркова рассматриваемой вычислительной системы (рис. 2.7) и решаем ее для каждого случая начального работоспособного состояния $k = 0, 1, 2$ и 3 :

$$\left\{ \begin{array}{l} \hat{P}_k(0) = 1; \quad \forall i \neq k : \hat{P}_i(0) = 0; \\ d\hat{P}_0(t)/dt = -(\lambda_1 + \lambda_2 + \lambda_3)\hat{P}_0(t) + \mu_1 \hat{P}_1(t) + \mu_2 \hat{P}_2(t) + \mu_3 \hat{P}_3(t); \\ d\hat{P}_1(t)/dt = \lambda_1 \hat{P}_0(t) - (\mu_1 + \lambda_2 + \lambda_3)\hat{P}_1(t); \\ d\hat{P}_2(t)/dt = \lambda_2 \hat{P}_0(t) - (\mu_2 + \lambda_1 + \lambda_3)\hat{P}_2(t); \\ d\hat{P}_3(t)/dt = \lambda_3 \hat{P}_0(t) - (\mu_3 + \lambda_1 + \lambda_2)\hat{P}_3(t). \end{array} \right. \quad (2.31)$$

Далее, для каждого случая численным методом вычисляем соответствующую вероятность безотказной работы в течение заданного интервала $\Delta t = 100$ часов, как сумму вероятностей работоспособных состояний $0, 1, 2$ и 3 , вычисленных в результате решения системы при соответствующем начальном работоспособном состоянии $k = 0, 1, 2$ и 3 :

$$\begin{aligned} P_{U,0}(\Delta t) &= \hat{P}_0(\Delta t) + \hat{P}_1(\Delta t) + \hat{P}_2(\Delta t) + \hat{P}_3(\Delta t) \approx 0,99968413; \text{ при } k = 0; \\ P_{U,1}(\Delta t) &= \hat{P}_0(\Delta t) + \hat{P}_1(\Delta t) + \hat{P}_2(\Delta t) + \hat{P}_3(\Delta t) \approx 0,98645611; \text{ при } k = 1; \\ P_{U,2}(\Delta t) &= \hat{P}_0(\Delta t) + \hat{P}_1(\Delta t) + \hat{P}_2(\Delta t) + \hat{P}_3(\Delta t) \approx 0,99607507; \text{ при } k = 2; \\ P_{U,3}(\Delta t) &= \hat{P}_0(\Delta t) + \hat{P}_1(\Delta t) + \hat{P}_2(\Delta t) + \hat{P}_3(\Delta t) \approx 0,99563758; \text{ при } k = 3. \end{aligned}$$

Тогда, коэффициент оперативной готовности рассматриваемой вычислительной системы, отражающий вероятность того, что рассматриваемая система будет находиться в одном из работоспособных состояний в момент времени $t = 100$ часов и, начиная с этого момента, безотказно проработает в течение заданного интервала времени $\Delta t = 100$ часов:

$$R(t, \Delta t) = P_0(t)P_{U,0}(\Delta t) + P_1(t)P_{U,1}(\Delta t) + P_2(t)P_{U,2}(\Delta t) + P_3(t)P_{U,3}(\Delta t) \approx 0,9996024076.$$

3. Математические модели задач комбинаторной оптимизации

3.1. Задача псевдобулевой оптимизации о рюкзаке

3.1.1. Описание модели задачи о рюкзаке и метода ее решения

Одним из ярких примеров задач псевдобулевой оптимизации является задача о рюкзаке. Суть задачи заключается в следующем: задано множество предметов $\{P_j\}$, где $j = 1 \dots n$, различного веса a_j и ценности c_j , а также рюкзак с определенной вместимостью b по весу и требуется найти такой набор предметов, при котором рюкзак их мог бы вместить, а суммарная ценность помещенных вещей была максимальной.

Математическая модель задачи выглядит следующим образом:

$$\begin{cases} \sum_{j=1}^n a_j x_j \leq b; \\ L = \sum_{j=1}^n c_j x_j \rightarrow \max; \\ \forall j \in [1, n]: x_j \in \{0, 1\}; \\ a_j \geq 0; b \geq 0; c_j \geq 0. \end{cases} \quad (3.1)$$

Глобальный оптимум задачи для любых исходных условий может быть гарантированно найден только полным перебором, поэтому чаще используются приближенные методы, дающие некоторое субоптимальное решение. Одним из наиболее известных приближенных методов решения является метод локального поиска.

Сначала введем несколько терминов комбинаторного анализа для упрощения дальнейшего описания метода локального поиска.

Пусть задано n -мерное булево пространство, состоящее из множества точек $X = (x_1, \dots, x_n)$ с булевыми координатами: $x_j \in \{0, 1\}, j = 1 \dots n$.

Определение 1. $D(X_A, X_B)$ будем называть расстоянием между точками X_A и X_B , численно равным числу координат, по которым эти точки отличаются. Согласно метрике Хэмминга, численно расстояние определяется по формуле: $D(X_A, X_B) = \sum_{j=1}^n |x_j^{(A)} - x_j^{(B)}|$.

Определение 2. $S_r(X_0)$ будем называть n -мерной сферой поиска с радиусом r и центром X_0 , состоящей из множества точек, отличающихся от точки X_0 ровно r координатами, то есть: $\forall X \in S_r(X_0) \rightarrow D(X, X_0) = r$.

Определение 3. $Z_r(X_0)$ будем называть зоной поиска с радиусом r и центром X_0 , состоящей из всех сфер с радиусами от 0 до r . Иными словами $\forall X \in Z_r(X_0) \rightarrow \exists q \in [0, r]: X \in S_q(X_0)$, т.е. $Z_r(X_0) = S_0(X_0) \cup \dots \cup S_r(X_0)$.

Рассмотрим теперь кратко суть метода локального поиска для решения задачи псевдобулевой оптимизации.

1) Выбирается некоторая начальная точка X_0 , вычисляется в ней целевая функция. Начальная точка принимается за текущий оптимум $X_{\text{опт}}$.

2) Далее ведется перебор всех точек $X \in Z_1(X_{\text{опт}})$ и в них вычисляется значение целевой функции (в базовом варианте метода локального поиска, радиус зон поиска не задается, он равен строго 1, и на каждом шаге просматриваются точки, отличающиеся не более одной координатой от точки $X_{\text{опт}}$). В процессе перебора возможны две стратегии:

- Просмотр всей зоны $Z_1(X_{\text{опт}})$ целиком и нахождение наилучшей точки X^* , в которой значение целевой функции лучше, чем во всех остальных точках зоны (включая центр зоны – точку $X_{\text{опт}}$). Эта точка принимается за новый текущий оптимум и далее ведется уже поиск вокруг нее.
- Перебор до первого улучшения, когда в ходе перебора, как только находится первая точка, в которой значение целевой функции улучшилось, то эта точка принимается за новый текущий оптимум и далее ведется поиск вокруг нее.

3) Для обеих стратегий работа алгоритма завершается, когда в зоне поиска нельзя найти ни одну точку, в которой значение целевой функции лучше. Очевидно, что при использовании второй стратегии задача решится быстрее, поскольку в этом случае перебирается меньшее количество точек, однако, по этой же причине результат в случае многоэкстремальной целевой функции может быть хуже, чем при использовании первой стратегии поиска в зонах.

Мы рассмотрим метод локального поиска в следующей модификации:

- На уровне поиска внутри зон поиск не останавливается при первом улучшении целевой функции, и просматриваются все нерассмотренные точки зоны поиска.
- Радиус зон поиска не равен жестко единице, а может быть задан как дополнительный параметр r .
- Поскольку при переходе от старой зоны поиска к очередной зоне, центр новой зоны поиска всегда находится в старой зоне: $D(X_T, X_{CT}) \leq r$, а значит старая и очередная зоны пересекаются, то во избежание повторного просмотра точек старой зоны $Z_r(X_{CT})$ при рассмотрении очередной зоны $Z_r(X_T)$, просматриваются только точки $X^* \in Z_r(X_T): D(X^*, X_{CT}) > r$, иными словами просматривается «разностная» область $Z_r(X_T) \setminus Z_r(X_{CT})$. Заметим, что первая зона всегда просматривается целиком.

Для использования локального поиска в качестве выбранного метода решения, преобразуем задачу условной псевдобулевой оптимизации (3.1) в следующий вид:

$$\begin{cases} F = \sum_{j=1}^n c_j x_j - R \cdot \max\{0, \sum_{j=1}^n a_j x_j - b\} \rightarrow \max; \\ \forall j \in [1, n]: x_j \in \{0, 1\}. \end{cases} \quad (3.2)$$

В преобразованной математической модели (3.2) ограничение уже явным образом отсутствует, но оно учитывается при помощи штрафной функции. При рассмотрении задачи (3.1) было отмечено, что для этой задачи элементы a_j , b и c_j – неотрицательные величины. Соответственно, скорректированная целевая функция F в этом случае для всех допустимых точек всегда неотрицательна (штраф в таких случаях равен нулю). В случае же недопустимых точек F может быть как неотрицательной, так и отрицательной величиной – это зависит от выбора коэффициента R . Во избежание ошибок при решении задач, коэффициент R лучше всего выбрать настолько большим, чтобы для любой недопустимой точки скорректированная целевая функция F была отрицательной.

После этого задача (3.1) может быть решена при помощи метода локального поиска.

На рисунке 3.1 представлен алгоритм локального поиска с использованием заданного радиуса r зон поиска, а также с учетом вышерассмотренного подхода к исключению повторного просмотра точек.

В алгоритме используются следующие основные и вспомогательные обозначения:

q – радиус текущей просматриваемой сферы поиска в текущей зоне.

X_T – центр текущей зоны поиска (текущий оптимум).

X_{CT} – центр предыдущей зоны поиска (предыдущий оптимум).

X^* – текущая наилучшая точка в текущей зоне поиска. При переходе в новую зону поиска X_T совпадает с последней наилучшей точкой X^* в предыдущей зоне. Поэтому, когда поиск заканчивается (в последней рассматриваемой зоне не нашлось более лучших точек), последней наилучшей точкой является X_T .

F^* – значение целевой функции в точке X^* .

Flag1 – признак того, что просматривается первая зона поиска, соответственно, X_{CT} еще не существует, поэтому в зоне поиска просматриваются все без исключения точки.

Flag2 – признак того, что в зоне поиска была найдена лучшая точка.

