Bezruk V. M., Semenets V. V.,
Chebotarova D. V., Kaliuzhniy N. M.,
Guo Qiang, Zheng Yu

OPTIMIZATION AND MATHEMATICAL MODELING OF COMMUNICATION NETWORKS

Monograph

Kharkiv 2019

UDC 381.324:621.394.74
B 40

Reviewers:
Klymash Mykhaylo, Doctor of Technical Sciences, Heaf of Department of Telecommunication; Lviv Polytechnic National University;
Lemeshko Oleksandr, Doctor of Technical Sciences, Heaf of Department of Infocommunication engineering; Kharkiv National University of Radio Electronics

ISBN 978-617-7319-22-01 2nd edition PC «Technology Center», Kharkiv

ISBN 978-617-621-035-1 1st edition LLC «SMIT Company», Kharkiv

Bezruk V. M., Semenets V. V., Chebotarova D. V., Kaliuzhniy N. M., Guo Qiang, Zheng Yu

B 40 Optimization and mathematical modeling of communication networks, Monograph. 2nd edition, revised and supplemented. – Kharkiv: PC «Technology Center», 2019. – 192 p.

ISBN 978-617-7319-22-01

The main attention is paid to optimization issues, as well as mathematical modeling of systems that are important for the initial stages of the design of communication networks. Examples of solving scalar and vector optimization problems are given taking into account mathematical models of various types of communication networks. Information is given about standard software packages that can be used for optimization and computer simulation of communication networks.

Translated into English by Entrepreneur Kravchenko Anna

ISBN 978-617-7319-22-01

LIST OF ABBREVIATIONS

DM – decision maker (person who makes the decision).
OP – optimization problem.
UAC – unconditional advantage criterion (or the Pareto criterion).
MPC – multidimensional potential characteristics.
MED – multidimensional exchange diagrams.
CAC – conditional advantage criterion.
CN – communication network.
IS – information sources.
IC – information consumers.
EP – end points.
SN – switching nodes.
CC – communication channels.
QS – queuing system.
SD – service devices.
IDS – information decision system (in communication network).
TMID – technical means of implementing decisions.
CICS – control information collection system.
DCS – dynamic network resource control system.
MS – maintenance system.
ACS – administrative control system.
TMIC – technical means for the implementation of control decisions.
PTC – probability-time characteristics.
CSW – circuit-switched networks.
CS – capacity selection.
TCCDF – topology, capacity and distribution flows.
DF – distribution of flows.
CERM – concave edge removal method.
CCN – cellular communication networks.
BS – base station.
CA – control agents.
MOS – mean opinion score.
MSD – mean-square deviation.

CONTENT

INTRODUCTION

An important stage in the creation of any technical system is the design process. Designing is understood as a complex of works consisting of research, calculations and design in order to obtain a description of the system in the form of design documentation necessary to create a new or modernize an existing system. In the conditions of rapid development of information systems and communication networks, the growing demands on the quality and terms of their design can't always be satisfied by a simple increase in the number of designers. The solution to this problem is possible through the use of computer-aided design, which means the implementation of design procedures with the close interaction of a person with computers. Due to the automation of design, a reduction in the time and cost of design is achieved, as well as qualitatively new opportunities for performing design procedures, which in some cases is almost impossible to achieve without the use of computers.

Previously, heuristic and experimental methods were used in the design of systems. The design process was reduced to the selection of a small number of system options and only satisfying the given restrictions on the performance characteristics of the system. With the complication of systems and the growth of their cost, the need arose to create optimal systems. The phrase «We are not as rich as to design suboptimal» became winged. At present, when designing, they are no longer satisfied with the analysis of only one version of the system with specified technical characteristics. They are trying to compare as many alternative options for building a system as possible in order to choose among them the best – the best in the given sense.

Therefore, in modern conditions, automation of system design is relevant, in particular, at the initial stages – at the system-technical level. This is ensured by the development of new information technologies in the automation of system design. The basis of these technologies is the methods of mathematical modeling and optimization of systems, as well as their implementation on a computer in the form of appropriate application packages in order to obtain optimal design solutions.

Optimization is selection (synthesis) of the best in the given understanding of the structure and parameters of the system. The criterion of optimality determines a formalized procedure for choosing the best design option for a system.

When designing optimal systems, mathematical methods are used, the set of initial data for the design is formulated in the form of strict mathematical principles, in particular, mathematical models of the system are built, system quality

indicators are determined, a system optimality criterion is selected, and problems of optimizing the structure and parameters of the system are solved. The selection of optimal design options for the system is carried out by optimizing some target functions, showing the dependence of quality indicators on the structure and parameters of the systems.

The task of choosing the optimal design options for systems from the perspective of system analysis is a typical task of the branches of operations research and decision-making, including, in particular, the following stages:

1. Building a mathematical model of the system and formalizing the decision-making process on the optimal design options.
2. Finding optimal solutions on the set of feasible design decisions using mathematical optimization methods.

This monograph summarizes the issues of optimization and mathematical systems with specification for communication networks. The main attention is paid to the optimization and mathematical modeling of systems, which are important for the initial stages of the design of communication systems and networks. The theoretical foundations of scalar and multi-criteria optimization of systems and methods for solving various types of optimization problems that are concretized taking into account mathematical models of communication networks are considered. Examples of solving problems of optimizing communication networks are given. Information is given about standard software packages for modeling and optimization of computer communication networks that can be used in their computer-aided design.

Generalizations of well-known publications and some results of scientific studies of the authors regarding scalar and vector optimization, as well as mathematical modeling of communication systems and networks, are used in preparing the materials of the monograph.

The monograph may be useful to specialists in the design of telecommunication systems and networks.

1 THEORETICAL BASES OF SYSTEM OPTIMIZATION

This section discusses the main provisions of the branches of decision making, operations research, mathematical analysis, necessary for the mathematical formalization of system optimization problems. In general, an optimization problem is formulated and a classification of types of optimization problems is given.

In preparing the materials in this section, the works [1, 4, 6, 20, 24, 32, 36, 39] are used, which should be used additionally for an in-depth study of these issues.

1.1 Application of mathematical methods in the initial stages of system design

In the process of designing a complex system, it is necessary that at each stage optimal design decisions are made. Therefore, when designing a significant role is played by the following basic design procedures, such as optimization, synthesis, analysis of options for constructing a system. Optimization is the selection (synthesis) of the optimal (best in a certain sense) structure and parameters of the system. In this case, there are problems of synthesis and analysis. The choice of optimal system parameters is parametric synthesis, the choice of optimal structure is structural synthesis. The optimality criterion determines a formalized procedure for choosing the best design option. Analysis is an assessment of the potential capabilities of the system, determining the nature of changes in the values of system quality indicators, depending on how the structure and internal parameters of the system change. Quality indicators characterize the consumer properties of the system (for example, accuracy and speed of data transfer, message delay duration). Internal system parameters include system characteristics, in particular, modulation type, transmitter signal power, signal operating frequency, modulation coefficient, frequency deviation, etc., at which optimal values of quality indicators are achieved.

When using mathematical methods for designing optimal systems, the set of initial data for design is formulated in the form of strict mathematical principles, in particular, mathematical models of the system are built, system quality indicators and objective functions are determined, their dependence on the structure and system parameters is determined by the optimality criterion, and structural optimization problems are solved and system parameters.

The task of choosing the optimal design options for systems from the perspective of system analysis is a typical task of well-known areas of operations research, in particular, decision theory. The decision-making problem is the pair $\langle X,OP\rangle$, where X is the set of acceptable variants of the system, OP is the principle (criterion) of optimality, which defines the concept of the best (optimal) options. The solution to the problem $\langle X,OP\rangle$ is a subset of optimal options obtained using the given criterion of the optimality principle.

The mathematical expression of the optimality principle is a certain selection function, a subset of the acceptable variants X of its part $X_0 = C_0(X)$, which is a solution to the formulated selection problem. The tasks of making optimal decisions with the known X and OP are called optimization problems.

Let a certain property of alternatives from a set be expressed as a number, that is, there is an expression $X \to E_1$. Then such a property is called a quality indicator, and a number $k = f(x)$ is called an alternative assessment X by the objective function (criterion) $f(x)$. As a rule, alternative options of the system are characterized not by one but by several properties; it determines the need to characterize the system with a vector of quality indicators $\vec{K} = (k,\ldots,k_m)$. At the same time, the design solution x is evaluated by the totality of the objective functions $\vec{f}(x) = (f_1(x), ..., f_m(x))$, and the set X is expressed into the criteria space R^m, where each alternative $x \in X$ has its own evaluation vector $\vec{f}(x) \in R^m$.

The concept of optimality is associated with the selection of the best in the established understanding of the system options. This choice is made by optimizing some objective functions related to the quality indicators of systems. The question arises whether it is possible to find a design solution that satisfies the maximum (or minimum) at the same time of all objective functions. This is usually not possible. Therefore, for example, the wording «to achieve maximum system efficiency at minimum cost» does not make sense.

The selected indicators of system quality can be of three types: neutral, that is, independent of each other; interconnected but agreed upon; related and competing with each other. In the first two cases, system optimization can be performed independently with respect to each of the indicators. In the third, most difficult case, an improvement of some quality indicators leads to a deterioration of other quality indicators. Therefore, when solving the optimization problem, a consistent optimum of quality indicators should be sought.

There are many methods for optimizing systems; they put a different understanding into the concept of optimality. However, most of the rules for choosing the best solutions have a common feature: the choice is made on the basis of information about pairwise (binary) comparison of system options. Of course, such a comparison can be carried out on a variety of acceptable alternatives, but more

often it is convenient to perform in the criteria space, since here design decisions are compared using numerical estimates of quality indicators.

When designing systems, there are direct and inverse problems. Direct tasks are the tasks of analysis, they answer the question: what value does the optimality criterion of the chosen admissible option take? Inverse problems are synthesis problems, they answer the question: how to choose a solution for which the optimality criterion has reached an extreme value, in particular, if the analytical expression for the objective function is known (in the scalar case), then the optimal design solution is found alternative by solving some variational task:

$$x_0 = \arg \underset{x \in X}{opt} \left\{ f(x) \right\}.$$

The difficulties of solving optimization problems are associated with a priori uncertainty, in particular, about the operating conditions of the system. If this uncertainty is parametric, estimates of unknown parameters can be obtained, which are used to solve the optimization problem. In the general case, a priori uncertainty leads to complex systems.

The second type of a priori uncertainty is associated with the lack of sufficient information to formalize the vague idea of the customer of the system about the optimality of the system. This is a consequence of an insufficiently conscious, and therefore unclearly formulated global goal of the functioning of the system. Only the requirements for individual properties (a set of quality indicators) of the system are known. This leads to multicriteria (vector) optimization problems, in which it becomes necessary to search for the optimum of the vector objective function $\vec{f}(x) = \left(f_1(x), ..., f_m(x) \right)$.

Only in the case of neutral and coordinated quality indicators solution of such an optimization problem can be found by independent optimization of individual objective functions:

$$x_{0i} = \arg \underset{x \in X}{opt} \left\{ f_1(x) \right\}, \; i = \overline{1, m}.$$

In the case of interconnected and competing indicators of the quality of the system, the coincidence of individual decisions $x_{01} = x_{02} = ... = x_{0m}$ is more a case than a rule. In this case, the solution to the problem:

$$x_0 = \arg \underset{x \in X}{opt} \left\{ \vec{f}(x) \right\}$$

is a coordinated optimum, which corresponds to the best values of each of the quality indicators that can be achieved with fixed (but arbitrary) values of other indicators. By solving such optimization problems, as a rule, there is not one

alternative, but a certain set of alternative design solutions. The set of solutions satisfying the conditions of a consistent optimum of the vector objective function is called Pareto-optimal solutions.

Taking into account that quality indicators are interconnected, the potential values of each indicator, for example, the first k_{10} depends on the values of other $(m-1)$ quality indicators $k_2,...,k_m$. This dependence $k_{10} = f(k_2,...,k_m)$ for various acceptable combinations $k_2,...,k_m$ is called the working surface. Provided that the dependence is strictly monotonic for each indicator $k_2,...,k_m$, it is an optimal surface. All points of this surface satisfy the conditions of a multiple consistent optimum and therefore determine the multidimensional (m-dimensional) potential characteristics of the system. In terms of multicriteria (vector) optimization of the optimal surface, the set of Pareto-optimal values of the vector of quality indicators $\vec{K}$, as well as their corresponding Pareto-optimal systems, corresponds. The optimal surface also determines a family of multidimensional diagrams of the exchange of quality indicators, that is, the dependences of the potentially possible values of each of the quality indicators on the values of other $(m-1)$ indicators.

It should be noted that the one-dimensional potential characteristics of systems widely used in practice that characterize the potential value of a single quality indicator are, as a rule, a hidden form of a multidimensional potential characteristic. This is because when determining a one-dimensional potential characteristic, all quality indicators, except for one (the most important), are transferred to the rank of restrictions or completely ignored (i. e. are not taken into account). However, in fact, completely ignoring all quality indicators, except for one, is unacceptable, since in this case the potential value of this single optimized quality indicator will reach zero, that is, the optimization problem to be solved will degenerate into a trivial one. Thus, when optimizing, the selected main quality indicator should definitely take into account the value of at least one more antagonistic indicator. In practice, it is necessary to $X_0 \subseteq X$ take into account, as a rule, several quality indicators. It follows that in the general case, all the optimization problems of designing complex systems is essentially multi-criteria.

For the further selection of a unified design solution for the obtained set of Pareto-optimal options, additional information must be involved that clarifies and formalizes the initial vague idea of the customer about the optimality of the system. The use of such information allows to determine some constructive procedure for choosing a single design option for the system. Thus, to design an optimal system means to find the structure that is optimal according to the vector criterion and the optimal values of the system parameters. To do this, it is necessary to build a mathematical model of the system and solve mathematical problems in the general case of multicriteria optimization with given objective functions and

constraints. Compared to scalar optimization problems, it is much more complicated than mathematical problems.

In many cases, the synthesis and analysis of the effectiveness of systems is performed by analytical methods. But often solving these problems encounters difficulties not only of a technical, but also of a fundamental nature. In such cases, when designing complex systems, numerical methods for optimizing and modeling computer systems are also widely used. Modeling a computer system includes building and implementing a mathematical model of a computer system and conducting its research. The purpose of system simulation is verification even at the initial stages of designing the correctness of the selected design solutions, to assess the multidimensional potential characteristics of the system, as well as the possibility of fulfilling the technical requirements for obtaining the necessary tactical and technical characteristics of the system. It should be noted that the correction of design errors at the stage of preparation of technical documentation is much cheaper than at the stages of manufacturing research samples of the system, serial production and operation of the system.

Thus, computer simulation and optimization of the system are important and effective stages of the initial stages of system design. They are performed, as a rule, using a computer. This design method, when design procedures are carried out in close interaction between the designer and the computer, is called computer-aided design. Corresponding software packages and organizational and technical tools that implement design procedures are called computer-aided design systems. Computer-aided design is characterized by a rational distribution of functions between a person and a computer. With the help of computers, those tasks that are subject to formalization are solved, but provided that their machine solution is more effective than «manual». This design method is characterized by the use of computers in the automation of individual design procedures that were previously performed by the designers «manually» (calculations, optimization, studies of the potential characteristics of the system and multidimensional diagrams of the exchange of quality indicators, processing the results of system studies, issuing project documentation, etc.).

For the automated design of a complex system, not only the simple «operation» of the significant technical capabilities of the computer (high speed and memory), but also a deep knowledge of the subject area is necessary. This is necessary to build the appropriate mathematical models of the system and use the modern mathematical apparatus of the theory of system optimization to select design solutions that are optimal according to the vector optimality criterion. This raises a number of questions. How to form a set of acceptable options for building a system? How to formalize the goals for which the system is created? How, among all the possible options, how to find the optimal one – «the most suitable for the set goals»? To answer these questions, it is necessary to build a mathematical

model and mathematically rigorously pose the problem of synthesis of the optimal system, taking into account the totality of the requirements of the technical task.

1.2 Basic provisions of the field of operations research

Operations research is the theory of using quantitative methods of analysis in decision-making in targeted activities.

Under the operation refers to the totality of actions aimed at achieving a specific goal. The researcher of the operation helps on the operating side – the decision maker (DM) in a particular choice problem gives the scientific basis for the choice of solutions, formalization of this task, including the construction of a mathematical model for choosing the optimal solution.

The mathematical model of an operation is a formal relationship that establishes the relationship between the adopted criterion for the effectiveness of decisions on the factors of the operation –parameters that are controlled.

The main objective of operations research: to find, within the framework of the adopted model, such solutions that correspond to the extreme values of the performance criteria. The criterion becomes the equivalent of the goal of the operation.

At the same time, the following stages of operations research take place:

1. The construction of a mathematical model, that is, the formalization of the decision-making process, (identification of the model, verification of its adequacy).
2. The statement of the goal of the operation and the task, finding the optimal solutions to the operation on the set of possible solutions X and a variety of factors Ξ.
3. The solution to the optimization problem based on well-proven optimization methods.

Factors that are controlled by the operating side are called controlled. Factors that are not controlled by the operating side are called uncontrolled. Uncontrolled factors are divided into two groups: uncertain factors for which only the set of possible values of factors are known, as well as random factors for which a set of random values Z and the law of their distribution are specified.

The desire on the part of the parties to achieve the goal is described by the values of the objective function $F(x,y,z)$, which is also called the criterion of the effectiveness (optimality) of decisions.

In some tasks, they seek to maximize the objective function, in some they minimize it. In the general case, one speaks of finding the extremum of the objective function. The information that the operating side has about the factors is reflected in the operation research model.

Depending on the degree of completeness of information about the factors of the operation, various strategies are selected on the operating side. This can be a search for an extremum of the average efficiency function or a search for its lower or upper face.

Thus, for the researchers of the operation, it is first necessary to compile a mathematical model of the operation, that is, at a formalized level, describe controlled and uncontrolled factors, set a performance criterion (optimality) and a variety of strategies on the operating side, as well as describe estimates of the effectiveness of the strategy. Then, using appropriate mathematical optimization methods, optimal strategies are found for the prevailing transaction conditions.

1.3 Decision-making model for the criteria approach

Any decision-making process contains the following elements:
- the person who makes the decision (DM);
- a lot of controlled variables, the value of which the decision maker chooses;
- many uncontrollable variables;
- restrictions on controlled and output variables;
- objective function (optimality criterion);
- rules for making optimal decisions.

The task of decision-making is selection of values of controlled variables that render the objective function an extreme value.

However, the possibility of choosing a controlled variable is always limited by external conditions relative to the operation (energy, time, money factors, etc.). Ideally, these restrictions can be mathematically described by some system of equalities or irregularities on uncontrolled and control variables.

From a mathematical point of view, the problem of making optimal decisions is formulated as an optimization problem (OP), which is a tuple:

$$< f(\vec{x}), X, Y >, \tag{1.1}$$

where $f(\vec{x})$ – objective function which extremum must be found by varying the values of the controlled variables, in particular, the components of the parameter vector $\vec{x} = (x_1, x_2, ..., x_n)$, X – area of the objective function $f(\vec{x})$, Y – area of restrictions imposed on the vector of parameters $\vec{x}$.

Area $D = X \cap Y$ is called an admissible set, vectors $\vec{x} \in D$ are called admissible OP vectors. If the admissible set D coincides with the Euclidean space R^n, then such an OP is called the unconditional optimization problem.

An admissible vector $\vec{x}_0$ is called an absolute (global) minimum if the condition:

$$< f(\vec{x}), X, Y > . \tag{1.2}$$

An admissible vector $\vec{x}_{0l}$ is called a local minimum if there exists such $\delta > 0$ that for all $0 \le |\Delta\vec{x}| \le \delta$ the valid condition:

$$\Delta f = f(\vec{x}_{0l} + \Delta\vec{x}) - f(\vec{x}_{0l}) \ge 0. \tag{1.3}$$

For the case of finding the maxima of the objective function $f(\vec{x})$, the inequality sign in (1.2) and (1.3) is reversed. Thus, the solution of the optimization problem $\vec{x}_0$ is reduced to a search on the set of feasible solutions D of the extremum of the objective function $f(\vec{x}) \Rightarrow extr, \ \vec{x} \in D$.

A local minimum is called an internal or limit point if the point $\vec{x}_0$ is respectively the internal or limit point of the area D. For example, the scalar function $f(x) = x$ for $-\infty < x < +\infty$ does not have extrema, while $f_1(x) = x$ for $x \le 1$ has a limit maximum at a point $x = 1$, while for $0 \le x \le 1$ also has a limit minimum at a point $x = 0$. Considered approach to making optimal decisions is called criteria-based, since decisions are evaluated and compared by the value of the objective function (optimality criterion).

There is another approach to making optimal decisions, based on the introduction of binary relations.

1.4 Model for choosing optimal solutions in the language of binary relations

The language of binary relations is the second, more general than the original, language for describing the benefits of DM.

A binary relation is a set R of ordered pairs if the elements (alternatives) $x^*, x^{**} \in X$ are in a relation R where they write this: $(x^*, x^{**}) \in R$ or $x^* R x^{**}$.

The concept of a binary relation R allows to formalize the operations of pairwise comparison of alternatives. It is believed that each pair of solutions x^*, x^{**} can be in one of the following relationships:

- x^* dominates (or strictly dominates) x^{**};
- x^{**} dominates (or strictly dominates) x^{**};
- x^* no less dominates x^{**};
- x^{**} no less dominates x^*;
- x^* equivalent x^{**};
- x^*, x^{**} are incomparable.

A solution $x_o \in X$ is called optimal in the choice model (X,R) if it is no less predominant than any other element x from the set X, that is, there are no other solutions $x \in X$ so that they dominate x_o.

In the case of the introduction of a vector criterion for evaluating the effectiveness of comparison solutions $\vec{K}(x) = (K_1(x),...,K_m(x))$, the selection of optimal solutions can also be performed in the criterion space Y-space of vector estimates of the set of system quality indicators $\vec{y} = \vec{K}(x) = (K_1(x),...,K_m(x)) \in Y$.

The following binary relations are used for vector estimates:

– strict dominance relation – Slater relation for which conditions are satisfied:

$$\vec{y}'\Phi_1\vec{y}'' \leftrightarrow \vec{K}(x') > \vec{K}(x''), \quad K_i(x') > K_i(x''), \quad i = \overline{1,m}; \tag{1.4}$$

– non-strict dominance relation – Pareto relation for which conditions are satisfied:

$$\vec{y}'\Phi_1\vec{y}'' \leftrightarrow \vec{K}(y') \geq \vec{K}(y''), \quad K_i(x') \geq K_i(x''),$$

$$i = \overline{1,m}, \quad \vec{K}(x') \neq \vec{K}(x''). \tag{1.5}$$

In the criteria space, the set of optimal estimates includes those for which there are no other dominant estimates for the selected advantage ratio Φ. In particular, when introducing the Pareto binary relation in the space of vector estimates Y, the procedure for choosing a subset of Pareto optimal estimates $P_{\geq}(Y)$ is formulated as follows: the vector estimate is included in the subset $\vec{y}_o \in P_{\geq}(Y)$, provided there are no other vector estimates $\vec{y} \in Y$ for which there would be a fair advantage relation $\vec{y} \geq \vec{y}_0$.

Thus, optimal solutions can be chosen both on the set X and in the criteria space Y. The solution $x^0 \in X$ is called R-optimal on the set X, if there are no other solutions $x \in X$ that prevailed solutions x^o in the binary relation R. In particular, when introducing a Pareto binary relation in a space Y, a subset of Pareto-optimal estimates $P_{\geq}(Y)$ corresponds to a subset of Pareto-optimal solutions $X_0 = P_{\vec{k}}(X)$ according to the vector criterion for the effectiveness of solutions $\vec{K}(x) = (K_1(x),...,K_m(x))$. This subset includes solutions $x_0 \in X_0 = P_{\vec{k}}(X)$ for which there are no dominant solutions according to the condition $\vec{K}(x) \geq \vec{K}(x_0)$, $x \in X$. According to the Pareto axiom, there is a relationship between the choice of solutions in the space of estimates Y and the set X: if $\vec{y} \in P(Y)$, then $x \in P_{\vec{k}}(X)$.

If the strict dominance relation (1.4) is introduced, then there are many optimal estimates that are Slater-optimal (or weakly Pareto-optimal). These estimates correspond to Slater-optimal solutions.

1.5 Fundamentals of mathematical analysis used in optimization theory

1.5.1 Euclidean space

Set. Under the set usually understand a certain set of elements of an arbitrary nature, characterized by a common property. Examples of the set can serve as a collection of pages in this book, printed characters, formulas. The combination of real, natural and integer numbers is example of numerical sets.

The phrase «x is an element of the set X» («x belongs to the set X») is written briefly in the form $x \in X$. If x it does not belong to the set X, then write $x \notin X$.

Sets X and Y are called equal if they consist of the same elements.

If each element of the set X is an element of the set Y, then they say that X is a subset of the set Y and write $X \subset Y$.

Symbols $\in$ and $\subset$ are called signs of inclusion, and the ratio of the form $x \in X$ and $X \subset Y$ – inclusions.

Equality $X = Y$ takes place if and only if at the same time $X \subset Y$ and $Y \subset X$.

Notation $X = \{x_1, x_2, \ldots\}$ means that the set X consists of elements x_1, x_2 and, possibly, some others set in one way or another. If the set X consists of elements x having a certain property $\mathrm{P}(x)$, then write $X = \{x | \mathrm{P}(x)\}$. For example, $(0,1] = \{x | 0 < x \le 1\}$.

In order to define a set X which elements belong Y and, in addition, have a property $\mathrm{P}(x)$, let's use the notation $X = \{x \in Y | \mathrm{P}(x)\}$.

A set containing no elements is called an empty set and denoted by $\varnothing$.

The union of two sets X, Y isdenoted by $X \cup Y$, the elements of which belong to at least one of the sets X or Y. The operation of combining sets has the following properties:

$$\begin{aligned} &1)\ X \cup Y = Y \cup X, \\ &2)\ (X \cup Y) \cup Z = X \cup (Y \cup Z), \\ &3)\ X \cup X = X,\ \ X \cup \varnothing = X, \end{aligned} \tag{1.6}$$

where X, Y, Z are arbitrary sets. The first two properties are similar to the properties of the addition operation for numbers.

The section of sets X, Y is called the set, which is denoted by $X \cap Y$, the elements of which belong to both the set X and the set Y. The intersection operation of sets has the following properties:

$$\begin{aligned} &1)\ X \cap Y = Y \cap X, \\ &2)\ (X \cap Y) \cap Z = X \cap (Y \cap Z), \\ &3)\ X \cap X = X,\ \ X \cap \varnothing = \varnothing. \end{aligned} \tag{1.7}$$

The following properties are performed for the entered union and intersection operations:

1) $(X \cup Y) \cap Z = (X \cap Z) \cup (Y \cap Z)$.

2) $(X \cap Y) \cup Z = (X \cup Z) \cap (Y \cup Z)$. (1.8)

The difference between sets X and Y is the set consisting of elements that belong to the set X, but do not belong to the set Y ($X \setminus Y$ designation).

Euclidean space. An ordered set of n real numbers written as:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{1.9}$$

is called n-dimensional vector. Numbers $x_1, x_2, ..., x_n$ are called the components (or coordinates) of the vector $\vec{x}$. Using the transpose matrix operation, the coordinates of the vector are written as a row $\vec{x}^{tr} = (x_1, x_2, \ldots, x_n)$.

Often for reduction, if this does not lead to inaccuracy, the transpose sign is omitted. Two n-dimensional vectors $\vec{x}$ and $\vec{y}$ are considered equal if their respective components coincide, that is, if $x_i = y_i$, $i = 1,2\ldots,n$.

The sum of the vectors $\vec{x}$ and $\vec{y}$ is the vector whose components are found by the formula:

$$\vec{x} + \vec{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n). \tag{1.10}$$

The difference of vectors $\vec{x}$ and $\vec{y}$ is determined in this way:

$$\vec{x} - \vec{y} = (x_1 - y_1, x_2 - y_2, \ldots, x_n - y_n). \tag{1.11}$$

A zero vector (denoted by $\vec{0}_n$) is called n-dimensional vector, all components of which are equal to zero. Multiplication of a vector by a real number is a vector whose components are found by the formula:

$$\lambda\vec{x} = (\lambda x_1, \lambda x_2, \ldots, \lambda x_n). \tag{1.12}$$

The definitions directly imply the validity of the following equalities:

$$\vec{x} + \vec{y} = \vec{y} + \vec{x}, \quad (\vec{x} + \vec{y}) + \vec{z} = \vec{x} + (\vec{y} + \vec{z}), \quad 0\vec{x} = \vec{0},$$

$$\lambda(\vec{x} + \vec{y}) = \lambda\vec{x} + \lambda\vec{y}, \quad (\lambda_1 + \lambda_2)\vec{x} = \lambda_1\vec{x} + \lambda_2\vec{x}. \tag{1.13}$$

Each pair of vectors $\vec{x}, \vec{y}$ is associated with a number, denoted by $\langle \vec{x}, \vec{y} \rangle$ and called the scalar product of vectors $\vec{x}$ and $\vec{y}$, which is determined by the formula:

$$\langle \vec{x}, \vec{y} \rangle = \sum_{k=1}^{n} x_k y_k. \tag{1.14}$$

Let's note the properties that characterize the scalar product:

1) $\langle \vec{x}, \vec{y} \rangle = \langle \vec{y}, \vec{x} \rangle$;
2) $\langle \vec{x} + \vec{y}, \vec{z} \rangle = \langle \vec{x}, \vec{z} \rangle + \langle \vec{y}, \vec{z} \rangle$;
3) $\langle \lambda \vec{x}, \vec{y} \rangle = \lambda \langle \vec{x}, \vec{y} \rangle$;
4) $\langle \vec{x}, \vec{x} \rangle \geq 0$ (equality takes place if and only if $\vec{x}$ is the zero vector).

The validity of the above properties can easily be verified by relying directly on the definition of a scalar product.

The set of all n-dimensional vectors for which the operations of addition, subtraction, and multiplication by a real number are introduced, as well as a scalar product by the formulas (1.11)–(1.14), are called n-dimensional real space and are denoted by R^n. In short, this space is simply called Euclidean.

The set of real numbers is further denoted by the letter R. This set is an example of one-dimensional Euclidean space. Let's geometrically interpret the space R^2 as a set of points on a plane with a fixed rectangular coordinate system or as a set of vectors on this plane whose origin coincides with the beginning of a rectangular Cartesian coordinate system. Space R^3 has a similar interpretation in the space of three dimensions.

Taking into account the above interpretation, the elements of space R^n are also called points.

If $\langle \vec{x}, \vec{y} \rangle = 0$, then the vectors $\vec{x}$ and $\vec{y}$ are called orthogonal. In particular, the zero vector is orthogonal to any vector.

The norm (or length) of a vector $\vec{x} \in R^n$ is a number, denoted by $\|\vec{x}\|$ and determined by the formula:

$$\|\vec{x}\| = \sqrt{\langle \vec{x}, \vec{x} \rangle} = \sqrt{\sum_{k=1}^{n} x_k^2}. \tag{1.15}$$

Let's formulate the main properties of the norm of a vector:

1) $\|\vec{x}\| \geq 0$, and $\|x\| = \vec{0}$ if and only if $\vec{x} = 0$;
2) $\|\lambda \vec{x}\| = |\lambda| \cdot \|\vec{x}\|$, where λ – the number;
3) $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$ (triangle inequality);
4) $|\langle \vec{x}, \vec{y} \rangle| \leq \|\vec{x}\| \cdot \|\vec{y}\|$ (Cauchy-Bunyakovsky inequality).

The distance between the points $\vec{x}$ and $\vec{y}$ of the Euclidean space is denoted by $\rho(\vec{x},\vec{y})$ and determined as follows:

$$\rho(\vec{x},\vec{y}) = \|\vec{x}-\vec{y}\| = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}. \tag{1.16}$$

A system of vectors $\vec{x}^{(1)}$, $\vec{x}^{(2)}$, …, $\vec{x}^{(m)}$ is called a linearly independent system if equality $\lambda_1\vec{x}^{(1)} + \lambda_2\vec{x}^{(2)} + \ldots + \lambda_m\vec{x}^{(m)} = \vec{0}_n$ is possible only in the case $\lambda_1 = \lambda_2 = \ldots = \lambda_m = 0$. A system of vectors, which is not linearly independent, is called a linearly dependent system.

In a *n*-dimensional space there exists a linearly independent system of *n*-vectors, and any of $n+1$ (and more) vectors is linearly dependent. Any linearly independent system $\left\{\vec{e}^{(k)}\right\}\cdots k = \overline{1,n}$ of vectors in *n*-dimensional space forms a basis. Moreover, each vector of space $\vec{x}$ is uniquely represented as a linear combination of basis vectors: $\vec{x} = \lambda_1\vec{e}^{(1)} + \lambda_2\vec{e}^{(2)} + \ldots + \lambda_n\vec{e}^{(n)}$ for some λ_1, λ_2, …, λ_n.

For vectors $\vec{x} \in R^n$, $\vec{y} \in R^n$, it is believed that the notation $\vec{x} \geq \vec{y}$ means $x_i \geq y_i$, $i = 1,2,\ldots,n$, and at least one inequality is strict, and the notation $\vec{x} > \vec{y}$ means: $x_i > y_i$, $i = 1,2,\ldots,n$.

Linear sets. An empty subset L of space R^n is called a subspace if, as a result of addition of any two vectors of L, as well as multiplication of an arbitrary vector of L, by any real number of resulting vectors belonging L. If the maximum number of linearly independent vectors that can be found in L is equal r, then it is possible to say that there are *r*-dimensional subspace is in L. Space R^n itself can be considered as *n*-dimensional subspace.

A one-dimensional linear set is called a straight line, a two-dimensional set is called a plane, and $(n-1)$-dimensional one is called a hyperplane. Any line in R^n can be set in the form:

$$\left\{\vec{x} \in R^n \middle| \vec{x} = \vec{a} + \lambda\vec{c}, \text{ for some } \vec{x} \in R^n\right\}, \tag{1.17}$$

choosing vectors $\vec{a}$ and $\vec{c}$ from R^n. If in (1.17) the number λ is bounded above or below, then obtain a ray. If λ is bounded both above and below, then the set (1.17) defines a line. A line connecting points $\vec{x}'$, $\vec{y}' \in R^n$ is a set of the form:

$$\left\{\vec{x} \in R^n \middle| \vec{x} = \vec{a} + \lambda\vec{x}' + (1-\lambda)\vec{y}', \text{ for some } 0 \leq \lambda \leq 1\right\}. \tag{1.18}$$

Many solutions $\vec{x} \in R^n$ found with the equation:

$$\langle\vec{a},\vec{x}\rangle = b, \tag{1.19}$$

The converse is also true: any hyperplane in R^n can be specified in the form of a set of solutions of equation (1.19), choosing a vector $\vec{a}$ and a number b accordingly.

1.5.2 Sets in Euclidean space

The set of species:

$$U_\varepsilon\left(x^{(0)}\right)=\left\{\vec{x}\in R^n \,\middle|\, \left\|\vec{x}-\vec{x}(0)\right\|<\varepsilon\right\}$$

is called an open sphere of radius ε centered at a point $\vec{x}^{(0)}\in R^n$ or ε-neighborhood of a point.

Let $\vec{x}^{(0)}\in R^n$. A point $\vec{x}\in X$ is called an interior point of the set if there is such $\varepsilon>0$ that $U_\varepsilon\left(\vec{x}^0\right)\subset X$, that is, if the point x belongs to the set X together with its certain margin.In the case where each point of the set X is internal, this set is called an open set. For example $U_\varepsilon(\vec{x}^{(0)})$ is an open set.

A point in space R^n, which does not necessarily belong to a set X, is called a limit point of a set $X\subset R^n$ if in its neighborhood of any radius $\varepsilon>0$ there is at least one point from X and at least one point does not belong X. The set of boundary points of a set forms its boundaries.

A set X is called convex if for any pair of points from X the entire segment connecting these points also belongs to X.

1.5.3 Functions of many variables

Continuous differentiated functions. Let X and Y are two sets. If a rule is specified according to which each element of the set X is assigned a certain element of the set Y, then they say that a function is displayed f that displays X in Y. This is written in the form $f:X\to Y$ or $y=f(x)$, where $x\in X$, $y\in Y$. The set X is called the task domain or function f determination domain, and the set Y is called the set of function values.

A function continuous at each point of a set X is called continuous on the set X (or simply continuous).

As examples of functions continuous on $X=R^n$, let's give:

– linear function:

$$f_1(\vec{x})=\langle\vec{c},\vec{x}\rangle+b=c_1x_1+c_2x_2+\ldots+c_nx_n+b,$$

– quadratic function:

$$f_2(\vec{x}) = \frac{1}{2}\langle Q\vec{x}, \vec{x}\rangle + \langle \vec{c}, \vec{x}\rangle + b,$$

where Q – a numerical symmetric matrix of size $n \times n$, $\vec{c}$ – a vector with R^n and b – some numbers, $Q\vec{x}$ means the product of the matrix and the vector according to the matrix multiplication rules adopted in linear algebra.

Let $\vec{x}^{(0)}$ – interior point of the set X. A function $f(\vec{x})$ is called differentiated at a point $\vec{x}^{(0)}$ if there exists a vector $\vec{p} \in R^n$ such that for all $\vec{h} \in R^n$ satisfying the condition $\left(\vec{x}^{(0)} + \vec{h}\right) \in X$ the formula is:

$$f\left(\vec{x}^{(0)} + \vec{h}\right) = f\left(\vec{x}^{(0)}\right) + \left\langle \vec{p}, \vec{h} \right\rangle + \left\| \vec{h} \right\| \left(\vec{x}^{(0)} + \vec{h}\right). \tag{1.20}$$

If the specified vector $\vec{p}$ exists, then it is called the gradient of the function $f(\vec{x})$ at the point $\vec{x}^{(0)}$ and denoted by $\Delta f\left(\vec{x}^{(0)}\right)$. It is known that if a function $f(\vec{x})$ is differentiated at a point $\vec{x}^{(0)}$, then it is continuous at this point and is characterized by a gradient:

$$\Delta f\left(\vec{x}^{(0)}\right) = \left(\frac{\partial f(x_1)}{\partial x_1}, \frac{\partial f(x_2)}{\partial x_2}, \ldots, \frac{\partial f(x_n)}{\partial x_n} \right)^{\tau}, \tag{1.21}$$

which is a vector with coordinates in the form of first-order partial derivatives calculated at a point $x^{(0)}$.

A function $f(\vec{x})$ that differentiates at each point $\vec{x}$ of an open set X is called differentiated on the set X.

The linear and quadratic functions given above are continuously differentiated, moreover $\nabla f_1(\vec{x}) = c$, $\nabla f_2(\vec{x}) = Q\vec{x} + c$.

*Convex, pseudo-convex, and quasi-convex function*s. Convex functions and their generalizations (pseudo-convex and quasi-convex functions) play an important role in optimization theory. Using these functions, let's formulate sufficient optimality conditions.

A numerical function $f(\vec{x})$ defined on a convex set $X \subset R^n$ is called convex if, for any two points $\vec{x}^{(1)}$, $\vec{x}^{(2)} \in X$ and an arbitrary number $\lambda \in [0,1]$, the inequality is satisfied:

$$f\left(\lambda\vec{x}^{(1)} + (1-\lambda)\vec{x}^{(2)}\right) \le \lambda f\left(\vec{x}^{(1)}\right) + (1-\lambda) f\left(\vec{x}^{(2)}\right). \tag{1.22}$$

For example, a function of one variable, the property of convexity of the function $f(\vec{x})$ geometrically means that the line of its graph corresponding

to the points of the segment connecting the points $x^{(1)}$ and $x^{(2)}$, is placed no higher than the line connecting the points of the graph $(x^{(1)}, f(x^{(1)}))$ and $(x^{(2)}, f(x^{(2)}))$. A simple example of a convex function of one variable is parabola.

1.6 Mathematical statement of the optimization problem

Valid set and objective function. The statement of the optimization problem includes a set X of feasible solutions X and a numerical function $f(\vec{x})$ defined on the set X, which is called the objective function. The set X is also called the feasible set or set of possible solutions.

The concept of solution is identified with a vector (point) of n-dimensional Euclidean space R^n. In accordance with this admissible set X is a certain subset of space R^n, that is $X \subset R^n$, and the objective function $f(\vec{x})$ is a function n of variables, x_1, x_2, …, x_n. The case of equality $X = R^n$ is not excluded. For the elements of the set X, along with the term «solution», the terms «vector» and «point» are used below.

Not strictly speaking, the optimization task consists in choosing among the elements of the set X of such a solution that would be most preferable from a certain point of view. The comparison of solutions is predominantly carried out using the values of the objective function. There are two options for comparing an arbitrary pair of solutions $x^{(1)} \in X$, $x^{(2)} \in X$ using a function $f(\vec{x})$. It is possible to assume that the solution $\vec{x}^{(1)}$ prevails $\vec{x}^{(2)}$ if the inequality $f(\vec{x}^{(1)}) < f(\vec{x}^{(2)})$ holds. Then the search for the most preferable solution among all elements of the set X consists in finding a solution that delivers the least possible value of the objective function $f(\vec{x})$ on the set X. In this case, the optimization problem is the minimization problem. If it is believed that the solution $\vec{x}^{(1)}$ prevails $\vec{x}^{(2)}$ when the inequality $f(\vec{x}^{(1)}) > f(\vec{x}^{(2)})$ is satisfied, then the search for the most preferable solution is the problem of maximizing the function $f(\vec{x})$ on the set X.