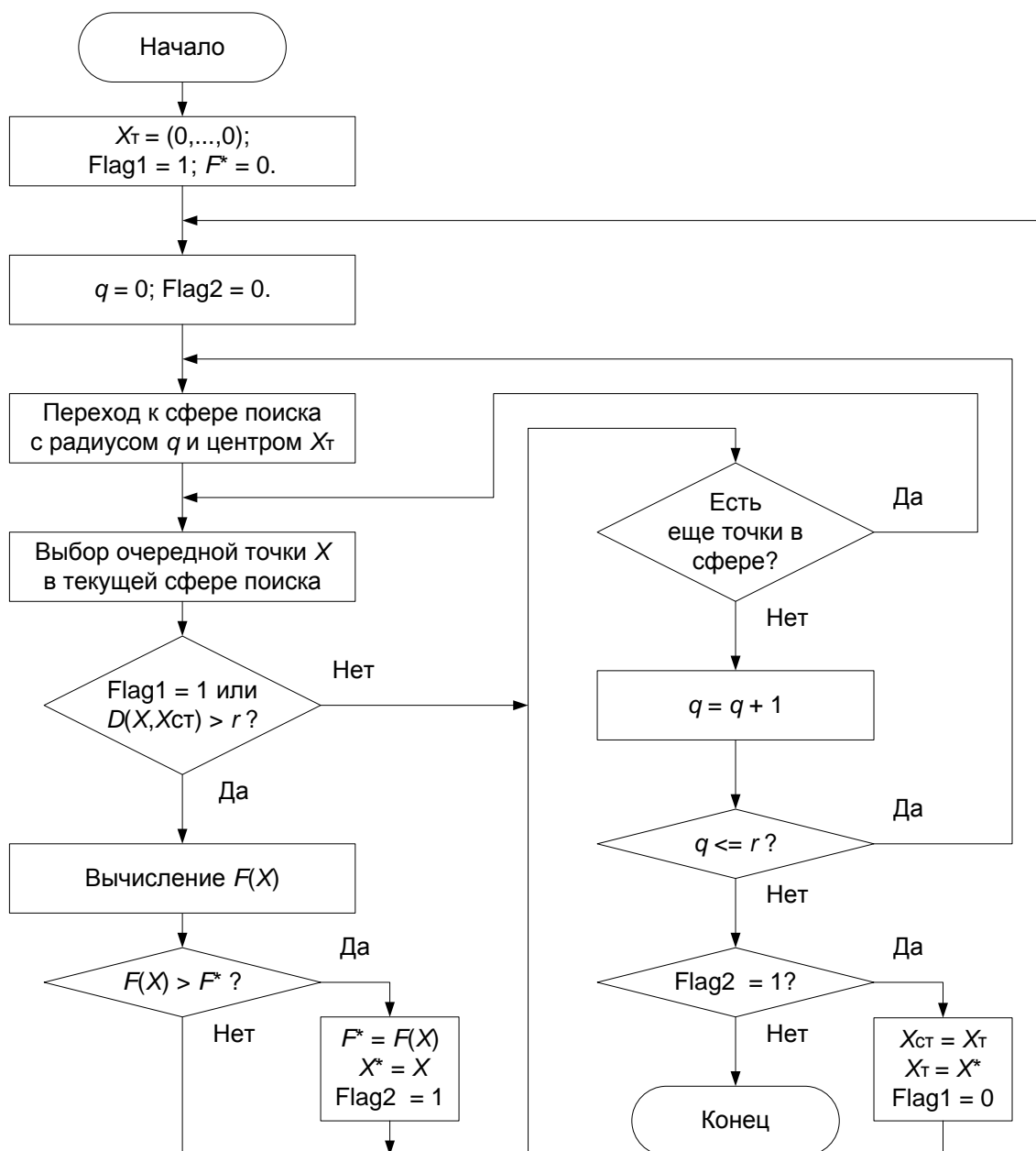


Рис. 3.1. Схема алгоритма модифицированного локального поиска.

Ниже на рис. 3.2. также приведена программная реализация модифицированного метода локального поиска в среде Maple 15. Помимо поиска решения, в программе также реализован подсчет общего числа точек, перебранных методом локального поиска в процессе поиска решения, а также числа просмотренных зон поиска.

3.1.2. Пример решения задачи о рюкзаке методом локального поиска

Рассмотрим следующий простейший пример задачи о рюкзаке при $n = 5$:

$$\begin{cases} 10x_1 + 10x_2 + 30x_3 + 20x_4 + 20x_5 \leq 40; \\ L = 5x_1 + 2x_2 + 4x_3 + 3x_4 + x_5 \rightarrow \max; \\ \forall j \in [1, 5]: x_j \in \{0, 1\}. \end{cases} \quad (3.3)$$

Глобальным оптимумом задачи является решение $X^* = [1, 1, 0, 1, 0]$ и при нем целевая функция достигает значения $L^* = 10$.

Теперь выполним поиск решения задачи, используя программную реализацию рассмотренного выше метода локального поиска, при различных радиусах r зон поиска.

```

> OptLocalSearch:= proc(n,r,A,B,C) local q,j,Comb,D,flag1,flag2,
      X,XOPT,XT,XCT,F,FOPT,temp,combscount,zonescount:
FOPT:= 0: combscount:= 0: zonescount:= 0:
if ((n >= 0) and (r >= 0) and (r <= n)) then
  XT:= Vector[row](n,0): XOPT:= Vector[row](n,0):
  XCT:= Vector[row](n,0): X:= Vector[row](n,0):
  flag1:= 1: flag2:= 1:
  while(flag2 = 1) do
    flag2:= 0:
    for q from 0 to r do
      for Comb in choose(n,q) do
        for j from 1 to n do X[j]:= XT[j]: end do:
        for j in Comb do X[j]:= 1 - XT[j]: end do:
        D:= 0:
        for j from 1 to n do
          if (X[j] <> XCT[j]) then D:= D + 1: end if:
        end do:
        if ((D > r) or (flag1 = 1)) then
          temp:= 0:
          for j from 1 to n do temp:= temp + A[j]*X[j]: end do:
          if (temp <= B) then
            F:= 0:
            for j from 1 to n do F:= F + C[j]*X[j]: end do:
            if ((F > FOPT)) then
              FOPT:= F:
              for j from 1 to n do
                XOPT[j]:= X[j]:
                flag2:= 1:
              end do:
            end if:
          end if:
          combscount:= combscount + 1:
        end if:
      end do:
    end do:
    if (flag2 = 1) then
      for j from 1 to n do
        XCT[j]:= XT[j]:
        XT[j]:= XOPT[j]:
        flag1:= 0:
      end do:
    end if:
    zonescount:= zonescount + 1:
  end do:
end if:
return (XOPT,FOPT,combscount,zonescount):
end proc:

```

Рис. 3.2. Программная реализация метода локального поиска.

Ниже на рис. 3.3 приводятся результаты решения задачи при радиусах $r = 1, 2, 3, 4$ и 5 зон поиска. Помимо самих решений, также выводятся значения целевой функции и общее число точек, перебранных методом при поиске решения, и число просмотренных зон поиска.

```

> n:= 5: A:= [10,10,30,20,20]:
  B:= 40: C:= [5,2,4,3,1]:

> r:= 1:
  LRES:= OptLocalSearch(n,r,A,B,C);
                                LRES:= [ 1 0 1 0 0 ], 9, 14, 3

> r:= 2:
  LRES:= OptLocalSearch(n,r,A,B,C);
                                LRES:= [ 1 0 1 0 0 ], 9, 22, 2

> r:= 3:
  LRES:= OptLocalSearch(n,r,A,B,C);
                                LRES:= [ 1 1 0 1 0 ], 10, 32, 2

> r:= 4:
  LRES:= OptLocalSearch(n,r,A,B,C);
                                LRES:= [ 1 1 0 1 0 ], 10, 32, 2

> r:= 5:
  LRES:= OptLocalSearch(n,r,A,B,C);
                                LRES:= [ 1 1 0 1 0 ], 10, 32, 2

```

Рис. 3.3. Результаты решения задачи при различных радиусах зон поиска.

Нетрудно заметить, что при радиусах $r = 1$ и $r = 2$ метод локального поиска находит субоптимальное решение $X = [1, 0, 1, 0, 0]$, при котором целевая функция достигает значения $L = 9$, просматривая при этом 14 и 22 точек, соответственно.

Кроме того, начиная с радиуса $r = 3$ зон поиска, метод уже выходит на глобальный оптимум $X^* = [1, 1, 0, 1, 0]$, при котором целевая функция достигает значения $L^* = 10$. При этом метод просматривает 32 точки, что уже фактически равносильно полному перебору всех 2^5 точек 5-мерного булевого пространства.

Таким образом, данный пример показывает, что метод локального поиска может найти оптимум при небольших радиусах зон поиска, перебрав ограниченное количество точек, но при этом решение может оказаться субоптимальным.

Соответственно, на глобальный оптимум метод локального поиска может выйти при больших радиусах зон поиска, однако, неизбежной расплатой при этом становится рост объема перебора, вплоть до полного перебора.

3.1.3. Анализ верхней границы объема перебора

Одна из важнейших характеристик метода решения задачи о рюкзаке – это объем перебора, равный общему количеству перебираемых точек n -мерного булевого пространства в процессе решения задачи. От объема перебора напрямую зависит время решения задачи на ЭВМ, имеющей заданное быстродействие.

Перейдем к теоретическому анализу объема перебора для худших случаев исходных данных. Для начала рассмотрим два частных случая:

- 1) Радиус зон поиска $r = 1$: случай, когда имеем дело с классическим вариантом метода локального поиска.
- 2) Радиус зон поиска $r = n$: случай, когда локальный поиск фактически обращается полный перебор.

В первом случае задача решается один раз для одной стартовой точки, пусть это будет точка $(0, 0, \dots, 0)$, поскольку очевидно, что она при любых исходных условиях является допустимой, поскольку $b \geq 0$.

В наихудшем случае, когда по условию задачи $b > a_1 + \dots + a_n$, то нетрудно заметить, что глобальный оптимум – это точка $(1, 1, \dots, 1)$, поскольку $c_j \geq 0, j = 1 \dots n$, и метод локального поиска всегда выйдет на оптимум, поскольку все точки пространства будут являться допустимыми, поскольку $b \geq 0$ и $a_j \geq 0, j = 1 \dots n$. Поскольку радиус зон поиска равен 1, то на каждой итерации метод будет находить лучшую точку, отличающуюся только одной координатой от точки центра зоны поиска. Таким образом, метод начнет с точки $(0, 0, \dots, 0)$, пройдет точки $(0, 0, \dots, 1), (0, 0, \dots, 1, 1) \dots (0, 1, \dots, 1)$ и достигнет точки $(1, \dots, 1)$. Достаточно несложно заметить, что на это понадобится n итераций, поскольку за итерацию «улучшается» только одна координата, и еще 1 итерация, когда будет просматриваться последняя зона с центром $(1, \dots, 1)$ и будет установлено, что лучших точек нет, и алгоритм решения задачи завершится. Что касается, размера областей поиска, то на 1-й итерации зона $Z_1(X_0)$ просматривается целиком и число точек в ней: $|Z_1(X_{(0)})| = C_n^0 + C_n^1 = 1 + n$, на последующих n итерациях просматриваются зоны $Z_1(X_T) \setminus Z_1(X_{CT})$, причем расстояние между центрами зон: $D(X_T, X_{CT}) = 1$. Тогда, на примере, когда старая зона с центром $(0, \dots, 0)$ содержит точки: $(0, \dots, 0), (0, \dots, 0, 1) \dots (1, 0, \dots, 0)$, а текущая зона с центром $(0, \dots, 1)$ содержит точки: $(0, \dots, 1), (0, \dots, 0), (0, \dots, 1, 1) \dots (1, 0, \dots, 1)$, нетрудно заметить, что в новой зоне первые 2 точки принадлежат старой зоне. Тогда, учитывая, что $|Z_1(X_T)| = |Z_1(X_{CT})| = n + 1$, то $|Z_1(X_T) \setminus Z_1(X_{CT})| = n + 1 - 2 = n - 1$.

Теперь мы, зная число просматриваемых точек на каждой итерации: на первой итерации $n + 1$ точек, на последующих n итерациях по $n - 1$ точек, можем окончательно оценить объем перебора в худшем случае для $r = 1$, которое составляет $n^2 + 1$.

Во втором случае также рассмотрим наихудший вариант исходных данных, когда оптимум – это точка $(1, \dots, 1)$, а стартовая точка $(0, \dots, 0)$. Так как радиус зон $r = n$, то, очевидно, что первая же зона будет содержать все точки пространства: $|Z_n(X_0)| = C_n^0 + C_n^1 + \dots + C_n^n = 2^n$, и задача решится фактически за одну итерацию, поскольку на второй итерации не найдется ни одной точки, не принадлежащей старой зоне: $|Z_n(X_T) \setminus Z_n(X_{(0)})| = 0$. Тогда объем перебора в худшем случае для $r = n$ составляет 2^n .

Теперь оценим объем перебора при произвольном радиусе зон поиска $1 \leq r \leq n$. Для рассмотрения наихудшего случая ситуации, снова обратимся к варианту исходных данных, когда, когда оптимум – это точка $(1, \dots, 1)$, а стартовая точка $(0, \dots, 0)$. Что касается числа итераций, за которое решится задача, здесь следует отметить, что при рассмотрении частного случая $r = 1$ мы видели, что за одну итерацию «улучшается» одна координата и требуется n итераций, чтобы прийти от стартовой точки $(0, \dots, 0)$ к оптимальной точке $(1, \dots, 1)$. При $r = n$, все n координат «улучшаются» за одну итерацию. Тогда можно сделать обобщение, что при $1 \leq r \leq n$ за одну итерацию может «улучшиться» не более r координат, и в таком случае потребуется не менее $\lceil n / r \rceil$ циклов «улучшения».

Мы рассмотрим именно случай, когда за итерацию «улучшаются» r координат, поскольку локальный поиск согласно первой стратегии просматривает зоны целиком, и находит наилучшую, в нашем случае наилучшая точка будет находиться «на краю зоны», на «самом отдалении», $D = r$, от центра рассматриваемой зоны. Таким образом, 1 итерация будет затрачена на рассмотрение первой зоны поиска, и еще $\lceil n / r \rceil$ итераций на рассмотрение последующих зон, включая последнюю зону, причем центры последующих зон будут находиться на предельном расстоянии $D = r$ от центра предыдущих зон.

Что же касается размеров самих зон, то первая зона просматривается целиком, и число точек в ней, то вычисляется как:

$$|Z_r(X_{(0)})| = \sum_{p=0}^r C_n^p. \quad (3.4)$$

В последующих зонах исключаются из рассмотрения точки предыдущей зоны, причем расстояние между последующей и предыдущей зоной равно $D(X_T, X_{CT}) = r$.

Количество просматриваемых точек в последующих зонах можно вычислить, как размер «разностной зоны» $Z_r(X_T) \setminus Z_r(X_{CT})$, по следующей формуле:

$$|Z_r(X_T) \setminus Z_r(X_{CT})| = \sum_{p=0}^r \left(\sum_{q=0}^{p/2-1} (C_r^q C_{n-r}^{p-q}) \right); \quad (3.5)$$

$$0 \leq r \leq n; D(X_T, X_{CT}) = r.$$

Тогда учитывая все вышесказанное, итоговый объем перебора в худшем случае может быть оценен по следующей формуле:

$$V_{\max} = \sum_{p=0}^r C_n^p + \left\lceil \frac{n}{r} \right\rceil \cdot \sum_{p=0}^r \left(\sum_{q=0}^{p/2-1} C_r^q C_{n-r}^{p-q} \right). \quad (3.6)$$

3.1.4. Анализ нижней границы качества решений

Метод локального поиска относится к классу приближенных методов, который дает некоторое, так называемое, «субоптимальное» решение, значение целевой функции L в котором может быть хуже, чем в глобальном оптимуме. Глобальный оптимум может быть найден только полным перебором.

Мы рассматриваем задачу псевдобулевой оптимизации, конкретно – максимизации, модифицированным методом локального поиска. Соответственно, качество субоптимального решения, обозначим ее ξ , мы можем определить как отношение значения целевой функции \tilde{L} в субоптимальном решении к значению функции L^* в глобальном оптимуме:

$$\xi = \tilde{L} / L^*. \quad (3.7)$$

Перейдем теперь к теоретическому анализу качества субоптимальных решений для специально подобранных худших случаев исходных данных.

Пусть стартовая точка для поиска: $X_0 = (0, \dots, 0)$.

1) Пусть радиус зон поиска равен $r = 1$ и $n \geq 3$. Рассмотрим следующие специально подобранные исходные данные для задачи:

- $b = n$.
- $a_1 = n, a_2 = 1 \dots a_n = 1$, т. е. $a_2 + \dots + a_n = n - 1$.
- $c_1 = 1001, c_2 = 1000 \dots c_n = 1000$, т. е. $c_2 + \dots + c_n = 1000(n - 1)$.

Тогда, при решении задачи методом локального поиска в первой зоне с центром в точке $X_0 = (0, \dots, 0)$ будут рассмотрены следующие точки: $(0, \dots, 0)$, $(1, 0, \dots, 0)$, ..., $(0, \dots, 0, 1)$, так как радиус поиска $r = 1$. В первой зоне будет выбрана наилучшая точка $X_1 = (1, 0, \dots, 0)$, так как выполняется ограничение $a_1 = b$, и $c_1 = 1001 > c_j = 1000$ для $\forall j = 2 \dots n$, и в ней целевая функция L имеет наилучшее значение 1001.

Далее во второй зоне с центром в точке $X_1 = (1, 0, \dots, 0)$ будут рассмотрены точки: $(1, 1, 0, \dots, 0)$, $(1, 0, 1, 0, \dots, 0)$, ..., $(1, 0, \dots, 0, 1)$, но ни одна из них не пройдет ограничение $a_1 + \dots + a_n \leq b$, так как $a_1 = b$.

В результате работа алгоритма завершится и в качестве решения будет принята точка $\tilde{X} = (1, 0, \dots, 0)$, в которой целевая функция $\tilde{L} = 1001$.

Нетрудно заметить, что существует также глобальный оптимум $X^* = (0, 1, \dots, 1)$, удовлетворяющая ограничению, так как $a_2 + \dots + a_n < b$, и в ней целевая функция $L^* = 1000(n - 1)$ и, тогда качество субоптимального решения составляет:

$$\xi = \tilde{L} / L^* = 1001 / (1000(n - 1)).$$

Если же радиус поиска в данной задаче задать $r = 2$, то в первой зоне с центром в точке $X_0 = (0, \dots, 0)$ будут рассмотрены точки $(0, \dots, 0)$, $(1, 0, \dots, 0)$, ..., $(0, \dots, 0, 1)$, $(1, 1, 0, \dots, 0)$, $(0, 1, 1, 0, \dots, 0)$..., $(0, \dots, 0, 1, 1)$, и точка $X_1 = (0, 1, 1, 0, \dots, 0)$ будет выбрана наилучшей. Далее, через несколько шагов мы гарантированно выйдем на глобальный оптимум $X^* = (0, 1, \dots, 1)$.

2) Пусть радиус зон поиска равен $r = 2$ и $n \geq 4$. Рассмотрим следующие специально подобранные исходные данные для задачи:

- $b = n$.
- $a_1 = n, a_2 = 1 \dots a_n = 1$ (то есть, $a_2 + \dots + a_n = n - 1$).
- $c_1 = 2001, c_2 = 1000 \dots c_n = 1000$, т. е. $c_2 + \dots + c_n = 1000(n - 1)$.

Тогда, при решении задачи методом локального поиска в первой зоне с центром в точке $X_0 = (0, \dots, 0)$ будут рассмотрены следующие точки: $(0, 0, \dots, 0), (1, 0, \dots, 0), \dots, (0, \dots, 0, 1), (1, 1, 0, \dots, 0), (0, 1, 1, 0, \dots, 0) \dots, (0, \dots, 0, 1, 1)$, так как радиус зон поиска $r = 2$.

В первой зоне будет выбрана наилучшая точка $X_1 = (1, 0, \dots, 0)$, так как выполняется ограничение $a_1 = b$, и $c_1 = 2001 > c_i + c_j = 2000$ для $\forall i, j = 2 \dots n, i \neq j$, и в ней целевая функция L имеет наилучшее значение 2001.

Далее во второй зоне с центром в точке $X_1 = (1, 0, \dots, 0)$ будут рассмотрены точки: $(1, 1, 0, \dots, 0), (1, 0, 1, 0, \dots, 0), \dots, (1, 0, \dots, 0, 1), (1, 1, 1, 0, \dots, 0), (1, 0, 1, 1, 0, \dots, 0) \dots, (1, 0, \dots, 0, 1, 1)$, но ни одна из них не пройдет ограничение $a_1 + \dots + a_n \leq b$, так как $a_1 = b$.