To solve the problem of optimizing a function $f(\vec{x})$ on a set X means to find such a vector $\vec{x}^{(0)} \in X$ (and also the corresponding value $f(\vec{x}^{(0)})$ so that the inequality $f(\vec{x}^{(0)}) \le f(\vec{x})$ holds for everyone $\vec{x} \in X$. Moreover, the solution $\vec{x}^{(0)}$ is called optimal (more precisely, the minimum), and the value $f(\vec{x}^{(0)})$ is called optimum (minimum). The fact that the solution $\vec{x}^{(0)}$ is optimal, that is, it delivers the smallest possible value of the function $f(\vec{x})$ on the set, X is written as:

$$f\left(\vec{x}^{(0)}\right) = \min_{\vec{x} \in X} f(\vec{x}). \tag{1.23}$$

When setting the minimization problem, one also uses the notation:

$$f(\vec{x}) \to \min_{x \in X}. \tag{1.24}$$

Similarly, the maximization problem consists in finding such a vector $\vec{x}^{(0)} \in X$ (and the corresponding value $f(\vec{x}^{(0)})$) for which the inequality $f(\vec{x}^{(0)}) \geq f(\vec{x})$ holds for all $\vec{x} \in X$. If $\vec{x}^{(0)}$ is the solution to the maximization problem, then the notation is used:

$$f\left(\vec{x}^{(0)}\right) = \max_{x \in X} f(\vec{x}). \tag{1.25}$$

In this case $\vec{x}^{(0)}$ is the maximum (optimal) solution and the value $f(\vec{x}^{(0)})$ is the maximum (optimum).

Thus, to solve the optimization problem means to find the optimal point $\vec{x}^{(0)}$ and optimal value $f(x^{(0)})$. If the optimal point is found, then the determination of the optimal value of the function is usually not difficult. If the extreme value of the function is found, then to find the optimal point $\vec{x}^{(0)}$ it is necessary to solve the equation $f(\vec{x}) = f(\vec{x}^{(0)})$, it can be a difficult computational task.

Local and global minimums. During optimization, two types of optimum are considered: local and global. It is said that a point $\vec{x}^{(0)} \in X$ delivers functions on a set X of local minimums if there exists a neighborhood $U_\varepsilon(\vec{x}^{(0)})$ $(\varepsilon > 0)$ of the point $\vec{x}^{(0)}$ such that the inequality $f(\vec{x}^{(0)}) \leq f(\vec{x})$ holds for all $\vec{x} \in X \cap U_\varepsilon(\vec{x}^{(0)})$. The global minimum of the function $f(\vec{x})$ is delivered by the point $\vec{x}^{(0)} \in X$ for which the inequality written above is satisfied for all $\vec{x} \in X$. The adjective «global» is used to emphasize the difference between this minimum and the local minimum.

In accordance with the above definitions, in the first case, the point $\vec{x}$ is called the local minimum point, and in the second case, the global minimum point.

The point of local minimum is not always the point of global minimum, which means that the local minimum does not always coincide with the global minimum. In the following statement, conditions are formed that are imposed on the set X and the function $f(\vec{x})$, upon which the indicated points coincide. If the set $X \subset R^n$ is convex, and the function $f(\vec{x})$ is convex or pseudo-convex on X, then every point of the local minimum is a point of global optimum.

Weierstrass theorem. If the set $X \subset R^n$ is not empty and compact, and the function $f(\vec{x})$ is continuous on it, then the set of global minimum points (as well as the set of global maximum points) is not empty and compact.

Generalized optimization problem. In optimization theory, it is sometimes convenient to consider a more general optimization problem in which the concept

of a solution is defined in such a way that it always exists. In order to formulate this generalized problem, it is used to determine the exact lower bound.

A number f^0 is called a lower bound or infimum of a function $f(\vec{x})$ on a set X if an inequality $f^0 \le f(\vec{x})$ holds for everyone $\vec{x} \in X$, and in addition, for any number $f' > f^0$ there is a point $\vec{x}' \in X$ such that the inequality $f(\vec{x}') < f'$ holds. The fact that f^0 is the exact lower bound of the function f^0 on the set X is written as:

$$f^0 = \inf_{\vec{x} \in X} f(\vec{x}). \tag{1.26}$$

It is not always possible to indicate the point at which the exact bound is reached, that is, the point $\vec{x}^0$ for which $f(\vec{x}^{(0)}) = \inf_{x \in X} f(\vec{x})$. Therefore, in a generalized minimization problem $f(\vec{x}) \to \inf_{\vec{x} \to X} f(\vec{x})$, a solution is understood not as a single point, as is the case in the usual optimization problem, but as a sequence of points $\left\{\vec{x}^{(k)}\right\}_{k=1}^{\infty}$, $\vec{x}^{(k)} \in X$, $k = 1, 2, ...,$ such that:

$$\lim_{k \to \infty} f\left(\vec{x}^{(k)}\right) = f^0. \tag{1.27}$$

This sequence always exists and is called minimizing sequence.

If, $X = R^n$ then it is possible to talk about the minimization problem without restrictions. Indeed, in this case it is necessary to find a point such $\vec{x}^{(0)}$ that the inequality $f(\vec{x}^{(0)}) \le f(\vec{x})$ holds for all points in space R^n without restriction. Often the minimization problem without restriction is also called the unconditional minimization problem. Moreover, to characterize the minimum point add the adjective «unconditional».

If $X \ne R^n$, then there is a minimization problem with constraint. In this case, they also talk about the problem of conditional minimization and the conditional minimum.

Appropriate terminology is also introduced for maximization problems.

If the feasible set X is given in the form:

$$X = \left\{\vec{x} \in R^n \middle| g_j(\vec{x}) \le 0,\ j = 1, 2, \ldots, k;\ g_j(\vec{x}) = 0,\ j = k+1, \ldots, m\right\}, \tag{1.28}$$

where all the numerical functions $g_j(\vec{x})$ are defined on R^n, then it is possible to say about the problem of mathematical programming. Among problems of this class, problems with restrictions of the inequality type are distinguished when the set X has the form (1.28) and $m = k$, as well as problems with restrictions of the equality type when there are no inequalities in (1.28), that is $k = 0$. There are problems with mixed constraints – when irregularities and equalities occur in set X.

1.7 Classification of types of optimization problems

Depending on a given principle of optimality, the solutions to problems of choosing optimal solutions can be performed both on a set of solutions and in the space of estimates of criteria for optimal solutions. In the first case, the corresponding ordinal approach to the selection of optimal solutions is used, based on the introduction of binary relations on a set of solutions. In the second case, a cardinalistic approach is used in the space of evaluations of the criterion for the effectiveness of decisions, when each alternative solution is evaluated by the numerical value of the objective function – $f(\vec{x})$ criterion for the effectiveness of the solution.

Depending on the number of objective functions by which the effectiveness of solutions is estimated and which are optimized, scalar optimization problems $(f(\vec{x}) \Rightarrow extr)$ and multicriteria (vector) optimization problems $(f(\vec{x}) = (f_1(\vec{x}),\ldots,f_i(\vec{x}),\ldots,f_m(\vec{x})) \Rightarrow extr)$ are considered.

Depending on the type of objective functions, linear optimization problems are considered when $f(\vec{x})$ is a linear function, and nonlinear optimization problems when $f(\vec{x})$ is a nonlinear function. These optimization problems, respectively, are also called linear and nonlinear programming problems.

Depending on the presence of restrictions under which optimization is carried out, problems of unconditional optimization and conditional optimization are distinguished. In unconditional optimization problems, optimal solutions are selected from a variety of solutions X, it coincides with Euclidean space $X = R^n$. In conditional optimization problems, optimal solutions are selected from a subset of feasible solutions $D \subset X$, determined by some solution constraints.

Depending on the number of parameters on which the objective function depends, one-parameter (scalar) optimization problems are distinguished when the objective function depends on one variable) and multidimensional optimization when the objective function depends on many variables.

Depending on the methods for finding the extremum of the objective functions, optimization problems are distinguished, in which classical (analytical) optimization methods and numerical optimization methods are used.

The following sections discuss the features of the statements and methods for solving the main types of optimization problems.

2 CLASSIC OPTIMIZATION PROBLEMS OF OBJECTIVE FUNCTIONS

When designing communication networks, it becomes necessary to solve various types of optimization problems, such as finding the minimum spanning tree, finding the shortest path, finding the critical path, optimal distribution of flows, minimizing the cost of the flow in a network with limited bandwidth, minimizing the average delay time of messages in the network for given restrictions on the cost of the network and others. This determines the need to use various optimization methods. Let's consider some classic problems and optimization methods that are widely used in the design of optimal communication networks.

In preparing the materials in this section works [4, 6, 8, 20, 24, 36, 39, 42, 44, 50] are used, which can be addressed in-depth study of the considered optimization methods.

2.1 Problems of unconditional optimization of objective functions

Classical problems of optimizing objective functions are solved using analytical methods for finding extrema (maximums or minimums) of objective functions. The conditions for the existence of internal extrema are as follows: if a derivative $f'(x)$ exists, then a function $f(x)$ can have an internal maximum or minimum at a point a only when:

$$f'(a) = 0.$$

This is a necessary condition for the existence of an extremum.

If there is a second derivative $f''(a)$, then the function $f(x)$ has at a:

$$\begin{aligned} &\text{maximum at } f'(a) = 0 \text{ and } f''(a) < 0, \\ &\text{minimum at } f'(a) = 0 \text{ and } f''(a) > 0. \end{aligned} \qquad (2.1, a)$$

A more general statement: if there is a derivative $f^{(n)}(a)$ and if $f'(a) = = f''(a) = \ldots = f^{(n-1)}(a) = 0$, then the function $f(x)$ has at the point:

$$\text{maximum at even } n \text{ and } f^{(n)}(a) < 0,$$
$$\text{maximum at even } n \text{ and } f^{(n)}(a) > 0. \qquad (2.1, b)$$

If n is odd, then the function $f(x)$ at the point a has neither a minimum nor a maximum, but has an inflection point $x = a$.

Conditions (2.1, *a*, *b*) are sufficient conditions for the existence of an extremum.

If $f'(a) = 0$, then in all cases they say that the function $f(x)$ at $x = a$ has a stationary value, and the point $x = a$ is called stationary.

In the general case, functions of several variables, the necessary optimality conditions are as follows: if a function $f(x_1, x_2, ..., x_n)$ is differentiated at a point $(a_1, a_2, ..., a_n)$, then it can have an internal maximum or minimum only at that point when its first differential df takes on zero at this point, that is, when:

$$\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = ... = \frac{\partial f}{\partial x_n} = 0. \qquad (2.2)$$

Sufficient conditions for optimality: if the function $f(\vec{x})$ has continuous second partial derivative at the point $\vec{a}$ and if the necessary conditions (2.2) are satisfied at this point, when the second differential:

$$d^2 f = \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_k}\Big|_{(a_1, a_2, ..., a_n)} \Delta x_i \Delta x_k = \sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik} \Delta x_i \Delta x_k \qquad (2.3)$$

is a negative or positive definite quadratic form.

The function $f(\vec{x})$ has a maximum at a point $\vec{a}$, if $d^2 f(\vec{x})$ is a negatively defined quadratic form. When $d^2 f(\vec{x})$ is a positive definite quadratic form, then the function $f(\vec{x})$ has a minimum point $\vec{a}$. If the quadratic form is not defined, there is no extremum at the point a.

In turn, the quadratic form is positive definite if all eigenvalues of the matrix $A = \|a_{ik}\|$ are positive, and negatively defined if all eigenvalues are negative. If part of the eigenvalues is positive and the other part is negative, the quadratic form is not defined. The eigenvalues are the roots of the algebraic equation:

$$\det(A - \lambda I) = \begin{Vmatrix} a_{11} - \lambda & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} - \lambda & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} - \lambda \end{Vmatrix} = 0. \qquad (2.4)$$

2.2 Problems of conditional optimization by the method of Lagrange multipliers

The problem of conditional optimization is formulated as follows:

– it is necessary to find the extremum of the function:

$$f(\vec{x}) = f(x_1, x_2, ..., x_n) \Rightarrow extr, \tag{2.5}$$

– for fulfilling the condition:

$$\varphi_i(\vec{x}) = \varphi_i(x_1, x_2, ..., x_n) = 0, \ i = \overline{1, m}, \ m < n. \tag{2.6}$$

The necessary conditions for local optimality for this problem are known as a rule of Lagrange multipliers and are formulated with respect to the Lagrange function:

$$\Phi(\vec{x}, \vec{\psi}) = f(\vec{x}) + \sum_{i=1}^{m} \psi_i \varphi_i(x) \tag{2.7}$$

in the following form: if $\vec{x}^*$ is a local solution to problem (2.5), (2.6), then there exists a vector $\vec{\psi}^* = (\psi_1^*, \psi_2^*, ..., \psi_B^*)$ such that:

$$\Phi'(\vec{x}^*, \vec{\psi}^*) = 0, \tag{2.8}$$

where ψ_i are the Lagrange multipliers;

$$\Phi'(\vec{x}, \vec{\psi}) = f'(\vec{x}) + \sum_{i=1}^{m} \psi_i \varphi'$$

is the vector of derivatives of the Lagrange function with respect to the components of the vector $\vec{x}$.

Any point $\vec{x}^*$ satisfying under certain $\vec{\psi}^*$ conditions (2.8), as well as conditions (2.6), is called a stationary point of the problem (2.5), (2.6). There are also sufficient optimality conditions involving second derivatives.

2.3 Linear programming problems

The problem of linear programming in an arbitrary form of writing is called the optimization problem in which it is necessary to minimize the objective function of a linear form:

$$f(\vec{x}) = \sum_{i=1}^{n} c_i x_i \to \min \tag{2.9}$$

for fulfilling the conditions:

$$\sum_{j=1}^{n} a_{ij} x_i \le b_j, \quad j = \overline{1,m}, \tag{2.10, a}$$

or

$$\sum_{i=1}^{n} a_{ij} x_i = b_j, \quad j = \overline{(m+1),p}, \tag{2.10, b}$$

and

$$x_i > 0, \quad i = \overline{1,n}. \tag{2.10, c}$$

This problem in matrix form is written as:

$$\left.\begin{array}{l} \vec{c}^{\,tr}\vec{x} \to \min, \\ A\vec{x} \le \vec{\vec{b}}, \\ \vec{x} \ge \vec{0} \end{array}\right\} \tag{2.11}$$

and is called the *symmetric notation form* of the linear programming problem.

Linear programming problem of the form:

$$\left.\begin{array}{l} \vec{c}^{\,tr}\vec{x} \to \min, \\ A\vec{x} = \vec{\vec{b}}, \\ \vec{x} \ge \vec{0} \end{array}\right\} \tag{2.12}$$

is called the *canonical notation form* of a linear programming problem.

If the restriction system is specified in the form:

$$A\vec{x} \le \vec{b},$$

then it can be brought into canonical form by introducing additional variables.

Let's consider an equality $A\vec{x} = \vec{b}$ constraint problem. This system of constraints can be represented as a system of equations

$$\left.\begin{array}{l} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n + x_{n+1} = b_1, \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n + x_{n+2} = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m,1}x_1 + a_{m,2}x_2 + \ldots + a_{m,n}x_n + x_{n+m} = b_m. \end{array}\right\} \tag{2.13}$$

Let's introduce the following notation:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad \vec{A}_1 = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \dots \\ a_{m,1} \end{bmatrix}, \dots, \quad \vec{A}_n = \begin{bmatrix} a_{1,n} \\ a_{2,n} \\ \dots \\ a_{m,n} \end{bmatrix},$$

$$\vec{A}_{n+1} = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}, \dots, \quad \vec{A}_{n+m} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix}, \quad \vec{A}_0 = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}. \qquad (2.14, a)$$

Then the linear programming problem can be written as:

$$\sum_{i=1}^{n} c_i x_i \to \min, \qquad (2.14, b)$$

$$x_1\vec{A}_1 + x_2\vec{A}_2 + \dots + x_n\vec{A}_n + x_{n+1}\vec{A}_{n+1} + \dots + x_{n+m}\vec{A}_{n+m} = \vec{A}_0, \quad \vec{x} \ge \vec{0}.$$

Vectors $\vec{A}_i$ are called *condition vectors*, and the linear programming problem itself is called extended relative to the original. Let D and D_1 be admissible sets of solutions to the original and extended problems, respectively.

Then any point of the set D_1 corresponds to a single point of the set D and vice versa. In the general case, an admissible set D of the initial problem is a projection of the set D_1 of the extended problem onto the subspace of output variables.

A set of numbers $\vec{x} = (x_1, x_2, \dots, x_n)$ that satisfies the constraints of a linear programming problem is called its plan. The solution to the linear programming problem is called its plan, minimizes the linear form.

Let's introduce the concept of a basic solution. From the matrix of the extended problem $A_p = \left[\vec{A}_1, \vec{A}_2, \dots, \vec{A}_{n+m}\right]$, let's choose m linearly independent column vectors, which denote as the matrix $B_{m\times m}$, and as $D_{m\times n}$ denote the matrix from the remaining columns. Then $A_p = [B, D]$ and the limitations of the extended linear programming problem can be written as:

$$A_p\vec{x} = B\vec{x}_B + D\vec{x}_D = A_0. \qquad (2.15)$$

Obviously, the columns of the matrix B form a basis of m-dimensional space. Therefore, a vector A_0 and any matrix column D can be represented as a linear combination of matrix columns D.

Multiplying (2.15) on left B^{-1}, let's find $\vec{x}_B$ at $\vec{x}_D = \bar{0}$:

$$\vec{x}_B = B^{-1}\vec{A}_0. \tag{2.16}$$

The solution (2.16) is called the *basic solution* of a system of m equations with $m+n$ unknowns. If the resulting solution contains only positive components, then it is called *basic admissible.*

A feature of admissible basic solutions is that they are the extreme points of an admissible set D_1 of an extended problem.

If among the components $\vec{x}_B$ are not zero, then the basic admissible solution is called *non-degenerate*.

The plan $\vec{x}$ of the linear programming problem will be called *support* if the condition vectors $\vec{A}_i$ with positive coefficients are linearly independent.

A non-degenerate support plan is formed by the intersection n of hyperplanes forming an admissible region. In the case of degeneration at a corner point of a polyhedron of solutions, more than n hyperplanes intersect.

The main theorem of linear programming:

1. The linear form $z = \vec{c}^{tr}\vec{x}$ reaches its minimum at the corner point of the polyhedron of solutions (Fig. 2.1).
2. If it makes a minimal decision at more than one corner point, then it reaches the same value at any point that is a convex combination of these corner points.

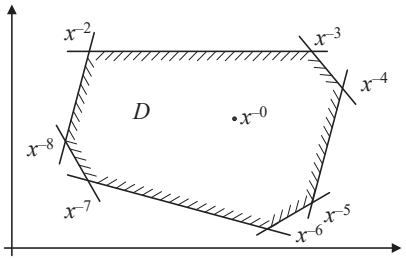

Fig. 2.1 Polyhedron of solutions

Each corner point of the polyhedron of solutions corresponds to m linearly independent vectors from the given system $\vec{A}_1,...,\vec{A}_n$.

To solve the general problem of linear programming, the simplex method (or the method of successive improvement of the plan) is used.The method assumes that the following problem is being solved:

$$Q(\vec{x}) = c_1x_1 + c_2x_2 + ... + c_nx_n \to \min, \tag{2.17}$$

with a restriction system of the following form:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + ... + a_{1,n}x_n = b_1, \\ ....., \\ a_{m,1}x_1 + a_{m,2}x_2 + ... + a_{m,n}x_n = b_m. \end{cases} \quad (2.18)$$

Let's solve this system with respect to variables $x_1,...,x_m$:

$$\begin{cases} x_1 = a'_{1,m+1}x_{m+1} + ... + a'_{1,n}x_n + b'_1, \\ ..... \\ x_m = a'_{m,m+1}x_{m+1} + ... + a'_{m,n}x_n + b'_m. \end{cases} \quad (2.19)$$

The condition vectors corresponding $x_1,...,x_m$ form a basis. Variables $x_1,...,x_m$ are called basic variables. Other problem variables are non-basic.

The formalized simplex method algorithm consists of two main steps:

– building a reference plan;
– building an optimal plan.

Construction of an optimal plan. In order for the reference plan to be optimal, while minimizing the objective function, it is necessary that the coefficients in the row of the objective function are non-positive (in the case of maximization, non-negative). That is, when searching for a minimum, it is necessary to get rid of the positive coefficients in the row $Q(\vec{x})$.

The choice of the decisive element. If the coefficients are greater than zero when searching for the minimum in the row of the objective function, then the column with the opposite coefficient in the row of the objective function is selected as the decisive one. To select a decisive line (decisive element) among the positive coefficients of the decisive column, one (row) is selected for which the ratio of the coefficient in the column of free terms to the coefficient in the decisive column is minimal:

$$\frac{b_r}{a_{r,l}} = \min\left\{\frac{b_i}{a_{i,l}} \middle| a_{i,l} \geq 0\right\}, \quad (2.19)$$

where $a_{r,l}$ is the untying (directing) element; r is the decisive line.

To move to the next simplex table (to the next supporting plan with a lower value of the objective function), a step of a modified Jordan exception is made from the decisive element a_{rl}. The step of a modified Jordan exception over a simplex table includes the following procedures:

1. 1 is put in place of the decisive element and is divided by the decoupling element.
2. The last elements of the decisive column change the sign to the opposite and are divided by the decisive element.

3. The last elements of the decisive line are divided by the decisive element.
4. All other elements of the simplex table are calculated by the following formula:

$$a_{i,j} = \frac{a_{ij} \cdot a_{rl} - a_{rj} \cdot a_{il}}{a_{rl}} = a_{ij} - \frac{a_{rj} \cdot a_{il}}{a_{rl}}. \quad (2.20)$$

When using the simplex method, it is assumed that the linear programming problem is non-degenerate, that is, each support plan m consists of exactly positive components, where m is the number of constraints in the problem. In a non-degenerate reference plan, the number of positive components is less than the number of restrictions: some basic variables corresponding to this reference plan take zero values. In a degenerate problem, more than two lines intersect at one vertex of the polyhedron of conditions.

If the linear programming problem turns out to be degenerate, then with a poor choice of the vector of conditions, infinite movement along the bases of the same support plan may occur, the so-called looping phenomenon.

If the total number of variables of linear programming problems $n = 2$ or it can be reduced to the corresponding problem with the number of independent variable $k = 2$ s, then such a problem can be easily solved graphically.

2.4 Transport problems of linear programming

In the general case, the transport linear programming problem is formulated as follows. Let's minimize transportation costs:

$$Q(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min \quad (2.21)$$

for restrictions:

$$\left.\begin{aligned} &\sum_{i=1}^{m} x_{ij} = b_j, \ j = \overline{1,n}, \\ &\sum_{j=1}^{n} x_{ij} = a_i, \ i = \overline{1,m}, \\ &x_{ij} \geq 0, \ i = \overline{1,m}, \ j = \overline{1,n}, \end{aligned}\right\} \quad (2.22)$$

where c_{ij} is the cost of transporting a unit of product from point i to point j; x_{ij} is the planned amount of traffic from point i to point j (transportation plan X is matrix $m \times n$); b_j is product requirements in point j; a_i are stocks of products in point i.

This is a general mathematical model of the transport problem of linear programming in communication networks is specified in terms of the cost of transmitting information from m information sources to n information consumers.

It is envisaged that there is a *closed* type model, that is

$$\sum_{j=1}^{n} b_j = \sum_{i=1}^{m} a_i.$$

If the model is open type

$$\left(\sum_{j=1}^{n} b_j \neq \sum_{i=1}^{m} a_i \right),$$

then it can always be reduced to closed type by introducing a fictitious production point or a fictitious consumption point:

– if $\sum_{j=1}^{n} b_j < \sum_{i=1}^{m} a_i$, then $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$, so $\sum_{j=1}^{n+1} b_j = \sum_{i=1}^{m} a_i$, and $c_{i,n+1} = 0, \forall i$;

– if $\sum_{j=1}^{n} b_j > \sum_{i=1}^{m} a_i$, then $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$, $\sum_{j=1}^{n} b_j = \sum_{i=1}^{m+1} a_i$ and $c_{m+1,j} = 0, \forall j$.

The transport problem is a linear programming problem and, of course, it can be solved using the method of sequential improvement of the plan or the method of sequential refinement of estimates. In this case, the main difficulties are associated with the number of problem variables $m \times n$ and the number of constraints $m + n$.

Therefore, special algorithms are more efficient. Such algorithms include the *potential algorithm* and the *Hungarian algorithm*.

The potential algorithm (also called the modified distribution method) begins with a certain basic plan of the transportation problem (an acceptable transportation plan). To construct a support plan, one of three methods is usually used: *the northwest corner method*, *the minimum element method*, or *the Vogel method*.

An acceptable reference plan for a transportation problem is called *non-degenerate* if the number of filled cells in the transportation table, that is, the number of positive transportations $x_{ij} > 0$, is equal to $m + n + 1$, where m is the number of departure points and n is the number of destinations.

If an admissible reference plan $m + n + 1$ contains fewer $x_{ij} > 0$ elements, then it is called *degenerate*, and the transportation problem is called a *degenerate transportation problem*.

2.5 Problems of dynamic programming

Dynamic programming problems are computational methods for solving optimization problems of the following form:

$$z = \max_{\bar{x}} \sum_{i=1}^{n} f(x_i) \tag{2.23}$$

when fulfilling restrictions:

$$\sum_{i=1}^{n} a_i x_i \le b, \quad a_i > 0, \quad x_i \ge 0. \tag{2.24}$$

If all functions $f_i(x_i)$, $i = \overline{1,n}$, are convex, then the Lagrange multiplier method can be used for the solution. However, if there are many local maxima, then such a method gives only one of these solutions. If it is necessary to find a global maximum, then you need to take all the local maxima. In this case, the application of the Lagrange multiplier method is problematic.

Let's consider a method that provides a solution to problem (2.23)–(2.24) for the case when all $\{a_i\}$, $i = \overline{1,n}$, and b are integers. It is also assumed that all variables $\{x_i\}$, $i = \overline{1,n}$, in the problem can take only integer values.

Let's introduce the following notation. Let's denote by z^* the absolute maximum z at provided condition $\sum_{i=1}^{n} a_i x_i \le b$. Let's select the value x_n and, fixing it, maximize z in all other variables $x_1, x_2, \ldots, x_{n-1}$. Let's suppose that such maximization is carried out for all possible values x_n. Then z^* will be the largest of all possible values z. Formally, this process can be written as follows:

$$\max_{x_1, x_2, \ldots, x_{n-1}} \left\{ \sum_{i=1}^{n} f_i(x_i) \right\} = f_n(x_n) + \max_{x_1, x_2, \ldots, x_{n-1}} \left\{ \sum_{i=1}^{n-1} f_i(x_i) \right\}, \tag{2.25}$$

so

$$\sum_{i=1}^{n-1} a_i x_i \le b - a_n x_n.$$

Since

$$\max \sum_{i=1}^{n-1} f_i(x_i)$$

for non-negative integers satisfying the condition

$$\sum_{i=1}^{n-1} a_i x_i \le b - a_n x_n$$

depends on $b - a_n x_n$, let's denote:

$$\max_{x_1, x_2, \ldots, x_{n-1}} \sum_{i=1}^{n-1} f_i(x_i) = \Lambda_{n-1}(b - a_n x_n). \tag{2.26}$$

Let's assume that $\Lambda_{n-1}(b - a_n x_n)$ is calculated for all valid integer values:

$$x_n = \left\{0, 1, \ldots, \left[\frac{b}{a_n}\right]\right\}.$$

Then it is obvious that:

$$z^* = \max_{x_n \geq 0} f_n(x_n) + \Lambda_{n-1}(b - a_n x_n). \tag{2.27}$$

To determine (2.27), let's find the values $f_n(x_n) + \Lambda_{n-1}(b - a_n x_n)$ for all admissible values x_n and choose the maximum among them. This corresponds to the maximum x_n^*. If the function were known $\Lambda_{n-1}(b - a_n x_n)$, then the whole problem would be reduced to a problem with one variable.

The solution to the dynamic programming problem is a directed sequential enumeration of options, which necessarily leads to a global maximum.

The considered dynamic programming problem (2.23-2.24) can be interpreted as the distribution of raw materials (information) with one limited source of raw materials (information):

$$\sum_{i=1}^{n} a_i x_i \leq b,$$

where x_i is the amount of raw materials (information) used in the *i*-th method of production (transmission of information). Then $f_i(x_i)$ is income from processing (transmitting information) by the *i*-th way of x_i units of raw materials (information).

3 NUMERICAL METHODS OF SCALAR OPTIMIZATION

In some cases, when analytical methods for optimizing objective functions can't be used, numerical methods for optimizing functions implemented on a computer are used. A universal numerical method with which it would be possible to successfully solve all optimization problems does not exist. Therefore, to solve each specific type of optimization problem, its own numerical method is used. This section discusses some numerical methods for solving scalar nonlinear objective function optimization problems.

Works [2, 4, 26, 36, 41] are used in preparing the materials, which can be addressed in the course of an in-depth study of these issues.

3.1 The classic method of minimizing the function by one variable

Let's consider a class of functions that, from a computational point of view, have an important property, namely, unimodality. A function f is called unimodal on a segment $[a,b]$ if it has a single point of global minimum $x_{\min}$ on this segment and to the left of this point is such that it strictly arrives, and the matter strictly increases. In other words, the function f is unimodal if the point $x_{\min}$ exists and is unique. Moreover, for any two points $x_1, x_2 \in [a,b]$ such that for $x_1 < x_2$, it always follows $f(x_1) < f(x_2)$ from inequality $x_1 > x_{\min}$ that, and inequality $x_2 < x_{\min}$ follows from inequality $f(x_1) > f(x_2)$. The considered property of a unimodal function is illustrated in Fig. 3.1.

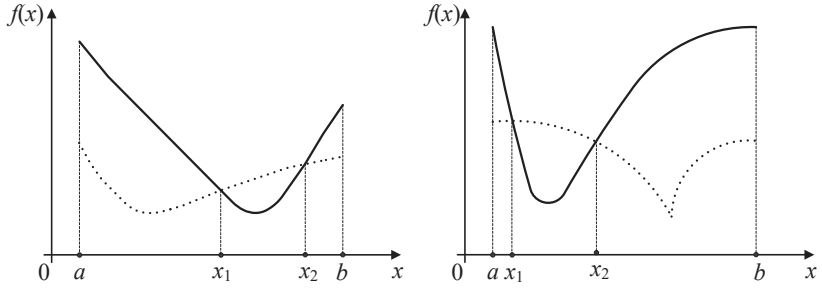

Fig. 3.1. Unimodal function

In the classical method of finding the minimum of a function f on a segment $[a, b]$, it is assumed that a continuous function has a continuous derivative on the entire segment $[a, b]$, except for a finite number of points. According to the classical method, the derivative $f'(x)$ is calculated and critical points are determined, that is, such internal points of the segment $[a,b]$ at which the derivative takes the value zero or not. Further, the sign of the derivative is examined at each critical point and those points are selected from them, when passing through which the derivative changes sign from – to + (these are local minimum points). Finally, at each of the selected points, including the ends of the segment $[a, b]$, the value of the objective function is calculated. Comparing the found values, determine the minimum. The point corresponding to this minimum value of the objective function is the point of the global minimum of the function f on the segment $[a, b]$.

If the function f has a continuous derivative at each internal point of the segment $[a, b]$, then the procedure is simplified: find the points at which the derivative is zero by finding the roots of the equation $f'(x) = 0$. Having calculated the values of the function at the found points (including the ends of the segment), a global minimum point is selected.

The main disadvantage of this classical method is the narrow scope of its applicability. So, if the numerical values of the objective function are determined from observations or as a result of experiments, it is difficult to obtain an analytical expression for its derivative. But even if a derivative is found, then finding the root of the equation $f'(x) = 0$ can be a complicated computational task, for which it can take a lot of time to decouple.

Let's consider numerical optimization methods, the use of which does not require knowledge of derivatives and in which, in addition, the amount of computation of the values of the objective function in a certain sense is the smallest.

3.2 Golden section method

Let's search for the global minimum point of a unimodal function f on a segment $[a,b]$ so that the number of calculations of the values of this function necessary to ensure a given accuracy is as small as possible.

Let's consider the point x_1 on the segment $[a,b]$ and calculate the value $f(x_1)$. Knowing the value of the objective function at one point, it is impossible to narrow the search area of the point $x_{\min}$. Therefore, let's choose the second point x_2, so $a < x_1 < x_2 < b$ to calculate $f(x_2)$. One of two cases is possible: $f(x_1) \le f(x_2)$ either $f(x_1) \ge f(x_2)$. According to the property of unimodal function, in the first case, the desired point $x_{\min}$ can't be on the segment $[x_2, b]$, and in the second, on the segment $[a, x_1]$. In Fig. 3.2 these segments are marked by hatching. So, now

the search area is narrowing: the next search point x_3 should be taken in one of the shortened segments $[a,x_2]$ or $[x_1,b]$. Since at first nothing is known about the position of the point x_1, then both of the above cases are equivalent, that is, any of the segments $[x_2,b]$ and $[a,x_1]$ can be «redundant» and it follows that the points x_1 and x_2 must be located symmetrically relative to the segment $[a,b]$. Further, in order to narrow the search zone as much as possible, these points should be «closer» to the middle of the output segment. However, they should not be taken very close, since it is necessary to build an algorithm for the implementation of which a minimum number of calculations of the function values is necessary. This occurs when, at the second stage, the narrowing of the search area will need to calculate only one value, which should be compared with the existing value $f(x_1)$ or $f(x_2)$ depending on which of the two cases was implemented. Therefore, if to take the points x_1 and x_2 near the middle of the output segment, then at the second stage the narrowing of the search zone will be insignificant. Thus, on the one hand, the points x_1 and x_2 should be chosen near the middle of the segment, and on the other, they can't be taken very close.

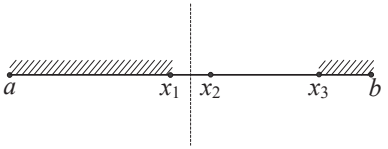

Fig. 3.2. Search for the global minimum point of a unimodal function *f* on a segment [*a*,*b*]

First, for simplicity, let's consider a segment $[0,1]$ of unit length instead $[a,b]$. In order for the point x to be «profitable» both at this and at the next stage, it must divide this segment in the same respect as:

$$\frac{x}{1}=\frac{1-2x}{1-x}. \tag{3.1}$$

The equation has one root equal $(3-\sqrt{5})/2 \approx 0.382$. About a point *x* located at a distance of $(3-\sqrt{5})/2$ % length from one of the ends of the segment $[0,1]$, they say that it implements the *golden section of the segment* $[0,1]$. Obviously, each segment has two such points located symmetrically with respect to the middle.

In the general case, the lengths of a segment of a point x_1 and x_2 must implement the golden section of the initial segment $[a,b]$. Rejecting the part $[a_1,b_1]$ in which $x_{\min}$ obviously can't be, let's apply similar reasoning to the shortened segment with the only difference that there is one internal point, it implements the golden section.

Let's give an exact description of the «golden section» method. For convenience, let's introduce the following notation: $a_0 = a$, $b_0 = b$:

Step 0. Calculate the coordinates of the points implementing the golden section of the output segment:

$$y_0 = a_0 + \frac{3-\sqrt{5}}{2}(b_0 - a_0), \quad z_0 = a_0 + b_0 - y_0.$$

In addition, let's calculate the value of $f(y_0)$ and $f(z_0)$.

Step 1. As a result of the previous step, the quantities y_{k-1}, z_{k-1}, a_{k-1}, b_{k-1}, $f(y_{k-1})$, $f(z_{k-1})$ are known. Compare the value of $f(y_{k-1})$ and $f(z_{k-1})$. If $f(y_{k-1}) \le f(z_{k-1})$, then let's consider $a_k = b_{k-1}$, $b_k = z_{k-1}$, $z_k = y_{k-1}$, and using these numbers, let's calculate the coordinates of the symmetric point (to the left of the one that is) $y_k = a_k + b_k - z_k$ and the section $f(y_k)$.

If the opposite inequality $f(y_{k-1}) > f(z_{k-1})$ holds, then it should accept $a_k = y_{k-1}$, $b_k = b_{k-1}$, $y_k = z_{k-1}$ and calculate the coordinate of the symmetric point (to the right of the one that is) $z_k = a_k + b_k - y_k$ along with the value $f(z_k)$.

Next, it is necessary to calculate the length of the next $(k+1)$ segment, that is, the value $\Delta_{k+1} = b_k - y_k = z_k - a_k$.

The steps of the algorithm are carried out until an inequality $\Delta_{k+1} = \varepsilon$ is obtained, where $\varepsilon > 0$ is the specified accuracy of the calculations. Choose the smallest of the numbers $f(y_k)$ and $f(z_k)$, which will be the approximate value of the minimum of the objective function. And the point corresponding to it gives an approximate value to the sought $x_{\min}$. In this case, the deviation of the approximate minimum point from the actual minimum point $x_{\min}$ does not exceed the specified calculation accuracy ε.

Let's establish that the segments $[a_k, b_k]$, $k = 1,2,\ldots$, constructed using the golden section method are indeed charged to a point $x_{\min}$. At the k-th step, the value of the length of the segment $[a_k, b_k]$ is:

$$b_k - a_k = \frac{\sqrt{5}-1}{2}(b_{k-1} - a_{k-1}), \quad k = 1,2,\ldots. \tag{3.2}$$

So,

$$\Delta_k = b_k - a_k = \left(\frac{\sqrt{5}-1}{2}\right)^k (b_0 - a_0) < 0.7^k (b_0 - a_0)_{k\to\infty} \to 0. \tag{3.3}$$

So, the segment $[a_k, b_k]$ on which the point is located $x_{\min}$, with unlimited magnification k, is contracted to a point that belongs to all segments

simultaneously $[a_k, b_k]$, k=1,2,… . The function $f(x)$ is unimodal, so only a point can be such a point $x_{\min}$.

Let's note that in the golden section method at the zero stage two values of the objective function are calculated, and at each subsequent stage only one.

3.3 Fibonacci method

In practice, the number of calculations of the objective function is often limited to a certain number n. Thus, the number of calculation steps by the golden section method is also limited, not exceeding $n-1$. The Fibonacci method differs from the golden section method only in the choice of the first two symmetric points and guarantees more accurate approximations to the point $x_{\min}$ in $(n-1)$ steps.

According to the Fibonacci method, at the zero step, the coordinates of the first two symmetric points are calculated by the formulas:

$$y_0 = a_0 + \frac{F_n}{F_{n+2}}(b_0 - a_0), \quad z_0 = a_0 + b_0 - y_0, \tag{3.4}$$

where F_{n+2} denotes the $(n+2)$ Fibonacci number, which is determined by the recurrence relation:

$$F_{n+2} = F_n + F_{n+1}, \quad n = 1,2,3,\ldots; \quad F_1 = F_2 = 1. \tag{3.5}$$

The first ten Fibonacci numbers matter:

$$F_1 = F_2 = 1, \quad F_3 = 2, \quad F_4 = 3, \quad F_5 = 5, \quad F_6 = 8, \quad F_7 = 13,$$

$$F_8 = 21, \quad F_9 = 34, \quad F_{10} = 55.$$

Further calculations using the Fibonacci method coincide with the corresponding steps of the golden section method. The difference is that finding the values Δ_k becomes superfluous, since for $k = n-1$ the calculation process they finish and y_{n-1} take for an approximate value $x_{\min}$.

Let's consider in more detail the situation arising at $k = n-1$. Using mathematical induction and the definition of Fibonacci numbers, it is possible to prove that:

$$\Delta_k = b_k - a_k = \frac{F_{n-k+2}}{F_{n+2}}(b_0 - a_0), \quad k = 1,2,\ldots,n-1. \tag{3.6}$$

Using these formulas for symmetric points y_k and z_k write the following expressions:

$$y_k = a_k - \Delta_{k+2} = a_k + \frac{F_{n-k}}{F_{n+2}}(b_0 - a_0), \tag{3.7}$$

$$z_k = a_k - \Delta_{k+1} = a_k + \frac{F_{n-k+1}}{F_{n+2}}(b_0 - a_0). \tag{3.8}$$

This shows that for $k=n-1$ two symmetric points merge into one (since $y_{n-1} = z_{n-1}$) and divide the segment $[a_{n-1}, b_{n-1}]$ into two equal parts. According to formula (3.6), the length of this segment is equal:

$$\Delta_k = \frac{2}{F_{n+2}}(b_0 - a_0).$$

So, taking y_{n-1} asapproximate value of the minimum, there is such an estimate of the deviation of this value from the true value $x_{\min}$:

$$|y_{n-1} - x_{\min}| \le \frac{b_0 - a_0}{F_{n+2}}. \tag{3.9}$$

Thus, the error of calculations by the Fibonacci method for a fixed n does not exceed the value of the right-hand side of inequality (3.9).

In practice, it can be specified not by the number of calculations n of the values of the objective function, but some calculation error $\varepsilon > 0$. In this case, in order to determine the number n necessary to ensure a given accuracy, it is possible to use inequality (3.9).

Indeed, if $(b_0 - a_0)/F_{n+2} \le \varepsilon$, then the required accuracy will be achieved. Hence the conditions for determining n:

$$F_{n+1} < \frac{b_0 - a_0}{\varepsilon} \le F_{n+2}. \tag{3.10}$$

It can be proved that $\lim\limits_{n\to\infty} F_n / F_{n+2} = (3 - \sqrt{5})/2$, therefore, with significant n calculations by the Fibonacci method and the golden ratio method, they begin with almost the same pair of symmetrical points.

In substantiating the golden section method and the Fibonacci method, an important role is played by the property of unimodality of the function, it is minimized. If the objective function is not unimodal, then the numerical implementation of both methods will lead, generally speaking, only to the neighborhood

of the local minimum point. Moreover, the value of the local minimum may turn out to be very far from the value of the global minimum.

3.4 Uniform brute force method

Ideologically (and from the point of view of computer implementation), the most simple method for finding the global minimum is the method of uniform enumeration. According to this method, the step $h>0$ value is fixed, the value of the objective function f is calculated at points $x_1 = a$ (or at a point close to the right a), and $x_2 = x_1 + h$ the calculated values are compared. Remember the smaller of the two values. Next, calculate the value of the function$f$ with $x_3 = x_2 + h$ and compare it with the value that is stored in memory. Again, remember a lower value. Thus, they sequentially sort the value f in points $x_k = x_{k-1} + h,\ k = 4,5,\ldots,$ untilat some point $k = n$ the next point x_{n+1} leaves the segment $[a,b]$. The value of the objective function that remains in memory after stopping (this is $\min\limits_{i=1,2,\ldots,n} f(x_i)$) is considered an approximate value of the global minimum.

In the practical implementation of such a method, the main problem is to establish the step size h. Even with a relatively small h, there is the possibility of slipping through the global minimum.

In the general case, it is impossible to solve the question of how small should be chosen h so that the value of the approximate minimum differs from the true value $x_{\min}$ by no more than $\varepsilon > 0$:

$$\left| \min_{i=1,2,\ldots,n} f(x_i) - \min_{x\in[a,b]} f(x) \right| \le \varepsilon. \tag{3.11}$$

However, there is a fairly wide class of functions for which this problem can be solved. These are functions f that satisfy the Lipschitz condition with a constant $L \ge 0$:

$$\left| f(x) - f(x') \right| \le L|x - x'| \ \text{ for all } \ x, x' \in [a,b]. \tag{3.12}$$

A function satisfying the Lipschitz condition reaches its smallest value on $[a,b]$. There may be several local minima.

Let the function f satisfy the Lipschitz condition with constant L and $x_1 = a + h/2,\ x_k = x_{k-1} + h,\ k = 2,3,\ldots,n-1,\ x_n = \min\{x_{n-1} + h;\ b\}$, where n is selected so that the condition $h/2 < b - x_{n-1} \le 3h/2$ is satisfied. Then the choice of the step by the formul $h = 2\varepsilon / L$ guarantees the fulfillment of inequality (3.11).

3.5 Numerical methods for minimizing the objective functions of many variables

The general scheme by the descent method. Let's consider the problem of unconditional minimization, that is, the problem of minimizing the objective function $f(\vec{x})$ in the whole space. The essence of all methods for the approximate solution of this problem is to build a sequence of points $\vec{x}^{(0)}, \vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(k)}$ that monotonically reduce the value of the objective function:

$$f\left(\vec{x}^{(0)}\right) \geq f\left(\vec{x}^{(1)}\right) \geq f\left(\vec{x}^{(2)}\right) \geq \ldots \geq f\left(\vec{x}^{(k)}\right) \geq \ldots . \tag{3.13}$$

Such methods are called descent methods. In the course of using these methods, the following scheme is used. Let the point $\vec{x}^{(k)}$ be in the k-th iteration, then the direction of descent $\vec{p}^{(k)} \in R^n$ and the step length along this direction $a_k > 0$ are determined. The next point in the sequence is calculated by the formula:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + a_k \vec{p}^{(k)}, \quad k = 0,1,2,\ldots . \tag{3.14}$$

According to this formula, the magnitude of the advancement from the point $\vec{x}^{(k)}$ to the point $\vec{x}^{(k+1)}$ depends on a_k and $\vec{p}^{(k)}$. Value a_k is traditionally called stride length. Formally, different descent methods differ from each other in the way they select a number a_k and a vector $\vec{p}^{(k)}$. If to determine a_k and $\vec{p}^{(k)}$ it is necessary to calculate only the values of the objective function, the corresponding methods are called zero-order methods or search methods. First-order methods also require the calculation of the first derivatives of the objective function. If the method involves the use of second derivatives, then it is called a second-order method, etc.

However, methods of zero order, as a rule, require significant calculations to achieve a given accuracy, since using only the values of the objective function does not allow to accurately determine the direction to the minimum point.

The most important characteristic of any descent methods is their convergence. As a rule, the type of convergence of the same method depends on the specific type of the objective function, that is, in different tasks the method can converge in different ways. With fairly stringent requirements for the function f, using this method it is possible to build a sequence that converges at the point of global minimum.

Coordinate descent method. According to this method, the descent direction is chosen parallel to the coordinate axes. First, a descent is carried out along the first axis Ox_1, then along the second axis Ox_2 and so on to the last axis Ox_n.

Let's denote the *i*-th unit vector of the space R^n by $\vec{e}(i)$, i. e., the vector for which all coordinates are zero, except for the *i*-th, equal to one. Let $\vec{x}^{(0)}$ is the starting point and a_0 is some positive number. The point $\vec{x}^{(1)}$ is determined as follows. The function $f(\vec{x})$ value is calculated at $\vec{x} = \vec{x}^{(0)} + a_0\vec{e}^{(1)}$ and the inequality is checked:

$$f\left(\vec{x}^{(0)} + a_0\vec{e}^{(1)}\right) < f\left(\vec{x}^{(0)}\right). \tag{3.15}$$

If the inequality is true, then along the axis directionof the value of the function $f(x)$ decreases and therefore it is believed that:

$$\vec{x}^{(1)} = \vec{x}^{(0)} + a_0\vec{e}^{(1)}, \quad a_1 = a_0. \tag{3.16}$$

If (3.15) does not hold, then take a step in the opposite direction and check the inequality:

$$f\left(\vec{x}^{(0)} - a_0\vec{e}^{(1)}\right) < f\left(\vec{x}^{(0)}\right). \tag{3.17}$$

If this inequality holds, let's consider:

$$\vec{x}^{(1)} = \vec{x}^{(0)} - a_0\vec{e}^{(1)}, \quad a_1 = a_0. \tag{3.18}$$

It is possible that both inequalities turn out to be unfulfilled. Then it should be considered:

$$\vec{x}_1 = \vec{x}_0, \quad a_1 = a_0. \tag{3.19}$$

The second step is performed along the coordinate axis Ox_2: if $f(\vec{x}^{(1)} + a_0\vec{e}^{(2)}) < f(\vec{x}^{(1)})$, then consider $\vec{x}^{(2)} = \vec{x}^{(1)} + a_0\vec{e}^{(2)}$, $a_2 = a_1$. If the last inequality does not hold, then the inequality $f(\vec{x}^{(1)} - a_0\vec{e}^{(2)}) < f(\vec{x}^{(1)})$ is checked and, if it is satisfied, it is considered that $\vec{x}^{(2)} = \vec{x}^{(1)} - a_0\vec{e}^{(2)}$, $a_2 = a_1$. If none of the inequalities is satisfied, it is considered that $\vec{x}^{(2)} = \vec{x}^{(1)}$, $a_2 = a_1$. So go through all the n directions of the coordinate axes. This completes the first iteration; at *n*-th step, a certain point $\vec{x}^{(n)}$ is obtained. If at the same time $\vec{x}^{(n)} \neq \vec{x}^{(0)}$, then similarly, starting with another iteration $\vec{x}^{(n)}$. If $\vec{x}^{(n)} = \vec{x}^{(0)}$ (this is the case when at each step no pair of irregularities of the test can be found complete), then the step size should be reduced, taking, for example, $a_{n+1} = a_n/2$ and in the next iteration, use the new step size.

Further iterations are performed similarly. In practice, calculations continue until some condition for the end of the search for the minimum point of the function is fulfilled. Often use the following conditions:

$$\left\|\vec{x}^{(k+1)} - \vec{x}^{(k)}\right\| \le \delta \text{ or } \left|f\left(\vec{x}^{(k+1)}\right) - f\left(\vec{x}^{(k)}\right)\right| \le \varepsilon, \tag{3.20}$$

where δ and ε are some positive numbers characterizing the accuracy of the solution to the minimization problem.

The convergence of the method of coordinate descent is guaranteed if the starting point $x^{(0)}$ is chosen correctly.

This method belongs to the class of methods of zero order and for its implementation it is not required to calculate derivatives. However, there is a requirement for continuous differentiation of a function f.

Sometimes, in an effort to accelerate the convergence of the method, the value a_k is selected so that when moving from $x^{(k)}$ to $x^{(k+1)}$ along the direction of descent, the greatest possible decrease in the objective function is ensured. Generally speaking, such a choice a_k leads to the fact that to achieve a given accuracy, fewer steps are required. However, the implementation of each step will be associated with the solution of the problem of minimizing the function of one variable, which will lead to additional calculations. In addition, finding the exact value a_k in this problem is not always possible. If, instead of the exact value a_k, an approximate one is used in this problem, then the decrease condition $f(x^{(k+1)}) \le f(x^{(k)})$ may be violated.

Gradient methods. A nonzero anti-gradient $\nabla f(\vec{x}^{(0)})$ indicates a direction, a small movement along which $\vec{x}^{(0)}$ does not lead to a value of a function f smaller than $f(\vec{x}^{(0)})$. This remarkable property of the anti-gradient forms the basis of gradient methods, according to which the k-th iteration are considered $\vec{p}^{(k)} = -\nabla f(\vec{x}^{(k)})$, i. e.

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - a_k \nabla f\left(\vec{x}^{(k)}\right), \quad a_k > 0, \quad k = 0, 1, 2, \ldots .$$

These methods differ from each other in the way they select the step size a_k. A sufficiently small step a_k ensures a drop in the objective function:

$$f\left(\vec{x}^{(k+1)}\right) = f\left(\vec{x}^{(k)} - a_k \nabla f\left(\vec{x}^{(k)}\right)\right) < f\left(\vec{x}^{(k)}\right), \tag{3.22}$$

but can lead to a very large number of iterations to achieve the required accuracy. On the other hand, the choice of a significant step size can lead to a violation of inequality (3.22).

Often, it is recommended to choose a value a_k so that there is a strict descent condition:

$$f\left(\vec{x}^{(k)}\right) - f\left(\vec{x}^{(k)} - a_k \nabla f\left(\vec{x}^{(k)}\right)\right) \ge \varepsilon a_k \left\|\nabla f\left(\vec{x}^{(k)}\right)\right\|. \tag{3.23}$$

In the steepest descent method, the value $a_k > 0$ is determined by minimizing the function $\varphi_k(a) = f(\vec{x}^{(k)} - a\nabla f(\vec{x}^{(k)}))$ of one variable a:

$$\varphi_k(a_k) = \min_{a>0} \varphi_k(a). \tag{3.24}$$

Thus, the motion curve in the steepest descent method is a broken line, the neighboring links of which are mutually orthogonal. Moreover, the link connecting $\vec{x}^{(k)}$ to $\vec{x}^{(k+1)}$ lies in the hyperplane tangent to the level surface $f(\vec{x}) = f(\vec{x}^{(k+1)})$.

When implementing gradient methods, in addition to (3.22), a condition of the form are used:

$$\left\|\nabla f\left(\vec{x}^{(k)}\right)\right\| \geq \gamma, \tag{3.25}$$

where $\gamma > 0$ is the fixed accuracy of the calculations.

4 MULTICRITERIA OPTIMIZATION PROBLEMS

Designing optimal communication networks requires taking into account at a strictly formalized level the aggregate of technical and economic quality indicators. In order to select the optimal design solutions, it is necessary to use multicriteria optimization methods, which in essence differ from scalar optimization methods. This section discusses the features of the formulation and methods for solving such problems.

In preparing the materials in this section, the works [1, 11, 18, 20, 27, 32, 36, 38, 39, 42, 44, 51] are used, which can be addressed in the course of an in-depth study of multicriteria optimization methods.

4.1 Formulation of a multicriteria optimization problem

The task of making a decision is choosing among many possible solutions (they are also called options, plans, etc.) such a solution that would be, in a sense, the best (optimal) taking into account the totality of quality indicators.

It is convenient to assume that the decision is made by the DM (decision maker) to achieve a specific goal. Depending on the specific situation, the role of DM can be played by either an individual person (engineer, researcher) or an entire team (a group of specialists engaged in solving one problem).

Each possible decision is characterized by a certain degree of goal achievement. In accordance with this, DM has its own idea of the advantages and disadvantages of decisions, on the basis of which one solution is preferred over another. The optimal solution is a solution that, from the point of view of the decision maker, prevails over other possible solutions. The advantages of solutions in practice are expressed in different forms, and their mathematical formalization can be a difficult task, since DM, as a rule, can't clearly articulate them in mathematical form.

The goal of decision theory is development of the methods that would help DM to most fully and accurately reflect their advantages within the framework of the corresponding mathematical model and, in the end, reasonably choose a truly optimal solution.

In the multicriteria optimization problems considered in this section, let's assume that the set of possible solutions is represented by a vector $\vec{X} \in R^n$. Each solution $\vec{X}$ is evaluated by a set of objective functions $f_1(\vec{x})$, $f_2(\vec{x})$, …, $f_m(\vec{x})$,

defined on the set X. The set of objective functions forms a vector objective function, which will be denoted by $f(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x}))$.

Along with the set of feasible solutions X, let's consider the set of estimates of vector objective functions:

$$Y = f(X) = \{\vec{y} \in R^m \mid \vec{y} = f(\vec{x}), \text{ for some } \vec{x} \in X\}. \quad (4.1)$$

Often, the space R^n in which the set X is contained is called the solution space, and the space R^m in which the estimates $\vec{y}$ are contained is called the *estimation space* or *criteria space*.

Each decision $\vec{x} \in X$ corresponds to one specific estimation $\vec{y} = f(\vec{x}) \in Y$. On the other hand, each estimation corresponds to a solution $\vec{x} \in X$ (there may be more than one) in which $f(\vec{x}) = \vec{y}$. Thus, there is a close connection between the set X and Y therefore the choice of optimal solutions in space X in the indicated sense is equivalent to the choice of the corresponding estimates in the criteria space.

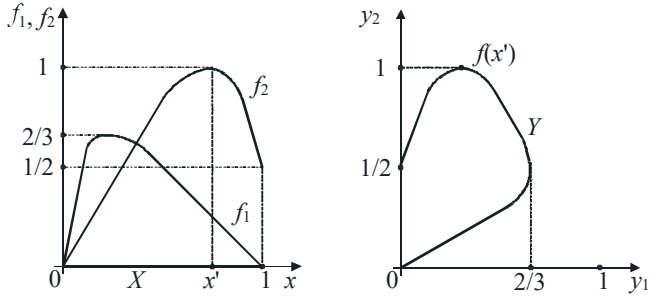

Fig. 4.1 Solution space and criteria space

In the multicriteria optimization problem, certain information about the DM benefits is considered known. This is the idea that it is desirable to maximize (or minimize) objective functions. In this case, additional information on the advantages of one solution over another can also be known and used.

To describe the DM preferences let's use such a mathematical concept as relations.

Determination of relations. Simple examples of relations have already been encountered when $<$, $\leq$, $>$, $\geq$, $=$ signs were used to compare real numbers. The use of these symbols implies the presence of a pair of numbers, one of which is written to the left of the symbol, and the rest is the case. It is said that these numbers are in some relation to each other (the first number is greater than the second, the first number is greater than or equal to the second, etc.).

In certain relations, there can be not only numbers, but also more complex objects, and the relation between them is of a different nature.

Let's give an exact definition of the relation. Let A be some set. Let's create a Cartesian product $A \times A$ – the set of all ordered pairs of elements (a,b), where $a \in A$, $b \in A$ are in relation R. A relation R is a subset of the set $A \times A$, that is $R \subset A \times A$. If there is a relation $(a,b) \in R$, then it is possible to say that the elements a and b are in the relation R. This can also be written as aRb. Notation aRb and bRa does not mean the same thing, except when $a = b$. On the same set A, different relations can be given depending on which pairs (a,b) make up the set R. In particular, a set R may not contain a single pair, contain all possible pairs, that is $R = A \times A$, include only pairs of identical elements (a,a) (equality relation).

Let's consider examples of relations. Let $A = R^n$. Already met the ratio $>$ and $\geq$, given on R^n:

$$\vec{a} \geq \vec{b}, \text{ which means that } a_i \geq b_i, \ i = 1,2,\ldots,n;$$

$$\vec{a} > \vec{b}, \text{ which means that } a_i > b_i, \ i = 1,2,\ldots,n,$$

where $\vec{a} = (a_1,a_2,...,a_n)$, $\vec{b} = (b_1,b_2,...,b_n)$.

One more relation is used on R^n: $\vec{a} \geq \vec{b}$, which takes place if and only if $a_i \geq b_i$, $i = 1,2,\ldots,n$, and at least for one number $i \in \{1,2,\ldots,n\}$ a strict inequality holds $a_i > b_i$.

When $n = 1$ the ratio $\geq$ is the same as $>$ for numbers.

When $n = 2$ the inequality $\vec{a} \geq \vec{b}$ geometrically means that the point a is in the shaded area, which has the shape of a right angle with a punctured vertex b, the sides of which are parallel to the coordinate axes (Fig. 4.2). And the point a for which the inequality $a > b$ holds is the internal point of such an angle.

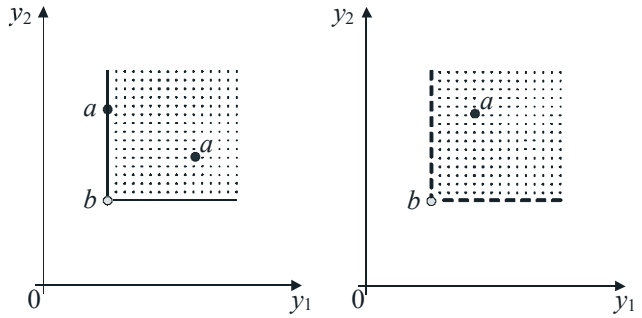

Fig. 4.2 Geometrically means of the inequality $\vec{a} \geq \vec{b}$ when $n = 2$

If A – the set of all lines on a certain plane, but R are pairs of lines, have no common point, then R is nothing but a parallelism relation of lines.

If A – the set of people, then on this set one can expect, for example, the relation «is a relative», etc.

In the set A automotive internal combustion engines of the same volume, you can enter the ratio R: «economical than». Namely: aRb the ratio is correct if and only if the engine a has less fuel consumption than the engine b. Since the relation is a certain set, then in different relations given on the same set, all known set-theoretic operations can be applied. In particular, it is possible to consider the union, intersection and difference of relations $>$ and $=$. For example, a relation $\geq$ for numbers is a union of and. The relation $>$ and $=$ is the intersection of the relations $\geq$ and $\leq$, and the relation $\geq$ for vectors with R^n itself is the difference of the relations $\geq$ and $=$.

Types of relations. Elements a and b of the set A are called comparisons in relation R, if relations aRb or bRa are necessarily fulfilled (maybe both, together), and not comparable in relation R, if neither ratio is correct: aRb no bRa. For example, any two real numbers $\geq$ are comparable in relation, but may not be comparable in relation $>$ (since the inequality $a > a$ is not correct). If any two elements of a set A are compared with a relation R, then such a relation is called complete. If in the set A there is at least one pair of elements that are not comparable in relation R, then R call a partial relation. In a set of real numbers, the relation $\geq$ is complete, and the relation $>$ is a partial relation. For set $A = R^n$, the relation $\geq$ is no longer complete, because, for example, the vectors $(1,-1)^{\mathrm{T}}$ and $(-1,1)^{\mathrm{T}}$ are not comparable in relation $\geq$. Hence, in this case, the ratio $\geq$ is partial.

4.2 Sets of optimal solutions

The ratio of advantages and distinguishability. The choice of a solution from the set of possible solutions X is equivalent to the choice of an estimate Y from a set of estimates, thereforein this section, for convenience, the set of possible solutions is denoted by Z, believing that Z can be taken as X and Y.

For definiteness, let's fix DM. If DM chooses a solution b from two given solutions a and b, then it is possible to say that a solution a is preferable to solution b. Pairs of the form (a,b) where $a,b \in Z$, for which a solution a is preferable to a solution b, form a certain set, which is called a strict preference relation and is denoted by a symbol $\succ$. The indicated set is a relation defined on the set Z. Accordingly, a notation $a \succ b$ means that a decision a for a decision maker is preferable to a solution b.

When a pair of solutions a and b is compared predominantly, such a case is possible when none of them will be preferred. This is the case if, for example $a = b$. Therefore, the following definition is introduced. It is possible to say that the solution a and b, where $a,b \in Z$ is incomparable, if neither $a \succ b$ nor $b \succ a$ relations are satisfied. In other words, solutions a and b are incomparable if they can't be comparable in relation $\succ$. Set of pairs of the kind (a,b) in which solutions a and b are incomparable are called the ratio of indistinguishability (the ratio of indifference) and are denoted by the symbol ~.

It should not be assumed that the relation $a \sim b$ means equality $a = b$. If, for example, $Z = R^n (n > 1)$ and the relation is taken with respect to the relation $\geq$, then $(1.0)^{\mathrm{T}} \sim (0.1)^{\mathrm{T}}$ is correct, however $(1,0)^{\mathrm{T}} \neq (0,1)^{\mathrm{T}}$. The relation $a \sim b$ can take place when the decision maker considers that in the sense of advantage for him there is no difference between the solutions a and b (in particular, when $a = b$). In addition, indistinguishability can also occur if the solution a and b the DM can't compare at all with each other.

So, for an arbitrarily selected pair of solutions $a,b \in Z$ one and only one of the given relations $a \succ b$, $b \succ a$, $a \sim b$ is satisfied.

Finding a set of optimal solutions. Let DM when choosing a solution from the set Z be guided by some strict advantage relation $\succ$, which is asymmetric and transitive. Let's use the relation $\succ$ in order to single out solutions that can be «better», that is, optimal solutions. All those solutions for which there are preferred solutions should be deleted from Z; they consciously can't be considered optimal. As a result of such an exception Z, there will remain a solution (the only solution), each of which can be considered optimal according to this ratio $\succ$.

Thus, a solution $z^{(0)} \in Z$ is called optimal with respect $\succ$ to the set Z if there is no other solution $z \in Z$ for which a fair relation $z \succ z^{(0)}$ exists. Using a relation $\succeq$ constructed on the basis of relations $\succ$ and ~ it is possible to give an equivalent formulation: relation $z^{(0)} \in Z$ is called optimal with respect to the relation $\succ$ if the relation $z \succ z^{(0)}$ is not satisfied for any other $z \in Z$.

The set of all optimal solutions of the set Z is denoted by $opt_{\succ} Z$. Depending on the structure Z and type of relation $\succ$, the set $opt_{\succ} Z$ may contain a single element, a finite or infinite set of elements, and also not contain a single element.

Theorem 4.1. If the set Z is not empty and contains a finite number of elements, and the relation $\succ$ is asymmetric and transitive, then the set of optimal solutions is nonempty $opt_{\succ} Z \neq 0$.

The proof of this statement is constructive and in fact is an algorithm for finding the whole set of optimal solutions. Let's introduce the notation:

$$Z = Z_1 = \left\{ z^{(11)}, z^{(12)}, \ldots, z^{(1n_1)} \right\}.$$

If $n_1 = 1$ that:

$$Z_1 = \left\{ z^{(11)} \right\} = opt_{\succ} Z_1.$$

Therefore, let's further consider $n_1 > 1$. The first step of the algorithm is a pairwise comparison of the solutions $z^{(1i)}$ for each of the latest solutions. If a relation $z^{(11)} \succ z^{(1i)}$ holds for some $i \in \{2,3,\ldots,n_1\}$, then the solution $z^{(1i)}$ is removed from the set Z_1: it can't be optimal. Otherwise, when $z^{(11)} \approx z^{(1i)}$ or $z^{(11)} \succ z^{(1i)}$ the solution $z^{(1i)}$ is ever stored. After all comparisons are complete, the solution $z^{(11)}$ should also be excluded from Z_1. Moreover, if there is no relation $z^{(1i)} \succ z^{(11)}$ for which the relation $i = 2,3,\ldots,n_1$ turned out to be fulfilled, then the solution $z^{(11)}$ is optimal and must be remembered. The set of decisions left as a result of the withdrawal is denoted by

$$Z_2 = \left\{ z^{(21)}, z^{(22)}, \ldots, z^{(2n_2)} \right\}, \quad n_2 < n_1.$$

If $Z_2 = \varnothing$, then the solution $z^{(11)}$ is optimal (it is stored in memory), because through asymmetry of the relation $\succ$ turns out that the relation $z^{(1i)} \succ z^{(11)}$ $i = 2,3,\ldots,n_1$ can't take place. In this case $Z_2 \neq \varnothing$, the procedure for finding the set is completed. If, then go to the next step of the algorithm.

The second step is similar to the first and consists in pairwise comparing the solutions $z^{(21)}$ for each of the solutions $z^{(22)},\ldots,z^{(2n_3)}$. All solutions $z^{(2i)}$ for which $z^{(2i)} \succ z^{(21)}$ are excluded from the set Z_2. In addition, decision $z^{(21)}$ is excluded. At the same time, if there is no relation $i = 2,3,\ldots,n_2$ for which $z^{(21)} \in opt_{\succ} Z_2$ turned out to be fulfilled, then, moreover, $z^{(21)} \in opt_{\succ} Z_1$ the decision $z^{(21)}$ should be remembered. In fact, the relation $z^{(11)} \succ z^{(21)}$ can't take place, since the solution $z^{(21)}$ is not removed from Z_1 in the first step. The relation $z^{(1i)} \succ z^{(21)}$ for $z^{(1i)} \in Z_1 \setminus Z_2$, $i \neq 1$ also can't be fulfilled, since $z^{(11)} \succ z^{(1i)}$ and relation $\succ$ is transitive: from $z^{(11)} \succ z^{(1i)}$ and $z^{(1i)} \succ z^{(21)}$ it follows that $z^{(11)} \succ z^{(21)}$. Set of decisions remaining after the exception are denoted by

$$Z_3 = \left\{ z^{(31)}, z^{(32)}, \ldots, z^{(3n_3)} \right\}, \quad n_3 < n_2.$$

If $Z_3 \neq \varnothing$, then go to the next step, etc.

The algorithm is such that according to the transitivity of the relation $\succ$, the solution $z^{(k1)}$ that is optimal on the set Z_k is optimal on Z_{k-1}, $k = 2,3,\ldots$, and therefore on the original set Z_1.

Since the set Z_1 contains a finite number of elements, the procedure will end in a finite number of steps.

The solutions stored in memory form the desired nonempty set of optimal solutions with respect to $\succ$: $opt_{\succ} Z$.

Let's estimate the «laboriousness» of the formulated algorithm, that is, let's determine the smallest and greatest possible number of pairwise comparisons that will be required to find the entire set $opt_{\succ} Z$. The smallest number of comparisons $n_1 - 1$ takes place if $z^{(11)} \succ z^{(1i)}$, $i = 2,3,\ldots,n_1$. In the «long version», all possible pairs of solutions will have to be compared with each other and therefore the maximum number of comparisons is equal $n_1(n_1 - 1)/2$.

4.3 Pareto optimal estimates and solutions

Consistency of relation of advantage on the sets of decisions and estimates. Let the objective vector function $\vec{f}(x) = (f_1(x), f_2(x),\ldots,f_m(x))$ be defined on the set of feasible solutions $X \subset R^n$. Given a set X in a mapping $\vec{f}(\bullet)$ corresponds to a set of estimates Y; it is determined by equality (4.1). Let's assume that on sets X and Y given the relation is a strict advantage $\underset{X}{\succ}$ and $\underset{Y}{\succ}$ respectively. Each decision $\vec{x} \in X$ corresponds to a specific estimation $\vec{y} = \vec{f}(\vec{x}) \in Y$ and, conversely, to each estimate $\vec{y}$ correspond such decisions $\vec{x}$ for which $\vec{f}(\vec{x}) = \vec{y}$. Therefore, these relations are consistent with each other: $\vec{y} \underset{Y}{\succ} \vec{y}'$ takes place if and only if $\vec{x} \underset{X}{\succ} \vec{x}'$, where $\vec{y} = \vec{f}(\vec{x})$, and $\vec{y}' = \vec{f}(\vec{x}')$. Thus, the results formulated in terms of estimates can be reformulated with respect to solutions, and vice versa.

Multicriteria problems. Let's consider two arbitrary estimates $\vec{y}$, $\vec{y}' \in Y$, which are interconnected by inequality $\vec{y} \geq \vec{y}'$ (that is $\vec{y} \neq \vec{y}'$). Moreover, the estimation $\vec{y}$ may be better for the decision maker than $\vec{y}'$.

The multicriteria maximization problem is characterized by the fact that estimation $\vec{y}'$ always prevails $\vec{y}'$, if only $\vec{y} \geq \vec{y}'$. In other words, in the multicriteria maximization problem, the following axiom is considered fulfilled.

Pareto axiom (in terms of estimates). For any two estimates $\vec{y}$, $\vec{y}' \in Y$ corresponding inequalities $\vec{y} \geq \vec{y}'$, the relation $\vec{y} \underset{Y}{\succ} \vec{y}'$ is always satisfied.

Pareto axiom (in terms of solutions). For any two decisions $\vec{x}$ for which $\vec{f}(\vec{x}) \geq \vec{f}(\vec{x}')$ is correct, there is always a relation $\vec{x} \underset{X}{\succ} \vec{x}'$.

In the multicriteria minimization problem, it is believed that for two arbitrary estimates $\vec{y}$, $\vec{y}' \in Y$, connected by inequality $\vec{y}' \geq \vec{y}$, relation $\vec{y} \underset{Y}{\succ} \vec{y}'$ is always satisfied. For definiteness, we restrict ourselves to the consideration of maximization problems. The results and conclusions can be easily reformulated as applied to minimization problems.

The Pareto axiom imposes certain requirements on the nature of the relation of advantage in the multicriteria maximization problem. Namely: for a decision maker, it is advisable to obtain the greatest possible value for each of the criteria $f_1, f_2,\ldots,f_m$, that is, maximize each of the criteria. The maximum point in

the set X at the same time for all functions $f_1, f_2, \ldots, f_m$ is consciously the optimal solution to the multicriteria maximization problem. However, in practice this case is extremely rare, since such a maximum point, as a rule, does not exist. Therefore, in the absence of additional information about the advantages $\succ_X$ and $\succ_Y$ in the multicriteria problem, it is possible to find only a certain upper bound for the desired set of optimal solutions.

Pareto optimality. According to Pareto axiom, relation $\geq$ plays an important role in multicriteria problems. Therefore, the set of optimal estimates with respect to $\geq$ in the set Y has a special name: *the set of Pareto optimal or effective estimates*. This set is denoted by $P(Y)$. Using the symbols adopted in the previous section, by definition, you can write: $P(Y) = opt_{\geq} Y$. Further let's use the notation $P(Y)$. Thus, inclusion $\vec{y}^{(0)} \in P(Y)$ takes place if and only if there is no other estimate $\vec{y} \in Y$ for which the inequality $\vec{y} \geq \vec{y}^{(0)}$ holds.When $m = 1$, the relation $\geq$ turns into a relation $>$ for numbers, the Pareto optimal estimate coincides with the maximum element of the number set $y \subset R$. If $m = 2$ (that is, two criteria), then the set $P(Y)$ has a simple geometric interpretation in the criteria space (Fig. 4.3).

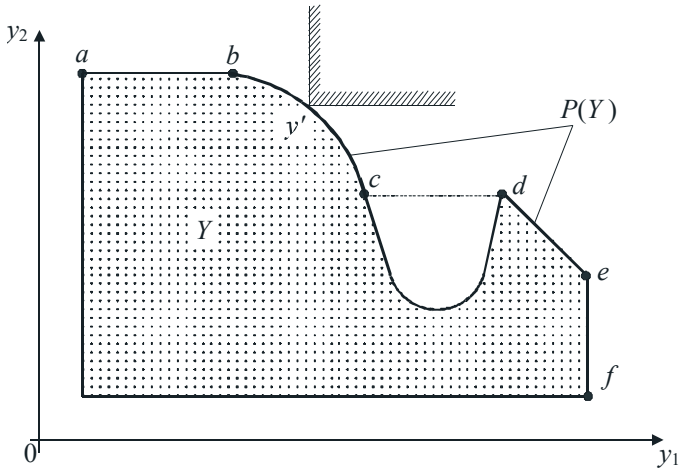

Fig. 4.3 Geometric interpretation in the criteria space

For the set Y shown in this figure, the Pareto optimal estimates consist of the points of the curve bc (excluding the point c) and the line de. $P(Y)$ is the «northeastern» boundary of the set Y without those parts of it that are parallel to one of the coordinate axes or lie in «deep dips». In order to verify this, it is enough to recall that all points $y \in R^2$ for which the correct inequality $\vec{y} \geq \vec{y}'$

forms a right angle, the sides of which are parallel to the coordinate axes, and the vertex is a point $\vec{y}'$. Therefore, if for a point $\vec{y}' \in Y$ the indicated angle is located outside the set, then this point is Pareto optimal, otherwise it will not be like that.

A solution $\vec{x}^{(0)} \in X$ for which inclusion $\vec{y}^{(0)} = \vec{f}(\vec{x}^{(0)}) \in P(Y)$ is true is called a *Pareto optimal or an effective solution* with respect to a vector function $\vec{f}(\bullet)$ on the set X. The sets of all such solutions are denoted by $P_f(X)$. Thus, inclusion $\vec{x}^{(0)} \in P_f(X)$ takes place if and only if there is no $\vec{x} \in X$ such that the inequality $\vec{f}(\vec{x}) \geq \vec{f}(\vec{x}^{(0)})$ holds.

If $m = 1$, then formulated definition of the Pareto optimal solution turns into the definition of the maximum point of a numerical function $f_1(\bullet)$. Thus, the concept of the Pareto optimal point can be considered as a generalization of the concept of the maximum point of the objective function to the case of several objective functions.

4.4 Practical features of choosing the optimal design options for systems, taking into account the totality of quality indicators

Multicriteria optimization defines the rules for choosing the optimal design solutions – the optimal options for building the system, taking into account the totality of the system quality indicators. Vector optimization methods have gained rapid development both in the field of system-wide analysis and in the field of radio engineering systems, in particular, telecommunication systems. This was caused by the objective need to take into account when designing the totality of, as a rule, conflicting technical and economic requirements for the system. The main principles of multicriteria optimization are used in the synthesis and analysis of a system when a project of an optimal system is created using mathematical models of messages, signals, interference, optimality criteria, as well as methods for choosing optimal design solutions.

Let's consider the practical features of a formalized formulation and the main methods for solving multicriteria optimization problems that can be used at the initial stages of designing optimal communication systems and networks, taking into account the totality of quality indicators.

Features of the formulation of the problem of designing optimal systems. When designing optimal systems, the statement of the problem plays an important role. No wonder they say that the correct formulation of the problem by half gives its successful solution. Let's consider the problems of a formalized formulation of the problem of designing an optimal system taking into account the totality of quality

indicators. Let's assume that an alternative design solution is an option to build a system ϕ.

In general terms, the system can be represented as an ordered set of elements, relations between them and their properties. Their unambiguous task completely determines the construction and effectiveness of the system. Let's assume that the system $\phi = (S, \vec{p})$ construction option is determined by the structure S and vector of parameters $\vec{p}$.

These abstract system definitions should be specified in the design process. When designing, the structure of the optimal system should be determined as a set of relevant elements and the relations between them, and the values of the optimal parameters of this system should be obtained.

Initial data for the design of the system include: a set of working conditions of the system $\{Y\}$; restrictions on working conditions, structure and system parameters $\{Q_s\}$; a set of quality indicators $\vec{K} = (k_1, ..., k_m)$ and restrictions on the values of these quality indicators $\{Q_{\bar{k}}\}$; system optimality criterion.

Let's consider examples of input data in the design of communication systems. The working conditions $\{Y\}$ of a communication system may include: the type and characteristics of messages, signals, interference, communication channels. Restrictions on the structure of the system $\{Q_s\}$, depending on the specific task, can be set both weak and hard. In particular, these are general requirements for a class of communication systems, for example, requirements for a system to be single-channel, have free access, and do not include a return (service) channel, repeaters. For more stringent restrictions, the principle of the system's operation, the type of modulation, the decomposition of the system and even the complete structure are set, and during the optimization process, only the vector of internal parameters of the system varies. Moreover, restrictions on system parameters (for example, modulation parameters, signal power or interference, number of channels) can be of the type of equalities, inequalities, or some functional connection.

The vector of quality indicators $\vec{K} = (k_1, ..., k_m)$ includes a set of external parameters of the communication system that characterize the main tactical and technical characteristics of the system, for example, message transmission speed, probability of transmission errors, throughput, reliability, message delay time, probability of false message delivery. When setting the task, it is not numerical values that are set, but only the composition of quality indicators that should be taken into account when optimizing the system. At the initial stages of design, as a rule, only the most important quality indicators are taken into account. The restrictions $Q_{\bar{k}}$ imposed on the numerical values of quality indicators can also be of the type of equalities, inequalities or functional relations.

Systems that satisfy the totality of data $\{Y\}, \{Q_s\}$ are called valid, and systems that satisfy the restrictions on the values of quality indicators $\{Q_{\bar{k}}\}$ are called

strictly valid. Previously, when designing systems, they were limited to choosing strictly acceptable systems. With the complication and increase in the cost of the systems designed, it becomes relevant to search for optimal quality indicators for the system as a whole.

Of all strictly admissible systems, the optimal (best) is the system to which the best (in the previously established sense) value of the vector $\vec{K}$ corresponds. To select an optimal system, one should choose or justify a criterion for the advantage of one system over another (optimality criterion), that is, a rule based on which one value of the vector of quality indicators $\vec{K}$ should be recognized as the best in comparison with another value.

Thus, the task of designing an optimal system is formulated as follows: to find a system that satisfies the totality of the source data $\{Y\}, \{Q_s\}, \{Q_{\bar{k}}\}$ and does not have a vector of quality indicators $\vec{K}$, it is better according to the chosen advantage criterion.

Designing, carried out taking into account the totality of quality indicators $\vec{K} = (k_1, ..., k_m)$, is called vector synthesis (vector optimization, optimization by vector criterion, multi-criteria optimization). In contrast, the synthesis of a system, taking into account one quality indicator $(m = 1)$, is called scalar.

Depending on the formulated initial data, finding the optimal system can be reduced to solving various mathematical problems of optimization problems:

1. Synthesis of the optimal structure of the system, which means finding the optimal structure of the system.
2. Parametric optimization, that is, the choice of optimal values of the system parameters for a given structure.
3. Discrete selection of optimal system options with a finite set of valid options.

Mathematical methods for optimizing parameters and discrete selection are well developed and are widely used in system design. Synthesis of the structure of the system is a difficult task and often encounters difficulties not only mathematical, but also fundamental in nature, associated with information uncertainty in formulating the conditions of the system, as well as the choice of a generalized objective function of the system.

When forming the target function of the system and its optimization, the difficult task of «approximating» the function of choosing the optimal system, which is in the imagination of the customer of the system, another function of choice is formalized in the form of a certain criterion of optimality using strict mathematical methods. As a rule, it is not immediately possible to select a global optimality criterion in the form of a scalar objective function, it includes a set of quality indicators and the optimization of which would lead to the selection of

a single optimal variant of the system. Therefore, there is a need to introduce a set of objective functions related to the corresponding quality indicators, which leads to the need to solve multicriteria optimization problems. With the introduction of the vector objective function $\vec{K}(\phi) = \left(k_1(\phi), k_2(\phi), ..., k_m(\phi)\right)$, the set of valid variants Φ_v of the system, each version of which is characterized by a corresponding vector of estimates $\vec{v} = (v_1, v_2, ..., v_m)$ and is mapped into the criteria space of vector estimates $\Phi_v \to Y \in R^m$. This makes it possible to compare the system options with each other in the criteria space Y and select the optimal vector estimates and the corresponding optimal system options according to a certain optimality criterion.

Formation of the set of valid variants of the system based on the morphological approach. When defining the set of acceptable options for technical systems of widespread use, he acquired a morphological approach, which is characterized by the following factors:

- identification of the maximum list of the basic functions of the system and the decomposition of the system into subsystems according to functional features $\left\{\phi_l, l = \overline{1,L}\right\}$;
- determination of various alternative ways of implementing each subsystem and setting acceptable options for their construction $\Phi_l = \left\{\phi_{l1}, \phi_{l2}, \ldots, \phi_{lK_l}\right\}$;
- formation of various options for constructing the system as a whole on the basis of morphological classes – the set of options for constructing each subsystem for which the conditions are satisfied: $\sigma(l) = \Phi_l$, $l = \overline{1,L}$, $\sigma(l) \cap \sigma(j) = 0$.