В результате работа алгоритма завершится и в качестве решения будет принята точка $\tilde{X} = (1, 0, \dots, 0)$, в которой целевая функция $\tilde{L} = 2001$.

Нетрудно заметить, что существует также глобальный оптимум $X^* = (0, 1, \dots, 1)$, удовлетворяющая ограничению, так как $a_2 + \dots + a_n < b$, и в ней целевая функция $L^* = 1000(n - 1)$, и, тогда, качество субоптимального решения составляет:

$$\xi = \tilde{L} / L^* = 2001 / (1000(n - 1)).$$

Если же радиус поиска в данной задаче задать $r = 3$, то в первой зоне с центром в точке $X_0 = (0, 0, \dots, 0)$ будут рассмотрены точки $(0, 0, \dots, 0), (1, 0, \dots, 0), \dots, (0, 0, \dots, 0, 1), (1, 1, 0, \dots, 0), (0, 1, 1, 0, \dots, 0) \dots, (0, \dots, 0, 1, 1), (1, 1, 1, 0, \dots, 0), (0, 1, 1, 1, 0, \dots, 0), \dots, (0, \dots, 0, 1, 1, 1)$, и точка $X_1 = (0, 1, 1, 1, 0, \dots, 0)$ будет выбрана наилучшей. Далее, через несколько шагов мы гарантированно выйдем на глобальный оптимум $X^* = (0, 1, \dots, 1)$.

3) Обобщая рассмотренные примеры, можно показать, что для $\forall n \geq 3$ и $1 \leq r \leq n - 2$, можно подбирать специальные исходные данные:

- $b = n$.
- $a_1 = n, a_2 = 1 \dots a_n = 1$, т. е. $a_2 + \dots + a_n = n - 1$.
- $c_1 = 1000r + 1, c_2 = 1000 \dots c_n = 1000$, т. е. $c_2 + \dots + c_n = 1000(n - 1)$.

При решении задачи методом локального поиска в первой зоне с центром в точке $X_0 = (0, 0, \dots, 0)$ будут рассмотрены множество точек $(0, 0, \dots, 0), (1, 0, \dots, 0), \dots, (0, \dots, 0, 1), \dots, (1, \dots, r, 0, \dots, 0), \dots, (0, \dots, 0, 1, \dots, r)$. В первой зоне будет выбрана наилучшая точка $X_1 = (1, 0, \dots, 0)$, так как выполняется ограничение $a_1 = b$, и $c_1 = 1000r + 1 > 1000r$, и в ней целевая функция L имеет наилучшее значение $1000r + 1$.

Далее во второй зоне с центром в точке $X_1 = (1, 0, \dots, 0)$ будут рассмотрены точки $(1, 1, 0, \dots, 0), (1, 0, 1, 0, \dots, 0), \dots, (1, 0, \dots, 0, 1), \dots, (1, 1, \dots, r, 0, \dots, 0), \dots, (1, 0, \dots, 0, 1, \dots, r)$, но ни одна из них не пройдет ограничение $a_1 + \dots + a_n \leq b$, так как $a_1 = b$.

В результате работа алгоритма завершится и в качестве решения будет точка $\tilde{X} = (1, 0, \dots, 0)$, в которой целевая функция $\tilde{L} = 1000r + 1$.

Нетрудно заметить, что существует также глобальный оптимум $X^* = (0, 1, \dots, 1)$, удовлетворяющая ограничению, так как $a_2 + \dots + a_n < b$, и в ней целевая функция $L^* = 1000(n - 1)$, и, тогда, качество субоптимального решения составляет:

$$\xi = \tilde{L} / L^* = (1000r + 1) / (1000(n - 1)).$$

Особо отметим, что при $r = n - 1$, коэффициент $c_1 = 1000(n - 1) + 1$ и он перевешивает сумму всех остальных коэффициентов $c_2 + \dots + c_n = 1000(n - 1)$, и в этом случае глобальный оптимум $X^* = (1, 0, \dots, 0)$ совпадает с решением, которое находит метод локального поиска: $\tilde{X} = (1, 0, \dots, 0)$, и, очевидно, качество решения в таком случае $\xi = \tilde{L} / L^* = 1$.

Наконец, при $r = n$, метод локального поиска фактически выполняет полный перебор, поскольку зона $Z_n(X_0)$ радиусом n – это есть множество всех точек n -мерного булевого пространства и глобальный оптимум будет гарантированно найден, а значит, качество решения также $\xi = \tilde{L} / L^* = 1$.

Теперь, если в рассмотренных выше примерах задач вместо значения 1000 использовать достаточно большое число $Q > 0$, такое что $Qr \gg 1$, то, очевидно, что качество решений $(Qr + 1) / (Q(n - 1)) \sim r / (n - 1)$.

Тогда, с учетом всего сказанного, имеем следующую нижнюю границу для качества субоптимальных решений, получаемых описанным выше методом локального поиска для задачи о рюкзаке, при заданной размерности задачи $n \geq 3$ и радиусе поиска $1 \leq r \leq n$:

$$\xi_{\min} = \begin{cases} \frac{r}{n-1}, & 1 \leq r \leq n-2; \\ 1, & n-1 \leq r \leq n. \end{cases} \quad (3.8)$$

3.1.5. Экспериментальное исследование объема перебора и качества решений

Используя программную реализацию описанного выше метода локального поиска, можно провести экспериментальное исследование объема перебора и качества решений, которое подтверждает достоверность полученных ранее аналитических оценок для верхней границы объема перебора и нижней границы качества решений. Ниже приведены примеры экспериментального исследования и полученные результаты.

1) Для исследования объема перебора проводились отдельные серии экспериментов для различных частных случаев задач размерности n , радиуса поиска r и по 10000 экспериментов в серии для различных случайных значений для b , a_j , и c_j . В каждом эксперименте задача решалась методом локального поиска, и подсчитывался объем перебора точек в процессе поиска решения. Затем среди значений объемов переборов во всех экспериментах в серии выбиралось наибольшее значение.

При решении задач использовались следующие значения для b , a_j и c_j :

- b = целое случайное числа в диапазоне от 1 до 1000.
- a_j = целые случайные числа в диапазоне от 1 до b .
- c_j = целые случайные числа в диапазоне от 1 до 1000.

Ниже в таблице 3.1 приведены результаты экспериментальной оценки наибольших значений объемов перебора при заданных n и r , среди всех значений объемов перебора при различных случайных значениях b , a_j и c_j .

Таблица 3.1. Наибольшие значения объемов перебора среди всех значений при различных случайных значениях b , a_j и c_j .

n	2	3	4	5	6	7	8	9
$r = 1$	5	10	17	26	37	50	65	82
$r = 2$	–	9	17	34	52	89	121	186
$r = 3$	–	–	17	36	74	148	238	388
$r = 4$	–	–	–	33	69	145	273	546

Результаты экспериментов показали, что наибольшие значения объемов перебора на различных случайных исходных данных близки к аналитическим оценкам верхней границы для объема перебора и, что самое важное, не выше аналитической оценки.

2) Для исследования качества решений проводились отдельные серии экспериментов для различных частных случаев задач размерности n , радиуса поиска r и по 10000 экспериментов в серии для различных случайных значений для b , a_j , и c_j . В каждом эксперименте задача решалась методом локального поиска и методом полного перебора, и вычислялось качество получаемого решения. Затем среди значений качества решений во всех экспериментах в серии выбиралось наименьшее значение.

При решении задач использовались следующие значения для b , a_j и c_j :

- b = целое случайное числа в диапазоне от 1 до 1000.
- a_j = целые случайные числа в диапазоне от 1 до b .
- c_j = целые случайные числа в диапазоне от 1 до 1000.

Ниже в таблице 3.2 приведены результаты экспериментальной оценки наименьших значений качества решений при заданных n и r , среди всех значений качества при различных случайных значениях b , a_j и c_j .

Таблица 3.2. Наименьшие значения качества решений среди всех значений при различных случайных значениях b , a_j и c_j .

n	2	3	4	5	6	7	8	9
$r = 1$	1	0,508	0,344	0,262	0,212	0,171	0,149	0,128
$r = 2$	–	1	0,675	0,516	0,407	0,345	0,292	0,261
$r = 3$	–	–	1	0,761	0,615	0,521	0,434	0,381
$r = 4$	–	–	–	1	0,821	0,673	0,579	0,512

Результаты экспериментов показали, что наименьшие значения качества решений на различных случайных исходных данных близки к аналитическим оценкам нижней границы качества решений и, что самое важное, не ниже аналитической оценки.

3.2. Специализированная задача распределения вычислительных ресурсов

В современном мире существует большое количество организаций, имеющих корпоративную сеть, состоящую из множества конечных рабочих мест пользователей и некоторого, так называемого, серверного парка. Многолетняя практика эксплуатации серверных систем показывает, что на сегодняшний день большинство компьютеров серверного парка достаточно слабо загружены по вычислительным ресурсам.

На сегодняшний день технология виртуальных машин является одним из наиболее эффективных подходов к повышению загрузки вычислительных ресурсов корпоративной сети. Технология виртуальных машин позволяет на физическом компьютере под управлением некоторой, так называемой базовой операционной системы, обеспечивать работу виртуальных машин, на которых также как и на обычных реальных компьютерах могут функционировать, так называемые, логические серверы. Логический сервер – это серверная операционная система со всеми работающими под ее управлением сетевыми службами. Технология виртуальных машин обеспечивает функционирование нескольких изолированных логических серверов на одном компьютере, обходя проблемы совместимости программного обеспечения, работающего на различных логических серверах.

Соответственно, при использовании технологии виртуальных машин серверный парк со слабой загрузкой ресурсов можно реорганизовать – перенести логические серверы на виртуальные платформы и распределить их на компьютеры в соответствии с некоторым вариантом распределения, что в конечном счете должно привести к уменьшению объема используемого оборудования и затрат на его поддержку.

Ниже показан пример сети предприятия с исходным серверным парком (рис. 3.4), и с реорганизованным серверным парком (рис. 3.5) после внедрения технологии виртуальных машин, перевода логических серверов S1-S15, функционирующих на компьютерах Н1-Н15, на виртуальную платформу и размещения на компьютеры Н4, Н5, Н6, Н11, Н12, Н13.

Однако, при применении технологии виртуальных машин неясно, каким образом перераспределять и объединять логические серверы среди компьютеров. При этом необходимо оценивать требования логических серверов, технические характеристики компьютеров, после чего каким-то образом размещать логические серверы на компьютеры, чтобы добиться хорошей загрузки ресурсов используемого оборудования. Соответственно здесь требуется специализированная математическая модель задачи поиска оптимального распределения логических серверов на физические компьютеры, а также метод ее решения.

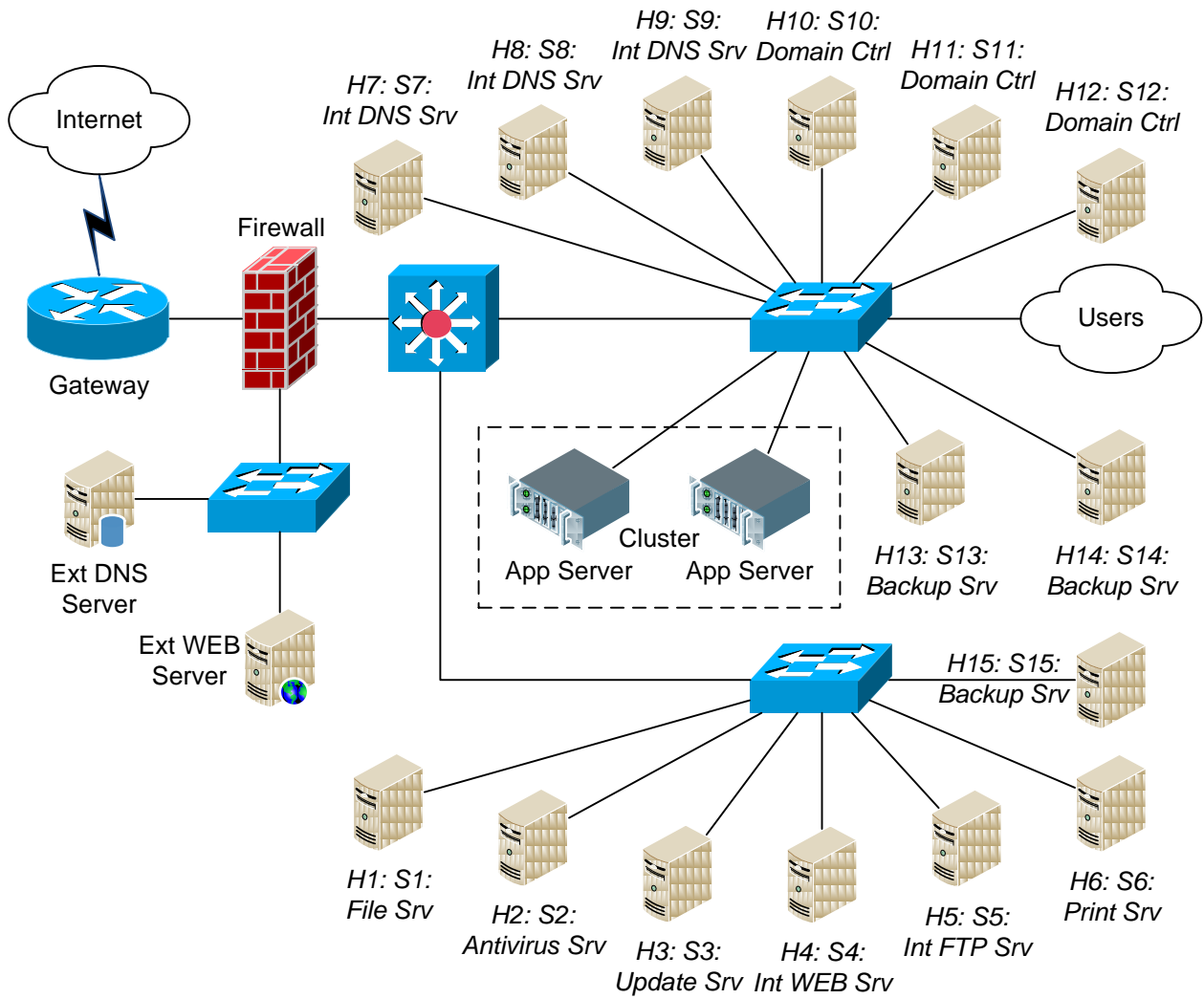


Рис. 3.4. Сеть предприятия с исходным серверным парком.

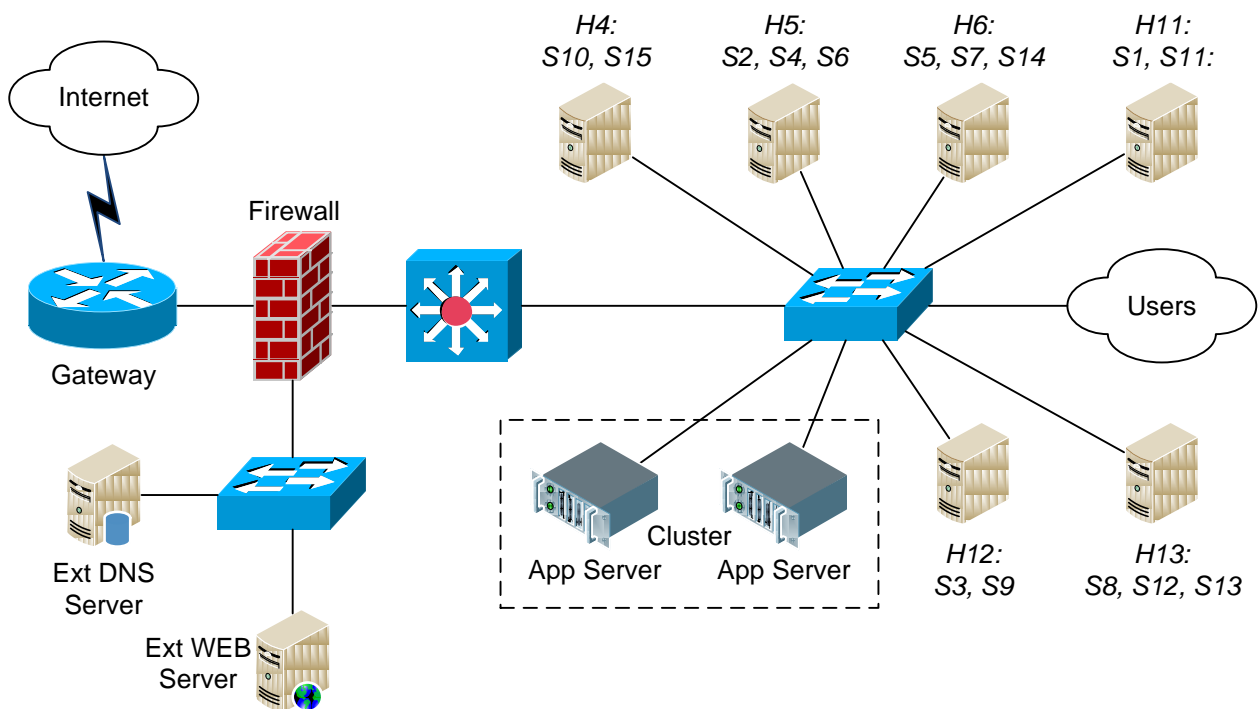


Рис. 3.5. Сеть предприятия с реорганизованным серверным парком.

3.2.1. Математическая модель задачи распределения вычислительных ресурсов

Задача поиска оптимального распределения логических серверов по компьютерам является задачей дискретной оптимизации.

Входными данными задачи являются:

- Множество типов ресурсов $\{R_i\}$, $i = 1..NC$.
- Множество компьютеров $\{H_k\}$, $k = 1..NH$, и матрица базовых уровней их ресурсов $\{R_{i,k}\}$, $k = 1..NH$, $i = 1..NC$.
- Множество логических серверов $\{S_j\}$, $j = 1..NS$, и матрица их требований $\{Q_{i,j}\}$, $j = 1..NS$, $i = 1..NC$.
- Базовая операционная система компьютера и ее требования $\{V_i\}$, $i = 1..NC$.
- Дополнительные ограничения $\{E_{d,j}\}$, $d = 1..NX$, $i = 1..NC$, на одновременное размещение логических серверов на один и тот же компьютер.
- Двоичный вектор маски $\{O_i\}$, $i = 1..NC$, определяющий то, по каким типам ресурсов требуется повышение эффективности их использования.

Выходными данными задачи (решением задачи) является матрица распределения $\{X_{k,j}\}$, $k = 1..NH$, $j = 1..NS$, логических серверов по физическим компьютерам.

Целью является повышение эффективности использования вычислительных ресурсов.

Поскольку рассматриваемая задача оптимизации является достаточно сложной и не может быть приведена к какой-либо известной задаче оптимизации, то мы рассмотрим сначала родственную ей простейшую задачу дискретной оптимизации – задачу о рюкзаке. Далее попытаемся, постепенно усложняя математическую модель задачи о рюкзаке, подойти вплотную к нашей задаче оптимизации, построить для нее математическую модель, а также предложить метод ее решения.

Первое приближение. В задаче о рюкзаке задано множество вещей $\{P_1, \dots, P_N\}$ различного веса $\{a_j\}$ и ценности $\{c_j\}$, рюкзак с определенной вместимостью b по весу и требуется найти такой набор вещей, при котором рюкзак их мог бы вместить, а суммарная ценность помещенных вещей была максимальной. Математическая модель задачи о рюкзаке выглядит следующим образом:

$$\begin{cases} \sum_{j=1}^N a_j x_j \leq b; \\ L = \sum_{j=1}^N c_j x_j \rightarrow \max; \\ \forall j \in [1, N]: x_j \in \{0, 1\}. \end{cases} \quad (3.9)$$

Второе приближение. Во втором приближении – отсутствуют ценности вещей, но их роль выполняют уровни загрузки ресурсов, для задачи о рюкзаке это отношение веса вещи к вместимости рюкзака по весу. Соответственно, во втором приближении математическая модель немного преобразуется:

$$\begin{cases} \sum_{j=1}^N a_j x_j \leq b; \\ L = \sum_{j=1}^N \frac{a_j}{b} x_j \rightarrow \max; \\ \forall j \in [1, N]: x_j \in \{0, 1\}. \end{cases} \quad (3.10)$$

Третье приближение. В третьем приближении, мы имеем дело с многомерным ресурсом. Соответственно, рюкзак и вещи становятся многомерным. Можно представить это следующим образом – в рюкзаке есть множество отсеков различной вместимости по весу $\{b_i\}$, $i = 1..M$, а вместо вещей теперь однородные комплекты более мелких вещей весом $\{a_{ij}\}$, причем i -я вещь любого j -го комплекта может быть помещена только в i -й отсек рюкзака и не иначе. Кроме того, комплект может размещаться только целиком со всеми входящими в него вещами, разделение комплекта не допускается.