A morphological table is formed (Table 4.1). Each variant of building a system is determined by various possible variants of subsystems.

Table 4.1

Morphological table for specifying the set of valid system options

Morphological classes	Possible ways to implement the subsystem	Number of ways to implement the system
$\sigma(1)$	$\phi_{11}\ \phi_{12}\ \phi_{13}\quad \phi_{1K_1}$	K_1
$\sigma(2)$	$\phi_{21}\ \phi_{22}\ \phi_{23}\quad \phi_{2K_1}$	K_2
....	...	...
$\sigma(l)$	$\phi_{l1}\ \phi_{l2}\ \phi_{l3}\quad \phi_{lK_1}$	K_l
...	...	...
$\sigma(L)$	$\phi_{L1}\ \phi_{L2}\ \phi_{L3}\quad \phi_{LK_L}$	K_L

When forming the set of acceptable variants of a system, restrictions on the structure, parameters and technical implementation of individual subsystems and systems as a whole, as well as acceptable combinations of combinations of individual variants of subsystems among themselves should be taken into account. The number of possible system options is defined as:

$$Q = \prod_{l=1}^{L} K_l.$$

The choice of the optimality criterion of the system. When solving optimization problems, the question of choosing a system optimality criterion is very important. It is the criterion of optimality that determines the true value of the designed system. No convenience of a mathematical or other nature can compensate for the harmful consequences of applying an inadequate criterion for the optimality of the system.

The choice of the criterion of optimality, as already noted, is associated with the formalization of the imagination of the customer of the system (DM) about the advantages of the system and its optimality. There are two approaches to describing the advantages of one variant of a system over another: ordinalistic and cardinalistic.

Cardinal approach to describing customer preferences ascribes to each system $\phi \in \Phi_v$ a numerical value of the utility function $U(\phi)$. The utility function determines the corresponding order (or advantage) R on the set Φ_v if and only if the inequality $U(\phi') > U(\phi'')$ holds for the various options $\phi' R \phi''$. In this case, it is said that the utility function $U(\phi)$ is an indicator of advantage R. In fact, this approach is associated with the task of such a scalar objective function, the optimization of which in the general case can lead to the choice of a single best (optimal) version of the system:

$$\phi_0 = \arg \underset{\phi \in \Phi_v}{extr} \{U(\phi)\}.$$

However, at the initial stages of system design, it is rather difficult to define a scalar utility function. Therefore, a set of quality indicators and related objective functions are first introduced. This is due to the following reasons: the versatility of the technical requirements for the designed system; the need to ensure the optimality of the system under various conditions of its operation; the system consists of several interconnected subsystems and the optimality of the system as a whole is determined by the efficiency of its components.

Due to the fact that the system ϕ has to be characterized by a combination of quality indicators and related objective functions, this complicates the process

of choosing the optimal system options. There are three cases: quality indicators are not related; quality indicators are interconnected but consistent; quality indicators are interconnected and are competing (antagonistic).

In the first case, finding the optimal system options is performed by optimizing for each of the objective functions independently:

$$\phi_{0i} = \arg\underset{\phi\in\Phi_v}{extr}\left\{k_i(\phi)\right\},\ i=\overline{1,m}. \tag{4.2}$$

In the second case, the best options can also be found by optimizing individual objective functions, that is, this case is close to the first.

In the third case, the extrema for different objective functions do not coincide. The solution to this optimization problem is a consistent optimum of objective functions. The agreed optimum is that the minimum (maximum) value of each of the objective functions is achieved, provided that the other objective functions take fixed but arbitrary values.

Ordinary approach appeals to order (better-worse) and is based on the introduction of certain binary relations on the set of valid variants of the system. In this case, the concept of preference for the customer of the system is a binary relation R on the set of admissible systems Φ_v, which reflects the imagination of the customer of the system, the system ϕ' is better than the system ϕ'': $\phi' R \phi''$.

In practice, often when choosing a system on a set Φ_v, one can be guided by the relation of strict advantage $\succ$, which is asymmetric and transitive. Moreover, the system $\phi_0 \in \Phi_v$ is called optimal in relation $\succ$ if there is no other system $\phi \in \Phi_v$ for which relation $\phi \succ \phi_0$ is rightly. Set of the optimal systems with respect to $\succ$ are denoted by $opt_{\succ}\Phi_v$. Depending on the structure of the admissible set Φ_v and properties of the relation $\succ$, the set of optimal systems may include a single element, a finite or infinite number of elements. If the inseparability relation coincides with the equality relation $=$, then the set $opt_{\succ}\Phi_v$ (if it is not empty) consists of a single element.

With the introduction of the set of objective functions, each system is characterized by a vector of estimates $\vec{\nu} = (\nu_1, \nu_2, ..., \nu_m)$ and is mapped to the criteria space. Moreover, the indicated strict advantage relation also exists for vector estimates. The consistency of the relation of preference on the set of design decisions Φ_v and in the space of vector estimates V establishes the Pareto axiom. According to it, for any two vector estimates $\vec{v}', \vec{v}'' \in V$ satisfying the vector inequality $\vec{v}' \geq \vec{v}''$, the relation $\phi' \succ \phi''$ always holds.

The set of optimal estimates $\geq$ with respect to space V is called the set of Pareto optimal or effective estimates and denote $P(V) = opt_{\geq}V$. Inclusion $\vec{v}^0 \in P(V)$ takes place if and only if there is no estimate for which the inequality $\vec{v} \geq \vec{v}^0$ holds. Such a criterion for choosing optimal solutions is called the unconditional advantage criterion (UAC) or the Pareto criterion.

Design decisions, that is, options for constructing a system $\phi_0 \in \Phi_v$ for which inclusions $\vec{v}_0 = \vec{K}(\phi_0) \in P(V)$ are justified, are called Pareto optimal with respect to the vector objective function $\vec{K}(\phi)$ defined on the set Φ_v, and are denoted as $P_{\vec{K}}(\Phi_v)$. In other words, $\phi_0 \in P_{\vec{K}}(\Phi_v)$ if and only if there is no such system $\phi \in \Phi_v$ for which the vector inequality holds:

$$\vec{K}(\phi) \geq \vec{K}(\phi_0). \tag{4.3}$$

Relation (4.3) means that the inequalities $k_j(\phi) \geq k_j(\phi_0)$ of all $j = \overline{1, m}$ are fulfilled, and at least one of the quality indicators satisfies strict inequality.

It should be noted that the strict advantage relation $\geq$, which takes place for vector estimates, turns at $m = 1$ into a relation $>$ for scalar estimates. In this case, the Pareto optimal estimate coincides with the extremum of the scalar objective function $k(\phi)$. Thus, the concept of Pareto optimality should be considered as a generalization of the concept of optimum in the case of several objective functions. Moreover, the Pareto optimum is a consistent optimum of interconnected and competing system quality indicators.

The following properties are characteristic of Pareto optimal design solutions:

1. All elements of the set of admissible variants of the system Φ_v that do not belong to the Pareto optimal set $P_{\vec{K}}(\Phi_v)$ are certainly worse.
2. Neither the Pareto optimal system from the set $P_{\vec{K}}(\Phi_v)$ can be recognized unconditionally worse or better in comparison with other systems of this set. This means that they are all incomparable according to the Pareto criterion.
3. If the set $P_{\vec{K}}(\Phi_v)$ is consistent, that is, it contains only one element (system), then the corresponding version of the system is the best.
4. Each Pareto optimal system corresponds to the potentially possible value of each of the quality indicators $(k_1, k_2, ..., k_m)$, which can be achieved with fixed but arbitrary values of other $(m-1)$ quality indicators. This is a multiple *m*-optimum property. The totality of such optimal values of quality indicators is the multidimensional potential characteristics of the system (MPC).
5. The optimal surface is a geometric site of Pareto optimal estimates has a strictly monotonic character, that is, each of the functions:

$$\begin{aligned} k_{10} &= f_1(k_2, k_3, ..., k_m), \\ k_{20} &= f_2(k_1, k_3, ..., k_m), \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ k_{m0} &= f_m(k_1, k_2, ..., k_{m-1}) \end{aligned} \tag{4.4}$$

for Pareto optimal estimates monotonically decreases for each of the arguments. These dependencies are called multidimensional exchange diagrams (MED) for Pareto optimal systems.

Compared with the one-dimensional MPC potential characteristics of the system and the associated MEDs, they are characterized by two important properties. Firstly, they give the best (potential possible) value of not one, but each of the selected quality indicators. Secondly, they indicate how the value of some quality indicators should be changed to improve other quality indicators and this can be done by changing the structure or parameters of the system.

Some methods for finding Pareto optimal solutions. Most methods for finding Pareto optimal solutions are based on certain Pareto optimality conditions. In the general case, sufficient and necessary Pareto optimality conditions are used. In particular, a solution is Pareto optimal if it is a solution to the problem of maximizing a certain function growing in relation. In fact, the solution of the Pareto optimization problem reduces to the set of corresponding scalar optimization problems with some restrictions. If the optimality conditions are used is also sufficient, then the set of solutions found in this way is the set of Pareto optimal solutions. Otherwise, the set found may include unnecessary solutions that must be rejected.

Finding the set of Pareto optimal systems can be carried out either by directly sorting out all strictly admissible system variants and checking condition (4.3), or using special methods, for example, the method of successive assignments, the weight method, and the method of performance characteristics. The choice of a suitable optimization method depends on the content of the formulated input data and the type of the design task. Let's consider the features of some methods.

Discrete brute force method. When solving the optimization problem by enumeration according to condition (4.3), it is assumed that the set Φ_v has finite power. Such tasks arise, for example, when choosing from already known («available» or in the form of technical projects) system options. In particular, many feasible systems can be formed on the basis of the well-known morphological approach as various feasible combinations of a certain number of subsystems. It is important to note here that even for relatively simple systems consisting of only a few subsystems, the number of permissible combinations of the latter can be significant (tens and hundreds of thousands). Therefore, although there are no fundamental difficulties in using the enumeration method, in practice, computational difficulties are possible.

Performance method. The method consists in finding an extremum of one of the objective, for example, the first function on the set of strictly admissible systems, provided that all other objective functions are constrained by equality type:

$$\underset{\phi\in\Phi_v}{extr}\, k_1(\phi) \text{ at } k_2(\phi)=k_{2f},\ ...,\ k_m(\phi)=k_{mf}. \tag{4.5}$$

Pareto optimal variants of the system are found by solving many scalar optimization problems with various allowable combinations of fixed values of quality indicators $k_{2f}, ..., k_{mf}$.

Obviously, the optimal value of the indicator k_{1o} in the general case will depend on the fixed values of other quality indicators $k_{1o}=f_p\left(k_{2f}, k_{3f},..., k_{mf}\right)$. The dependencies found in this way for admissible combinations of fixed values $k_{2f}, k_{3f},..., k_{mf}$ in the criteria space are the working surface. The working surface corresponds to a family with $(m-1)$ one-dimensional performance characteristics:

$$\begin{aligned}
k_{1o} &= f_p\left(k_2, \underline{k_3,..., k_m}\right),\\
k_{1o} &= f_p\left(\underline{k_2}, k_3,\underline{..., k_m}\right),\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
k_{1o} &= f_p\left(\underline{k_2, k_3,..., k_m}\right).
\end{aligned} \tag{4.6}$$

Variables underlined here are considered as fixed parameters.

The working surface has the following characteristic properties:

1. The working surface includes all Pareto optimal points, but along with them it has a number of certainly worst points. They should be discarded from further consideration.
2. A necessary and sufficient condition for the working surface to coincide with a Pareto optimal set is its strict monotonicity, that is, a monotonously descending character with respect to each of the arguments. In this case, the working surface determines the MPC of the system.

The main difficulties when using the method of performance characteristics are solving the scalar optimization problem under conditions of $(m-1)$ equality type constraint. But in many practical cases, this problem can be brought to a specific structure of the system with arbitrary parameters.

Weight method. During its application, Pareto optimal solutions are found by optimizing the weighted sum of objective functions of the form:

$$\underset{\phi\in\Phi_v}{extr}\left\{k_w = k_1(\phi)+a_1k_2(\phi)+...+a_{m-1}k_m(\phi)\right\} \tag{4.7}$$

with valid combinations of positive weights $a_1, a_2,..., a_{m-1}$. In this case, the optimal values k_{w0} and the corresponding values of the quality indicators $k_{1w}, k_{2w},...,\cdots k_{mw}$ are found:

$$k_w = k_{1w} + a_1 k_{2w} + \ldots + a_{m-1} k_{mw}. \tag{4.8}$$

In general, the values $k_{1w},\ldots,k_{mw}$ depend on the selected weights $a_1, \ldots, a_{m-1}$:

$$\begin{aligned} k_{1w} &= f_{1w}(a_1, \ldots, a_{m-1}), \\ k_{2w} &= f_{2w}(a_1, \ldots, a_{m-1}), \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ k_{mw} &= f_{mw}(a_1, \ldots, a_{m-1}). \end{aligned} \tag{4.9}$$

To solve the optimization problem (4.7), as well as to find the dependencies (4.9), it is necessary to perform optimization for all possible combinations of coefficients $0 < a_j < \infty,\ j = \overline{1, m-1}$.

Having solved the system with m equations (4.9), it is possible to obtain the dependence:

$$k_{1w} = f_{1w}(k_{2w}, k_{3w}, \ldots, k_{mw}). \tag{4.10}$$

In the m-dimensional space of vector estimates, this dependence is considered as the equation of the weight surface. It is easy to see that the use of the weighted method reduces to many scalar optimization problems, in particular, the well-known method of Lagrange multipliers.

The weight surface has the following properties:

1. Includes only Pareto optimal points, that is, none of the worst worst points can belong to this surface.
2. In many cases, the weight surface is completely defined and continuous in the entire range of quality indicators $k_2, k_3, \ldots, k_m$.

In such cases, the weight surface coincides with the Pareto optimal set.

So, when using the considered methods, as well as their modifications, the vector optimization problem reduces mathematically to solving many scalar optimization problems, taking into account various kinds of restrictions.

In the general case, when solving optimization problems (4.5), (4.7), the system operator $\phi = (S, \vec{p})$ varies, that is, both the structure S and the parameters $\vec{p}$ of the system. In this case, methods of calculus of variations, functional analysis, theory of statistical solutions, and theory of information can be used. With a fixed structure of the system S, the problem of finding the optimal version of the system is reduced to the problem of optimizing the vector of parameters $\vec{p}$. In some cases, this problem can be solved by linear, nonlinear, or dynamic programming methods.

If the Pareto set $P(Y)$ is relatively narrow, then any Pareto optimal estimate and the corresponding system can be taken as the optimal solution. In such cases,

it is possible to assume that the strict advantage relation $\succ$ coincides with the relation $\geq$ on the set of vector estimates, and $opt_{\succ} Y = P(Y)$. At the same time, they often do not resort to searching for the whole set of Pareto optimal systems, but immediately choose one of the Pareto optimal options.

However, often set $P(Y)$ is too big. This indicates that the relations $\succ$ and $\geq$, although related by the Pareto axiom, does not coincide. To narrow the set of Pareto optimal estimates, one should use the conditional advantage criterion (CAC), which reduces to the formation of some scalar function of choosing the only option, which can be set after receiving additional information from the decision maker about his understanding of the advantages of one system over another.

This raises the question: does it make sense to make the choice of system options based on the unconditional advantage criterion – the Pareto criterion, if at the final stage you still have to introduce a conditional advantage criterion. In support of the feasibility of searching for Pareto optimal system options using UAC at the initial stages of optimal design, we note the following:

1. UAC allows to find all Pareto optimal systems, that is, reject the worst-case system variants.
2. UAC allows to find the potential (best possible) values of each of the quality indicators and the relation between them, that is, MPC and MED.
3. Methods for finding Pareto optimal systems are reduced mathematically to the optimization of scalar objective functions, that is, they reduce the solution of the vector synthesis problem to a certain set of scalar synthesis problems.
4. In a degenerate case, UAC allows one to find the best single system.
5. In the non-degenerate case of finding Pareto optimal systems often leads to the same system structure, but with different parameters.
6. Even when at the final stage of the synthesis it is necessary to introduce CAC to select a single system, it is better to introduce various conventions at a later stage of design.

Narrowing methods for the set of Pareto optimal solutions. The formal model of the Pareto optimization problem does not contain information for choosing a single alternative. Moreover, the set of admissible variants of the system only narrows to the Pareto set by eliminating the certainly worst-case variants with respect to $\succ$. However, for subsequent stages of system design, as a rule, a single version of the system should be selected. Therefore, it becomes necessary to narrow the set of Pareto optimal solutions with the use of additional information about the relation $\succ$. Such information appears as a result of a comprehensive analysis of the structure and parameters of Pareto optimal system options,

multidimensional diagrams of the exchange of system quality indicators, the relative importance of quality indicators, and a comparative analysis of the obtained system variants among themselves. Such an analysis is carried out with the involvement of decision maker.

The additional information obtained in this case can be used to construct some scalar objective function $U\left(k_1(\phi),..., k_m(\phi)\right)$, it depends on a set of quality indicators. Optimization on the set of Pareto optimal solutions $P_{\bar{k}}\left(\Phi_v\right)$ leads to the choice of a single optimal variant of the system:

$$\phi_0 = extr\left(U\left(k_1(\phi),..., k_m(\phi)\right)\right),\ \phi \in P_{\bar{k}}\left(\Phi_v\right),\ j = \overline{1, m-1}. \tag{4.11}$$

The general requirement for a function $U\left(k_1, ..., k_m\right)$ is ensuring that it is monotonous (increasing or decreasing) for each of its arguments.

There are both objective and subjective approaches to the construction of such a function. In some cases, on the basis of considering the purpose of the system designed as part of a more complex supersystem (complex), objective methods can establish the relation of the quality indicators of the system $\left(k_1,..., k_m\right)$ with some global quality indicator K of the supersystem in the form of an appropriate function $K = U\left(k_1,..., k_m\right)$. However, in most cases it is not possible to objectively introduce such a function, and it is necessary to resort to its construction to a large extent by subjective methods. Let's consider some of them.

Selection of optimal solutions using value functions. One of the widely used methods of narrowing the set of Pareto optimal solutions is the use of a scalar value (utility) function, the optimization of which leads to the selection of one of the optimal options for the system. A numerical function $U\left(v_1,..., v_m\right)$ is called a value function for a strict advantage relation $\succ$, if for arbitrary estimates $\vec{v}'$, $\vec{v}'' \in Y$ in the criteria space Y inequality $U\left(\vec{v}'\right) > U\left(\vec{v}''\right)$ occurs if and only if $\vec{v}' \geq \vec{v}''$. Let's suppose that a strict advantage relation $\succ$ satisfies the Pareto axiom. Moreover, the relations $\vec{v}' \succ \vec{v}''$ follows from inequality $\vec{v}' \geq \vec{v}''$, which means $U\left(\vec{v}'\right) > U\left(\vec{v}''\right)$ that there is a value function $U\left(\vec{v}\right)$ that is growing in relation $\geq$. If a value function $U\left(\vec{v}\right)$ is constructed, then the optimal estimate is found by maximizing this function on the Pareto set:

$$\vec{v}_0 \in Y:\ U\left(\vec{v}_0\right) = \max_{\vec{v} \in opt_{\geq} V} U\left(\vec{v}\right). \tag{4.12}$$

Thus, finding the optimal estimate reduces to solving the scalar optimization problem for the function of many variables $U\left(\vec{v}\right)$. In this case, additive, multi-

plicative, and multiline value functions can be constructed. The procedure for constructing a value function $U(\vec{v})$ is sometimes called a convolution of a vector criterion $\vec{K} = (k_1, k_2, ..., k_m)$.

The convolution operation is possible if:

- particular criteria are quantitatively total in importance, that is, each of them corresponds to a certain number C_i, which determines its relative importance in accordance with other criteria;
- particular criteria are homogeneous, that is, they are quantitatively compared in one dimension.

There are various forms of representing the generalized scalar criterion and choosing the appropriate optimal solutions. In particular, these are such methods of convolution of particular criteria:

- a generalized criterion is formed, the numerator of which is the product of the criteria to be maximized, and the denominator is the product of the criteria to be minimized;
- a generalized criterion is formed on the use of elements of the theory of additive utility, that is, the summation of particular criteria with certain weighting factors;
- a generalized criterion is formed with respect to all particular criteria.

The generalized value function can take the following form:

$$U(v_1, ..., v_m) = \sum_{j=1}^{m} c_j \varphi_j(v_j), \tag{4.13}$$

where $\varphi_j(\cdot)$ – the one-dimensional value functions that characterize the value of the system with the *j*-th quality indicator; c_j – weight factors.

The task of constructing the value function (4.13) is reduced to evaluating the coefficients c_j, choosing the type of functions $\varphi_j(v_j)$, checking their independence in advantage $\geq$, checking the consistency of the constructed value function. In some cases, the value function (4.13) can be used in the form of an equivalent sum of partial estimates:

$$U(\vec{v}) = \sum_{j=1}^{m} c_j v_j. \tag{4.14}$$

In this case, various methods are used to obtain additional information on the value of the weighting coefficients c_j. In particular, these are well-developed expert estimation methods. They come down to a survey of a selected group of experts on the value of the obtained Pareto optimal system options, the relative importance of quality indicators, and the like. There are well-developed

methods for accounting for the information received, which are implemented in the Saaty method.

Sometimes, to select a single option, they are limited to the so-called threshold optimization: the most significant criterion is subjected to optimization, others are included in the constraint system. It should be noted that there are also many other principles and approaches to choosing a single option using scalar optimality criteria. In fact, relation (4.14) determines the Bayesian deterministic optimality criterion. In conditions of uncertainty about the conditions for choosing decisions, he uses the methods of game theory. Such situations when choosing design decisions when creating systems are often called «games with nature». To make decisions, find the best strategy using the Wald criterion, the Savage criterion, the Hurwitz criterion, the Laplace criterion, etc.

Selection of optimal solutions based on the theory of fuzzy sets. This approach is based on the fact that from the a priori uncertainty of the concept of «best version of the system» it is impossible to determine exactly. It is possible to assume that this concept is diffuse by the set, and to evaluate the system, the basic principles of the theory of blurry sets can be used. In the general case, the fuzzy set G on the set X is set by the membership $\xi_G: X \to [0,1]$, which compares with each element $x \in X$ a real number ξ_G on the interval $[0,1]$. This number is called the degree to which the element x belongs to a fuzzy set G. The closer it is to 1, the higher the degree of belonging. The function $\xi_G(x)$ is a generalization of the characteristic function of sets, which takes on only two values: 1 – for $x \in G$ and 0 – for $x \notin G$. In the case of discrete sets, the notation of a fuzzy set as a set of pairs $G = \{x, \xi_G(x)\}$ is used.

According to these basic provisions, each quality indicator of a system can be set in the form of a fuzzy set $k_j = \{k_j, \xi_{kj}(k_j)\}$, where $\xi_{kj}(k_j)$ is the membership function of a particular *j*-th quality indicator of a fuzzy set of best value.

Such a notation of a separate quality indicator has a high information content, since it gives an idea of the physical nature of the quality indicator, its specific value and value relative to the best (extreme) value that characterizes the membership function. The universal form of the membership function, which can be used as a scalar objective function, has the following form:

$$U(k_1,\ldots,k_m) = \frac{1}{m}\left\{\sum_{j=1}^{m}\left[\xi_k(k_j)\right]^{\beta}\right\}^{\frac{1}{\beta}}. \qquad (4.15)$$

The advantage of such an objective function is that by choosing a parameter β a wide class of functions from linear additive at to purely nonlinear at $\beta = 1$ can be realized for $\beta \to \infty$.

Selection of the best option with strictly ordered quality indicators. Sometimes, for the customer of the system, based on the analysis of Pareto optimal options, as well as their MEDs, it turns out to be desirable to obtain the highest possible value of one of the quality indicators, for example k_1, even due to the deterioration of other quality indicators. This means that the indicator k_1 is more important than other quality indicators.

It is also possible that the entire set of quality indicators $k_1,....,k_m$, strictly ordered by importance, that is, an indicator k_1 is more important than indicators $k_2, ..., k_m$, an indicator k_2 is more important than indicators $k_3, k_4,..., k_m$, etc. This corresponds to the situation when lexicographic relations are used when comparing system estimates. Let's give a definition of this relation and features of use when choosing the only version of the system in the criteria space of estimates.

Let there be two vectors of estimates $\vec{v}'$, $\vec{v}'' \in V$. A lexicographic relation $\vec{v}' \overset{lex}{\succ} \vec{v}''$ takes place if and only if one of the following conditions is satisfied:

$$\begin{gathered} \vec{v}'_1 > \vec{v}''_1, \\ \vec{v}'_1 = \vec{v}''_1,\ \vec{v}'_2 > \vec{v}''_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \vec{v}'_j = \vec{v}''_j,\ j = 1, 2..., m-1; \vec{v}'_m > \vec{v}''_m. \end{gathered} \tag{4.16}$$

For $m = 1$ lexicographical relation coincides with the relation $>$ on filings of real numbers. During execution of the relation $\vec{v}' \overset{lex}{\succ} \vec{v}''$ it is possible to say that the vector $\vec{v}'$ is lexicographically dominated by the vector $\vec{v}''$.

If a lexicographic relation is used when choosing a single system, it means that with a pair of estimates (and the systems corresponding to them), preference is given to that estimate (system) in which the first component of the vector $\vec{v}'$ (that is, the quality indicator k_1) is greater, regardless of the ratio of other components of the vector. If the first components of the ratings are the same, then preference is given to that rating (system) in which the large second component of the vector $\vec{v}'$ (quality indicator k_2). The following components of the vector $\vec{v}'$ can significantly lose the corresponding components of the vector $\vec{v}''$.

Similar conclusions take place when the first two components, three components, and so on up to $(m-1)$ the components of the vectors $\vec{v}'$ and $\vec{v}''$ are equal. In such cases, it is said that the components $v_1, v_2,..., v_m$, that is, the estimates of the quality indicators of the system $k_1(\phi), k_2(\phi), ..., k_m(\phi)$ are strictly ordered by importance.

In determining the lexicographic relation, an important role is played by the order of enumeration of quality indicators. Changing the numbering of quality indicators leads to a second lexicographic relation. In addition to the methods mentioned above for constructing a scalar objective function and choosing an option from the set of Pareto optimal ones, there are many others. The choice of a suitable method is determined by the initial data and the type of specific optimization problem. But be that as it may, the optimal system options should be sought among the Pareto optimal solutions to the problem. That is, the Pareto optimization stage is mandatory when designing systems taking into account the totality of quality indicators.

5 SELECTION OF MATHEMATICAL MODELS OF COMMUNICATION NETWORKS

An important stage in the design and solution of optimization problems is the construction of adequate mathematical models of communication networks. The type of mathematical model of the network depends on the problem being solved, that is, on what characteristic properties of the network the selected model should describe. This section briefly discusses only some mathematical models that can be used to describe information flows, application servicing processes at network nodes, network structure and network subsystems [1, 3, 5, 18, 26, 45]. More detailed information about various types of mathematical models of communication networks can also be found in [1, 3, 5, 10, 15, 16, 18, 26, 35, 45, 48].

5.1 Features of the communication network as an object of design

Let's consider the general structure of a communication network (CN). Information from information sources (IS) to information consumers (IC) can be delivered using a communication system or network. A communication network is a set of end points (EP), switching nodes (SN), communication channels (CC), computer centers (CCS) and a control system (CS), which are designed to transmit information from some set of information sources to multiple information consumers with using electrical signals.

End points are a set of technical means for converting information into electrical signals that can be transmitted via CCs.

SNs are divided into nodes with circuit switching, message switching, packet switching and are intended for information distribution. Communication channels are designed to carry signals in the space between individual points of the network. In fact, CC is a single or multi-channel system of a signal transmission system.

A control system is a set of software and hardware designed to ensure the normal operation of individual systems and devices and deliver messages to an address with specified quality indicators. SNs are used to provide information services, that is, the collection, storage and processing of information.

Thus, the communication network is a complex system of transmission and distribution of information, that is, a communication system in the broad sense. The choice of switching methods is determined based on the characteristics of message flows, user classes and quality of service indicators.

The process of transmitting messages to the CN, when the sources of information are independent of each other, generates a system of random (in time and space) data flows. Communication occurs at random times, their durations are also random variables.

A significant number of subscribers and the randomness of the message flows generated by them requiring service, gives reason to consider the CN as a queuing system (QS). The mathematical side of processes in the QS is investigated by queuing theory, as well as queuing theory. It establishes the relationship between the nature of the flows of applications, the performance of a single communication channel (as a serving apparatus), the number of channels and the efficiency of service.

There are various types of service systems when applications are received in the system when service devices (SD) are busy: *systems with losses* (the requirements arriving at the system do not find any free service devices and are lost) *systems with waiting* (waiting for any number of applications is possible, which cannot be serviced immediately and with the help of some service discipline it is determined in what order the expected requirements are selected from the queue for servicing); combined systems with waiting and losses (only a finite number of applications will be able to be determined by the number of places to wait, the application may be lost even when the waiting time for applications exceeds specified limits) priority systems (applications received have different priorities. If the application received has a high priority, and all servicing devices are busy, then it either takes one of the first places in the queue, or temporarily stops servicing a low-priority application, or even stops servicing.

Communication networks are also divided according to the type of transmission; depending on the served territory; by departmental affiliation; by the method of distribution and delivery of messages; by type of communication channels and their capacity.

Thus, the design of communication networks is a complex scientific technical and economic problem, requiring the solution of a large number of interrelated tasks. These include, first of all, tasks related to determining the network topology, choosing its architecture and switching method, choosing communication channels, managing information flows and the network as a whole, developing and implementing standard procedures (interface protocols) for the interaction of network hardware and software, after-sales service and the like. To solve these problems, a systematic approach is used that allows

to consider network design as a complex communication system from a single perspective.

In the process of creating communication networks, various design problems arise: building a new network, developing existing networks with the introduction of new points and communication channels, building new communication networks based on existing points and communication channels of a given primary network. In any case, when designing communication networks, it is necessary to build a mathematical model; select quality indicators and optimality criteria; choose the optimal structure and parameters of communication networks and its components.

Designing a communication network includes solving the following main tasks: determining the network topology; choice of capacity channels; definition of routing procedures; definition of control procedures. In general, the content of designing a communication network is in synthesis of the network structure for given flows, at a minimal cost, and in compliance with certain requirements for the characteristics of the network, could serve these flows. Of course, network design is carried out in several stages: making decisions regarding the topology of the subscriber network (total number of subscribers, number and location of access concentrates); subscriber network design; backbone network design.

Various approaches to solving the problems of network design are determined by the choice of a global quality indicator by which networks are evaluated, for example, the cost of building a network; network operating cost; network reliability and performance; average time delay in servicing applications of network subscribers. Of great importance for design is also the selection of a set of private performance indicators characterizing the consumer properties of the network. The network design process includes the selection of the network structure (structural synthesis) and the selection of numerical values of the network parameters (parametric synthesis).

To solve the problems of optimal design of a communication network, it is mandatory to build its mathematical model. The introduction of a network model based on the theory of queuing makes it possible to solve a number of topological design problems, as well as the problems of selecting the capacity of communication channels and the distribution of flows. However, the obtained optimal solutions to these problems can only be used for the output formulated in these problems. In a number of cases, the actual initial data for constructing a model of a communication network differ from those that were laid down in the formulation of the solvable design problem. Therefore, to solve the real problems of designing communication networks, many other methods of synthesis and analysis have been proposed. In particular, problems of structural synthesis of the system in the following formulations are of particular interest.

Task 1. On the set of possible structures of the system $Q = \{Q_j\},\ j = 1,...,J$, find the structure Q_{opt} that corresponds to the minimum cost criterion $\min\limits_{Q_j \in Q} C(Q_j)$, subject to the fulfillment of restrictions such as inequalities on other performance indicators.

The task in this formulation is typical for the design of switching centers and networks of the most mass types (urban and rural exchanges, zone centers and message and packet switching networks, channel and message concentrators, etc.).

Task 2. On the set of possible structures of the system, find a subset of structures ordered by increasing value indicator $Q^l = \{Q_l\},\ l = 1,...,L;\ C(Q_{l-1}) \le C(Q_l)$, for which $C(Q_l)$ may exceed the permissible level of value C_p no more than by value ΔC, that is $C(Q_l) - C_p \le \Delta C$.

The separation of such a variant of the problem into an independent task is due to the fact that the optimal variant of the system structure by the criterion $\min C(Q_i)$ may turn out to be such that it is difficult to implement. It is very difficult to take into account all the variety of connections between system parameters. Therefore, when designing large systems, it is possible to get not one best in the sense of $\min C(Q_i)$ a variant of the structure of the system, but several close in terms of efficiency options that can be considered as alternative.

Task 3. Synthesize the structure of the system that provides service of the incoming flow of maximum intensity $\max\limits_{Q_j \in Q} P(Q_j)$, subject to the implementation of restrictions such as inequality on other performance indicators.

The analysis of these and many other problems of designing communication networks shows the complexity of their strictly formal solution by analytical methods for a large number of variables characterizing the structure and parameters of the system. Given the above during the study of such systems, it is advisable to apply the methods of phased optimization, suboptimization of subsystems, traditional for the theory of complex systems, a combination of analytical and simulation modeling methods that complement each other and provide the opportunity to study the structure of the system with initial data corresponding to various degrees of adequacy of the model and the object of research.

Modern communication networks are built using a significant number of heterogeneous subsystems, switching nodes and transmission lines. Therefore, it is not always justified to assess the CN effectiveness as systems for the transmission and distribution of information by methods suitable for assessing the effectiveness of individual communication systems. So, for messaging systems, the main indicators of quality are the speed of information transfer and the probability of information transfer. In communication networks, such comprehensive performance indicators as the total number of nodes or subscriber end points, the total length of transmission lines (channels), the total amount of information transmitted cost, the actual load, the time of information delivery, the number of lost applications

and their delays come to the forefront availability factors for the main elements of the network, indicators of connectivity and survivability, indicators of the use of nodes and channels and the like. In any communication network, a change in the transmission rate of information, an increase in the time it takes to establish a subscriber's connection, an increase in the level of interference in channels and signal distortion, and hardware failure on other transmission processes will inevitably affect the delay of messages in the network. Therefore, the average message delay time is often considered as a comprehensive measure of effectiveness.

The difficulties that arise during the practical solution of the problems of designing communication networks due to the following main factors: the «curse of dimensionality» and the «curse of uncertainty». The first factor is determined by the fact that the number of nodes in the network can reach several thousand. The second factor is determined by the fact that there are no reliable a priori data on traffic – the amount of information for transmission over the network. In addition, in many cases it is not possible to submit in a strictly formalized form all the requirements for the designed network.

Therefore, when designing communication networks, there is a need to perform multivariate network calculations when changing the source data in wide ranges. In this case, one also has to carry out mathematical modeling of the network to verify the obtained design solutions and evaluate some network parameters that can't be calculated by analytical methods.

Therefore, let's consider some features and methods of constructing mathematical models of communication networks.

5.2 Mathematical models of the structure of communication networks

A communication network is a complex system of a higher hierarchical level than a separate communication system. The network structure, that is, its topology is determined by the combination of points (end and switching nodes) and communication channels (lines) that connect. The purpose of the network is to transmit messages from sources to consumers of information. A characteristic feature of a communication network is a significant number of sources and consumers of information, as well as possible message transfer routes. Therefore, it is important for the network to control messaging processes with optimal quality indicators. The communication network model is determined by the mathematical description of the network structure, as well as the processes of receipt of applications to end points and the processes of their servicing in the network. Services include processes for distributing information in switching nodes and processes

for delivering messages to consumers on specific routes. Moreover, for a large number of applications, as well as the limited physical capabilities of switching systems and communication channels (lines), there are various ways of servicing applications on switching nodes: with losses (when the application receives a denial of service), with waiting (when the application is waiting for line release or switching device), with limited waiting (when either the number of applications waiting or the waiting time is limited). Thus, for the mathematical description of communication networks, a different mathematical apparatus is used compared to the description of simple communication systems; in the mentioned structure, networks are often used to connect various nodes.

Let's consider the features of the mathematical description of the structure of the communication network using the network mathematical model. In this case, a graph $G = (X, A)$ is used as a model, where $X = \{x_i\}$ is the set of graph vertices that are mapped to network points (end points, switching nodes), and $A = A = \{a_{ij}\}$ is the set of graph edges that are mapped to lines, communication channels. In accordance with the fact that the communication channels can be unidirectional and bidirectional, the edges of the graph can be oriented and non-oriented. Thus, oriented, undirected, mixed graphs, and also a multigraph can be used as a model of a communication network. Network models are widely used in practice in the design of telecommunication systems, space and radio communication systems, broadcast networks, computer systems, transport networks.

Network analysis is playing an ever-increasing role, since with the help of graphs it is quite simple to build a model of not only communication networks, but also other complex systems. The expansion of the scope of using the network model is due to the fact that network analysis methods make it possible to: build a model of a complex system as a set of simple subsystems; draw up formal procedures to determine the qualitative and quantitative characteristics of the system; show the mechanism of interaction of system components in order to describe the latter in terms of its main characteristics; determine what data is needed to study the system.

When constructing models of network structure, it is convenient to use the algebraic image of graphs, determined by topological matrices and matrices of characteristics of graph edges (network branches).

The topological matrix that determines the structure of the network can be defined as an adjacency matrix and a structural matrix. The adjacency matrix (message) of a graph is a square matrix of size (the number of vertices of the graph). It will be defined as follows:

$$A = \left[a_{ij}\right] = \begin{cases} 1, & \text{if } G \text{ has edge } (i, j), \\ 0, & \text{if } G \text{ hasn t edge } (i, j). \end{cases} \qquad (5.1, a)$$

Elements of the main diagonal of a matrix $A = \left[a_{ij} \right]$ are of course assigned equal to zero $a_{ii} = 0$, unless in some vertices there are «loops». The matrix $A = \left[a_{ij} \right]$ for the opientated graph is asymmetric with respect to the main diagonal; it will be symmetric only for a non-orientated graph.

The structural matrix is used to simplify the recording of the network structure when special designations are assigned to the edges of the graph, for example $a, b, c.$ These designations are used as elements of a structural matrix. The structural matrix of a graph is a square matrix of size, which is defined as follows:

$$b_{ij} = \begin{cases} 1, \text{if } i = j, \\ 0, \text{if } G \text{ hasn't edge } (i, j), \\ x, \text{if edge } (i, j) \text{ in } G \text{ is at } i < j, \\ \bar{x}, \text{if edge } (i, j) \text{ in } G \text{ is at } i > j. \end{cases} \qquad (5.1, b)$$

In addition to the considered topological matrice, incidence matrices «vertex-arcs», «arcs-arcs» can be used.

The matrix of quantitative characteristics of the edges of the graph is used for various quantitative estimates of the network. At the same time, a certain weight is assigned to each edge of the graph – a number that characterizes any power of a given edge, for example, length, cost, capacity, channel capacity, information transfer time, reliability, and the like. The indicated characteristics of the edges of the graph are presented in the form of corresponding square matrices of size n – lengths, values. If G is an undirected graph, then all matrices are symmetric with respect to the main diagonal.

For example, to build a matrix of path lengths in a network $L = \left[l_{ij} \right]$, let's use the following rule:

$$l_{ij} = \begin{cases} 0, \text{if } i = j; \\ \infty, \text{if } G \text{ hasn't edge } (i, j); \\ \text{edge length } (i, j), \text{if edge } (i, j) \text{is in } G. \end{cases} \qquad (5.1, c)$$

The matrix of channel capacities of the edges is obtained according to the rule:

$$c_{ij} = \begin{cases} \infty, \text{if } i = j; \\ 0, \text{if } G \text{ hasn't edge } (i, j); \\ \text{number of edge channels } (i, j), \text{in edge } (i, j) \text{is in } G. \end{cases} \qquad (5.1, d)$$

Similarly, other matrices of characteristics of the edges of the graph are obtained. The indicated network characteristics can be used to solve various problems of synthesis and analysis of communication networks, in particular, to search

for optimal message transmission paths. Since the purpose of the communication network is to provide subscribers with connecting paths for transmitting messages in accordance with the address and specified quality indicators, it is therefore necessary to make the optimal choice of connecting paths. At the same time, such paths should be selected in order to ensure the most efficient use of network equipment, or to ensure the minimum possible path length and the number of transit sections in the paths, or to provide the necessary number of channels in the paths or the maximum transmission speed.

So, when solving the problems of designing communication networks, it becomes necessary to search for the many paths that exist between a given pair of communication nodes (vertices of the graph). The methods for finding multiple paths in a network can be divided into two classes: matrix and network. Matrix methods are based on the transformation of various matrices – topological or matrices of characteristics of the edges of the graph, and network methods – on assigning designations to the vertices of the graph are called marks (or indices). Network methods for determining the set of paths between given network nodes is the graphic equivalent of matrix methods. The determination of the set of paths is based on the construction of a path tree from a fixed vertex-leak (tree root) to the remaining vertices-sinks of the graph.

5.3 Mathematical models of application flows in a communication network

In addition to the structure, the mathematical model of the communication network should describe the flows of applications and their servicing in the network. These processes are stochastic. Let's consider their mathematical models, which are based on the theory of random processes and the theory of mass service [1, 92, 94, 99].

The main characteristics of random application flows. A random flow of applications is considered as a sequence of random variables, which can be set in various ways, including in the form of:

- a sequence of random time instants of the application $t_i,\ i = 1,2,\ldots,n$;
- a sequence of random time intervals between applications $\Delta t_i = t_i - t_{i-1}$, $i = 1,2,\ldots,n$;
- a sequence of random numbers $k_i,\ i = 1,2,\ldots,n$ that determine the number of applications at given time intervals $\left[t_0, t_i\right),\ i = 1,2,\ldots,n$.

In the first two methods of specifying, the application flow is considered as a random point process, and in the third – as a random integer process $k(t)$ with an initial value $k(t_0) = 0$. The probabilistic description of such random processes

uses the following characteristics: the distribution law or the corresponding probability density of the time instants of applications or time intervals between applications, as well as the law of distribution of the number of applications at given time intervals.

Call flows are classified according to properties such as uniformity, stationarity, ordinariness and aftereffect. Let's consider these properties.

I. In a heterogeneous flow, each call has two or more characteristics. For example, calls from subscribers of the telephone network are characterized by the moments of their arrival, the directions of the establishment of connections, the duration of the service, and other characteristics.

A homogeneous flow is characterized only by the moments of incoming calls.

In practice, call flows are generally heterogeneous. However, the theory considers homogeneous flows if there are no special reservations.

II. The stationary property of the flow means that its probabilistic characteristics do not change over time. The call flow is called stationary if, for any n joint probability $P\left\{K\left(t,t+\tau_i\right)=k_i,\ i=\overline{1,n}\right\}$ of calls arriving at time intervals $[t,t+\tau_1)$, $[t,t+\tau_2),\ldots,[t,t+\tau_n)$ does not depend on the initial moment of time t.

For a stationary flow, its parameter and intensity are constant: $\lambda(t)=\lambda$, $\chi(t)=\chi$.

A flow that does not have the stationary property is called non-stationary.

III. A call flow is called ordinary if it is not possible to simultaneously receive two or more calls, that is, the following condition is fulfilled:

$$\lim_{\tau\to 0}\frac{P_{k\geq 2}\left(t,t+\tau\right)}{\tau}=0. \tag{5.2}$$

A flow that does not have this property is called extraordinary.

An example of an ordinary flow is a call flow arriving at a PBX from a large group of subscribers, and telegram flows to several addresses are examples of extraordinary flows.

IV. A call flow is called a flow without aftereffect, if for any the probability of receipt $k_i,\ \ i=1,2,\ldots,n$ of calls is compatible for the corresponding intervals $\left[t,t_i\right),\ i=1,2,\ldots,n$:

$$P\left\{K\left(t,t_i\right)=k_i,i=\overline{1,n}\right\} \tag{5.3}$$

independent of the process of incoming calls to the initial moment of time t.

In other words, the absence of a flow aftereffect means the independence of the flow of a random call flow after a certain point in time t from its flow to this point.

An example of a flow without aftereffect can be a flow of telephone calls coming from a large group of sources. This is due to the fact that only a small part (10–20 %) of the subscriber group is simultaneously involved in the connections.

Flows that do not have the aftereffect property are called aftereffect flows. Call flows from paired telephones and call flows from small groups of subscribers are examples of flows with aftereffect.

Let's consider the main classes of flows.

Simple flows. The simple flow is the stationary ordinary flow without aftereffect. This flow is the main model in the theory of teletraffic.

For the simplest flow of incoming k calls for a time interval of duration t in the simplest flow, it obeys the Poisson distribution with probabilities:

$$p_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad k = 0,1,2,..., \tag{5.4}$$

where $p_k(t)$ is the probability that t calls will be received over a time interval t; $\lambda > 0$ is a parameter of the simplest flow.

The simplest flow is also called Poisson flow.

In the simplest flow, the time intervals between adjacent calls are statistically independent random variables with an exponential distribution law. Their distribution function:

$$F(\Delta t) = \begin{cases} 1 - e^{-\lambda \Delta t}, & \Delta t > 0, \\ 0, & \Delta t \le 0. \end{cases} \tag{5.5}$$

The probability density of the exponential distribution is described by the expression:

$$p(\Delta t) = \lambda \cdot e^{-\lambda \Delta t}, \quad \Delta t \ge 0. \tag{5.6}$$

The property of exponential distribution by another formulation of the property of the absence of aftereffect.

Non-stationary Poisson call flow. A non-stationary Poisson flow (or a simple flow with a variable parameter) is an ordinary flow without aftereffect, for which the parameter $\lambda(t)$ depends on time t.

For this flow, the probability of receipt of k applications on the time interval is determined by the formula:

$$p_k(t) = \frac{\left[\int_0^t \lambda(t')\mathrm{d}t'\right]^k}{k!} \cdot \exp\left\{-\int_0^t \lambda(t')\mathrm{d}t'\right\}, \quad k = 0,1,2,\ldots. \tag{5.7}$$

Flows with limited aftereffect. A flow with limited aftereffect is a flow in which the sequence of time intervals between adjacent calls $\Delta T_1, \Delta T_2, ..., \Delta T_n, ...$ is a sequence of independent random variables. Such a call flow is described by a sequence of distribution functions of intervals between calls:

$$F_k(\Delta t) = P\{\Delta T_k < \Delta t\}.$$

Special cases of a flow with limited aftereffect are:

1) recurrent flow characterized by equally distributed intervals between calls:

$$F_1(\Delta t) = F_2(\Delta t) = ... = F(\Delta t);$$

2) recurrent flow with delay, for which:

$$F_2(\Delta t) = F_3(\Delta t) = ... = F(\Delta t), \quad F_1(\Delta t) \neq F(\Delta t).$$

In reliability theory, a recurrent flow is called a recovery process, and a delayed recurrent flow is a common recovery process.

Palm flows. The stationary ordinary recurrent flow with delay is called the Palm flow. Palm flows describe the challenges lost in the IDS.

The simplest flow is a case of the Palm flow, in which all time intervals between calls, including the first, have an exponential distribution.

In the theory of random flows, it can be seen that for the Palm flow, random variables ΔT_1 have a continuous distribution with a probability density:

$$p_1(\Delta t) = \frac{1 - F(\Delta t)}{\overline{\Delta T}}, \tag{5.8}$$

where

$$\overline{\Delta T} = M[\Delta T_k], \quad k \neq 1. \tag{5.9}$$

Erlang flow. The *m*-th order Erlang flow is the Palm flow, in which the intervals between adjacent calls are statistically independent random variables distributed according to the *m*-th order Erlang law, that is, they have a continuous distribution with probability density:

$$p(\Delta t) = \begin{cases} \dfrac{\lambda^{m+1} \Delta t^m}{m!} e^{-\lambda \cdot \Delta t}, & \text{if } \Delta t \geq 0; \\ 0, & \text{if } \Delta t < 0, \end{cases} \tag{5.10}$$

where $\lambda > 0$ is the Erlang distribution parameter.

The Erlang flow belongs to the class of flows with limited aftereffect, for which the time intervals between adjacent calls are statically independent random variables with arbitrary and generally different distribution laws.

The m-th order Erlang flow can be obtained from the simplest flow with a parameter λ using the regular sieving operation. The essence of this operation is that each $m+1$ application is saved in a simple flow, and the rest are eliminated.

When applied to the simplest flow with the thinning operation parameter λ (removing part of the orders from it), a recurrent flow with restoration is obtained. If at the same time m applications are lost in a row, and only each $(m+1)$ remains, then the thinning flow $\lambda/(m+1)$ has a parameter and probability density for time intervals between applications:

$$p(\Delta t) = \frac{\lambda^{m+1}\Delta t^m}{m!}\exp(-\lambda\Delta t). \tag{5.11}$$

Such a distribution is called the m-th order Erlang distribution, and the corresponding flows are called Erlang. With the help of the Erlang distribution, it is possible to describe a wide class of flows – from the simplest (at $m=0$) to the determinate with a constant duration of intervals between applications (at $m\to\infty$).

Erlang distribution coincides with the exponential distribution at $m=0$.

The waiting and variance of the time interval between adjacent calls in the m-th order Erlang flow are equal:

$$\mathrm{M}[\Delta T] = \frac{m+1}{\lambda},\quad D[\Delta T] = \frac{m+1}{\lambda^2}. \tag{5.12}$$

Erlang flows in different order create flows with varying degrees of randomness: from the simplest at m to the deterministic at $m=\infty$.

The Erlang flow model is used to describe actions in information distribution systems when the call flow itself is divided into $m+1$ directions according to a regular thinning operation.

Flows with a simple aftereffect. A flow with a simple aftereffect is an ordinary flow, for which the flow parameter $\lambda(t)$ depends only on the IDS state $S(t)$ at a point in time t. A more rigorous definition: a flow with a simple aftereffect is an ordinary flow, for which at any time there is a finite parameter of the flow depending on the state of the system $S(t)$:

$$\lambda_{s(t)} = \lim_{\tau\to 0}\frac{P_{i\geq 1}(t,t+\tau/S(t))}{\tau}. \tag{5.13}$$

Special cases of flows with a simple aftereffect include a symmetric flow, a primitive flow, and a flow with repeated calls.

A symmetric flow is a flow with a simple aftereffect, the parameter of which at any moment of time depends only on the macrostate of the system:

$$\lambda(t) = \lambda_i, \tag{5.14}$$

where i is the number of occupied devices (outputs) of the system at a time t.

A primitive flow is a symmetric flow in which the parameter λ_i is directly proportional to the number of free sources at a given time:

$$\lambda_i = (n - i) \cdot \alpha, \tag{5.15}$$

where n is the total number of call sources; i is the number of sources served at a time t; α is source parameter in the free state.

For sources in a free state, an exponential distribution of intervals between adjacent calls is usually assumed.

In telephony, a primitive flow is called a call flow from a limited number of sources.

Such a flow is non-stationary and is a flow with aftereffect, since the probability of calls depends on the number of calls i received up to this point.

With increase n and decrease α, the aftereffect of the flow decreases.

In the extreme case, $n \to \infty$, $\alpha \to 0$ so that $\alpha \cdot n = \lambda = \text{const}$, the model of the primitive flow goes into the model of the simplest call flow.

The repeat call flow consists of primary calls and repeated calls arriving at the IDS again if the initial application has not been served. In the case of the simplest primary call flow, the parameter of such a flow:

$$\lambda_j = \lambda + j\beta, \tag{5.16}$$

where j is the number of call sources; β is parameter of one source of repeated calls; λ is parameter of the simplest flow of primary calls.

Release flow. The release flow is called the sequence of moments when the call service ends.

In the general case, the properties of the release flow depend on the properties of the input flow, the number of servicing devices and the law of distribution of the duration of servicing.

When servicing an incoming call flow without losses, in the case of a constant duration of service, the properties of the release flow coincide with the properties of the input flow.

5.4 Mathematical models of service processes in a communication network

5.4.1 Call service features and disciplines

Calls from subscriber devices occupy IDS devices for a certain time. There are mathematical models that correspond to fixed and random service times T_s.

The fixed value of the duration of one seizure T_s provides that for each call the duration of its service is determined. In particular, T_s the time may be constant if all calls are the same in duration of service. In telephony, a model of constant service time is used to describe the operation of control devices when establishing a connection.

The random service time T_s model is a random variable described by a probabilistic distribution law. A simple and common model of random service duration is a random variable with an exponential distribution. The distribution function of the exponential law has the form:

$$F(t) = P\{T_s < t\} = \begin{cases} 1 - e^{\mu \cdot t}, & t > 0; \\ 0, & t \le 0, \end{cases} \tag{5.17}$$

where

$$\mu = \frac{1}{M[T_s]}$$

isservice parameter.

In reliability theory, a function $\Phi(t) = 1 - F(t)$ is called a reliability function. It characterizes the probability that an element will not fail earlier than in time t.

A random variable model with an exponential distribution law is used to describe the duration of calls in telephone networks.

Load concept. When servicing call flows in IDS, each of them occupies a servicing device for a certain time interval. In the theory of teletraffic, the total time that the devices take up calls is called the load.

Distinguish between the load arriving is serviced and lost.

The load, the time interval is serviced $[t_1, t_2)$, is called the total occupation time of all IDS devices:

$$Y_o(t_1, t_2) = \sum_{i=1}^{v} \tau_i, \tag{5.18}$$

where v is the total number of IDS devices; τ_i is the occupation time of the i-th device per interval $[t_1, t_2)$.

The unit of load is one hour-seizure (1 hour-seiz.). One hour-seizure is such a load that it can be serviced by one device for an hour with continuous use of this device.

The load $Y(t_1,t_2)$ arriving for the time interval, $[t_1,t_2)$ is called the load, can be served by an ideal switching system if one of the service devices would be immediately provided to each call.

Lost load is that part of the load that arrived and was not served by the IDS.

$$Y_{lost}(t_1,t_2) = Y(t_1,t_2) - Y_o(t_1,t_2). \tag{5.19}$$

The mathematical waiting of the load per unit of time (usually in one hour) is called the load intensity. Let's affects the load by the symbol y. At the same time, it is possible to talk about the intensity of the loads that are received, maintained and lost.

Erlang is accepted as a unit of measurement of load. One Erlang is a load of one hour-seizure at 1 hour.

If the IDSs receive a stationary call flow, then the intensity of the served load, expressed in Erlang, is quantitatively equal to the average number of simultaneously occupied outputs of the system serving this load.

The intensity of the load created by a simple call flow is quantitatively equal to the mathematical waiting of the number of calls arriving in a time equal to the average duration of one seizure:

$$y = M[T_s]\cdot\chi = \frac{\lambda}{\mu}, \tag{5.20}$$

where χ and are the intensity and parameter of the call flow, respectively (for the simplest flow $\chi = \lambda$); $M[T_s]$ is average call service time; μ is service intensity ($\mu = 1/M[T_s]$).

The intensity of the load varies and depends on the season, day of the week, time of day. To ensure a satisfactory quality of customer service at any time of the day, the calculation of IDS equipment must be performed based on the intensity of the load at the hour when it is maximum. This hour is called the hour of highest load – HHL.

The hour of highest load is a continuous 60-minute interval during which the average load is at its maximum. It is recommended to measure the load on the working days of two consecutive weeks twice a year in the month of the highest load.

Call service disciplines. All call service disciplines can be divided into lossless service disciplines and loss service disciplines.

A lossless service discipline is a discipline in which calls arriving immediately are served. The loss service discipline – if calls arriving receive a denial of service or their service is delayed for a certain time.

For economic reasons, IDSs are usually designed with losses.

Distinguish between explicit and conditional losses. Discipline of service with obvious losses is a discipline in which a call, having received a denial of service, leaves the system and does not affect it in any way in the future. A conditional loss service discipline is one in which calls are not lost, but are served with the waiting, or after they are repeated.

Service quality characteristics. To assess the quality of call service with obvious losses, one of three types of losses is used:

1) call loss P_c;
2) time loss P_t;
3) load losses P_{load}.

Losses are random variables. In calculations, they are usually operated on with their average values, that is, their mathematical expecations. These averages are called loss probabilities.

The probability of losing calls can be determined by the formula:

$$p_c = \frac{M[n_l]}{M[n]}, \tag{5.21}$$

where n_l is the number of calls lost during the considered period of time; n is total number of calls.

The probability of call loss can also be determined through the ratio:

$$p_c(0,t) = \frac{\Lambda_l(0,t)}{\Lambda(0,t)}, \tag{5.22}$$

where in the numerator is the leading function of the flow of lost calls, and in the denominator is the leading function of the flow of incoming calls.

For stationary flows:

$$p_c = \frac{\chi_l}{\chi},$$

where χ_l is the lost call flow rate; χ is intensity of the flow of income calls.

In other words, p_c is part of the calls whose service does not end with the service.

The probability of time loss p_t is part of the time during which calls are blocked, since all devices are busy with maintenance, i. e.:

$$p_t = P\{i = v\},$$

where v is the total number of servicing devices; i is the number of occupied devices.

The probability of load losses p_{load} is the ratio of the mathematical waiting of the load, which is lost during the considered period of time to the mathematical waiting of the load arriving for the same period of time:

$$p_{load}(0,t) = \frac{M\left[Y_l(0,t)\right]}{M\left[Y(0,t)\right]}. \tag{5.23}$$

For stationary flows:

$$p_{load} = \frac{y_l}{y} = \frac{y - y_o}{y}, \tag{5.24}$$

where y_l is the intensity of the load is lost; y is load intensity of the incoming information.

The amount of loss may be expressed in fractions of a unit as a percentage or in ppm.

The service discipline with conditional losses assumes that calls received at the time all devices are busy do not disappear, but are delayed in service.

By the method of servicing delayed calls, it is distinguished:

1) service with waiting;
2) service with repeated calls.

In the first case, calls are queued and serviced as instruments become available. In the second case, delayed calls are repeated at certain intervals until the necessary connection is received (active queue).

To assess the quality of call service in the discipline with waiting, such characteristics are used.

1. The probability of waiting for calls arriving:

$$P_w = P\left(T_w > 0\right) = \frac{M\left[k_d\right]}{\left[k\right]},$$

where k_d is the number of delayed calls, and k is the number of calls received.

For stationary flows:

$$P_w = \frac{\chi_d}{\chi}.$$

2. The probability of waiting for any call is more than some time $P(T_w > t)$.
3. The probability of waiting for a delayed call for more than a while $P_d(T_w > t)$.
4. Average waiting time relative to calls $\overline{T}_w = M[T_w]$.
5. Average waiting time for delayed calls:

$$\overline{T}_{w.d.} = M[T_{w.d.}].$$

6. The average length of the queue $\overline{r}$.

The following characteristics are used to evaluate the quality of IDS service with repeated calls:

1) the probability of loss of the primary call – p;
2) the probability of loss of a repeated call – p_r;
3) the average number of repeated calls per primary call $\overline{c}$.

IDS capacity. One of the most important characteristics of IDS is capacity.

By SIDS capacity η, let's mean the maximum intensity of the served load at which the losses do not exceed the permissible ones.

In teletraffic theory, specific capacity is often used as the ratio of IDS capacity to the number of devices: $\eta_1 = \eta / v$.

5.4.2 Call service models in fully accessible IDS

Let's assume that IDS with v devices is fully accessible and serves calls that form a symmetrical flow with a simple aftereffect with a parameter λ_i, $i = \overline{0,n}$. The duration of the call service by the IDS device is a random variable distributed exponentially, that is, its distribution function has the form (5.17) and is characterized by a service parameter μ. It is necessary to determine the probabilities of IDS states p_i, $i = \overline{0,n}$, which differ in the number of occupied devices of the system or the number of calls in the queue.

The use of the mathematical apparatus of Markov processes. Let's denote by $S(t)$ the number of calls that are in the system at a time t. It is a random variable that varies over time. Therefore $S(t)$ is a random process with a finite set of values $S(t) = 0,1,2,...,n$.

Thus, the process determines the IDS state and takes $(n+1)$-th value. It can be shown that the $S(t)$ process is Markov.

Markov is a random process in which for any moment in time t the probability of any value in the future depends only on the value of the process $S(t)$ at the moment and does not depend on previous values of this process [5].

The process $S(t)$ is Markov in that the moments of the arrival of new calls are determined by the flow of incoming calls and do not depend on the state of the

system at times that precede the time t. In addition, the moments t of the end of calls (a property of the exponential distribution of the duration of service) do not depend on the flow of the process until the moment.

During the presentation of the theory of random processes with discrete states, oriented graphs of process states (system states) are used. On these graphs, the vertices are represented by circles in which the states of the system fit, and the arc drawn from the vertex S_i to the vertex S_j means the possibility of transition from one state to another.

It is generally accepted that the transition of a system from state S_i to state S_j is carried out under the influence of a Poisson flow with intensity $a_{ij}(t)$. Then the probability of transition from state S_i to state S_j in a small time interval:

$$p_{ij}(t,t+\tau) = a_{ij}(t)\cdot\tau + o(\tau), \tag{5.25}$$

where $o(\tau)$ is a quantity of a lower order in comparison with τ.

A Markov random process with discrete states and continuous time is called homogeneous if the probability of a transition from state S_i to state S_j in time τ does not depend on at what point in time t the system was in state S_i, but depends only on the quantity τ:

$$p_{ij}(t,t+\tau) = p_{ij}(\tau). \tag{5.26}$$

For a homogeneous Markov process:

$$p_{ij}(t,t+\tau) = a_{ij}\cdot\tau + o(\tau).$$

In this case, oriented graphs of system states can be used.

Let's consider a system that has $n+1$ possible states $S_0, S_1, ..., S_i, ..., S_n$. Let $p_i(t)$ be the probability that the system is in S_i state at a time t.

If the transition probabilities $p_{ij}(t,t+\tau)$ satisfy relation (5.25), then the probabilities of the states of the Markov process obey the Kolmogorov system of differential equations:

$$\frac{dp_i(t)}{dt} = \sum_{\substack{j=0\\ j\neq i}}^{n} p_j(t)\cdot a_{ji}(t) - p_i(t)\cdot\sum_{\substack{j=0\\ j\neq i}}^{n} a_{ij}(t), \quad i = 0,...,n. \tag{5.27}$$

The Kolmogorov equation consists of the following rule: the derivative of the probability of any state of the system is equal to the sum of the probability flows that bring the system into this state, minus the sum of all probability flows that take the system out of this state.

In order for the process that occurs in the system to be ergodic, it is necessary that its state graph be strongly connected and the transition probabilities satisfy the uniformity condition (5.26).

To solve the problems of servicing calls of a symmetric flow of completely accessible IDS, it is convenient to use a special case of Markov processes – the process of death and birth.

The process of death and birth is called such a Markov process with continuous time t, having a finite or countable set of states, in each of which an infinitesimal time interval $[t, t+\tau)$ with probabilities greater than zero, direct transitions only to neighboring states are possible. In other words, a transition from a state i is possible only to a states $i-1$ or $i+1$, or the process retains a state.

If a completely accessible IDS receives an ordinary call flow, then the call service process is a process of birth and death.

The birth process in this case is identified with the process of occupying the devices of the system, and the death process with the process of releasing the devices. The parameters of occupation flows and release flows are denoted by λ_i and j, $i = 1,...,n$, respectively.

5.4.3 Explicit loss service call service models

Call service in M/M/v/L IDS type. Let's formulate the problem statement.

The input of a fully accessible IDS with devices receives a simple call flow, the parameter of which depends on the IDS state: $\lambda_i = \lambda$, $i = \overline{0, v}$.

The duration of the call service by the IDS device is a random variable distributed exponentially, that is, its distribution function has the form (5.17) and is characterized by a service parameter μ.

Under these conditions, the probabilities of IDS states are determined by the expression:

$$p_i = \frac{\frac{y^i}{i!}}{\sum_{k=0}^{v} \frac{y^k}{k!}} = E_{i,v}(y), \quad i = \overline{0, v}, \tag{5.28}$$

where $y = \lambda / \mu$ is the intensity of the incoming load.

The sequence of probabilities $i = \overline{0, v}$, calculated according to (5.28), is called the Erlang distribution. For Erlang distributions, a fair recurrence relation:

$$p_i = p_{i-1} \cdot \frac{y}{i}, \quad i = \overline{1, v}, \tag{5.29}$$

where

$$p_0 = \frac{1}{\sum_{k=0}^{v} \frac{y^k}{k!}}. \tag{5.30}$$

Comparing formula (5.29) with a similar relation for the Poisson distribution, it follows that, up to a constant factor, the Erlang distribution coincides with the Poisson distribution. Therefore, the Erlang distribution is also called the truncated Poisson distribution.

Probability of loss. The probability of loss in time p_t is the period of time during which all IDS devices are occupied and, according to (5.28), is equal to:

$$p_t = p_v = \frac{\frac{y^v}{v!}}{\sum_{k=0}^{v} \frac{y^k}{k!}}. \tag{5.31}$$

The probability of loss on calls is defined as the ratio of the intensity of the flow of lost calls to the intensity of the flow of incoming calls:

$$p_c = \frac{\chi_l}{\chi}, \tag{5.32}$$

where

$$\chi = \sum_{i=0}^{v} \lambda_i \cdot p_i; \quad \chi_l = \lambda_v \cdot p_v.$$

In this case $\chi = \sum_{i=0}^{v} \lambda \cdot p_i = \lambda; \quad \chi_l = \lambda \cdot p_v.$

Therefore, the probability of loss on calls $p_c = p_t = p_v$.

The probability of load loss is defined as the ratio of the intensity of the secondary load to the intensity of the incoming load:

$$p_{lost} = \frac{y_{lost}}{y} = \frac{y - y_o}{y}. \tag{5.33}$$

Here the intensity of the served load:

$$y_o = \sum_{i=0}^{v} i \cdot p_i.$$

Given the type of Erlang distribution, let's obtain:

$$y_o = y \cdot [1 - p_v]. \tag{5.34}$$

Therefore, when servicing calls of the simplest IDS flow, the probabilities of losses in time, calls and load are equal to each other and equal to the probability that the IDS is in a state v:

$$p_t = p_c = p_{load} = E_v(y) = \frac{\frac{y^v}{v!}}{\sum_{k=0}^{v} \frac{y^k}{k!}}. \tag{5.35}$$

This formula for losses in fully accessible IDS is called Erlang first formula.

Erlang first formula is tabulated. With the modern development of computer technology, the value of the function $E_v(y)$ can be calculated using computer programs Mathcad, Matlab, etc. Moreover, with a large number of devices ($v \geq 100$), it is advisable to use the connection between the Erlang distribution and the Poisson distribution.

An analysis of Erlang formula shows that, with a fixed quality of service, the average use of one device (capacity of one device) increases with the number of devices. In telephony, this property is called the «advantage of large bundles» of lines serving calls.

The load of a fully accessible IDS served by each device during the ordered occupation of free devices, when each call is served by a free device with the lowest number, is equal to:

$$\eta_i = y_o(i) - y_o(i-1) = y\left[1 - E_i(y)\right] - y\left[1 - E_{i-1}(y)\right] =$$
$$= y\left[E_{i-1}(y) - E_i(y)\right]. \tag{5.36}$$

It is necessary to pay attention to the high use of the first device, equal to:

$$\eta_1 = \frac{y}{1+y}. \tag{5.37}$$

It is easy to show that the highest load is served by the first device. And then with an increase in the number of the device the served load comes. From a physical point of view, this is due to the fact that for each subsequent device loads of lower intensity is supplied than for the previous one. In addition, loads created by Palm flows, which are characterized by greater non-uniformity of intervals between calls than in simple flows, are supplied to the second and subsequent devices. Moreover, the larger the number of the device, the higher the uneven flow.

Call servicing tasks in Mi/M/v/L IDS type are formulated in the same way as in the previous case, with the difference that a simple call flow arrives at the system input, is a special case of a symmetric flow with a simple aftereffect. Its parameter:

$$\lambda_i = (n-i)\cdot\alpha, \quad 0 \le i \le v, \tag{5.38}$$

where α is the call flow parameter from one free source; n is number of call sources.

After substituting (5.38) into the expression for the probabilities of IDS states, and also because the parameter of the release flow $v_i = \mu \cdot i$, let's obtain the expression:

$$v = d + (y_o - y_d)/\sqrt[d]{p}, \quad i = 0,1,...,v. \tag{5.39}$$

The sequence of probabilities p_i, $i = \overline{0,v}$, calculated according to (5.39), is called the Engset distribution.

If the number of call sources $n \to \infty$, and the source parameter in the free state $\alpha \to 0$ so $n \cdot \alpha = \lambda$ that something:

$$\lim_{\substack{n\to\infty \\ \alpha\to 0}} C_n^i \cdot \alpha^i = \frac{\lambda^i}{i!}.$$

Therefore, the Engset distribution (5.39) coincides with the Erlang distribution (5.28).

The Engset distribution is called the truncated binomial distribution. The Engseth distribution coincides with the binomial distribution at $n = v$.

5.4.4 Waiting service call models

It is assumed that IDS with v devices is fully accessible; it serves the simplest flow of applications with a parameter λ.

In this case, the discipline of service with waiting in the $M/M/v/W$ system is used. Applications can form a queue of unlimited length. Applications that are pending are served in turn. The duration of the device is considered a random variable with an exponential distribution law and a service intensity parameter μ.

Let's denote the IDS state by $S(t)$. Moreover, if $S(t) = i$, where $i \le v$, then all applications are serviced. If $i = v + r$, then v applications are being serviced, while other $r = i - v$ calls are in the queue.

In compliance with the conditions specified in the given statement of the problem, the IDS state can be described by the process of death and birth with the parameter of the flow of incoming requests (birth flow) $\lambda_i = \lambda$ and the parameter of the release flow (death flow):

$$\nu_i = \begin{cases} \beta \cdot i, \ 0 \le i \le v; \\ \beta \cdot v, \ i > v. \end{cases} \tag{5.40}$$

After substituting expression (5.40) into the basic system of formulas of the theory of teletraffic, let's obtain:

$$p_i = \begin{cases} \frac{y^i}{i!} p_0, \ 0 \le i \le v; \\ \frac{y^v}{v!} (\frac{y}{v})^{i-v} p_0, \ i > v, \end{cases} \tag{5.41}$$

where

$$p_0 = \frac{1}{\sum\limits_{i=0}^{v} \frac{y^i}{i!} + \frac{y^v}{v!} \sum\limits_{i=v+1}^{\infty} \left(\frac{y}{v}\right)^{i-v}}, \quad y = \frac{\lambda}{\beta}. \tag{5.42}$$

After simple transformations in (5.41) and (5.42), let's obtain expressions for $M/M/v/W$ IDS states:

$$p_i = \begin{cases} \frac{E_{i,v}(y)}{1 + E_v(y) \frac{y}{v - y}}, \ 0 \le i \le v; \\ \frac{E_v(y) \cdot (\frac{y}{v})^{i-v}}{1 + E_v(y) \frac{y}{v - y}}, \ i > v. \end{cases} \tag{5.43}$$

This shows that $p_i < E_{i,v}(y)$ for all $0 \le i \le v$, that is, for systems with an waiting, the time spent in states when calls are immediately serviced is less than for systems with losses.

The recurrence relations for state probabilities for $M/M/v/W$ systems have the form:

$$p_i = p_{i-1} \cdot \left(\frac{y}{i}\right), \ i \le v; \tag{5.44}$$

$$p_i = p_{i-1} \cdot \left(\frac{y}{v}\right), \ i > v.$$

In the IDS, serving applications for discipline with waiting, time losses are equal to the probability that the application received will not be served immediately, is determined by the expression:

$$p_t = \frac{E_v(y)}{1 - \frac{y}{v}(1 - E_v(y))} = D_v(y), \tag{5.45}$$

what is called the second Erlang formula. This formula is widely used in telephony: it determines the likelihood that a call arriving on a bundle of lines will not catch a single free line and will be put on the service queue. This formula is often called the C-Erlang formula.

Erlang second formula is tabulated. This formula allows to determine the third parameter p, y, v by any two of the three parameters.

Since the denominator in the second Erlang formula is less than unity, then in systems with the waiting of time loss is greater than in systems with obvious losses.

For given losses p_t, the input load in the IDS with waiting should be less than in the IDS with losses.

Distribution of waiting time in case of FIFO queue discipline. In this case, the probability of waiting for the start of service in time t is determined by the formula:

$$P(\gamma > t) = P(\gamma > 0) \cdot e^{-\mu \cdot (v - y) \cdot t}. \tag{5.46}$$

Moreover, the distribution function is $F(t) = 1 - P(\gamma > t)$.

For single line system:

$$P(\gamma > t) = y \cdot e^{-\mu \cdot (1 - y) \cdot t}. \tag{5.47}$$

The quality of service of the incoming request flow in the IDS with waiting is also characterized by the likelihood that the waiting time for the start of service for the delayed application will be more than t:

$$P_a(\gamma > t) = \frac{P(\gamma > t)}{P(\gamma > 0)} = e^{-\mu \cdot (v - y) \cdot t}. \tag{5.48}$$

The average waiting time for the start of service applications:

$$\overline{\gamma} = \int_0^\infty t \mathrm{d}F(t) = \frac{P(\gamma > 0)}{\mu(v - y)}. \tag{5.49}$$

For queued applications, the average waiting time for the start of service:

$$\overline{\gamma}_a = \frac{1}{\mu(v - y)}. \tag{5.50}$$

Average queue length:

$$\overline{r} = P(\gamma > 0) \cdot \frac{y}{v - y} \tag{5.51}$$

or

$$\overline{r} = \overline{\gamma} \cdot \lambda. \tag{5.52}$$

Relation (5.52) is called the Little's formula and claims that the average queue length is equal to the product of the average waiting time and the flow of applications.

Little's formula. Little's formula establishes the relationship between the average queue length in the system, the input flow rate, and the average residence time of applications in the system. It is valid for any system with waiting that it is in a constant state.

The last relation means that the average number of applications $\overline{N}$ in the system is equal to the product of the intensity of receipt of applications λ in the system by the average time $\overline{T}$ they spend in the system:

$$\overline{N} = \lambda \overline{T}. \tag{5.53}$$

Interestingly, as IDS, it is possible only consider the queue of applications in the buffer. Then Little's formula takes the form (5.52) and has a different meaning – the average queue length is the product of the intensity of the flow of incoming requests by the average waiting time in the queue.

5.4.5 Service models of the simplest flow with an arbitrary distribution law of the duration of a seizure with waiting

Pollaszek and Khinchin investigated a single-line system with a waiting, which receives applications for the simplest flow with a parameter λ and an arbitrary distribution of the seizure duration in the $M/G/1/W$ system). Applications are served in turn (FIFO queue discipline).

For IDS with one device, the probability of conditional losses is $P(\gamma > 0) = 1 - p_0$. On the other hand $y = y_o = 1 - p_0$. It follows from this that in the $M/G/1/W$ system the probability of conditional losses:

$$P(\gamma > 0) = y, \quad 0 \le y < 1. \tag{5.54}$$

The Pollaszek – Khinchin formula for the average number of applications in the system looks:

$$\bar{q} = y + \frac{y^2}{2(1-y)}\left[1 + \left(\frac{\sigma(t)}{\bar{t}}\right)^2\right], \tag{5.55}$$

where $\bar{t}$ is the average class time (serving one application); $\sigma(t)$ is the average deviation of the seizure duration; y is incoming load intensity ($y = \lambda \cdot \bar{t}$).

It follows from the Pollaszek – Khinchin formula that in a single-line system, the average time a call spent in a queue with constant seizure duration is half as much. With an increase in the number of devices, this ratio decreases, but it always remains large 1.

5.5 An example of building a mathematical model of a communication network in the optimization of channel capacity

Let's consider the features of constructing a mathematical model when solving the problem of choosing the optimal capacity of communication channels in order to minimize the average delay time of messages in communication network (CN) with message switching. A solution to this problem was proposed by L. Kleinrock. This task can be considered as a conditional optimization problem with two quality indicators in the form of the average delay time of messages in the communication network and the rent for using the network. The problem was solved by minimizing one of the quality indicators – the average message delay time, provided that the second indicator (rent) takes a fixed value. Let's dwell on the main stages of constructing a mathematical solution to this optimization problem.

The process of transmitting messages in a communication network generates a system of random (in time and space) data flows. The quality of the network's functioning is determined primarily by the average message delay time, characterized by the time interval from the moment the sender connects to the network until the data message is sent to the recipient. Communication occurs at random times, their duration in time is also random variables. To transmit messages in a communication network, CC with certain capacities are used, the value of which affects the delay time of messages in the network. The optimality criterion for such a network is the minimum of the average message delay time. The optimization task is in selection of such channel capacities at which the minimum value of the average delay time of messages in the network is achieved with the given restrictions on the cost of the network (rent for using the network).

Let's consider the mathematical model of the network in terms of its use in solving such an optimization problem. The communication network model with message (packet) switching has V communication channels (CC) and W switching

nodes (*SN*). Let's assume that there are no errors and hardware failures in the CC, and the capacity of the *i*-th channel is c_i (bit/s). Let the duration of message processing in all SNs be constant and equal t_p. As a rule, the value t_p is much smaller relative to the duration of the transmission of messages on CCs.

Each CC may have a queue of service requests, which may cause delays in the transmission of messages. It is believed that network traffic is described by Poisson flows with an average value γ_{ik}. This is the number of messages per second that occur in the *SN* w_j and are intended for transmission in *SN* w_k. The total external traffic entering the network is determined by the ratio:

$$\gamma = \sum_{j=1}^{W} \sum_{k=1}^{W} \gamma_{ik}. \tag{5.56}$$

Let's assume that message lengths are independent and distributed according to exponential laws with an average value $1/\mu$ (bit). To place these messages in the *SN* there is a buffer memory of unlimited size. It is possible to assume that if the amount of buffer memory is so large that the probability of different users simultaneously accessing the same resource is less than 10^{-3}, then the assumption of an unlimited amount of buffer memory is quite acceptable. The message is sent over the network from the source node to the destination node in accordance with a fixed routing procedure.

If the message has a length n (bit), then the time during which it occupies the *i*-th channel will be n / c_i (s). Each CC in the network is considered as a separate service device. Let's denote by λ_i the average number of messages per second passing through the *i*-th channel. Then full internal network traffic:

$$\lambda = \sum_{i=1}^{V} \lambda_i. \tag{5.57}$$

The cost of renting of *i*-th channel with capacity c_i is set by an arbitrary function $d_i(c_i)$, depending on the number and CC capacity. Let's denote by *D* the cost of the entire network, mainly determined by the cost of constructing the channels. Moreover, the *SN* cost can be included directly in the cost of channels. Then the cost of the network:

$$D = \sum_{i=1}^{V} d_i(c_i). \tag{5.58}$$

Of greatest interest is the average network message delay $\overline{T}$, considered by the user to be one of the main characteristics of the network. Let's denote by $\overline{Z}_{jk}$

delay the message that occurred in the *SN* w_j and is transmitted to the CC w_k. Then the values $\overline{T}$ and Z_{jk} are related by the equality:

$$\overline{T} = \sum_{j=1}^{W} \sum_{k=1}^{W} (\gamma_{jk}/\gamma) \overline{Z}_{jk}, \tag{5.59}$$

where γ_{jk}/γ is the fraction of the total input traffic, which has a delay $\overline{Z}_{jk}$.

Relation (5.59) reflects the network decomposition into source-receiver pairs.

Let's define as a criterion of service quality – the average delay of messages $\overline{T}$ in the network and its minimized value due to an appropriate choice of channel capacity values taking into account restrictions on the cost of the network (5.58). Thus, in this optimization problem, the characteristics of the communication network $\overline{T}$, cost restrictions D, as well as parameters $\{c_i\}$, $\{\lambda_i\}$ and the network topology, of which $\{c_i\}$ are variable parameters, are set.

Let's consider the delay $\overline{T}$, which is determined by equality (5.59), to clarify the nature of its random nature. Let's denote by the path l_{jk} along which messages are transmitted from *SN* w_i to *SN* w_k. It is said that traffic γ_{ik} for the *i*-th CC with capacity c_i is included in the path l_{jk} if the messages transmitted along this path pass the specified CC. In this case, the application flow intensity λ_i in the *i*-th CC is equal to the sum of the call flow intensities along all paths passing through this channel:

$$\lambda_i = \sum_j \sum_k \gamma_{jk}. \tag{5.60}$$

Since *c* is the sum of the average delays incurred by a message when transmitting it in different l_{jk}, then:

$$\overline{Z}_{jk} = \sum_{i\,:\,c_i \in l_{jk}} \overline{T}_i, \tag{5.61}$$

where $\overline{T}_i$ is the average time spent in the message in the *i*-th CC, that is, the average delay.

It is defined as the time spent waiting and transmitting over the *i*-th CC. With this in mind, from (5.59), (5.60) let's obtain:

$$\overline{T} = \sum_{i=1}^{V} (\gamma_i/\gamma) \overline{T}_i. \tag{5.62}$$

It is also possible to come to the same result using the Little's formula. Indeed, there is an average number of services in the *i*-th CC is $n_a = \lambda_i \overline{T}_i$.

Then the average number of messages on the network will be

$$\sum_{i-1}^{V} \lambda_i \overline{T}_i.$$

On the other hand, the average number of messages on the network is $\gamma\overline{T}$. Equating both quantities:

$$\gamma\overline{T} = \sum_{i=1}^{V} \lambda_i \overline{T}_i,$$

let's obtain:

$$\overline{T} = \sum_{i=1}^{V} \lambda_i \overline{T}_i / \gamma. \tag{5.63}$$

Let's express the average time spent on a message in the network through the CC characteristics. In the considered model of a communication network, the intervals between the moments of receipt of messages depend on the service time in the CC. This means that the service time for a given message in different CCs is associated with the length of the message and the fixed parameters of the channels: the length l_{ij} and propagation time of the signal in the CC t_i. So, if v is the energy propagation velocity of one burst (bit) of a signal in a CC, then $t_i = l_{ij}/v$. If the message has n bit, then the time during which it takes the i-th CC will be $t_i + n/c_i$. Accounting for the values t_i needed for networks with a large geographical length. Under the assumption that CCs are error-free and reliable, the source of randomness will be only a random value of message length – n.