Очевидно, в таком случае помимо того, что число ограничений увеличивается до M , число критериев оптимизации также может быть равно M , но в то же время, если число ограничений строго фиксировано, то оптимизацию проводить вовсе необязательно по всем отсекам рюкзака, можно, в частности, оптимизировать только один конкретный i^* -й отсек, где $1 \leq i^* \leq M$. Тогда математическую модель можно представить следующим образом:

$$\left\{ \begin{array}{l} \sum_{j=1}^N a_{ij} x_j \leq b_i, i = 1..M; \\ L = \sum_{j=1}^N \frac{a_{i^*j}}{b_{i^*}} x_j \rightarrow \max, 1 \leq i^* \leq M; \\ \forall j \in [1, N]: x_j \in \{0,1\}. \end{array} \right. \quad (3.11)$$

Четвертое приближение. В четвертом приближении, критериев может быть несколько – предпочтительно решение проблемы неэффективного использования по многим (или даже по всем M) типам ресурсов (отсеков рюкзака). То, по каким именно отсекам проводить оптимизацию можно задать при помощи вектора маски $\{O_i\}$, где O_i – булева переменная, равная «1» – если по i -му типу ресурса требуется повышение эффективности его использования, «0» – в противном случае.

Однако, как известно, большинство точных и приближенных методов решения задач оптимизации рассчитаны только на одну целевую функцию, и поэтому нам необходимо пойти на некоторый компромисс и выработать некоторую общую целевую функцию для множества заданных критериев. Общую целевую функцию можно представить как сумму целевых функций для каждого отдельного оптимизируемого по эффективности использования типа ресурса. Кроме того, для удобства можно нормировать эту сумму, поделив ее на число целей по отдельным типам ресурсам, для которых проводится оптимизация. Тогда математическая модель принимает следующий вид:

$$\left\{ \begin{array}{l} \sum_{j=1}^N a_{ij} x_j \leq b_i, i = 1..M; \\ L = \frac{1}{\sum_{i=1}^M O_i} \left(\sum_{i=1}^M \frac{O_i}{b_i} \left(\sum_{j=1}^N a_{ij} x_j \right) \right) \rightarrow \max; \\ \forall j \in [1, N]: x_j \in \{0,1\}. \end{array} \right. \quad (3.12)$$

Пятое приближение. В пятом приближении, при решении задачи оптимизации необходимо еще учитывать так называемые требования базовой ОС – для задачи о рюкзаке это некоторый специальный комплект, который не участвует в размещении, но обязательно находится в рюкзаке, занимая некоторое место $\{V_i\}$ в каждом его отсеке. Тогда математическая модель принимает следующий вид:

$$\left\{ \begin{array}{l} \sum_{j=1}^N a_{ij} x_j \leq b_i - V_i, i = 1..M; \\ L = \frac{1}{\sum_{i=1}^M O_i} \left(\sum_{i=1}^M \frac{O_i}{b_i - V_i} \left(\sum_{j=1}^N a_{ij} x_j \right) \right) \rightarrow \max; \\ \forall j \in [1, N]: x_j \in \{0,1\}. \end{array} \right. \quad (3.13)$$

Следует заметить то, что если для какого-либо типа ресурса $b_i \leq V_i$, то сразу можно сказать то, что ни один комплект не разместится в рюкзаке, и задачу нет смысла решать. Второй важный момент – это то, что целевая функция ограничена снизу нулем: случай, когда ни один комплект мы не размещаем в рюкзаке, $\forall j: x_j = 0$, и ограничена сверху единицей: в силу заданных ограничений сумма требований по каждому типу ресурсов не может превышать базовый уровень имеющихся ресурсов.

Шестое приближение. В шестом приближении, с целью учета вопросов надежности для логических серверов могли быть заданы специальные ограничения для исключения размещения определенных логических серверов (виртуальных машин содержащих в себе серверы), дублирующих функции друг друга, на один и тот же компьютер. Для задачи о рюкзаке – это специальные ограничения, оговаривающие то, что определенные сочетания комплектов вещей недопустимы для размещения в рюкзаке. Такие ограничения задать несложно: например, если мы не хотим, чтобы в рюкзак одновременно попали 1-й и 5-й комплект, то задаем ограничение следующего вида: $x_1 + x_5 \leq 1$. Однако, для задания дополнительных ограничений в матричной форме, где строки – ограничения, столбцы – номера (индексы) комплектов, удобнее использовать булеву матрицу $\{E_{dj}\}$, где E_{dj} – элемент матрицы, равный «1», если j -й комплект присутствует в d -м ограничении, «0» – в противном случае, $d = 1..D$. Тогда математическая модель принимает следующий вид:

$$\left\{ \begin{array}{l} \sum_{j=1}^N a_{ij} x_j \leq b_i - V_i, i = 1..M; \\ \sum_{j=1}^N E_{dj} x_j \leq 1, d = 1..D; \\ L = \frac{1}{\sum_{i=1}^M O_i} \left(\sum_{i=1}^M \frac{O_i}{b_i - V_i} \left(\sum_{j=1}^N a_{ij} x_j \right) \right) \rightarrow \max; \\ \forall j \in [1, N]: x_j \in \{0, 1\}. \end{array} \right. \quad (3.14)$$

Седьмое приближение. Наконец, в седьмом приближении мы имеем дело не с одним, а с множеством рюкзаков. Задача оптимизации в данном приближении резко усложняется, поскольку практически она из простой задачи о рюкзаке превращается в задачу поиска оптимального распределения множества комплектов вещей на множество рюкзаков с множеством отсеков. Основная проблема здесь в выборе правильного подхода к достижению цели. Целью реорганизации является то, что за счет использования технологии виртуальных машин, мы пытаемся освободить некоторое множество компьютеров, причем идем к этой цели, добиваясь максимально эффективного использования ресурсов компьютеров. Однако очень важно подчеркнуть то, что можно сделать ошибку и пытаться добиваться эффективного использования ресурсов физических компьютеров всех сразу и тем самым решить задачу балансировки нагрузки. В таком случае нагрузка будет распределена так, чтобы ни один физический компьютер не простаивал, и не было так, чтобы один компьютер сильно загружен, а другой очень слабо. Этим мы лишь добьемся приблизительно равномерной загрузки ресурсов среди множества физических компьютеров, и ни один физический компьютер не будет освобожден.

Истинная же цель – решить задачу, обратную задаче балансировки нагрузки, то есть сосредоточить логические серверы на некотором множестве компьютеров таким образом, чтобы задействованные компьютеры были загружены максимально, а все остальные незадействованные компьютеры были свободны.

Тогда, используя теперь обозначения нашей основной задачи поиска оптимального распределения логических серверов по физическим компьютерам, рассмотрим следующую математическую модель (в 3-х частях):

1) Для любого k -го физического компьютера ($k = 1..NH$), для которого $\forall i: R_{i,k} - V_i > 0$, справедлива следующая система ограничений:

$$\left\{ \begin{array}{l} \sum_{j=1}^{NS} Q_{ij} x_{kj} \leq R_{ik} - V_i, i = 1..NC; \\ \sum_{j=1}^{NS} E_{dj} x_{kj} \leq 1, d = 1..NX; \\ \forall j \in [1, NS]: x_{kj} \in \{0, 1\}. \end{array} \right. \quad (3.15a)$$

Где, NC – число типов ресурсов.

NH – число физических компьютеров.

NS – число логических серверов (общее число с учетом дополнительных логических серверов, которые были подготовлены для дублирования функций критичных по важности логических серверов)

NX – число дополнительных ограничений (подмножество дублирующих друг друга логических серверов).

$R_{i,k}$ – базовый уровень i -го типа ресурса k -го физического компьютера.

$Q_{i,j}$ – требование j -го логического сервера к i -му ресурсу.

$E_{d,j}$ – элемент булевой матрицы, равный «1», если j -й логический сервер присутствует в d -м ограничении, «0» – в противном случае.

$X_{k,j}$ – элемент булевой матрицы, равный «1», если j -й логический сервер (виртуальная машина с логическим сервером) размещается на k -м физическом компьютере, «0» – в противном случае.

Первая система ограничений – это ограничения по ресурсам, вторая – ограничения, исключающие одновременное размещение определенных логических серверов на один и тот же физический компьютер.

2) Для любого j -го, где $j = 1..NS$, логического сервера справедливо то, что он может быть размещен на одном и только одном физическом компьютере либо не размещен вообще:

$$\sum_{k=1}^{NH} x_{kj} \leq 1. \quad (3.156)$$

3) Целевая функция:

$$\left\{ \begin{aligned} L_{\gamma} &= \frac{1}{\sum_{i=1}^{NC} O_i} \left(\sum_{i=1}^{NC} \frac{O_i}{R_{i\gamma} - V_i} \left(\sum_{j=1}^{NS} Q_{ij} x_{j\gamma} \right) \right) \rightarrow \max; \\ \forall \gamma &\in \{k^*\}. \end{aligned} \right. \quad (3.15b)$$

Где, V_i – требование базовой ОС к i -му ресурсу.

O_i – элемент вектора маски, принимающий значение «1», если соответствующему i -му типу ресурса требуется повышение эффективности его использования, «0» – в противном случае.

$\{k^*\}$ – множество индексов физических компьютеров, которые окажутся задействованными после решения задачи оптимального распределения.

В данном приближении в математической модели фигурирует множество целевых функций, причем то, сколько именно их и каким именно физическим компьютерам они соответствуют неизвестно, это можно выяснить, лишь решив всю задачу целиком, чтобы получить множество $\{k^*\}$.

Выделить одну глобальную целевую функцию в задаче невозможно: можно было бы, например, поставить целью минимизацию числа задействованных компьютеров, однако, в общем случае компьютеры все различаются по уровням своих ресурсов. Более того, для различных задач один и тот же компьютер может иметь различную ценность. Наконец, неоправданно большое количество задействованных компьютеров – это всегда следствие неэффективного использования ресурсов компьютеров. Также можно было бы пытаться суммировать по каждому компьютеру общий объем простаивающих ресурсов и пытаться минимизировать эту сумму. Однако, здесь нам снова неизвестно сколько и какие именно компьютеры будут задействованы, а сумма простаивающих ресурсов по всем изначально заданным физическим компьютерам не зависит от распределения (при условии, конечно, что все логические серверы размещены), поскольку сумма требований по всем логическим серверам остается постоянной.