For communication networks with medium connectivity, SNs have more than one input channel and more than one output channel. Using the assumption of independence, a single CC can be represented as a single-line queuing system with a Poisson flow at the input and an indicative service time with an average value $1/\mu c_i$. For such a queuing system, the delay caused by the waiting time and the service time will be written in the form $\overline{T}_i = n_c/(\mu c_i) + 1/(\mu c_i)$, where $n_c = \lambda_i \overline{T}_i$. Then, substituting the values n_c found by the Little's formula, performing the corresponding transformations, let's obtain the expression for:

$$\overline{T}_i = 1/(\mu c_i - \lambda_i). \tag{5.64}$$

Then, (5.64) is substituted into (5.63) and finally an expression is obtained for the average delay time of messages in the communication network:

$$\overline{T} = \sum_{i=1}^{V} (\lambda_i / \gamma)[1/(\mu c_i - \lambda_i)]. \tag{5.65}$$

By analyzing relation (5.65), it is possible to draw certain conclusions regarding the average message delay in the CN. In the case when the set $\{c_i\}$ is relatively homogeneous, with an increase in the load on the network, no term in expression (5.65) will dominate until the flow in one of the CCs (for example, in *i*-th) reaches the capacity of this channel, corresponds to a bottleneck online. Moreover, the value $\overline{T}$ is growing rapidly.

The analysis of the causes of message delay during transmission through the CN allows to directly go to the solution of the minimization problem by choosing the appropriate CC. The task of choosing the optimal capacity (OC) of channels in CN is one of the most important in the design (synthesis) of communication networks. This task is formulated as follows:

Flows $\{\lambda_i\}$ and network topology are given It is necessary to minimize $\overline{T}$ due to variation in CC capacities $\{c_i\}$ taking into account restrictions on the CN cost.

$$D = \sum_{i=1}^{V} d_i(c_i). \qquad (5.66, \text{a})$$

Let's consider the case of linear cost functions for capacities:

$$d_i(c_i) = d_i c_i, \qquad (5.66, \text{b})$$

where d_i is the cost per unit of capacity for the 1-th CC.

Cost coefficients d_i vary depending on the CC parameter, in particular, d_i are taken proportionally to the CC physical length. In the case of a linear relationship, maintaining the total cost at a fixed level will be equivalent to supporting the overall CN capacity at a certain constant level.

From (5.65) it follows that any solution to the OC problem should be implemented such that the *i*-th channel has capacity $c_i > \lambda_i / \mu$. From the point of view of message delay in the CN, it is not so significant as the allocation of excess of the CC capacity over the incoming traffic. It is only important that the condition indicated above is met.

Thus, an analytical expression is obtained for the objective function, the quality criterion (5.65). Now it is necessary to find the minimum value with restrictions on the CN cost (5.66, *a*).

To minimize the Lagrange functional:

$$G = \overline{T} + \beta\left[\sum_{i=1}^{V} d_i c_i - D\right],$$

where β is the Lagrange multiplier.

If to find the minimum value G when varying the capacity of the channels $\{c_i\}$ and taking into account the limitations, then the problem will be solved. Using the method of uncertain Lagrange multipliers, it is possible to arrive at an optimal solution:

$$c_{iopt} = \lambda_i / \mu + D_e \sqrt{\lambda_i d_i} / d_i \sum_{j-1}^{V} \sqrt{\lambda_i d_i}, \tag{5.67}$$

where D_e is the additional cost:

$$D_e = D - \sum_{i=1}^{V} \lambda_i d_i / \mu. \tag{5.68}$$

From (5.68) it follows that at the beginning of the design, capacity is allocated for each CC, corresponds to the load λ_i / μ. The remaining reserve capacity is distributed between the CCs in proportion to the square root of their loads. In order to be able to get the final value for the average delay of messages in the designed CN, its total cost should be greater than the sum of the costs for all V CCs. Then the difference (5.68) will be added value, which is aimed at supporting the necessary characteristics of the CN in real conditions.

Substituting expression (5.67) into (5.65), let's obtain an expression defining the minimum message delay time in the CN:

$$\overline{T}_{\min} = \left[\overline{l} / (\mu D_e)\right]\left[\sum_{i=1}^{V} \sqrt{\lambda_i d_i / \lambda}\right]^2, \tag{5.69}$$

where $\overline{l}$ is the average path length in the CN.

Relation (5.69) makes it possible to calculate the minimum average delay of the CN, if the CC capacity is chosen optimally according to (5.67). When $D_e \to 0$ the average message delay increases unlimitedly. If $D_e > 0$, the OC problem has a practical solution, that is $\overline{T} < \infty$. If $D_e \le 0$, the problem has no realized solution. Relations (5.67) and (5.69) give a complete solution to the OC problem in the case of a linear cost function.

By analyzing expression (5.69), a number of useful conclusions can be drawn. First of all, $\overline{T}$ is a strictly increasing function of the average path length $\overline{l}$. So, the CN topology has to be chosen so as to obtain the minimum average length of message transmission paths in the network. The latter is naturally achieved in a fully connected CN, that is, where each pair of nodes is connected by a CC.

The above optimization results were obtained under the condition that external traffic γ_{jk} is known and constant, and the routing procedures are fixed.

If they are either unknown or change in time, then it is impossible to find the traffic parameters in the channels λ_i, which means that it is optimal to distribute traffic according to (5.67). Therefore, it is necessary to introduce an adaptive routing procedure; it allows alternatives for selecting routes in order to search for paths with underloaded channel capacity.

Thus, the general conclusion for solving this optimization problem is as follows. If the parameters of the incoming traffic γ_{jk} are known and become, then, assuming a linear cost function, it is possible to design optimal communication networks from the point of view of selecting capacity that exactly correspond to the CN traffic with a fixed routing procedure. If the traffic parameters γ_{jk} are either unknown or change over time, then an adaptive route selection procedure should be used, which allows real traffic to adapt to an unsuccessful set of CC capacity.

Therefore, when designing complex CN, it becomes necessary to solve individual particular problems of CN optimization. These are the tasks of choosing connecting paths between CN users in such a way as to ensure the most efficient use of network equipment; provide the minimum possible path lengths and the number of transit sections in the tracks; provide the necessary amount of CC in the ways. In particular, during the analysis and synthesis of CN, it becomes necessary to solve the problem of finding the many paths that exist between a given pair of *SN*. All methods for finding paths in the CN are divided into two classes: matrix and network. Matrix methods are based on the transformation of various matrices – topological or matrix characteristics of the edges of the graph describing the CN. Network methods are based on constructing a tree of paths from a fixed source vertex to the remaining vertices of the graph. Network methods are the graphic equivalent of matrix methods.

Other problem that arise in the CN design are the problems of distributing channels and information flows, taking into account various criteria for the weight characteristic of paths. The most common distribution methods are methods that use the «shortest» paths. Short ones mean paths that are the shortest in length, with a minimum number of transit sections, with maximum capacity, minimum cost paths, maximum reliability. Today, there are a number of methods that allow to streamline the procedure for determining the length of a rank or other characteristics of paths. Conventionally, these methods can be divided into two groups: matrix and network. The matrix is Floyd algorithm, which makes it possible to obtain both the shortest distance matrix and the route matrix, which determines the vertices of the graph that make up the paths. Flow distribution algorithms in the CN based on the Ford and Fulkerson theorem on maximum flow and minimum cross section in the network. When designing, both matrix and network algorithms for distributing flows in networks are used.

5.6 Mathematical models of communication network control systems

5.6.1 Principles of building a communication network control system

First, let's outline the general provisions for optimal control of objects. The control system (CS) includes a control object, a control device, implements control solutions, and a measuring system. The control object receives useful input effects $X(t)$ and interference $\xi/(t)$. The state of the object is determined by some reaction $V(t)$. The process of extreme control consists in developing, according to estimates of the state of an object $\widehat{V}^*(t)$ obtained with the help of a measuring system and an estimation algorithm, optimal control actions $U(t)$ that bring the control object to the required state $V(t)$. The effectiveness of the control system is determined by the selected optimality criterion. The quality of the control system is determined by the degree of deviation of the real state of the system $V^*(t)$ from the desired $V(t)$: $\psi(t) = V(t) - V^*(t)$.

It is necessary to develop such control actions $U(t)$ in order to achieve the extremum of some target functional, in particular,

$$J = \overline{\left[V_1(t) - \widehat{V}^*(t)\right]^2} \to \min_U .$$

The general scheme of extreme control is shown in Fig. 5.1

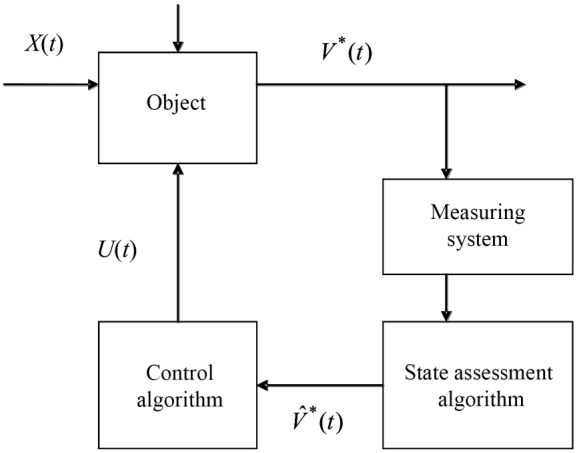

Fig. 5.1 General scheme of extreme control

In accordance with the purpose, the communication network control system is a set of services and software and hardware that provide the network administration with information about the operation of the network and make it possible to automatically or automatically influence its operation. A communications network is a distributed system. Network control organization includes:

a) collection of control information about the state of network elements;

b) analysis of the qualitative characteristics of the network for their compliance with user requirements;

c) development of control decisions;

d) bringing this decision through the control technical means of implementing decisions (TMID) to the network elements.

In this regard, the following are fundamental for network control:

- control of a real network is discrete, that is, the control system makes changes to the characteristics of the network to achieve its qualitative indicators of some critical values;
- reaction of the control system to possible critical situations during network operation (overload, equipment failures, etc.) occurs with some delay associated with the speed of control algorithms and the performance of technical equipment of the control system. If this delay is above a certain value, then the control actions will be inadequate to the situation, which may have negative consequences. In other words, stochastic stability of control is fundamental. The thresholds and stochastic stability of control are determined at the design stage of the control system. In CS network can be allocated such relatively autonomous, functional subsystems (Fig. 5.2).

The development of network control applications is closely related to the mathematical modeling of network control processes and its elements. The autonomous design of each of the selected control subsystems is a solution to two problems: optimization of the structure and optimization of functioning algorithms. Since the structure of the DCS is determined by the structure of the information delivery system (i. e., telecommunication network), when modeling the DCS, the main task is in optimization of its algorithms, the main of which is routing, which ensures the distribution of information flows in accordance with a specific plan, and controls the flows, control of input and transit loads in the network. In the process of solving these problems, threshold characteristics of qualitative indicators of network performance can be obtained, the passage of which requires either operator intervention or automatic redistribution of load flows. When simulating ACSs and MSs, their own structure and recommendations to the operator for controlling the primary and secondary networks (models and algorithms for managing the network structure) are subject to calculation.

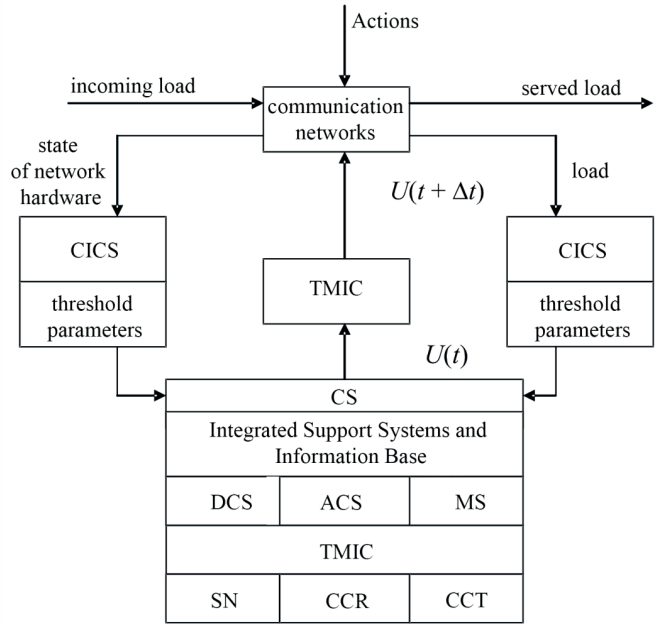

Fig. 5.2 Organization scheme for communication network control: control information collection system (CICS); dynamic network resource control system (DCS); maintenance system (MS); administrative control system (ACS); technical means for the implementation of control decisions (TMIC)

Mathematical modeling of the network is necessary both for solving the above problems and for developing numerous control applications. This direction is the most important in the design and implementation of network control of modern communication networks.

5.6.2 Features of the mathematical model in the problem of optimizing network resource control

The main task of DCS resources is in collection of statistical data on the operating modes of communication equipment and to ensure the efficient operation of the network and network elements. The following main classes of DCS functions can be distinguished by resources.

Dynamic control. Dynamic control of the probability-time characteristics (PTC) of information delivery (quality of service) and development of

solutions to support them within the specified limits, including monitoring and documenting network parameters regarding the requirements for delaying messages, the magnitude of denial of service, connection support, connection quality, dependent from criteria for evaluating network performance and message priority; operational monitoring of network equipment operation parameters, the decrease of which relative to the norm can cause poor quality of service in the network; collection of load data in network elements.

Load control. Control of subscriber load and access to the network (using the tariff schedule, priority service, threshold control, limiting the subscriber load by automatically disconnecting part of subscribers, assigning payment for repeated calls, organizational measures to notify subscribers, etc.) controlling inter-nodal and internally nodal load flows (organization of bypass directions, changing routes by threshold characteristics, protection against calls that have a low probability of successful completion, about border load transit, equipment configuration changes in overload situations, adaptive scheduling of work programs, etc.); load control at the outlet of the network (measures to increase the value of the served load).

Formation of service messages and their transfer to the CCR and CCT. One of the problems during network operation is the optimization of the flow of information control carried out in the DCS. The main objectives of the DCS is the optimization of routing, provides the distribution of information flows and flow control, aimed at limiting the incoming and transit network loads. Optimal solutions in the general case depend on time, which makes it difficult to solve the corresponding optimization problems of mathematical programming (called optimal control problems). However, a characteristic feature of controlled communication networks is the relatively slow change in the state of networks over time. This allows to introduce the concept of the so-called control interval T_a, during which the state of the network remains practically unchanged, that is, it can be considered stationary. Consequently, the solution to the simplified mathematical programming problem remains unchanged, the formulation of which does not explicitly include time. The control process can be represented in the form of multiple (with a period T_a) solution of such simplified mathematical programming problems.

The specific formulations of mathematical programming problems (specific objective functions and constraint functions) are directly determined by the selected resource control algorithm, the nature of the network, the set of its structural parameters, the adopted routing method, etc. Determining the objective function in this case is a complex independent task that precedes the solution of the optimization problem control in general and routing in particular.

The least difficulties are associated with the solution of optimization problems of load control. It is known that an increase in load above a certain value

can lead to a loss of operability of the switched network. Routing reduces to a certain extent network losses during congestion, however, in order to avoid loss of network operability, it is necessary to control the input and transit loads. In other words, message flow control is required. The threshold value is fundamental to the methods of flow control in circuit-switched networks (CSW): the circuit bypass is chosen for use if the number of occupied channels on all transit sections of the circuit does not exceed the threshold. A characteristic feature of networks with a CSW is the close relationship of routing algorithms with flow control algorithms. A generalization of the results of analytical and statistical modeling of these algorithms allows to draw the following conclusions: when using only routing algorithms, there is always a critical load in the network at which the maintenance of bypass routes is better; the use of thresholds for overload at any overload provides a reduction in losses compared to losses in the network without workarounds.

5.6.3 Shortcuts and optimal distribution plan in the network

In branched switched communication networks, between any two network nodes (source and destination), as a rule, there are several independent ways in which messages can be transmitted. The main task of routing is to select a specific path from the specified set. The selection is made using matrices (tables) of routes that are stored in each switching node (SN). The route matrix M_i of i-thsets the priority for the selection of outgoing directions when establishing a node connection with any of the other network nodes.

The matrix M_i can be represented as follows:

$$M_i = \begin{array}{c} \\ \beta_{i1} \\ \beta_{i2} \\ \vdots \\ \beta_{ij} \\ \vdots \\ \beta_{iA} \end{array} \begin{array}{c} \begin{array}{cccccc} \mathrm{SN}_1 & \mathrm{SN}_2 & \cdots & \mathrm{SN}_r & \cdots & \mathrm{SN}_\mathrm{N} \end{array} \\ \begin{bmatrix} m_{i11} & m_{i12} & \cdots & m_{i1r} & \cdots & m_{i1\mathrm{N}} \\ m_{i21} & m_{i22} & \cdots & m_{i2r} & \cdots & m_{i2\mathrm{N}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{ij1} & m_{ij2} & \cdots & m_{ijr} & \cdots & m_{ij\mathrm{N}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{iA_{i1}} & m_{iA_{i2}} & \cdots & m_{iA_{ir}} & \cdots & m_{iA_{i\mathrm{N}}} \end{bmatrix} \end{array}, \tag{5.70}$$

where N is the number of network nodes, β_{ij} is j-th branch originating from i-th SN, A_i is the number of branches originating from SN_i, m_{ijr} – the element that determines the serial number of the branch β_{ij} when choosing message transmission paths from SN_r to SN_i – in the case of a simple matrix of routes, or the probability of choosing the branch – in the case of a stochastic matrix routes β_{ij};

$$\sum_{j=1}^{A_i} m_{ijr} = 1,\ r = \overline{1,N}.$$

Let a set of route matrices $\left\{M_i, i = \overline{1,N}\right\}$ be given. This means that for the entire network a given plan for the distribution of information. With a static plan for the distribution of information, static (fixed) routing is performed on the network. However, the most efficient use of network resources is achieved with adaptive routing, when the information distribution plan changes in accordance with the network operating conditions, changes (overloads in individual directions or sections of the network, damage to channels or their bundles, damage to CNs, etc.). A similar dynamic distribution of information flows was first proposed for telephone networks, but received practical use in the CN packet switching. Beginning with the ARPA network, adaptive routing has been applied to one degree or another in almost all created CNs.

Adaptive routing involves choosing the best ways to transmit information depending on the situation on the network. Routing optimization can occur both according to general network and local optimality criteria. The former include the average delay in the transmission of messages (packets), the average probability of timely delivery in EP networks, integral losses in the CSW networks, the maximum allowable values for the length or cost of the road, and the like. Local criteria may include a delay in transmission between a group (steam) of subscribers, the probability of loss in individual communication directions, etc.

In the general case, the choice of the optimality criterion for an algorithm in DCS is ambiguous. Preference should be given to criteria related to CC capacity factors. This means that such routing or flow control solutions are considered optimal that, while fulfilling the requirements for the information delivery characteristics, make it possible to maximize the use of the capacity of the network paths, or to obtain the maximum values of the capacity factors of the network paths.

In accordance with the optimality criterion for each branch, it enters one route or another, some of its weight (cost) is determined.

A route with a minimum or maximum weight, which is the linear sum of the branch weights, is considered optimal by this criterion, or by the shortest path.

Depending on where the calculation of route matrices is performed and a decision is made on the routes for this *SN*. With adaptive routing, the following are defined:

- *centralized routing*, in which the decision on routes is made in the network route center;
- *distributed (decentralized) routing*, in which each *SN* determines the shortest paths to all nodes of the network using certain algorithms for exchanging service information between nodes, characterizing the state of the output directions of the node;

– isolated routing, when each node itself decides on the transmission path according to individual criteria without exchanging service information with neighbors;
– *mixed routing*, which is one or another combination of centralized and distributed routing.

To calculate the routes of all types of routing, two main classes of algorithms are used:

– *algorithms for determining the shortest path* provides for a given source-destination pair a choice of the optimal (short) path according to a given criterion;
– *algorithms for the probabilistic (alternative) selection of possibleroutes* for a given source-destination pair that minimize (or maximize) a criterion for optimizing routing throughout the network, taking into account the average load arriving in the network, between all source-destination pairs. In this case, for a given pair, the source-destination is determined by the set of routes with the probability of their choice.

In the shortest path determination algorithms, as a rule, two solution methods are used: node and branch numbering methods and matrix methods. The numbering methods of nodes and branches are usually iterative and determine the shortest path from a given node to all other network nodes. Matrix methods allow to find the shortest paths between all nodes of the network at the same time, using operations on the matrix weight communication can miss a certain load with the increase in network branches. In packet-switched networks, two more methods have been most used; they are varieties of the above methods: Ford and Fulkerson methods and Dijkstra's methods.

5.6.4 Features of the mathematical model in the optimization problem of network traffic control

Network load above a certain value occurs by saturation of the network, and this, in turn, can lead to the loss of its performance. They talk about a situation when the input load exceeds the potential network performance or the transit load in the *SN* is growing also due to an increase in the input load. It is necessary to control the input load and maintain a certain ratio between the input and transit loads. Such control is carried out using flow control algorithms.

Flow control in networks with a EP is aimed:

1) prevent a decrease in network efficiency due to congestion;
2) eliminate the loss of network performance (full block);

3) optimally distribute resources between users;

4) ensure compliance between network performance and input load.

Formal statement of the routing optimization problem. The main task of routing is in selection of a specific path from the set of acceptable paths between any two SNs (source and destination) using route tables. The route matrix of the A_i SN sets the sequence or probability of choosing outgoing directions when establishing connections from the *i*-th node to any $(r-y)$ node of the network. A set of routing matrices $\left\{M_i, i=\overline{1,N}\right\}$ is a plan for the distribution of information. In accordance with the route optimality criterion, for each branch, it enters into different routes, a certain weight is determined. A route with a minimum (maximum) total weight is optimal according to this criterion. Different optimization criteria generate different routing plans. Similarly, the use of various common network and individual criteria for optimizing routing also leads to solutions that match. Therefore, there is no universal routing. However, the general statement of the problem of optimizing the information distribution plan (routing) can be formulated, if do not specify the type of criterion. It could be like that. Let a network graph $G(V, U)$ be given in which the set of vertices V corresponds to a SN with capacities $C_i(t)$, $i=\overline{1, N}$, the set of edges U corresponds to the communication branches with capacities $C_{ij}(t)$,

$$U=\left\{i, j;\ i=1, N,\ j=\overline{1, A_{ij}},\ A_i \in \left\{\left[1, N-1\right],\ i \neq j\ \forall\ i, j\right\}\right.,$$

A_i is the number of edges (branches) arising from SN_i.

Optimization is carried out on the control interval T_a. Its size is chosen so that the value is not exceeded T_m – the time interval between recorded changes in the network (functional and structural), that is, the condition must be met:

$$T_a \leq T_m. \tag{5.71}$$

It is also assumed that one more condition is followed, which includes the values T_a:

$$T_{cit} \ll T_a, \tag{5.72}$$

where T_{cit} is the total time of control information transmission about changes in the network, development of actions (solutions to the corresponding optimization problems), control and transmission of control service information.

Condition (5.72) allows to justify the requirements for the time characteristics of adaptive routing algorithms.

The input flow is characterized by a set of functions $\lambda_{ir}(t)$, $i, r=\overline{1,N}$. The value $\lambda_{ir}(t)$ is the intensity of the input flow at the node i addressed to the *r*-th node.

Consideration of conditions (5.71) allows to exclude the time for consideration and use the following parameters:

$$\left\|C_{ij}(t)\right\|_{t\in T_a} = C_{1a} = \left\|C_{ij}\right\|,\ \left(C_i(t)\right)\Big|_{t\in T_a} = C_{2a} = (C_i); \tag{5.73}$$

$$\Lambda_{in}(t) = \left\|\lambda_{ir}(t)\right\|_{t\in T_a} = \Lambda_{ina} = \left\|\lambda_{ir}\right\|. \tag{5.74}$$

Route coefficients in the route matrix are defined as:

$$P_{ij} = \left\{P_{ikj} \geq 0, (i,k,j) \in \{M\}\right\}, \tag{5.75}$$

where P_{ikj} is the part of the total flow $\lambda_{i(j)}$ directed along the k-th branch from the i-th node addressed to SN_i.

Conditions must be met:

$$\sum_{k=1}^{A_i} P_{ikj} = 1,\ P_{ikj} \geq 0,\ i,k = \overline{1,N},\ j = \overline{1,A_i}, \tag{5.76}$$

The flow intensities in the network branches are defined as follows:

$$\lambda_{ik} = \sum_{j} P_{ikj}\lambda_{i(j)},\ i,k = \overline{1,N},\ j = \overline{1,A_i}, \tag{5.77}$$

where λ_{ik} is the intensity of the flow in the branch connecting SN_i and SN_k.

The intensity of the full flow at SN_i addressed to SN_k be defined as:

$$\lambda_{i(r)} = \lambda_{ir} + \sum_{k=1}^{N} P_{kir}\lambda_{k(r)}. \tag{5.78}$$

Conditions must also be met:

$$\lambda_{ij} < \mu_{ij},\ i = \overline{1,N},\ j = \overline{1,A_i}, \tag{5.79}$$

where μ_{ij} is the intensity of service on the branch ij, depending on the values C_{ij} and C_i.

The question of giving conditions (5.79) of an algorithmic form deserves a separate consideration.

For networks routed by route variables (packet switching, messages, channels), the optimization problem is formulated as follows:

$$Q(\Lambda_{ina}, C_{1a}, C_{2a}, P_{ij}) \Rightarrow \max_{P_{ij}}. \tag{5.80}$$

Under conditions (5.71)–(5.79), the formulation of the optimization problem changes:

$$Q(\Lambda_{ina}, C_{1a}, C_{2a}) \Rightarrow \max_{U_a}, \tag{5.81}$$

where U_a is control when short and bypass routes are allowed for part or all of the flows in the network.

For real networks with unreliable channels and a finite time to bring the controllers to the controlled objects in the formulations of problems (5.80), (5.71)–(5.79) and (5.81), (5.71)–(5.79) (hereinafter, problems I and II, respectively), it is necessary include one more condition:

$$U_a = U_{a\,ex}, \tag{5.82}$$

where $U_{a\,ex}$ is executed control action.

If when solving problems I and II, condition (5.71) is not taken into account, that is, it is considered that the network does not change during the entire period of consideration T, then it is about static routing. In general, if the condition (5.71) is met:

$$T = \sum T_a \tag{5.83}$$

introduce the concept of adaptive routing.

Under certain conditions, problems I and II can be greatly simplified. The most important of the simplifications is the breakdown of the original task for the entire network into a set of nodal tasks, when each *SN* node makes decisions on message routing independently of the others, that is, a distributed routing algorithm can be built. Such a conversion may be exact or approximate.

In the first case, the set of nodal solutions provides an optimal plan for the distribution of flows as a whole in the network; in the second, it is only possible to talk about the degree of proximity to the overall optimal solution. Accurate partitioning is possible under a number of essential conditions. First of all, it is necessary that the objective function be adaptive or multiplicative with respect to variables P_{ijr}, that is, it must be a sum or a product of functions, each of which depends only on its nodal variables.

Secondly, which is also absolutely necessary, the variables P_{ijr} should not depend on *r*, that is, they should be unknown:

$$P_{ijr} = f_{ij}(\lambda_{ij}) = P_{ij}, \ \forall_{i,j,r} \in M. \tag{5.84}$$

Moreover, the exchange of service information between the *SNs* is not needed, that is, a routing algorithm can be completely distributed.

And finally, the input flow also should not depend on, since otherwise the functioning of each of the nodes depends on all the other nodes, and there can be no question of the exact breakdown. Examples of nodal problems are discussed below.

In a real network:

$$P_{ijr} = f(\Lambda_{\sum a}),\ \Lambda_{\sum a} = \sum_i \sum_r \lambda_{ir}.$$

Therefore, in order to build optimal distributed routing, in addition to fulfilling other conditions, the exchange of service information between SNs is necessary, while the condition $U_a = U_{a\,ex}$ that can be fulfilled with the corresponding time characteristics of adaptive routing algorithms is of fundamental importance.

The following statements follow from this:

1. By the objective function $Q(\bullet)$ of an arbitrary form, the optimal information distribution plan (optimal routing) can be found for the communication network with a given $\Lambda_{\sum a}$ in control interval T_a, with full network observability and condition $T_a \geq T_{cit}$ fulfillment. Moreover, if the network is observed at one point, then a centralized routing strategy will be implemented. If the network is observed at many points of the network during the exchange of service information managing between them, a decentralized (distributed) routing strategy.
2. Fully optimal distributed routing in the network, which does not require the exchange of service information between observation points, can be carried out in a network with an objective function $Q(\bullet)$, which is a separable function of routing variables at the observation points. In addition, the above requirements P_{ijr} to λ_{ir} must be observed. Since, in the general case $P_{ijr} = f(\Lambda_{\Sigma})$ for $\forall i, r, j \in M$, then there is no optimal distributed routing without the exchange of service information between the CCs. For such routing to exist, conditions (5.84) must be satisfied.
3. In static routing, the condition $T_a \geq T_{cit}$ is not taken into account; in adaptive routing, it acquires a determining value.

The problem of finding optimal static routing coincides with problem 1 in the absence of a condition and is, in fact, the task of distributing information flows in the network.

If

$$P_{ijr} = \begin{cases} 1, \text{ at } j = 1; \\ 0, \text{ at } j \neq 1, \end{cases}$$

then, when solving problem 1, the distribution of flows with fixed routing will be found.

The methods for solving problems 1 and 2 essentially depend on the type of the objective function $Q(\bullet)$ and, therefore, will be different for networks with different types of switching.

A feature of optimization algorithms for control systems of the problems 1 and 2 is the presence of uncertainty about the design environment and the states of system elements. This circumstance can lead to errors in making decisions that appear during the operation of control algorithms.

Therefore, such assumptions are considered a priori valid, allowing to consider the found control solutions as stochastic stable:

1) symmetry of the communication network uses the methods of optimal organization of processing measurements of the parameters of the design environment, which allow to minimize a priori uncertainty in their values;

2) statistical processing of observations allows replacing stochastic optimization problems with deterministic ones by replacing the components of random vectors of the design environment with their mathematical expectations;

3) to solve problems 1 and 2, if necessary, appropriate generalizations of the objective functions and space limitations of the design environment are applied;

4) the optimal control in tasks 1 and 2 can be determined using the assumption of full network observability (full information). This means that, provided that the time requirements for collecting and processing information about the network state are met, it does not exceed the amount of time the design environment changes, the accuracy of the solution will depend on how much the conditions for the instantaneous execution of the control action $(U_a = U_{a\,ex})$ on the interval $T_{a-1} > t$ can be considered fulfilled.

In other words, if the transmission speed of the control information is high, and during the transfer of the control decision there will be no change in the current situation in the network, then the necessary accuracy of real control can be obtained. It is believed that the CS reaction time is such that $U_a = U_{a\,ex}$ is true. This implies the requirement for the transmission rate of service information.

The formulated provisions can be considered as external requirements for indicators of symmetry and transmission speed of service information in the DCS.

6 SOLUTIONS OF SOME OPTIMIZATION PROBLEMS OF COMMUNICATION NETWORKS

This section provides practical features of the application of scalar and multicriteria optimization methods in the course of solving some problems of planning and designing communication networks. In particular, the problems of optimizing the topological structure of the communication network, choosing the optimal capacity and optimal flow distribution under the given restrictions in message switching networks, as well as the problems of choosing the optimal design options for the data network, optimal planning of cellular communication networks, optimal routing, selection optimal speech codecs, optimal distribution of network resources, taking into account the totality of quality indicators.

In preparing the materials of the unit, the works [10, 11, 15, 16, 35, 45, 48] are used, which can be addressed in the course of an in-depth study of these issues.

6.1 Optimization problems of the topological structure of a communication network

Designing communication networks requires solving a set of complex interrelated tasks, which include: optimization of the topological structure and capacity of communication channels; choice of routes; selection of flow control methods and determination of control parameters; analysis of the buffer memory volumes of switching and routing nodes and the choice of buffering strategies during congestion and the like.

Formally, the task of designing a global network comes down to finding the minimum of the functional of the present value:

$$E(H,\Omega,Y) \to \min \tag{6.1}$$

in the presence of restrictions on the probabilistic-temporal and structural characteristics of the network:

$$W_i(H,\Omega,Y) \le W_{i0} \tag{6.2}$$

and requirements for membership of many network architecture options satisfying the constraints (6.2) in the field of technically feasible solutions:

$$Q(H,\Omega,Y) \in Q_0. \tag{6.3}$$

Here H is a vector quantity that displays the network load parameters, including the intensity of message flows between each pair of network switching nodes, the distribution of message lengths, the priority of message flows, and the like; Ω is vector value, which is a set of parameters of technical means, including the performance of switching nodes and channel-forming equipment, the reliability of technical means, the reliability of information transfer, and the like; Y is vector quantity, which displays the parameters of the logical structure of the network.

The means from this general design problem is in creation of a complex of mathematical models, among which the models of queuing systems (QS) occupy an important place. At the same time, high design quality can be achieved only when individual methods and models are combined on the basis of a systematic approach into a single design system and cover all or most of the design problems.

The presence of difficult formalized facts and limitations, the proximity of some initial data and the multi-criteria nature of the general task of designing communication networks necessitates the use of an interactive design mode. This mode allows to combine modern mathematical models and optimization methods with the experience and intuition of the designer in a single process. This provides the designer with the ability to monitor the design process and actively intervene in the search for optimal solutions.

The practical impossibility of formulating and solving within the framework of one mathematical problem of the whole complex of problems of designing a communication network leads to the necessity of using a procedure based on decomposition. Such decomposition is possible both at the structural level and at the level of solving individual design problems and allows to move from tasks of large dimension to a sequence of tasks of smaller dimension.

Decomposition at the structural level means that the design of a communication network is reduced to the independent design of a number of subnets, subject to the conditions of coincidence or proximity of optimal solutions to the network design problem and the corresponding solutions for subnets. These conditions include: subnets beyond the scope of restrictions must be independent; the objective function of the network is clearly a monotonous function of the objective functions of the subnets.

As a criterion in the design of a communication network, a generalized economic criterion is often chosen – reduced costs, which include the cost of renting communication lines in the base and regional networks, as well as the reduced cost of switching nodes. Other criteria (average delay time, reliability, etc.) are used as a limitation in solving the design problem. Obviously, with this choice of criterion and restrictions, the above conditions are satisfied when the global communication network is decomposed into the base and regional networks due to the additivity of restrictions and the objective function, which is the sum of the reduced costs for the base and regional networks. This allows independent design of core and regional networks.

Decomposition at the design level of the base and regional networks means the creation of a multilevel hierarchy of interconnected models, the analysis of which allows to obtain a solution to the common design problem for each of these networks and thereby comply with the principle of independence of the solution to the design of the computer network as a whole.

6.1.1 Problem statement of network topology synthesis

The task of synthesizing a topological structure is one of the main ones in the design of a communication network and consists in choosing the optimal connection scheme for switching and concentration nodes, choosing the line capacity and the optimal information transmission routes. The choice of topological structure is carried out according to the criterion of the minimum total annual lease of communication channels in the presence of restrictions on the delay time and reliability of information transfer. The reliability requirement in the design of basic and terminal networks is taken into account by introducing restrictions on the network connectivity (the number of independent routes from source nodes to destination nodes) and the number of retranslations in the route (number of intermediate switching or concentration nodes). It is assumed that the number of retranslations is not more than two and the principle of biconnection is used. In accordance with this, each source-destination pair is connected by at least two paths that do not have common nodes and channels. Thus, when a node or a communication channel fails, the network remains operational.

The initial data for the topological design of the information network are based on the requirements of the technical specifications for the functional and technical and economic characteristics of the information network and include:

- technical and economic characteristics of switching nodes and concentration of information, channels and data transmission equipment;
- requirements for time delay reliability and reliability;
- matrix of message flows from sources to recipients;

– volumes of information and service messages transmitted by the network;
– dependence of rental cost on the length and capacity of communication channels.

6.1.2 Combinatorial algorithm for topological network optimization

Let's consider a combinatorial algorithm for solving the problem of choosing the optimal connection scheme for package switching nodes, selecting routes and capacity of lines, which was used at various stages of design and development of large-scale networks.

The combinatorial approach is based on the representation of the data transmission network in the form of a finite graph without loops and multiple edges, whose vertices correspond to the network nodes, and the edges correspond to communication lines. Such a representation is convenient for studying the characteristics of a network using well-developed graph theory.

The use of graph theory to solve the problem of topological optimization has long been considered unpromising because of the need to study a significant number of possible options for connecting network nodes.

The nonlinear dependence of cost on the length and capacity of communication channels greatly complicates the solution of the problem of synthesizing a topological structure, not allowing the direct use of simple analytical results. Therefore, for the design of a computer network, in accordance with the principle of independence, it is carried out independently for the base and regional networks, more complex numerical algorithms are used.

Nevertheless, recently the combinatorial approach has been finding wider application in solving the topological optimization problem. This is due, firstly, to an increase in the performance of computers used to calculate combinatorial problems, secondly, to the development of new effective algorithms for generating graphs with specified properties and, finally, the extensive experience in the development and operation of data transmission networks allows to reasonably formulate the requirements for designed network, which significantly narrow down the class of possible solutions to the topological optimization problem.

The following is a description of a number of combinatorial algorithms for topological optimization, taking into account the main real requirements for the design of computer networks. The need to develop combinatorial algorithms is caused by at least the following reasons:

1. Large-scale data transmission networks are usually designed in stages, and the network dimension at the first stages is small, which allows one to obtain an exact solution to the topological optimization problem using

combinatorial algorithms. At the same time, approximate algorithms are known that do not allow finding the exact solution even for networks of small dimension. Moreover, the best heuristic algorithms according to the developers give an error of up to 10 %.

2. The presence of an exact solution allows to evaluate the quality of well-known and newly developed heuristic algorithms.
3. Combinatorial algorithms provide ample opportunities to study the properties of optimal solutions and optimize functions, which in turn create the prerequisites for the development of new efficient algorithms.

The developed algorithms are software realized in the application package for synthesizing the topology of data transmission networks that implement:

- accurate and approximate algorithms for constructive enumeration of graphs to solve the problem of topological synthesis of a network of minimum cost in the presence of restrictions on reliability, are effectively used at the stage of pre-project network research;
- combinatorial algorithms for accurate and approximate solutions to the problem of synthesizing topology, selecting capacities and routes are used at the stage of technical design and in the process of developing a computer network;
- algorithm of «saturation of the section»;
- algorithms for solving the problem of topological synthesis of networks with increased reliability requirements, in particular, designing networks with the number of independent paths between each pair of nodes greater than two;
- heuristic algorithms for the synthesis of topology of large-dimensional networks.

6.1.3 Optimization of the topological structure according to the criteria of cost and reliability

In some cases, the simpler topological optimization problem is of interest, the statement of which does not take into account the requirement for information flows passing through the network.This problem is solved at the pre-design stage of creating a network without detailed information about network protocols, input intensity matrixes, etc.

Let the network be described by a graph $G_0(V,U_0)$, where $N=|V|$ is the number of vertices and $M_0=|U_0|$ is the number of edges of the graph G_0.

Let's consider a spanning subgraph $G(V,U)$ of a graph G_0 in which $N=|V|$, $M=|U|$. Let's denote: $D(G)$ is the diameter of the graph G, $D(G_u)$ and $D(G_v)$ are

the diameters of the graph G_u obtained from the graph G by deleting an arbitrary edge and the graph G_v obtained from the graph G by deleting an arbitrary vertex. The inherent weights and level of the cost of renting channels between pairs of switching nodes are assigned to all edges of the graph G. Under the cost of the graph G let's mind, the sum of the weights that go into the G edges (indicated). Let's denote by X – the set of all spanning subgraphs of the graph G_0.

Then the statement of the synthesis problem for the topological structure of the data transmission system is as follows.

– find such a subgraph G':

$$E(G') = \min_{G \in X} \{E(G)\} \tag{6.4}$$

under such conditions:

$$D(G) \le d_1, \tag{6.5}$$

$$D(G - u) \le d_2, \text{ for any } u \in U, \tag{6.6}$$

$$D(G - x) \le d_2, \text{ for any } x \in V. \tag{6.7}$$

Condition (6.5) for determining the diameter of a graph is equivalent to a restriction on the length of the shortest path between each pair of vertices. Conditions (6.6) and (6.7) limit the lengths of the shortest paths between each pair of vertices when removing an edge or vertex in the graph.

Problems (6.4)–(6.7) are a complex NP-complete discrete programming problem. Before describing the algorithm for solving problem (6.4)–(6.7), let's dwell on the method of obtaining a lower estimate of the network cost. On the one hand, it allows to estimate the preliminary costs of creating a network (which is very important at the pre-project stage of creating a network), on the other hand, the lower cost estimate will be used as an important stage for a combinatorial algorithm for solving the problem.

Let's denote by M the number of edges of the network and determine the lower estimate of the cost of the network with M edges. To determine the lower score, the following technique is used: conditions (6.5)–(6.7) are not taken into account, and the condition of the graph biconnection is replaced by the necessary condition:

$$\deg\left(v \in V(G)\right) \ge 2.$$

In addition, a new condition is introduced, which is that the network contains M edges:

$$\sum_{v\in V(G)} \deg v = 2M.$$

Therefore, it is necessary to solve the following problem. To find:

$$E(G') = \min_{G\in X}\{E(G)\},$$

if

$$\sum_{v\in V(G)} \deg v = 2M \text{ and } \deg\left(v \in V(G)\right) \geq 2.$$

To solve this problem, the following algorithm can be used. Let G_{ij} be the weight assigned to the edge for which the incidence of the vertices i and j.