В такой ситуации, модель задачи распределения представляется неразрешимой из-за неопределенности в целевых функциях, и, соответственно здесь требуется декомпозиция исходной задачи и сведение ее решения к решению множества более простых подзадач.

3.2.2. Декомпозиция задачи распределения вычислительных ресурсов

В математической модели (3.15) основная проблема – необходимо повысить уровень загрузки ресурсов задействованных физических компьютеров, но заранее неизвестно какие именно компьютеры окажутся задействованными. Количество и индексы (номера) задействованных физических компьютеров можно получить, лишь решив целиком задачу.

Получается неразрешимая ситуация, однако, выход из нее есть – для этого необходимо уйти от попытки охватывать в каждый момент времени решения все физические компьютеры и все логические серверы одновременно и попытаться разбить задачу на множество более простых оптимизационных подзадач.

Предлагается следующий подход: изначально задано исходное множество физических компьютеров и множество логических серверов, но в общем случае в процессе решения эти множества будут изменяться. При каждом очередном «большом шаге» мы будем пытаться распределить некоторое «наилучшее сочетание логических серверов» на «наилучшее подходящее» физическом компьютере – компьютер, у которого достигается наибольшая средняя загрузка по всем ресурсам среди остальных компьютеров при оптимальном распределении на него логических серверов. В случае успеха после этого мы будем исключать из множества физических компьютеров «наилучшее подошедший компьютер», а из множества логических серверов – «распределенное наилучшее сочетание логических серверов». Так будет продолжаться до тех пор, пока либо множество физических компьютеров или множество логических серверов не станет пустым, либо не наступит ситуация, когда не удастся распределить ни один из оставшихся логических серверов ни на один из оставшихся физических компьютеров.

Конечно, при использовании вышеуказанного подхода мы далеко не всегда будем находить глобальный оптимум. Тем не менее, мы разбиваем сложную задачу (с множеством физических компьютеров) на множество более простых задач (с одним физическим компьютером) и в то же время на каждом шаге «не теряем из виду» ни один из всех оставшихся физических компьютеров.

Рассмотрим теперь, как образуется математическая модель подзадач и как решается исходная задача в целом по «большим шагам».

Пусть, T – это порядковый номер «большого шага», $T \geq 0$.

Пусть, $\{K(T)\}$ – множество индексов физических компьютеров, оставшихся к моменту шага T , причем $\{K(0)\} = \{1, \dots, NH\}$.

Пусть, $\{J(T)\}$ – множество индексов логических серверов, оставшихся к моменту шага T , причем $\{J(0)\} = \{1, \dots, NS\}$.

Тогда, математическая модель подзадачи на шаге T при рассмотрении k -го физического компьютера, $k \in \{K(T)\}$, для которого $\forall i: R_{i,k} - V_i > 0$, представляется следующим образом:

$$\left\{ \begin{array}{l} \sum_{j \in \{J(T)\}} Q_{ij} x_{kj} \leq R_{ik} - V_i, i = 1..NC; \\ \sum_{j \in \{J(T)\}} E_{dj} x_{kj} \leq 1, d = 1..NX; \\ L(T, k) = \frac{1}{\sum_{i=1}^{NC} O_i} \left(\sum_{i=1}^{NC} \frac{O_i}{R_{ik} - V_i} \left(\sum_{j \in \{J(T)\}} Q_{ij} x_{kj} \right) \right) \rightarrow \max; \\ \forall j \in \{J(T)\}: x_{kj} \in \{0, 1\}. \end{array} \right. \quad (3.16)$$

Где, $L(T, k)$ – целевая функция на шаге T при решении подзадачи для k -го физического компьютера.

На каждом шаге T последовательно рассматриваются все физические компьютеры с индексами $k \in \{K(T)\}$, оставшиеся к моменту шага T , и выбирается тот, для которого в результате решения соответствующей подзадачи, достигается наивысшее значение целевой функции среди значений, полученных при решении подзадач для компьютеров с индексами $k \in \{K(T)\}$. Соответственно, наилучший физический компьютер с индексом k^* выбирается из следующего условия:

$$\begin{cases} L_{k^*}(T) = \max_{k \in \{K(T)\}} \{L_{\max}(T, k)\}; \\ L_{k^*}(T) > 0. \end{cases} \quad (3.17)$$

Где, $L_{\max}(T, k)$ – оптимальное значение целевой функции на шаге T при решении подзадачи для k -го физического компьютера.

$L_{k^*}(T)$ – наивысшее оптимальное значение целевой функции на шаге T среди значений $L_{\max}(T, k)$ для всех $k \in \{K(T)\}$.

В условии (3.17) особенно важно условие того, что $L_{k^*}(T)$ не должна быть нулевой, это гарантирует, что хотя бы один логический сервер распределится и хотя бы один физический компьютер будет задействован. Если же $L_{k^*}(T) = 0$, то это означает, что дальнейшее распределение логических серверов невозможно и решение исходной задачи должно быть прекращено.

Если же k^* успешно найден из условия (3.17), то выполняются следующие преобразования: из множества оставшихся физических компьютеров исключается компьютер с индексом k^* , а из множества логических серверов – множество серверов, распределенных на этот компьютер:

$$\begin{aligned} \{K(T+1)\} &= \{K(T)\} \setminus k^*; \\ \{J(T+1)\} &= \{J(T)\} \setminus \{j^*\}. \end{aligned} \quad (3.18)$$

Где, $\{j^*\}$ – множество индексов логических серверов, которые были распределены на k^* -й физический компьютер.

Если в результате преобразования (3.18), множество $K(T+1)$ или $J(T+1)$ окажется пустым, то решения задачи завершается, в противном случае переход к шагу $T + 1$. Результатом решения задачи является распределение $\{X_{k,j}\}$, $k = 1 \dots NH$, $j = 1 \dots NS$, логических серверов по компьютерам.

На рис. 3.6 представлен алгоритм решения задачи в целом (по «большим шагам»).

Таким образом, используя компромиссный подход, дающий схему решения всей задачи в целом (без детализации подзадач), нам удалось свести неразрешимую стационарную математическую модель (3.15) с неопределенными целевыми функциями к некоторому множеству (число подзадач заранее неизвестно) вполне разрешимых математических моделей (3.16). Число «больших шагов», в худшем случае (когда множество физических компьютеров опустошается раньше или вместе с множеством логических серверов) составит: NH , а общее число решаемых подзадач оптимизации по отдельным компьютерам: $(NH \cdot (NH + 1)) / 2$. Так что, на глобальном уровне оптимизации сложность компромиссного метода полиномиальная ($\sim n^2$).

В алгоритме решения задачи каждая подзадача, в свою очередь, представляет собой класс задач условной псевдобулевой оптимизации, которая сводится к задаче о рюкзаке.

Соответственно, подзадачу на шаге T для рассматриваемого компьютера $k \in K(T)$ можно представить в виде математической модели задачи о рюкзаке:

$$\begin{cases} \sum_{j=1}^N a_{ij} x_j \leq b_i, i = 1..M; \\ L = \sum_{j=1}^N c_j x_j \rightarrow \max; \\ \forall j \in [1, N]: x_j \in \{0, 1\}. \end{cases} \quad (3.19)$$

Соответственно, исходные данные для подзадачи на шаге T для рассматриваемого компьютера $k \in K(T)$ формируются из исходных данных задачи распределения логических серверов следующим образом:

- $M = NC + NX$.
- $N = |\{J(T)\}|$.
- Для каждого $j = 1 \dots N$ и $q = j$ -й элемент из $\{J(T)\}$:
 - Для всех $i = 1 \dots NC$: $a_{i,j} = Q_{i,q}$; $b_i = R_{i,k} - V_i$.
 - Для всех $i = NC + 1 \dots NC + NX$: $a_{i,j} = E_{i-NC,q}$; $b_i = 1$.
 - $c_j = \frac{1}{\sum_{i=1}^{NC} O_i} \left(\sum_{i=1}^{NC} \frac{O_i Q_{i,q}}{R_{i,k} - V_i} \right)$.

Следует отметить следующие особенности задачи: элементы матрицы ограничений a_{ij} неотрицательны, поскольку требования $\{Q_{i,j}\}$ логических серверов неотрицательные величины по определению. Коэффициенты c_j , очевидно, также являются неотрицательными величинами, поскольку O_i – булевы переменные, а $R_{i,k} - V_i > 0$, поскольку в противном случае задача не решается. Правые части b_i – должны быть неотрицательными, поскольку если для какого-либо ограничения значение $b_i = R_{i,k} - V_i$ окажется меньше либо равно нулю, то задача не будет иметь решения по определению – это случай явной нехватки ресурсов (базовые уровни ресурсов $\{R_{i,k}\}$ физических компьютеров и требования $\{V_i\}$ базовой ОС также неотрицательны по определению). Соответственно, в случае выявления нехватки ресурсов задача не должна решаться.

Теперь заметим, что математическая модель (3.19) представляет собой задачу о рюкзаке с множеством ограничений, принадлежащей классу задач условной псевдобулевой оптимизации. В рассматриваемой модели присутствуют линейные неравенства с булевыми переменными, причем сами коэффициенты при них и правые части неравенств в общем случае являются вещественными числами (по этой причине класс задач называют псевдобулевой, а не чисто булевой). Также задана целевая функция с булевыми переменными и с вещественными коэффициентами, которую требуется максимизировать: найти такое булево решение, при котором выполняются все ограничения и целевая функция достигает наивысшего значения при заданных ограничениях.

Для решения задач псевдобулевой оптимизации используются приближенные методы решения, например, метод локального поиска.

Также отметим, что благодаря декомпозиции исходной задачи объем перебора в худшем случае (когда на каждом «большом шаге» распределяется только один логический сервер, причем $NS \geq NH$) составит: $NH \cdot 2^{NS} + (NH - 1) \cdot 2^{NS-1} + \dots + 1 \cdot 2^{NS-(NH-1)} = (2^{-NH} + NH - 1) \cdot 2^{NS+1}$, при достаточно больших NH это $\sim NH \cdot 2^{NS+1}$, в случае если подзадачи решаются полным перебором. Такой объем перебора существенно меньше, чем при полном переборе при решении исходной задачи: $(NH + 1)^{NS}$, и немного больше, чем при последовательном рассмотрении подзадач оптимизации по отдельным физическим компьютерам без поиска на каждом шаге наилучшего среди них: $NH \cdot 2^{NS}$.

Таким образом, используя рассмотренные выше методы решения задачи поиска распределения и подзадач о рюкзаке, можно получить матрицу распределения логических серверов по компьютерам $\{X_{k,j}\}$, $k = 1 \dots NH$, $j = 1 \dots NS$.