Step 1. Sort matrix rows $\|E_{ij}\|$. Matrix rows $\|E_{ij}\|$ are sorted in ascending order of cost. Thus, for each node *i*, the *i*-th row of the matrix $\|\tilde{E}_{ij}\|$ contains the cost of connecting the *i*-th node with all other nodes in ascending order (assumed $E_{ij} = \infty$).

Step 2. For each row of the matrix $\|\tilde{E}_{ij}\|$, select the first two elements, that is, put:

$$E_1^* = \sum_{i=1}^{N}\sum_{j=1}^{2} E_{ij}.$$

Step 3. Arrange the remaining elements $N^2 - 2N$ of the matrix in ascending order of their cost, that is, form a vector $\tilde{E} = \{\tilde{E}_r\} : \overline{E}_i \leq \overline{E}_j$ from its other elements for $i < j$.

Step 4. Select the first $2 - 2N$. In the elements $\tilde{E}$ of the vector, that is,

$$E_2^* = \sum_{r=1}^{2M-2N} \overline{E}_r.$$

Step 5. Calculate the lower score for the cost of the network with M edges: $E^* = \left(E_1^* + E_2^*\right)/2$. The score is divided in half, since the solution obtained by the algorithm contains $2M$ edges, and the solution must have M edges.

Step 6. The end of the algorithm.

It is easy to see that the complexity of the algorithm is determined by step 1 and is equal to $N^2 \log N$. The number of edges M is usually unknown. Therefore, to obtain a lower estimate of the network cost, it is necessary to determine the boundary of the number of edges at which conditions (6.5)–(6.7) are satisfied.

Thus, to view feasible solutions, wit is possible to propose an algorithm, which consists in the following. First, the lower limit of the number of edges $M_{\min}$ is determined. For a given number of nodes N and edges $M = M_{\min}$, let's inves-

tigate all graphs that are biconnected and satisfy the constraints (6.5)–(6.7). Among these graphs, a graph G' is selected whose sum of weights is minimal. If in the process of analysis it is established that a cost equal to the lower cost estimate for the M edges is reached on a certain graph, then the search for the optimal solution stops. If all graphs with M edges are analyzed, then the number of edges increases by one; a lower estimate of the network cost is determined, and if it is not better than the optimal solution with fewer edges, then the algorithm ends. Otherwise, all graphs with $M+1$ edges are investigated, etc.

Machine experiments of synthesizing the topological structure of networks with the number of switching nodes not exceeding ten showed one important property of the proposed combinatorial algorithm: no more than 5 % of the total number of iterations is required to find the optimal solution. The remaining iterations are spent on improving the optimality of the solution. Given this property, on the basis of the combinatorial approach, effective algorithms have been developed that are heuristic for the synthesis of the topology of medium and large networks.

6.1.4 Algorithm for generating the main biconnected subgraphs of a given graph

The exact solution to problems (6.4)–(6.7) is based on algorithms for constructive enumeration (generation) of graphs with specified reliability properties. The main drawback of constructive graph enumeration algorithms is the impossibility of their application for the synthesis of large-dimensional networks, since the number of generated graphs grows exponentially as the number of network nodes grows. A mathematically well-justified method is proposed for reducing the number of generated graphs when solving the synthesis problem for topology of DCNs.

Let's consider the problem of finding all the main biconnected subgraphs of a graph G_0 with a given number of edges M for which the diameter $d_1 \le 3$ (since networks with a small diameter are of practical importance). The proposed algorithm, based on the general search method with returning and fulfilling the necessary and sufficient conditions for biconnected graphs with a diameter, does not exceed 3. The main ideas of the proposed algorithm are in the following.

Let the vector $(x_1, x_2, ..., x_i)$ be a partial solution to the problem. To expand the solution, a candidate x_{i+1} is selected to $(x_1, x_2, ..., x_i, x_{i+1})$. If x_{i+1} is impossible to choose, then a return to the vector $(x_1, x_2, ..., x_{i-1})$ occurs, a new element x_i' is selected, and the process of expansion of the solution is repeated for a partial solution $(x_1, x_2, ..., x_{i-1}, x_i')$. If the element x_i' can't be selected, then the expansion process of the solution is repeated for the partial solution $(x_1, x_2, ..., x_{i-2})$ and the

element x'_{i-1} is selected, etc. If it is not possible to select the element x'_i, then the solution can be expanded and the process ends.

6.1.5 Network topological synthesis algorithm

When solving the general problem of topological network synthesis, in addition to choosing the optimal connection scheme for switching nodes, it is necessary to simultaneously solve the problem of route optimization and choice of communication channel capacity.

Since this problem is investigated at each step of the topology synthesis algorithm during the generation of the next graph, the solution to the problem of selecting the capacity of channels and routes should be realized by a high-speed algorithm. Let's consider this heuristic algorithm, which is a special case of choosing fixed routes.

Let's recall that the task of selecting capacities and optimal routing can be represented in this form. Let given: network topology; matrix of information flows $\Lambda = \|\lambda_{ij}\|$; matrix of the cost of renting channels between each pair of network nodes $E = \|E_{ij}\|$. It is necessary to determine the number of communication channels $Y = \|y_{rs}\|$ in each connection (r,s) and the magnitude of the flows $F = \|f_{rs}\|$ in each connection (r,s) so that:

$$\sum_r \sum_s E_{rs} y_{rs} \to \min \tag{6.8}$$

under such restrictions:

1) the delay of the message (package) T_{ij} in any virtual connection (i,j) should not exceed the specified value T;

2) the matrix F must match the matrix Λ;

3) the flow rate in each connection f_{rs} should not exceed the capacity of this connection (r,s);

4) at each vertex in which a certain flow is directed, the only direction in which it will exit from the vertex must be selected.

When choosing an algorithm for solving problem (6.8), it is necessary to take into account the requirements for the algorithm to be sufficiently effective in terms of the cost of computer time for its implementation.

The following is a description of the approximate algorithm.

Step 1. For all pairs (i,j) with a direct route, distribute the flows along these routes. The resulting flows through these communication lines are denoted by F^0.

Step 2. Determine the minimum number of communication channels so that $y_{ij}B \geq f_{ij}^0$.

Step 3. Arrange pairs (i, j) for which there is no direct route in descending order of flows λ_{ij}; mark the resulting list of pairs by Ω.

Step 4. Take the next pair $(i, j) \in \Omega$ and choose the shortest route with the least load.

Step 5. Direct the entire flow λ_{ij} along the selected route.

Step 6. If all pairs of Ω are considered, then go to step 7, otherwise go to step 4.

Step 7. On each pair (i, j) select the number of channels so that $T_{ij} \le T$.

The above-described algorithm for selecting routes and capacities of communication lines is used at each step of solving the general problem of topological optimization. The combinatorial algorithm for solving the general problem is similar to the algorithm for solving the problem of synthesizing a topological structure according to the criteria of cost and reliability.

Here, the lower limit of the number of edges $M_{\min}$ is first determined. For a given number of switching nodes and edges $M = M_{\min}$, all graphs satisfying the constraints (6.5)–(6.8) are studied. For each such graph, the problem of choosing the capacity and the distribution of flows is solved. The graph G_0 for which this task gives the lowest cost is remembered as the optimal solution. Then the number of edges increases by one and the process continues.

In conclusion, let's note that the combinatorial algorithms described in this section use only structural constraints on reliability. It is advisable to supplement the general tasks of topological synthesis with restrictions on the probability of network connectivity, which take into account the failure and restoration of communication channels and switching nodes. In this case, at each step of generating biconnected graphs, it is necessary to check the restrictions on the probability of connectivity. In the presence of high-speed (polynomial) algorithms for assessing the probability of connectivity, this will lead to the rejection of classes of graphs that do not satisfy the restrictions on reliability and a corresponding increase in the speed of the combinatorial algorithm.

6.2 Problems for optimizing the parameters of package switched communication networks

6.2.1 Problem statement of package switched communication networks

The problems of designing package switched communication networks provides for the use of models of multi-pole queuing networks to solve a wide class of problems, which can conditionally be divided into the following 4 groups.

I. The tasks of capacity selection (CS) of communication channels $\{C_i\}$, which are formulated as follows. Specified: network topology in the form of

a graph $G(V,U)$ with N nodes and M channels, intensities of external package flows $\{\gamma_{j,k}\}$, intensities of internal package flows $\{\lambda_i\}$. It is necessary, by changing the channel capacity $\{C_i\}$, to minimize the average delay of packages in the network $\overline{T}$ while limiting the cost of the network:

$$E = \sum_{i=1}^{M} E_i(C_i) \le E_d$$

or to minimize the cost of the network:

$$E = \sum_{i=1}^{M} E_i(C_i)$$

while limiting the average delay time $\overline{T} \le \overline{T}_d$ of packages in it.

II. The problems of the distribution of flows (DF) on the communication network are formulated as follows. Specified: network topology in the form of its graph $G(V,U)$ with N nodes and M channels, intensities of external package flows $\{\gamma_{j,k}\}$ and channel capacity $\{C_i\}$. It is necessary to minimize the average package delay $\overline{T}$ in the network by changing the intensities of internal flows $\{\lambda_i\}$.

III. Problems for selecting capacities and flow allocation (CS DF), which are formulated as follows. Specified: network topology in the form of its graph $G(V,U)$ with N nodes and M channels, intensity of external package flows $\{\gamma_{j,k}\}$. It is necessary to minimize the average delay of packages in the network $\overline{T}$ while limiting the cost of the network:

$$E = \sum_{i=1}^{M} E_i(C_i) \le E_d,$$

or to minimize the cost of the network:

$$E = \sum_{i=1}^{M} E_i(C_i)$$

when limiting the average delay time of packages $\overline{T} \le \overline{T}_d$ in it by changing the channel capacity $\{C_i\}$ and the intensities of internal flows $\{\lambda_i\}$.

IV. The problems of choosing the topology, capacity and distribution flows (TCCDF), which are formulated as follows. Specified: the position of N network nodes and the intensity of the external package flows $\{\gamma_{j,k}\}$ between them. It is necessary to minimize the network cost:

$$E = \sum_{i=1}^{M} E_i(C_i)$$

while limiting the average package delay time $\overline{T} \le \overline{T}_d$ in it by changing the network topology in the form of its graph $G(V,U)$, channel capacity $\{C_i\}$, and internal flow intensities $\{\lambda_i\}$.

6.2.2 Solution of the CS problem by the criterion of the minimum average package delay time in the network with a restriction on its cost

Let's start by considering the linear cost functions of the capacity of network channels, namely:

$$E_i(C_i) = e_i C_i, \quad i = \overline{1,M}, \tag{6.9}$$

where e_i is the channel cost i per unit of capacity for the i-th channel. Let's note that the cost coefficient e_i can randomly vary depending on any channel parameter, for example, e_i is often taken proportional to the physical length of the channel.

When studying the theoretical properties of the optimal capacity choice $\{C_i\}$, let's use the average package delay time $\overline{T}$ as:

$$\overline{T} = \sum_{i=1}^{N} (\lambda_i / \gamma)[1/(\xi C_i - \lambda_i)], \tag{6.10}$$

where $\xi = 1/\overline{n}$ is the reciprocal of the average package length value.

To minimize $\overline{T}$, taking into account the restrictions on the cost of the network:

$$E = \sum_{i=1}^{M} e_i C_i \le E_d,$$

let's compose the Lagrange function:

$$W(\{C_i\}, \chi) = \overline{T} + \chi\left[\sum_{i=1}^{M} e_i C_i - E_d\right], \tag{6.11}$$

where χ is the indefinite Lagrange multiplier.

Obviously, if find the minimum value W for varying capacities $\{C_i\}$, then it will get a solution to the CS problem, since the expression in square brackets

is identically equal to zero, and the factor χ will be determined below. Using the Lagrange method, let's obtain the following system of M equations:

$$\frac{\partial W\left(\{C_i\},\chi\right)}{\partial C_i}=0,\ \ i=\overline{1,M}.$$

Its solution gives an expression for capacity in the form:

$$C_i=\frac{\lambda_i}{\xi}+\frac{1}{\sqrt{\gamma\chi\xi}}\sqrt{\frac{\lambda_i}{e_i}}. \tag{6.12}$$

If y find the Lagrange multiplier χ, then expression (6.12) will be a solution to the CS problem. Let's find χ, making up the constraints by multiplying the last equality by e_i and summing over i:

$$\sum_{i=1}^{M}e_iC_i=\sum_{i=1}^{M}\frac{\lambda_i e_i}{\xi}+\frac{1}{\sqrt{\gamma\chi\xi}}\sum_{i=1}^{M}\sqrt{\lambda_i e_i}. \tag{6.13}$$

It is seen that the left side of equality (6.13) is equal E_d, therefore:

$$\frac{1}{\sqrt{\gamma\chi\xi}}=\frac{E_d-\sum_{i=1}^{M}(\lambda_i e_i/\xi)}{\sum_{i=1}^{M}\sqrt{\lambda_i e_i}}. \tag{6.14}$$

Defining value added E_a as:

$$E_a=E_d-\sum_{i=1}^{n_b}\frac{\lambda_i e_i}{\xi} \tag{6.15}$$

and using (6.14), (6.15) together with (6.12), let's arrive at the optimal solution to the linear CS problem:

$$C_i=\frac{\lambda_i}{\xi}+\left(\frac{E_a}{e_i}\right)\frac{\sqrt{\lambda_i e_i}}{\sum_{j=1}^{M}\sqrt{\lambda_j e_j}},\ \ i=\overline{1,M}. \tag{6.16}$$

With this choice of capacities, each communication channel will have at least capacity λ_i/ξ, that is, its minimum required value, and, in addition, some additional capacity. As follows from formula (6.16), this additional cost is first normalized using the cost coefficient and then distributed over all channels e_i in

proportion to the square root of the weighted traffic intensity $\lambda_i e_i$. This optimal set of capacities is called the square root set of capacities.

If substitute the expression (6.16) for C_i in (6.10), let's obtain the expression for the minimum average delay time:

$$\overline{T} = \frac{\overline{R}}{\xi E_a}\left[\sum_{i=1}^{M}\sqrt{\frac{\lambda_i e_i}{\lambda}}\right]^2, \tag{6.17}$$

where $\overline{R} = \lambda/\gamma$ is the average rank of paths in the network.

This expression defines the minimum average package delay in the network, in which capacities are optimally selected. Let's note that E_a plays an important role here: when $E_a \to 0$ the average message delay increases unlimitedly. If $E_a > 0$, the CS problem has realizing solutions ($\overline{T} < \infty$), which is a condition for the network stability. If $E_a \geq 0$, the problem does not have an implementing solution.

The last two equalities (6.16), (6.17) give a complete solution to the CS problem in the case of linear values.

In the particular case, which is of great importance, the unit costs of the channels can be the same ($e_i = e$, $i = \overline{1,M}$), while it can be given $e = 1$. This case takes place when considering satellite communication channels in which the distance between any two points on the Earth in the satellite zone is the same regardless of the distance between these two points on the Earth's surface. Let's note that in this case, the cost of the network is the sum of all the capacities of its channels $\left(E = \sum_{i=1}^{M} C_i\right)$, which can be denoted by C and assumed that it is expressed in bits per second. Now the two main results of the CS – one for a set of capacities and the other for a delay – are of the form:

$$C_i = \frac{\lambda_i}{\xi} + C\left(1 - \overline{R}_0\right)\frac{\sqrt{\lambda_i}}{\sum_{i=1}^{M}\sqrt{\lambda_i}},\ i = 1,2,...,M,\ \ i = \overline{1,M}, \tag{6.18}$$

$$\overline{T} = \frac{\overline{R}\left(\sum_{i=1}^{M}\sqrt{\lambda_i/\lambda}\right)^2}{\xi C(1 - \overline{R}p)}, \tag{6.19}$$

where p is:

$$p = \frac{\gamma/\xi}{C}.$$

The expression (6.17) for the average delay $\overline{T}$ in this case is very clear. First, let's note that $\overline{T}$ is a strictly increasing function of the average rank of paths $\overline{R}$. In addition, if consider the sum in the numerator $\overline{T}$, it is possible to find that it is minimized over the set $\{\lambda_i/\lambda\}$ when one of these quantities is equal to one, and all the others to zero. The amount in the numerator can't be minimized in this way, since all traffic will go through one channel, and the other channels will be idle, that is, the network will be inefficient. Therefore, it is necessary to send a significant part of the traffic on several high-speed channels and only a small part – on other channels. Typically, communications require at least one input and one output channel for each node.

The above findings relate to fixed route selection procedures, so it might wonder if the use of the route selection procedure, which allows alternatives, will improve. Such a procedure offers more than one path for traffic to a given destination, and in addition, it gives ordering to the advantage of these paths – usually longer paths are less better than a direct path. Therefore, routing procedures, which allow alternatives, lead to an increase in the path length for packages and at the same time try to distribute traffic over many channels, rather than concentrating it on only a few channels. When choosing routes, it allows alternatives, both intuitive rules inferred in the analysis of equality are violated (6.19). At some time periods, the network will not be optimal for traffic transmission. If the inconsistency is significant, it is necessary to introduce an adaptive procedure, allows for alternatives, the choice of routes for finding paths with underloaded capacity at these intervals. Thus, let's conclude that when the intensities of external flows $\{\gamma_{j,k}\}$ are known and constant, then, assuming a linear cost function, it is possible to design optimal networks in the sense of choosing the capacity that exactly matches the network traffic, is guided by a fixed route selection procedure. At the same time, if $\{\gamma_{j,k}\}$ are either unknown or change in time, then it is possible to use the route selection procedure, which allows for alternatives that allow traffic to adapt to an unsuccessful set of capacities.

Let's now return to the more general case of a linear cost function with an arbitrary set $\{E_i(C_i)\}$. With the above minimization of the average delay time $\overline{T}$, a wide variation of values $\overline{T}_i$ is possible. As a result of this, a new CS problem may be considered, in which it is necessary to minimize the function:

$$\overline{T}^{(k)} = \left[\sum_{i=1}^{M} \frac{\lambda i}{\lambda} (\overline{T}_i)^k\right]^{1/k}. \qquad (6.20)$$

The choice of this function is due to the fact that large values $\overline{T}_i$, which are reduced to a large degree, increase it much more than before, so that any minimization procedure with $k > 1$ will reduce the difference between $\overline{T}_i$. If solve this new

problem of optimizing the CS using $\bar{T}^{(k)}$ instead $\bar{T}$, let's obtain a new optimal set of capacities $\left\{C_i^{(k)}\right\}$:

$$C_i^{(k)} = \frac{\lambda_i}{\xi} + \frac{E_f}{e_i} \frac{(\lambda_i e_i^k)^{1/(1+k)}}{\sum_{j=1}^{M} (\lambda_j e_j^k)^{1/(1+k)}}, \quad i = \overline{1, M} \tag{6.21}$$

and expression for characteristics $\bar{T}^{(k)}$:

$$\bar{T}^{(k)} = \frac{(\bar{R})^{1/k}}{\xi E_a} \left[\sum_{i=1}^{M} \left(\frac{\lambda_i e_i^k}{\lambda} \right)^{1/(1+k)} \right]^{(1+k)/k}. \tag{6.22}$$

Let's note that for $k = 1$ the last two results (6.21), (6.22) are reduced to the equalities (6.16) and (6.17) obtained earlier. It is more interesting, of course, to trace how the average delay $\bar{T}$ varies with k. The size $\bar{T}$ worsens (grows) only slightly with increasing k over a wide range of values, and the differences between $\bar{T}_i$ decrease very quickly with increasing k, leading to the desired result with a slight additional increase in the delay.

It is interesting to consider the behavior $C_i^{(k)}$ when $k \to \infty$:

$$\lim_{k \to \infty} C_i^{(k)} = \frac{\lambda_i}{\xi} + \frac{E_a}{\sum_{j=1}^{M} e_j}, \tag{6.23}$$

$$\lim_{k \to \infty} \bar{T}^{(k)} = \frac{\bar{R}}{\xi E_a} \sum_{j=1}^{M} e_j. \tag{6.24}$$

As can be seen from (6.23), (6.24), in this case, the specified set of capacities provides in each channel its minimum required capacity λ_i / ξ plus some constant addition. Let's note that all $\bar{T}_i$ will be the same and this corresponds to the Minimax solution of the CS problem for $k \to \infty$.

When $0 \le k < 1$ a new characteristic $\bar{T}^{(k)}$ behaves diametrically opposite and seeks to worsen (increase) the differences between $\bar{T}_i$. For the case $k \to 0$ let's obtain:

$$\lim_{k \to 0} C_i^{(k)} = \frac{\lambda_i}{\xi} + \frac{\lambda_i E_a}{\bar{R} \gamma e_i}, \tag{6.25}$$

$$\lim_{k \to 0} T^{(k)} = \frac{\bar{R}}{\xi E_a} \sum_{i=1}^{M} e_i. \tag{6.26}$$

Let's note that in this case the capacity is directly proportional at $e_i = 1$ to the traffic intensity $e_i = 1$ in the corresponding channels. Such a set of capacities is called a proportional set of capacities and it is very natural that it should be considered first of all. More interestingly, the value $\overline{T}^{(k)}$ has the same meaning in the two opposite cases ($k \to \infty$ and $k \to 0$), although the sets of capacities are significantly different.

Let's now carry out a generalization that goes beyond the linear cost functions of capacities.

For example, with a logarithmic cost function:

$$E = \sum_{i=1}^{M} e_i \log a C_i. \tag{6.27}$$

CS problems are given by a proportional set of capacities.

For a power cost function:

$$E = \sum_{i=1}^{M} e_i C_i^a, \quad 0 \le a \le 1, \tag{6.28}$$

It is possible to obtain such equations for C_i:

$$C_i - \frac{\lambda_i}{\mu} - g_i C_i^{(1-\alpha)/2} = 0, \quad i = \overline{1, M}, \tag{6.29}$$

where

$$g_i = \left(\frac{\lambda_i}{\gamma \xi \alpha \chi e_i} \right)^{1/2}, \tag{6.30}$$

and the indefinite Lagrange multiplier χ should be chosen to satisfy the constraints $E \le E_d$. These equations for $\{C_i\}$ can be solved using any iterative algorithm.

6.2.3 Solution of the CS problem by the criterion of the minimum cost of the network while limiting the average package delay time

To solve this dual CS problem in the case of a linear cost function (6.9), let's use the Lagrange function:

$$W(\{C_i\}, \chi) = \sum_{i=1}^{M} e_i C_i + \chi \left(\frac{1}{\gamma} \sum_{i=1}^{M} \frac{\lambda_i}{\xi C_i - \lambda_i} - \overline{T}_d \right), \tag{6.31}$$

where χ is the indefinite Lagrange multiplier.

Minimization of the Lagrange function (6.31) with their arguments:

$$\begin{cases} \dfrac{\partial W\left(\{C_i\},\chi\right)}{\partial C_i} = 0,\ i = \overline{1,M}; \\ \dfrac{\partial W\left(\{C_i\},\chi\right)}{\partial \chi} = \dfrac{1}{\gamma}\sum\limits_{i=1}^{M}\dfrac{\lambda_i}{\xi C_i - \lambda_i} - \overline{T}_d = 0 \end{cases} \tag{6.32}$$

leads to such an optimal solution to the dual CS problem:

$$C_i = \frac{\lambda_i}{\xi} + \frac{\lambda_i \sum\limits_{j=1}^{M}\sqrt{\lambda_j e_j}}{\xi\gamma\overline{T}_d\sqrt{\lambda_i e_i}},\quad i = \overline{1,M}, \tag{6.33}$$

which is also called the square root rule.

It corresponds to the minimum cost of the network as a whole:

$$E = \sum_{i=1}^{M}\frac{\lambda_i e_i}{\xi} + \frac{1}{\gamma\overline{T}_d}\left(\sum_{i=1}^{M}\sqrt{\frac{\lambda_i e_i}{\xi}}\right)^2. \tag{6.34}$$

6.2.4 Solution of the DF problem by the criterion of the minimum average package delay time in the network

The maximum flow theorem and minimum cross section is the basis on which the theory of flows in networks is built. It considers the problem of package flows with a nonlinear objective function. For each j-th and k-th node, it is necessary to deliver the specified traffic $\gamma_{j,k}$ over the network from the node-source j to the destination node k. This DF problem requires minimizing the nonlinear function of the average delay $\overline{T}$ with respect to the intensities of internal flows $\{\lambda_i\}$ in order to satisfy the external requirements for the intensities of external flows, and minimization is carried out under the assumption that the channel capacities $\{C_i\}$ are specified. In addition, it is not necessary to violate the usual law of conservation of flows in each node. Further, there is a limitation on the capacity of each channel, which consists in the fact that the flow λ_i/ξ in channel i must be non-negative and less than the capacity, i. e. $0 \le \lambda_i/\xi \le C_i$. This restriction shows that the average delay $\overline{T}$ has an interesting and obvious property of unlimited growth when a flow tends to the capacity of the corresponding channel, that is, when many flows in the network tend to the upper limit for these

flows, it is determined by the capacity restrictions. In mathematical programming speech, this means that the characteristic includes an additional restriction $\bar{T}$ on capacity as a function of the penalty. This important property ensures the feasibility of the decision to limit the capacity when using any minimization method, is presented as a sequence of «small steps» and at the initial stage operates with the solutions that are being realized. Thus, if to start with the solutions being realized, then it is possible to neglect the capacity limit, and as a result, the DF task, which looks like a constrained optimization problem, will actually be a simpler problem without restrictions on optimizing package flows.

Let's begin by considering expression (6.10) for the average delay $\bar{T}$. Let's note that this characteristic is a resolution in the sense that it is expressed simply as the sum of applications, each of which depends only on the flow in one channel. In addition, it follows from (6.10) that:

$$\frac{\partial \bar{T}}{\partial(\lambda_i / \xi)} = \frac{C_i}{\gamma[C_i - (\lambda_i / \xi)]^2}, \quad i = \overline{1, M}. \tag{6.35}$$

From this it can be seen that $\partial \bar{T} / \partial(\lambda_i / \xi) > 0$ and $\partial^2 \bar{T} / \partial(\lambda_i / \xi)^2 > 0$ for all i, while satisfying the restrictions on the capacity of the channels. Thus, it is possible to conclude that $\bar{T}$ is a convex function of flows $f_i = \lambda_i / \xi, \ i = \overline{1, M}$. In addition, the set of realized flows is a convex polyhedron. So, if the DF problem has a realized solution, then any local minimum is a global minimum for $\bar{T}$. So, any method of finding a local minimum can be used to solve the problem of finding a global minimum.

An example would be the flow deflection (FD) method, which is designed to look for such a global minimum. The best way to understand the FD method is if first recognize the important concept of flow along the shortest paths. Let's suppose there is a network, each channel of which has a weight l_i assigned to it. In such a network, it is natural to find the path with the smallest total weight between the source node *j* and the destination node *k* and try to send the necessary flow with intensity $\gamma_{j,k}$ in this way. If to use this for all pairs (j,k), then the result is a flow called a flow with shortest paths. To find many short paths $\{m_{j,k}^{\min}\}$, it is possible to use the Floyd algorithm.Returning now to the DF problem, let's note that the essence of the FD method is related to comparing the «length» with the *i*-th channel, the value of which is given by equality (6.35). It is clear that this is a linear growth rate $\bar{T}$ with an infinitely small increase in flow in the *i*-th channel. Such «lengths» or «weights» can then be used in formulating the problem of finding flows along the shortest routes, and the paths leading out are the best to reduce $\bar{T}$ along which some part of the flow can be rejected. The question now is how much of the output should be rejected along these new paths. After it is

determined, it is possible to repeat the process, regaining new «lengths» based on updated flows, solving the new problem of finding flows along the shortest routes, defining the corresponding part of the flow, is rejected, etc. This iterative procedure continues until an acceptable characterization is obtained.

Let's now give a specific algorithm that uses these ideas. To do this, let's introduce the flow vector at some n-th iteration of the algorithm:

$$\vec{f}^{(n)} = (f_1^{(n)}, f_2^{(n)}, ..., f_M^{(n)}), \tag{6.36}$$

where the i-th component $f_i^{(n)}$ of which is the total flow through the i-th channel at the n-th iteration. Let's assume that the initial flow $\vec{f}^{(0)}$ is realized. Now it is possible to submit the following step-by-step description of the optimal DF algorithm for selecting routes.

Optimal DF algorithm for route selection

Step 1. Let the iteration number $n = 0$.

Step 2 (length calculations). Find the «lengths» of channels for the flow $\vec{f}^{(n)}$:

$$l_i = \frac{C_i}{\gamma[C_i - f_i^{(n)}]^2}, \quad i = \overline{1, M}.$$

Step 3. Find an additional cost factor for this flow b_n:

$$b_n = \sum_{i=1}^{M} l_i f_i^{(n)}.$$

Step 4 (calculation of flows along the shortest routes). Solve the problem of finding flows along the shortest routes using lengths l_i. Let φ_i be the resulting flow along the i-th channel if the entire flow is directed along these shortest paths. Denote the vector of such flows by

$$\vec{\varphi} = (\varphi_1, \varphi_2, ..., \varphi_M).$$

Step 5. Find β_n – additional cost factor for flows along the shortest route:

$$\beta_n = \sum_{i=1}^{M} l_i \varphi_i.$$

Step 6 (stop rule). If $b_n - \beta_n < \varepsilon$, where $\varepsilon > 0$ – the tolerance is properly selected, then the algorithm terminates. Otherwise, go to step 7.

Step 7 (calculating of deviating flow part). Find the value α from the interval $0 \le \alpha \le 1$, for which the flow $(1-\alpha)\vec{f}^{(n)} + \alpha\vec{\varphi}$ minimizes $\overline{T}$. Denote this optimal value by α' in view of which can be found, for example, using the Fibonacci method.

Step 8 (determination of flow deviation). Find $\vec{f}^{(n+1)} = (1-\alpha')\vec{f}^{(n)} + \alpha'\vec{\varphi}$.

Step 9. Let the iteration number $n = n+1$ and go to step 2.

Let's note that the most important steps of this algorithm are step 2 (length calculation), step 4 (calculating the flows along the shortest routes), step 6 (stopping rule), step 7 (calculating a part of the deviating flow) and finally step 8 (determining the flow deviations). Let's note that the deviation of the flow is made so that there is a maximum decrease in average $\overline{T}$. In the general case, this leads to a deterministic route selection procedure and allows alternatives.

Let's now turn to the problem of finding the initial realized flow $\vec{f}^{(0)}$. Let's suppose an external flow is specified γ. Let's introduce the scale factor h so that the value $h\gamma$ is equal to the intensity of the flow with which we are dealing with a given value h. A step-by-step description of the algorithm for finding the initial flow has the following form.

The algorithm for finding the initial realized flow

Step 1. Let $h_0 = 1$ and assume that $\vec{f}^{(0)}$ – the solution to the problem of finding flows along the shortest routes in a network with «lengths» $l_i = 1/\gamma C_i$ at zero flow. At this point, the entire flow $h_0\gamma$ is routed over the network. Denote by $f_i^{(0)}$ the flow in the i-th channel at this stage. Let the iteration number $n = 0$.

Step 2. Let:

$$\sigma_n = \max_i \left(\frac{f_i^{(n)}}{C_i} \right).$$

If $\sigma_n / h_n < 1$, then let $\vec{f}^{(0)} = \vec{f}^{(n)} / h_n$ and the algorithm finishes with the initial realized flow. If $\sigma_n / h_n \ge 1$, then let $h_{n+1} = h_n[1 - \varepsilon_1(1-\sigma_n)] / \sigma_n$, where ε_1 is the corresponding accuracy parameter $(0 < \varepsilon_1 < 1)$.

Step 3. Let $\vec{g}^{(n+1)} = (h_{n+1} / h_n)\vec{f}^{(n)}$. This is a package of realized flows and carries full traffic with intensity $h_{n+1}\gamma < 1$.

Step 4. Carry out the deviation operation on the flow $\vec{g}^{(n+1)}$, that is, perform steps 2, 4, 7, and 8 of the FD algorithm and find $\vec{\varphi}$ (the flow with short routes with lengths based on the flow $\vec{g}^{(n+1)}$ and the optimal value α (that is α') such that the flow $\vec{f}^{(n+1)} = (1-\alpha')\vec{g}^{(n+1)} + \alpha'\vec{\varphi}$ minimizes $\overline{T}$. If $n = 0$ then go to step 6, in other cases go to step 5.

Step 5. If

$$\left| \sum_{i=1}^{M} l_i(\varphi_i - g_i^{(n+1)}) \right| < \theta \quad \text{and} \quad |h_{n+1} - h_n| < \delta,$$

where θ and δ are the selected positive tolerances, then the algorithm finishes work and the problem does not have a solution realized with tolerances θ and δ. Otherwise, go to step 6.

Step 6. Let the iteration number $n = n+1$ and go to step 2.

This algorithm either finds the initial realized flow, or explains that the problem does not have solutions that are realized within the selected tolerances.

The FD method provides the optimal choice of routes for traffic on the network and is relatively efficient from the point of view of calculations, however, it turns out that there is a simpler suboptimal method that gives a fixed procedure for selecting routes and often leads to very good results, requiring much less calculation. This suboptimal method bypasses the task of determining which part of the flow should be rejected, it simply decides whether to reject the entire flow, or not to reject anything for each traffic intensity.

This approximation is based on the fact that fixed route selection procedures have good properties for short average path lengths and very concentrated traffic. The class of networks for which such a fixed route selection algorithm is effective is called the class of large and balanced networks. It is said that a network is large if it has a large number of nodes, and a network is balanced if the intensities $\gamma_{j,k}$ for different pairs of nodes are practically the same.

Let's now consider a suboptimal algorithm for finding flows directed by a fixed routing selection procedure. Again, let's suppose that the initial realized flow $\vec{f}^{(0)}$ is known, which is directed by a fixed routing procedure.

Algorithm for finding flows directed by a fixed routing procedure

Step 1. Let the iteration number $n = 0$.

Step 2. Using the flow $\vec{f}^{(n)}$, find the set of short routes for the metric:

$$l_i = \frac{C_i}{\gamma[C_i - f_i^{(n)}]^2}, \quad i = \overline{1,M}.$$

Step 3. Let $\vec{g} = \vec{f}^{(n)}$. For each external flow having intensity $\gamma_{j,k}$, $j = \overline{1,N}$, $k = \overline{1,N}$ carry out such steps.

Step 3 a. Let $\vec{v}$ is the flow obtained from $\vec{g}$ by deviating the entire flow with intensity $\gamma_{j,k}$ from its path in the flow $\vec{f}^{(n)}$ to the shortest path $m_{j,k}^{\min}$.

Step 3, b. If two statements are true: $m_{j,k}^{\min}$ is the realized flow, and $\overline{T}$ which takes place at $\vec{v}$ is strictly less than $\overline{T}$ that which takes place at $\vec{g}$, then go to step 3, *c*. Otherwise, go to step 3, *d*.

Step 3, c. Let $\vec{g}$.

Step 3, d. If all flows with intensities $\gamma_{j,k}$ have been considered, then go to step 4. In other cases, select any unconsidered flow with intensities $\gamma_{j,k}$ and go to step 3, *a*.

Step 4. If $\vec{g} = \vec{f}^{(n)}$, then the algorithm terminates because it can no longer improve the flow sent by the fixed routing procedure. In other cases, let $\vec{g} = \vec{f}^{(n+1)}$, $n = n+1$, and go to step 2.

This algorithm converges after a finite number of steps, because it is necessary to consider only a finite number of flows directed by a fixed routing selection procedure; the same flow is not considered twice through the stopping condition of the algorithm. The initial flow $\vec{f}^{(0)}$ is implemented and directed by a fixed routing selection procedure; it can be found by a method that is similar to that used in the optimal FD algorithm.

6.2.5 Solution of the CS and DF problems according to the criterion of the minimum average package delay in the network with a restriction on its cost

Since it is necessary to choose capacities $\{C_i\}$ again, it is believed that fixed routing procedures (internal flow intensities $\{\lambda_i\}$) should also be optimal. This is correct with the linear cost function of the channels $E_i(C_i) = e_i C_i,\ i = \overline{1,M}$, when fixed routing procedures are also optimal. This is true due to the concavity property of the dependence on them of the mean delay function $\overline{T}$ (6.10). In addition, local minima $\overline{T}$ exit the flows behind short routes – a subclass of flows directed by the fixed routing procedure, since the minima must be located at the nodes of the convex polyhedron of many flows are realized.

The approach to finding these local minima is to, starting with the initial flow that is being realized, obtain the optimal set of capacities with linear cost functions, using the flow deviation (FD) algorithm, find the optimal flows, repeat the solution to the CS problem for these new flows and continue the iteration between solving the CS and the DF problem to find a local minimum. Let's note that the FD algorithm will be especially simple in the case when the parameter α value is always equal to unity, corresponding to the flow directed by the fixed routing selection procedure. Thus, when a known initial flow $\vec{f}^{(0)}$ is realized, a step-by-step description of a suboptimal algorithm for solving the CS DF problem is as follows.

Step 1. Put the iteration number $n = 0$.

Step 2. Execute the CS algorithm for the flow $\vec{f}^{(n)}$ and find the optimal set of capacities using linear values.

Step 3. Using the lengths $l_i = \partial\overline{T}/\partial(\lambda_i/\xi)$, execute the FD algorithm when $\alpha = 1$ exiting at each step. The resulting optimal flow is designated $\vec{f}^{(n+1)}$ later.

Step 4. If $\overline{T}$ for the flow is more than or equal $\overline{T}$ to the flow $\vec{f}^{(n)}$, then the algorithm stop and the flow $\vec{f}^{(n)}$ gives a local minimum $\overline{T}$. Otherwise, let the iteration number $n = n+1$ and go to step 2.

The algorithm converges, since there are only a finite number of flows along the shortest routes.

After completing step 2 (CS algorithm), the channel capacities are set by equality (6.16), and the average package delay time is set by equality (6.17), where λ_i are replaced by $\lambda_i^{(n)}$.

In this case, the conditional channel lengths are given by the equality:

$$l_i = \frac{\partial \overline{T}}{\partial(\lambda_i/\xi)} = \frac{\overline{R}\sum_{j=1}^{M}\sqrt{\lambda_j e_j/\lambda}}{E_a}\left[\sqrt{\frac{e_i}{\lambda\lambda_i}} + \frac{e_i}{\xi E_a}\sum_{j=1}^{M}\sqrt{\frac{\lambda_j e_j}{\lambda}}\right]. \qquad (6.37, \text{a})$$

It follows that $l_i \geq 0$ and negative cycles can't exist, which requires the algorithm for finding the shortest paths. Let's also note that $\lim_{\lambda_i \to 0} l_i = \infty$; this means that if at the end of the iteration the flow, and, consequently, capacity becomes zero.

When using the considered algorithm, two tasks take place:

1) find the initial realized flow $\vec{f}^{(0)}$;
2) looking through many local lows, find a global minimum.

Both of these tasks can be solved by repeating the CS and DF algorithms for many different initial flows. Each initial flow is realized, found by randomly assigning conditional output «lengths» to the channels. For each destination, an algorithm is then found for finding the flows along the shortest paths that uses the «lengths» selected in this way, and checking the conditions $E_a > 0$ for this flow. If the condition is satisfied, then the initial flow is found to be realized, and wit is possible to proceed to the CS and DF algorithm. Otherwise, the output flow is discarded and an attempt is made to randomly assign many other conditional «lengths».

6.2.6 Solution of the CS and DF problem according to the criterion of the minimum cost of the network with a restriction on the average package delay

The above-described suboptimal CS and DF algorithm can be applied to the solution of this problem if the definition of «length» $l_i = \partial\overline{T}/\partial(\lambda_i/\xi)$ is replaced by a new definition $l_i = \partial E/\partial(\lambda_i/\xi)$. In the case where the cost functions of the channel capacity are linear, that is $E_i(C_i) = e_i C_i$, $i = \overline{1,M}$, it can be shown that the solution of the CS problem at the end of step 2 will have the form:

$$C_i = \frac{\lambda_i}{\xi} + \left(\frac{\lambda_i}{\xi\gamma\overline{T}_d}\right)\frac{\sum_{j=1}^{M}\sqrt{\lambda_j e_j}}{\sqrt{\lambda_i e_i}},$$

and the minimum cost satisfying the maximum average delay limit is:

$$E = \sum_{i=1}^{M} \frac{\lambda_i e_i}{\xi} + \frac{1}{\gamma T_d}\left[\sum_{i=1}^{M}\sqrt{\frac{\lambda_i e_i}{\xi}}\right]^2.$$

It can be shown that this function with intensity concave in flows $\{\lambda_i\}$ and any flow corresponds to the local minimum of the CS problem, also a flow with short routes, and also that the value α in the FD algorithm is always equal to one. These three properties are also valid for any concave cost function of bandwidth under constraint $\overline{T} \le \overline{T}_d$. The expression for the conditional length of the channels in this problem has the form:

$$l_i = \frac{\partial E}{\partial(\lambda_i/\xi)} = e_i\left(1 + \frac{\sum_{j=1}^{M}\sqrt{\lambda_j e_j}}{\gamma \overline{T}_d \sqrt{\lambda_i e_i}}\right), \qquad (6.37,b)$$

where e_i is the slope of the bandwidth cost curve of the i-th channel, linearized at the current capacity value.