Кроме того, используя матрицу распределения $\{X_{k,j}\}$ также можно вычислить матрицу загрузки ресурсов компьютеров $\{\eta_{k,i}\}$, $k = 1 \dots NH$, $i = 1 \dots NC$:

$$\eta_{ki} = \begin{cases} \sum_{j=1}^{NS} \frac{X_{kj} Q_{ij}}{R_{i,k} - V_i}, & R_{i,k} - V_i > 0; \\ 0, & R_{i,k} - V_i \leq 0. \end{cases} \quad (3.20)$$

$i = 1 \dots NC, k = 1 \dots NH$.

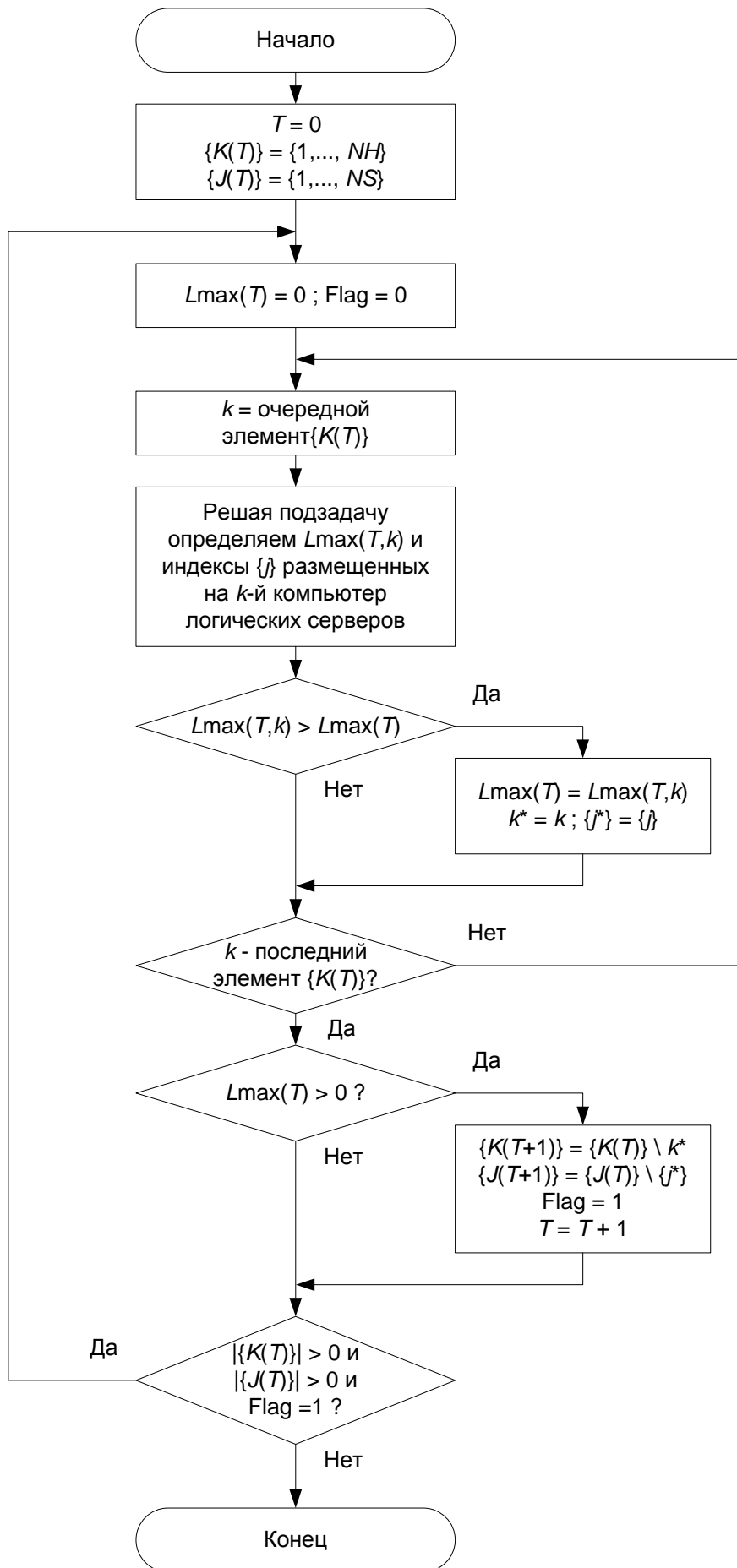


Рис. 3.6. Схема алгоритма решения задачи распределения логических серверов в целом.

3.2.3. Пример решения задачи распределения вычислительных ресурсов

Рассмотрим решение задачи распределения логических серверов по физическим компьютерам при применении технологии виртуальных машин на следующем примере.

Задан серверный парк корпоративной сети некоторого предприятия. Число компьютеров $NH = 15$, число логических серверов $NS = 15$, число типов ресурсов $NC = 3$. Число дополнительных ограничений $NX = 3$.

Базовые уровни ресурсов компьютеров, требования логических серверов и базовых ОС, вектор маски (оптимизация проводится по оперативной памяти и дисковому пространству), а также ограничения-исключения приведены в таблицах 3.3-3.6.

Таблица 3.3. Базовые уровни ресурсов компьютеров

	H1	H2	H3	H4	H5	H6	H7	H8
Память, MB	512	256	256	512	1024	1024	384	640
Диск, MB	20000	40000	30000	120000	120000	120000	60000	40000
Процессор, МТОPS	5200	5200	4500	4800	4300	3700	2200	2333
	H9	H10	H11	H12	H13	H14	H15	
Память, MB	512	512	384	384	768	640	512	
Диск, MB	80000	40000	25000	40000	50000	80000	120000	
Процессор, МТОPS	900	1200	2300	2300	3800	3800	4200	

Таблица 3.4. Требования логических серверов

	S1	S2	S3	S4	S5	S6	S7	S8
Память, MB	100	120	150	200	350	600	220	200
Диск, MB	9000	4500	2500	45000	32000	66000	20000	12000
Процессор, МТОPS	300	500	600	200	300	400	120	330
	S9	S10	S11	S12	S13	S14	S15	
Память, MB	125	220	120	200	270	340	120	
Диск, MB	35000	20000	10000	12000	22000	45000	90000	
Процессор, МТОPS	220	440	233	666	122	744	222	

Таблица 3.5. Требования базовой ОС и вектор маски оптимизации

	Базовая ОС			Маска
Память, MB	96		Память	1
Диск, MB	2000		Диск	1
Процессор, МТОPS	10		Процессор	0

Таблица 3.6. Матрица дополнительных ограничений-исключений

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Решая задачу с использованием рассмотренного выше метода, получаем матрицу распределения логических серверов по физическим компьютерам, а также матрицу загрузки ресурсов компьютеров (таблица 3.7).

Таблица 3.7. Матрица распределения логических серверов по компьютерам и матрица загрузки вычислительных ресурсов компьютеров.

Матрица распределения					Матрица загрузки ресурсов		
					Память	Диск	Процессор
Н1					0	0	0
Н2					0	0	0
Н3					0	0	0
Н4	S10	S15			81,73%	93,22%	13,82%
Н5	S2	S4	S6		99,14%	97,88%	25,64%
Н6	S5	S7	S14		98,06%	82,20%	31,54%
Н7					0	0	0
Н8					0	0	0
Н9					0	0	0
Н10					0	0	0
Н11	S1	S11			76,39%	82,61%	23,28%
Н12	S3	S9			95,49%	98,68%	35,81%
Н13	S8	S12	S13		99,70%	95,83%	29,50%
Н14					0	0	0
Н15					0	0	0

Из матрицы распределения видно, что задействовано 6 компьютеров: Н₄, Н₅, Н₆, Н₁₁, Н₁₂, Н₁₃, остальные 9 компьютеров освобождаются: Н₁, Н₂, Н₃, Н₇, Н₈, Н₉, Н₁₀, Н₁₄, Н₁₅.

Таким образом, серверный парк с достаточно слабой загрузкой вычислительных ресурсов может быть реорганизован с использованием технологии виртуальных машин, что в конечном счете приведет к существенному уменьшению объема используемого оборудования и затрат на его техническое обслуживание.

Список использованных источников

1. Самарский А.А. Михайлов А.П. Математическое моделирование. – М. Физматлит, 2005.
2. Демирчян К.С., Нейман Л.Р., Коровкин Н.В., Чечурин В.Л. Теоретические основы электротехники: в 3-х томах. – СПб.: Питер, 2003.
3. Черкесов Г.Н. Надежность аппаратно-программных комплексов. – СПб.: Питер, 2005.
4. Половко А.М., Гуров С.В. Основы теории надежности. – СПб.: БХВ-Петербург, 2006.
5. Гнеденко Б.В. Курс теории вероятностей. – М.: Едиториал УРСС, 2005.
6. Новиков Ф.А. Дискретная математика для программистов. СПб: Питер, 2009.
7. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985.
8. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся втузов. – 13-е изд. – М.: Наука, 1986.
9. Волков И.К., Канатников А.Н. Интегральные преобразования и операционное исчисление: учебник для вузов. – М.: МГТУ им. Н.Э. Баумана, 2002.
10. Дьяконов В.П. Maple 10/11/12/13/14 в математических расчетах. – М.: ДМК Пресс, 2011.
11. Рахман П.А. Топологический метод расчета среднего времени наработки до первого отказа систем управления на базе специального преобразования цепи Маркова // Экономика и менеджмент систем управления. – 2016. – Т. 22. – № 4.1. – С. 177-193.
12. Рахман П.А. Методика расчета коэффициента оперативной готовности систем управления в моделях надежности на базе цепей Маркова // Экономика и менеджмент систем управления. – 2018. – Т. 30. – № 4. – С. 90-99.
13. Рахман П.А. Модель задачи распределения вычислительных ресурсов в системах управления предприятием при применении технологии виртуальных машин // Экономика и менеджмент систем управления. – 2015. – Т. 18. – № 4.2. – С. 274-293.
14. Рахман П.А. Анализ вычислительной сложности решения задач псевдобулевой оптимизации методом локального поиска в системах управления предприятием // Экономика и менеджмент систем управления. – 2015. – Т. 18. – № 4.3. – С. 379-399.
15. Рахман П.А. Анализ нижней границы качества решения задач псевдобулевой оптимизации методом локального поиска в системах управления предприятием // Экономика и менеджмент систем управления. – 2016. – Т. 20. – № 2. – С. 92-99.