Let's note again that all the conditional lengths of the channels are not negative, and therefore, negative cycles are not. In addition, let's note that, as before, the conditional length of the channel tends to infinity when the flow in this channel tends to zero. Here again, local minima are found, and therefore, it is necessary to randomize the search in order to find several such minima.

6.2.7 Solution of the CS and DF problem according to the criterion of the minimum cost of the network with a restriction on the average package delay

Let's now turn to the complete design problem – TCC DF problem, which let's consider in its form, which consists in minimizing the cost E for a given average package delay $\overline{T} \le \overline{T}_d$. Let's consider an approximate (heuristic) solution, which, in all likelihood, is in 5–10 % of the optimum and significantly reduces computational difficulties.

The heuristic solution of the TCC DF problem has an iterative form. It uses the fact that the channels can be eliminated and therefore topological changes will be made as the CS and DF algorithm is executed. Such a shift of the edges (channels) is due to the fact that the cost E is a function concave in the flows $\{\lambda_i\}$.

Therefore, an iterative suboptimal algorithm is considered called the concave edge removal method (CERM).

Step 1. Select the initial topology, for example, fully connected.

Step 2. For each channel, according to the topology, conduct a linear approximation. At each iteration in the next step, use the value for capacity, linearize around the flow value for this channel.

Step 3. Run the CS and DF algorithm. If the connection restriction is violated during any iteration, then stop optimizing and go to step 4; otherwise, run the CS and DF algorithm to the end and then go to step 4.

Step 4. Discretize the continuous capacities obtained by using the suboptimal solution to the CS and DF problem. For example, continuous capacity can be rounded to the nearest discrete value with acceptable values $\lambda_i/\xi < C_i$, such that the condition $\overline{T} \le \overline{T}_d$ continues to be satisfied for it. In this case, obviously, the total cost E will change.

Step 5. Perform the final flow optimization by applying the FD algorithm and, if necessary, even adjusting the channel capacities and the intensities of internal flows.

Step 6. Repeat steps 3–5 for a series of random initial flows, realized by randomly selecting the initial lengths with flows directed along the shortest routes.

Step 7. Repeat steps 1–6 for a series of initial topologies.

Let's consider some other approaches to the TCC DF problem. One of these methods is called the edge replacement method. Based on an arbitrary topology that is implemented, the class of its local transformations (replacement of edges) is determined, in which one edge is eliminated, and some new edge is added, so that the biconnection is preserved. Then, the simplified CS and DF problem is solved with the help of the minimum coordination procedure; if the result is improvement, the transformation is fixed, otherwise it is canceled. This procedure is performed until the set of local transformations has been exhausted.

The development of the method of replacing edges is the method of saturation of sections. This method narrows down many local transformations to those that are suitable for improving capacity-cost characteristics. The procedure begins with a tree-like or some other loosely connected topology and a critical section is found that arises as a result of solving the joint venture problem. Then a channel is added to this section or the capacity of some channel from this section is increased. This is repeated until the topology of the products sold is obtained. Then the procedure continues, in addition, at each step, the least used network channel is eliminated if bilingualism persists. After a given number of iterations, the algorithm ends.

The success of all these heuristic algorithms is based on the availability of effective topology analysis methods and the random generation of several initial topologies that allow the search for many local minima.

6.3 Problems of multi-criteria optimization of communication networks

6.3.1 Selection of the best options for a data network taking into account a set of quality indicators

Modern communication networks, regardless of their organization and type of information transmitted, are becoming increasingly complex and are determined by the totality of technical and economic requirements, which are evaluated by the values of the corresponding quality indicators during their creation and operation. As a rule, there are a number of acceptable design decisions and it is necessary to choose the best (optimal by a predetermined criterion) in the problems of long-term planning, design and control of communication networks, taking into account the totality of quality indicators. Therefore, it is relevant to use, along with scalar methods, multicriteria optimization methods for making optimal design decisions.

The features of applying the multi-criteria optimization methodology when choosing the optimal design options for a package switched data network based on a set of quality indicators are considered. The obtained Pareto optimal network options, among which the only design solution, selected using the conditional advantage criterion.

In the considered problem, quality indicators are determined, which are determined by the delivery time and the probability of package loss in the framework of a datagram message transmission. These quality indicators are interconnected and are antagonistic, that is, when improving, the value of one of the indicators of another quality indicator deteriorates. This task of choosing the optimal design options for a data transmission network is relevant for practical applications that are critical to the timely delivery of messages, in particular, in video and voice messaging systems, bank terminal systems, alarm systems, and troubleshooting systems in communication networks.

During the research, a mathematical model of a fully connected topology of a data transmission network was built.

The structure of the mathematical model of the network includes simulators of message sources, procedures for packing messages into packages and transmitting them over communication channels, routing and maintenance procedures in switching nodes, error simulators in communication channels. A message source was modeled with a Poisson distribution law and various application intensities. It was also assumed that the simulation of various delays in the transmission of packages associated with the final propagation speed of the signals

in the communication channels, the fixed bandwidth of the channels, as well as the time the packages were in the queue for transmission over the communication channels.

Various network options were implemented, which differed in the disciplines of servicing data packages in queues, routing methods for package transmission, and the size of the transport connection window.

In this example, thirty-six valid network options were specified. As a result of simulation for each network option, estimates of quality indicators are found: average package delivery time $k_1 = \overline{T}$ and average probability of message loss $k_2 = \overline{P}$. An admissible set of network operation options was filed in the criteria space for evaluating quality indicators normalized to maximum values (Fig. 6.1). Here, a subset of the Pareto optimal network options is highlighted by eliminating the certainly worst cases according to the Pareto criterion. The Pareto subset corresponds to the lower left boundary of the set of valid options, including options: 1, 10, 11, 13, 17, 20.

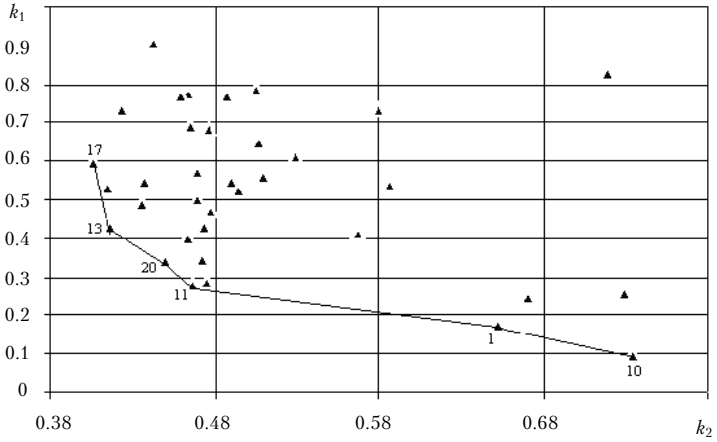

Fig. 6.1 Selection of Pareto optimal options for a data network in criteria space

Among the Pareto optimal network options, the only option was chosen using the conditional criterion of advantage – subject to a minimum scalar value function in the form of a resulting quality indicator $k_p = C_1 k_1 + C_2 k_2$. For the case $C_1 = 0.4$, $C_2 = 0.6$, the selected network operation option is numbered 11. This is a communication network option for which the discipline for servicing applications is random, the routing method is uniform according to weight, the size of the transmission «window» is 8.

6.3.2 Optimization of the nominal planning of cellular communication networks, taking into account the totality of quality indicators

Let's consider the practical features of the application of the multicriteria optimization methodology for the nominal planning of cellular communication networks (CCN).

Finding the best CCN options for nominal CCN planning, taking into account the totality of quality indicators, includes the following stages:

- assignment of the initial set of network options that differ in data on the territory of the served dedicated frequency band, the number of subscribers and etc.;
- highlighting the set of acceptable options, taking into account restrictions on the structure and parameters of networks, as well as restrictions on the values of quality indicators;
- selection of a subset of Pareto optimal network options using the unconditional advantage criterion;
- analysis of the obtained Pareto optimal network options, evaluation of their multidimensional potential characteristics and multidimensional diagrams of the exchange of quality indicators;
- selection of the only network option from Pareto subset using conditional advantage criterion.

A lot of acceptable second-generation CCN options were generated, determined by different data on the planned number of subscribers in the network, the size of the territory served by the activity of subscribers, the allocated frequency band, the size of the clusters, the power of the base station (BS) transmitters, the acceptable probability of blocking calls, the percentage of time deterioration in communication quality.

At the same time, the main technical parameters of CCN were calculated:

I. The total number of frequency channels allocated for the CCN deployment:

$$N_k = \text{int}(\Delta F / F_k), \tag{6.38}$$

where F_k is the frequency band occupied by one CCN frequency channel.

II. The number of radio frequencies necessary to serve subscribers in one sector of each cell:

$$n_S = \text{int}(N_k / C \cdot M). \tag{6.39}$$

III. The value of the allowable telephone load in one sector of one cell or in cell (for a BS having antennas with a circular radiation pattern), which is determined by the relations:

$$A = n_O\left[1-\sqrt{1-\left(P_{bl}\sqrt{\pi n_O/2}\right)^{1/n_0}}\right] \text{ at } P_{bl} \le \sqrt{\frac{2}{\pi n_O}}; \tag{6.40}$$

$$A = n_O + \sqrt{\frac{\pi}{2} + 2n_O \ln\left(P_{bl}\sqrt{\pi n_O/2}\right)} - \sqrt{\frac{\pi}{2}} \text{ at } P_{bl} > \sqrt{\frac{2}{\pi n_O}}, \tag{6.41}$$

where $n_O = n_S n_a$ is the number of subscribers who can simultaneously use one frequency channel.

IV. The number of served BS subscribers and which depends on the number of sectors, the allowable telephone load and the activity of subscribers:

$$N_{aBTS} = M\,\mathrm{int}(A/\beta). \tag{6.42}$$

V. The required number of BS in a given service area:

$$N_{BTS} = \mathrm{int}(N_a / N_{aBTS}). \tag{6.43}$$

VI. The radius of the cell, provided that the load is distributed throughout the area evenly:

$$R = \sqrt{\frac{1.21S_0}{\pi N_{BTS}}}. \tag{6.44}$$

VII. The size of the protective distance between BTS with the same frequency channels, provided that the load is distributed throughout the area moderately:

$$D = R\sqrt{3\mathrm{C}}. \tag{6.45}$$

VIII. The probability of error during the communication session:

$$P_{er} = \frac{1}{(\sqrt{3C}-1)^{2k}}. \tag{6.46}$$

IX. The effectiveness of the use of the radio spectrum, determined by the number of active subscribers per unit frequency band:

$$\gamma = 1.21\frac{S_0}{\pi R^2 F_k C}. \tag{6.47}$$

As a result of the calculations, an initial (nominal) frequency-territorial plan was developed. An example of nominal CCN planning is considered, in which the following quality indicators were selected: error probability, network capacity, number of base stations in the network, radio frequency spectrum efficiency, blocking probability, coverage area. For each variant of CCN, estimates of the values of quality indicators were found, their normalization to maximum values and reduction to a comparable form. Finding a subset of the Pareto optimal CCN options was performed in the space of estimates of the introduced quality indicators.

To implement the main stages of choosing the optimal CCN options using the multicriteria optimization methodology, a special software package was created. This software package performs the formation of the set of valid CCN variants using the morphological approach, the selection of a subset of the systems variants that are optimal according to the Pareto criterion, and the narrowing of the Pareto subset to a single variant with the introduction of a conditional advantage criterion.

Using the software package, the initial set of 100 CCN options has been formed. For each option, estimates of the values of these six quality indicators (marked with a corresponding sign in the windows on the interface ∨) were found. In the criteria space, a subset of Pareto optimal options was allocated, including 71 CCN variants. At the same time, 29 unconditionally worst CCN variants were rejected by the Pareto criterion (marked with 8 on the interface). With a minimum conditional criterion of advantage in the form of a weighted sum of the values of the selected quality indicators from Pareto subset, the only option – 72 is selected. This CCN option is characterized by the following data: number of subscribers in the network – 30,000; area of the territory served by – 320 sq. km; subscriber activity – 0.025 Erlang; bandwidth – 4 MHz; acceptable call blocking probability – 0.01; percentage of the time of deterioration in communication quality – 0.07; service density – 94 act. subscriber/sq. km; cluster size – 7; number of base stations in the network – 133; number of subscribers served by one BS – 226; radio frequency spectrum efficiency – $1.614 \cdot 10^{-4}$ act. subscriber/Hz; telephone load – 3.326 Erlang; probability of error – $5.277 \cdot 10^{-7}$.

As a result of Pareto optimization, multidimensional exchange diagrams (MEDs) of quality indicators were also obtained. To illustrate, some MEDs are shown in Fig. 6.2. Each MED point determines the potentially best values of each of the indicators that can be achieved with fixed but arbitrary values of other quality indicators.

The features of the application of the multicriteria optimization methodology when planning the CCN transport network, taking into account the totality of quality indicators, are also considered. At the same time, quality indicators were used, taking into account: the length of the relay span, the total network length, the used and reserve bandwidth; span reliability; transmission speed; frequency band; the probability of bit erroneous reception (BER), cost characteristics, etc.

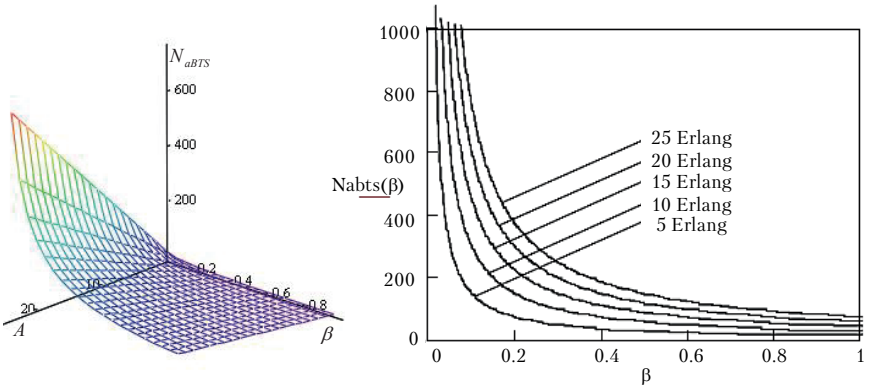

Fig. 6.2 Multidimensional diagrams of the exchange of quality indicators (The numbers of subscribers served by one BS, load, subscriber activity) for CCN

6.3.3 Optimal routing in communication networks, taking into account a set of quality indicators

A multiservice communication network is a complex system with many elements, and to ensure high quality service for various types of traffic, the optimal task is optimal routing, taking into account the totality of quality indicators. Therefore, there is a need to apply the methodology of multicriteria optimization when planning routing in such communication networks.

The problem of optimal routing, taking into account the totality of quality indicators, is represented by the model $\{X, F\} \to x^*$, where $X = \{x\}$ is the set of acceptable options for routing; $F(\bullet)$ is objective selection function; x^* is the optimal solution to the routing problem. A multi-criteria approach requires the decomposition of the objective function $F(\bullet)$, that is, its equivalent representation using a combination of individual selection functions $F_\nu(x), \nu = 1,\ldots,N$.

In this case, the following multicriteria routing problem can be formulated. A set of feasible solutions (routes) is given on the final graph of the network $G = (V, E)$, where V is the set of nodes, E is the set of communication lines. A valid set of routes are those subgraph $x = (V_x, E_x)$ solutions $x \in X$ for the graph $G = (V, E)$ that satisfy the constraints $V_x \in V$, $E_x \in E$. It is assumed that a vector objective function $\tilde{F}(x) = (F_1(x),\ldots,F_\nu(x),\ldots,F_m(x))$ is set on the set X, the components of which determine the values of the corresponding quality indicators of routes k_ν. Route quality indicators, as a rule, are interconnected and antagonistic. It is necessary to find the best route options for the aggregate of quality indicators. The solution to this problem is a subset of the Pareto optimal

routing options that correspond to the optimum of individual objective functions $F_1(x),\ldots,F_\nu(x),\ldots,F_m(x)$.

Each route is determined by the appropriate combinations of communication lines $E_x \in E$ and is characterized by a combination of quality of service indicators k_ν, $\nu = \overline{1,m}$ and their corresponding separate objective functions $F_1(x),\ldots,F_\nu(x),\ldots,F_m(x)$.

The choice of optimal routes, taking into account the totality of quality indicators, consists of highlighting a subset of Pareto optimal routing options. A route option $\tilde{x} \in X$ is Pareto optimal if there is no other route $x^* \in X$ for which inequalities hold $F_\nu(x^*) \le F_\nu(\tilde{x})$, $\nu = 1,\ldots,m$, and at least one of them is strict. When comparing routes using this vector criterion, the advantages from the set of acceptable options exclude the absolutely worst route options and remain incomparable – Pareto optimal route options. In particular, the weighting method can be used to find the Pareto optimal routing options. It reduces to finding the extreme values of the scalar objective function of routes for various admissible combinations of coefficient values λ_i $\left(\lambda_i > 0,\ \sum_{i=1}^{\nu}\lambda_i = 1\right)$.

$$\underset{\text{var}(\lambda_1,\lambda_2,\ldots,\lambda_\nu)}{extrem}\left(F_p(x) = \sum_{\nu=1}^{m}\lambda_\nu F_\nu(x)\right). \tag{6.48}$$

Some practical features of solving the multicriteria routing problem are considered using the example of a communication network structure (Fig. 6.3).

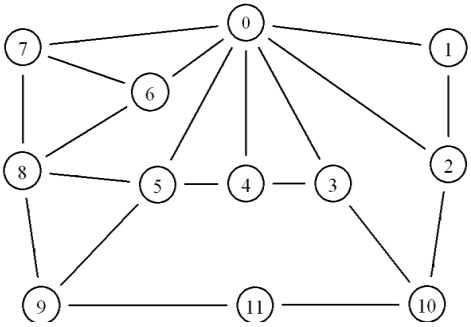

Fig. 6.3 The structure of the investigated communication network

Information is transmitted from node 0 to all other nodes. The following indicators of the quality of communication lines have been introduced: package

delay time, package loss rate, cost of using a communication line. The value of the normalized to maximum values of the quality indicators of communication lines are given in Table 6.1.

Table 6.1

Normalized values of the quality indicators of communication lines

Communication line	Transmission delay time	Package loss rate	Communication line cost
0–1	0.676	1	0.333
0–2	1	0.25	1
0–3	0.362	1	0.333
0–4	0.381	0.25	1
0–5	0.2	1	0.333
0–6	0.19	1	0.333
0–7	0.571	0.25	1
7–6	0.4	0.25	0.333
7–8	0.362	0.25	0.667
8–6	0.314	0.5	0.5
8–5	0.438	0.25	0.333
8–9	0.248	0.5	0.333
9–5	0.257	0.25	1
9–11	0.571	0.25	0.667
11–10	0.762	0.25	0.333
5–4	0.381	0.25	0.667
2–10	0.457	0.25	0.333
3–10	0.79	0.25	0.333
4–3	0.286	0.25	0.333
1–2	0.448	0.25	0.333

An analysis of this network shows that for each destination node there are a significant number of route selection options. For example, when transferring from node 0 to node 8, the number of routes is 22.

For illustration in Fig. 6.4 many options for routes between nodes 0 and 8 presented in the criteria space for evaluating quality indicators k_1 and k_2. A subset of the Pareto optimal route alternatives found by the weight method corresponds to the lower left boundary, including the three points marked with ▲. This subset corresponds to a Pareto-agreed optimum of quality indicators, that is,

the minimum possible value of one of the quality indicators when changing the values of another quality indicator.

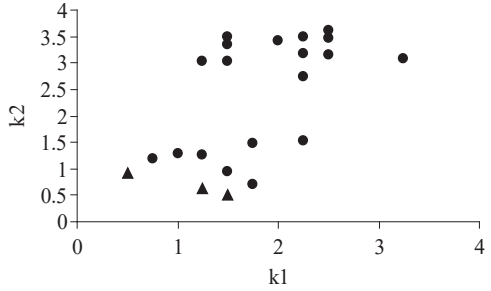

Fig. 6.4 Selection of a subset of Pareto optimal route options in criteria space

The found Pareto optimal route options are equivalent in terms of an unconditional advantage criterion – the Pareto criterion. The resulting subset of the Pareto optimal route options can be used to organize multi-way routing, which is used, in particular, by MPLS technology. This approach will allow for load balancing and traffic control, and will ensure optimal quality of service, taking into account the totality of quality indicators.

6.3.4 Selection of the best speech codecs based on a set of quality indicators

To conduct a comparative analysis of a certain set of existing speech codecs and select the best options, let's use data on 23 speech codecs, which are described by a combination of 5 quality indicators: coding speed, speech coding quality assessment, implementation complexity, frame size, and total delay. The value of these quality indicators of speech codecs is given in Table 6.2.

It is easy to notice that the quality indicators of speech codecs are interconnected and are antagonistic in nature.

The time delay increases with increasing frame size, as well as with increasing complexity of the encoding algorithm. When transmitting speech, the allowable delay in one direction can't be more than 250 ms.

The frame size affects the quality of the reproduced speech: the longer the frame, the more effectively speech is encoded. On the other hand, as the frame length increases, the delay in processing the transmitted information increases. The codec frame size is determined by a trade-off between these requirements.

The complexity of the coding algorithm is associated with the need for real-time computing. The complexity of the algorithm determines the processing speed, measured in millions of instructions per second (Millions of Instructions per second – MIPS). The complexity of the processing affects the physical dimensions of the encoding decoding or combined device, as well as its cost and power consumption.

The speech coding quality is evaluated using the MOS (Mean Opinion Score) feature. This is the average cumulative estimate for 5-point scale.

Table 6.2

The initial values of the quality indicators of speech codecs

No.	Codecs	Encoding Speed, Kbps	Evaluation of speech coding quality, MOS (1–5)	Difficulty of implementation, MIPS	Frame size, ms	Total delay, ms
1	G 711	64	3.83	11.95	0.125	60
2	G 721	32	4.1	7.2	0.125	30
3	G 722	48	3.83	11.95	0.125	31.5
4	G 722(a)	56	4.5	11.95	0.125	31.5
5	G 722(b)	64	4.13	11.95	0.125	31.5
6	G 723.1(a)	5.3	3.6	16.5	30	37.5
7	G 723.1	6.4	3.9	16.9	30	37.5
8	G 726	24	3.7	9.6	0.125	30
9	G 726(a)	32	4.05	9.6	0.125	30
10	G 726(b)	40	3.9	9.6	0.125	30
11	G 727	24	3.7	9.9	0.125	30
12	G 727(a)	32	4.05	9.9	0.125	30
13	G 727(b)	40	3.9	9.9	0.125	30
14	G 728	16	4	25.5	0.625	30
15	G 729	8	4.05	22.5	10	35
16	G 729a	8	3.95	10.7	10	35
17	G 729b	8	4.05	23.2	10	35
18	G 729ab	8	3.95	11.5	10	35
19	G 729e	8	4.1	30	10	35
20	G 729e(a)	11.8	4.12	30	10	35
21	G 727(c)	16	4	9.9	0.125	30
22	G 728(a)	12.8	4.1	16	0.625	30
23	G 729d	6.4	4	20	10	35

Table 6.3 shows the results of converting the initial values of quality indicators by normalizing them and bringing them to a comparable form. At the same time, for all quality indicators normalization operations $k_{in} = k_i / k_{i\max}$ are performed. Then, some quality indicators are converted into a comparable form so that they are of the same type depending on the technical characteristics of the codecs, in particular, for the indicators k_{3n} and k_{5n} the transformations $k'_{3n} = 1/k_{3n}$, $k'_{5n} = 1/k_{5n}$ are performed.

Table 6.3

Normative values of quality indicators of speech codecs

No.	Codec	k_{1n}	k_{2n}	k'_{3n}	k_{4n}	k'_{5n}	Pareto optimal variants
1	G 711	1	0.851	0.604	0.004	0.515	–
2	G 721	0.5	0.911	1	0.004	1	+
3	G 722	0.75	0.851	0.604	0.004	0.969	–
4	G 722(a)	0.875	1	0.604	0.004	0.969	+
5	G 722(b)	1	0.918	0.604	0.004	0.969	+
6	G 723.1(a)	0.083	0.8	0.439	1	0.818	+
7	G 723.1	0.1	0.867	0.424	1	0.818	+
8	G 726	0.375	0.822	0.748	0.004	1	–
9	G 726(a)	0.5	0.9	0.748	0.004	1	–
10	G 726(b)	0.625	0.866	0.748	0.004	1	+
11	G 727	0.375	0.822	0.727	0.004	1	–
12	G 727(a)	0.5	0.9	0.727	0.004	1	–
13	G 727(b)	0.625	0.866	0.727	0.004	1	–
14	G 728	0.25	0.889	0.281	0.021	1	+
15	G 729	0.125	0.9	0.317	0.333	0.879	+
16	G 729a	0.125	0.878	0.669	0.333	0.879	+
17	G 729b	0.125	0.9	0.309	0.333	0.879	–
18	G 729ab	0.125	0.878	0.626	0.333	0.879	–
19	G 729e	0.125	0.911	0.237	0.333	0.879	–
20	G 729e(a)	0.184	0.915	0.237	0.333	0.879	+
21	G 727(c)	0.25	0.889	0.727	0.004	1	–
22	G 728(a)	0.2	0.911	0.453	0.021	1	+
23	G 729d	0.1	0.889	0.359	0.333	0.879	+

Based on the data obtained in Table 6.3 using the unconditional advantage criterion in the criterion space with 23 variants of a selected subset of Pareto optimal variants of speech codecs, including 12 variants of codecs (marked with a +) are obtained.

The only design solution from the Pareto subset is chosen from the condition of the extremum of the scalar membership function:

$$U(k_1,\ldots,k_m)=\frac{1}{m}\left\{\sum_{j=1}^{m}\left[\xi_{\vec{k}}\left(k_j\right)\right]^{\beta}\right\}^{\frac{1}{\beta}}. \qquad (6.49)$$

In the Table 6.4 the values of this function for Pareto optimal variants of speech codecs with double coefficients $\beta=2$ and $\beta=3$ are given. It is found that the extreme value of the objective function for different values β is achieved for the same speech codec – G 722 (b).

Table 6.4

Value of membership function for Pareto optimal variants of speech codecs

No.	Codec	$U(\vec{k})$ value for different β	
		$\beta=2$	$\beta=3$
2	G 721	0.35099	0.24688
4	G 722(a)	0.35039	0.28188
5	G 722(b)	0.35476	0.28532
6	G 723.1(a)	0.31677	0.25791
7	G 723.1	0.32312	0.26308
10	G 726(b)	0.32863	0.26445
14	G 728	0.27801	0.24056
15	G 729	0.26904	0.22785
16	G 729a	0.29103	0.23837
20	G 729e(a)	0.26912	0.22898
22	G 728(a)	0.28812	0.24582
23	G 729d	0.26927	0.22716

Thus, for a given formulation of the multicriteria problem of choosing the optimal speech codec, taking into account the totality of quality indicators, is codec No. 5–722 (b). This is a speech codec that has the following values

of quality indicators: coding speed – 64 kbit/s, speech coding quality rating – 4.13 MOS, implementation complexity – 11.95 MIPS, frame size – 0.125 ms, total delay – 31.5 ms.

6.3.5 Optimal control of network resources based on a set of quality indicators

When planning communication networks, an important place is taken by algorithms for optimal control of network resources. The basic network resources include channel and information resources.

For example, the network control model is investigated, it consists of a certain set of control agents (CA) for each autonomous system (Fig. 6.5).

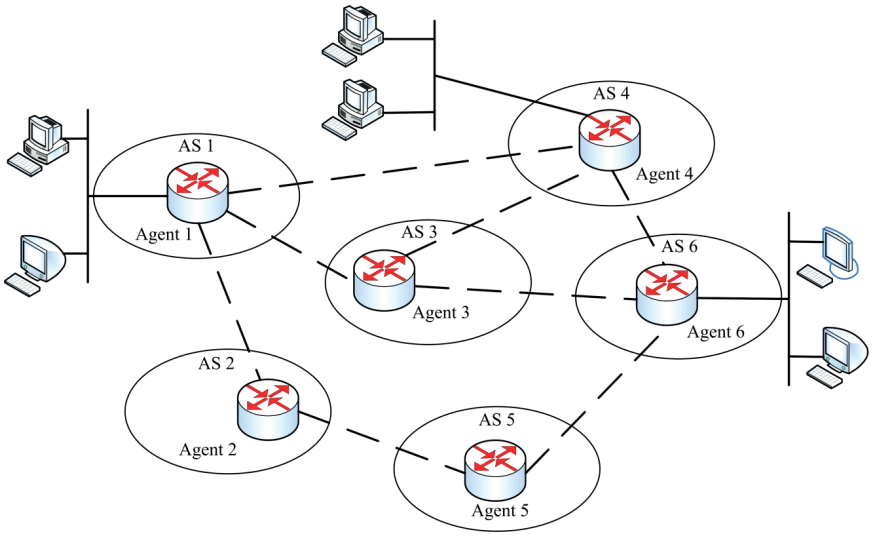

Fig. 6.5 The investigated model of network control of a telecommunication network

The process of managing network resources for an agent of a decentralized control architecture consists is in finding a flow distribution vector with corresponding restrictions:

$$\vec{Y} = (y_1, y_2, ..., y_l), \ \sum_i^l y_i = 1, \tag{6.50}$$

$$0 \le y_i \le 1, \ i = \overline{1...l}, \ \ \lambda_i^{out} \cdot y_i \le c_i, \ i = \overline{1...l}.$$

In the considered network control problem, it is not expected to select a subset of control options that are optimal according to the Pareto criterion. Multi-criteria optimization of the communication network is performed by finding the extremum of the scalar utility function, taking into account the totality of antagonistic quality indicators when managing the communication network.

Within the framework of this model, the optimal control of network functioning, the task of balancing the information resources of the local agent is solved by searching for the extremum of the target functional, taking into account the totality of quality indicators:

$$\varepsilon(\vec{Y}) = \min\ (q_1\Phi + q_2\sigma_1(\vec{Y}) + q_3\sigma_2(\vec{Y})), \tag{6.51}$$

where Φ is the metric of the standard routing protocol; $\sigma_1(\vec{Y})$ is mean-square deviation (MSD) of loading of channel agents;

$$\sigma_1(\vec{Y}) = \sqrt{\frac{1}{l-1}\sum_{i=1}^{l}(x_i - \bar{x})^2}, \quad \bar{x} = \frac{1}{l}\sum_{j=1}^{l} x_j; \tag{6.52}$$

$\sigma_2(\vec{Y})$ – MSD of loading of adjacent CAs:

$$\sigma_2(\vec{Y}) = \sqrt{\frac{1}{l-1}\sum_{i=1}^{l}(Z_i - \bar{Z})^2}, \quad \bar{Z} = \frac{1}{l}\sum_{j=1}^{l} Z_j, \tag{6.53}$$

q_1, q_2, q_3 are some weights characterizing the relative importance of quality indicators.

For research, a simulation model was used, including up to 18 CAs, with 6 of them being composite core of the network, and others – the boundary CAs. Studies were conducted for various connectivity nodes (from 2 to 6). In the course of the research, various models of network resource control and distribution of network resources were considered (Fig. 6.6): M1 – RIP one-way routing model; M2 – multi-way routing model along equal cost paths; M3 – multi-way routing model along the routes of different cost of the IGRP protocol; M4 – Gallagher flow model; M5 – proposed analytical model of managing network resources based on a distributed agent system proposed; M6 – for the case of fuzzy logic.

The dependences of the delay time (Fig. 6.7, *a*) and the probability of package loss (Fig. 6.7, *b*) on the value of the normalized subscriber load are obtained.

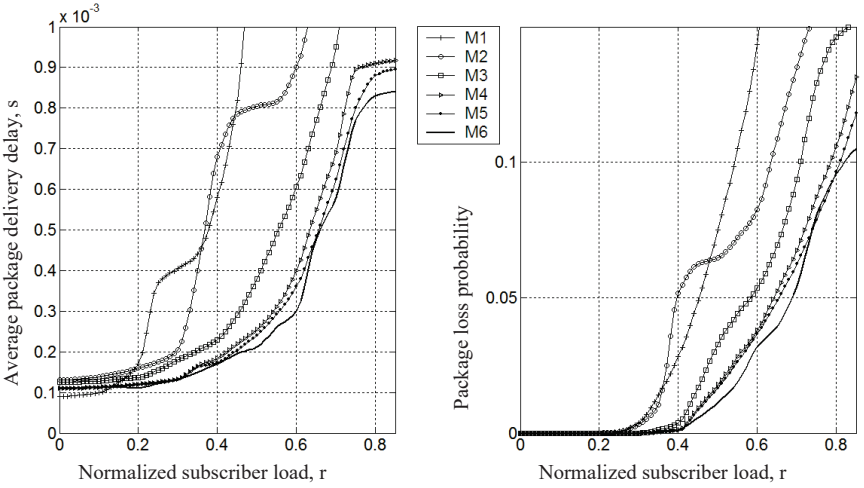

Fig. 6.6 Results of comparison of control models of the network resources

It can be seen that the control models M5 and M6 prevail, which allowed:

– reduce the average package transmission delay in the optimal way according to the best known M4 model by an average of 3–12 % (with a normalized subscriber load of more than 0.5);
– reduce the total probability of blocking packages along the optimal path by an average of 6–11 % (with a normalized subscriber load of more than 0.5).

7 SIMULATION AND OPTIMIZATION IN THE AUTOMATED DESIGN OF SYSTEMS AND COMMUNICATIONS NETWORKS

This section discusses the practical features of the study of communication systems and networks by the method of statistical computer simulation, and also analyzes various approaches to the software implementation of mathematical models of communication systems and networks on a computer. Information is given on some existing software packages that can be used to simulate and optimize the design of communication systems and networks.

In preparing the materials of the section, the works [3, 15, 16, 18, 19] are used, which can be addressed in the course of an in-depth study of these issues.

7.1 Stages and features of system design

Let's consider some general provisions and features of the initial stages of system design. The life cycle of a complex technical system, as a rule, includes the following stages:

1) formation of requirements for the system and the formation of technical specifications for its development;
2) system design;
3) manufacturing, research and refinement of prototypes of the system;
4) mass production;
5) operation;
6) system modernization;
7) disposal.

Stage 1 is also called external design. In this case, the purpose, for which the system is created, is clarified; the circle of solving problems by it is specified; the operating conditions of the system are determined, the requirements for technical characteristics and quality indicators of the system are formulated.

Stage 2 is also called internal design. In this case, the structure and parameters of the system are determined, construction options and methods of practical implementation, design, manufacturing technology of subsystems and the system as a whole. The purpose of internal design is in development of the necessary technical documentation, which is a project of the system.

At the stage of internal designing, in turn, the following stages are distinguished: development of a technical proposal, a preliminary design of a technical project, working design documentation. At the stage of developing a technical proposal, a construction concept is formed and the main system parameters are determined that satisfy the requirements of the technical task. In this case, the requirements of external design are actually consistent with the capabilities of internal design. The main task of outline design is in development of the structure and determination of the main characteristics of the system. At the stage of technical design, the system design is refined and detailed, a research model of the system is created and tested. At the final stage of design, a set of design documentation for the production and operation of the system is developed.

The process of internal system design in the general case includes the following activities: analysis of similar systems and justification of the source data and system restrictions; the choice of principles for constructing and determining the structure of the system; circuit synthesis system; system design; development of manufacturing technology for the system, preparation of equipment for testing the system.

When designing, research work is first carried out, in the framework of which a search for ways to create and study new principles for building a system is carried out. The result is the formulation of technical specifications for the development of a new system. Then, experimental design work is carried out, within the framework of which the methods of building the system are checked and specified, and a research model of the system is created. The result is a conceptual design of the system. During technical design, detailed development of all circuit, design and technological solutions necessary for the manufacture, testing and operation of the system is carried out.

The description of a complex technical system consists of several aspects: functional, design, technological. The functional aspect reflects the physical, informational processes that occur in the system during its functioning. The design aspect characterizes the structure, spatial arrangement and shape of the components of the system. The technological aspect characterizes the manufacturability, ability and means of manufacturing the system in the given conditions.

During the description of the system, the following levels of abstraction are distinguished: systemic, functional, circuitry, and component. At the system level, how the system appears is the object intended for design, for example, an

automatic telephone system or a data network between computers. The functional elements are blocks and devices, for example, a modem, transmitter, receiver, demodulator. The circuit elements are, for example, a signal generator, amplifier, counter, decoder. At the component level, the processes that occur in the components of the circuit, for example, integrated circuits, transistors are considered.

The process of internal system design in the general case, depending on the sequence in which the design is performed, consider the top-down and bottom-up method of system design. In a top-down design, flowing from top to bottom, tasks of higher hierarchical levels of design are disconnected earlier than tasks of lower hierarchical levels. In the case of bottom-up design, on the contrary, the tasks of the lower hierarchical levels of design are solved earlier than the tasks of the higher hierarchical levels. Functional design is, as a rule, downward, and design – upward.

A successful solution to the system design problem is possible only on the basis of a comprehensive, holistic review of the designed system and its development (change) in the process of interaction with the external environment and other systems. Only such an approach, called a systematic one, can lead to truly creative, innovative design solutions. The systems approach is based on the following principles:

1. Accounting for all stages of the developed «life cycle» (design, manufacture, operation, disposal).
2. Consideration of the history and especially the prospects for the development of systems of this and related classes of systems.
3. A comprehensive review of the interaction of the system with the environment (with nature and society in general).
4. Consideration of the main types of interaction of elements of the system itself (functional, constructive, energy, information, dynamic).
5. Taking into account the interaction between the development of the element base and system engineering.
6. Taking into account the possibility of changing the source data and even the problem to be solved in the processes of designing, manufacturing and operating the system.
7. Identification of the main indicators of the quality of the system, which should be taken into account and improved during the design.
8. Combination of the principles of composition, decomposition and hierarchy during the creation of subsystems, devices, blocks.
9. Disclosure of the main technical contradictions that impede the improvement of the main indicators of the quality of the system and the search for ways to eliminate them.

10. The correct combination of different design methods: mathematical, heuristic, experimental, and within the framework of mathematical methods – analytical and using computers.
11. Ensuring appropriate interaction in the design process of specialists of various profiles.

Since the complexity of communication systems is growing rapidly, for the analysis and synthesis of such complex systems, an approach to design is used, which is called decomposition of the system and means dividing the system into simpler subsystems and studying the set of their structures and the interaction between them. Such an approach to design is considered in such a scientific direction as system architecture. System architecture is a comprehensive concept that contains three important types of interconnected structures: physical, logical, and software. Each of these structures is determined by a set of elements and the nature of their interaction.

A complex communication system has the following distinguishing features:

- a large number of interconnected and interacting elements;
- complexity of the function that the system performs;
- possibility of dividing the system into subsystems, the functioning of which are subordinate to the general purpose of the functioning of the system;
- management of an extensive network (often hierarchical structure) and information flows;
- relationship with the external environment and functioning under the influence of random factors.

Modern communication systems and networks are typical representatives of complex systems with a hierarchical, functional and structural organization. The functional hierarchy reflects the specific tasks of each element of the system and the subordination of the elements due to their general functioning as part of the communication system.

In modern practice of creating complex systems for their synthesis, a combination of substantial (heuristic, intuitive) and formal (algorithmic) methods is used. The synthesis of complex systems is in determination of the structure of the synthesized system, and the processes of its functioning. The functional-structural approach is reduced to breaking down complex systems into separate structural components (subsystems) and determining their functional purpose. Such an organization reflects both the interaction of the system with the environment, and the internal relationships of the elements in the process of functioning of the system.

Previously, the system design process was reduced to choosing from a small number of system options and only satisfying the given restrictions on the per-

formance characteristics of the system. With the complication of systems and the growth of their cost, the need arose to create optimal systems. They are trying to compare as many design options as possible for building a system in order to choose the best one in the established sense.

When using mathematical methods of designing, the set of initial data for designing is formulated in the form of strict mathematical principles, in particular, mathematical models of the system are built, indicators of the quality of the system are determined, a criterion for optimality of the system is selected, and problems of optimizing the structure and parameters of the system are solved. The concept of optimality is associated with the selection of the best in the established understanding of the system options. When solving the optimization problem, one should look for a consistent optimum of quality indicators, which corresponds to the best value of each quality indicator that can be achieved with fixed but arbitrary values of other quality indicators. This condition corresponds to the Pareto optimal design warrants of the designed system and the corresponding multidimensional potential characteristics of the system. It should be noted that the widely used in practice one-dimensional potential characteristics of the systems used earlier and characterizes the potential value of a single quality indicator are, as a rule, a hidden form of a multidimensional potential characteristic. This is because in practice, when determining a one-dimensional potential characteristic, all quality indicators, except for one (the most important), are translated into the rank of restrictions or completely ignored (i.e. are not taken into account). However, in fact, completely ignoring all quality indicators, except for one, is unacceptable, since in this case the potential value of this single optimizing quality indicator will reach zero (that is, the solved optimization problem will degenerate into a trivial one). Thus, in practice, it is necessary to take into account, as a rule, several quality indicators, that is, all optimization tasks in the design of systems are essentially multi-criteria vector.

In many cases, the synthesis and analysis of optimal systems is performed by analytical methods. However, often solving these problems encounters difficulties. Therefore, when designing complex systems, their computer simulation is also widely used. System simulation includes the construction and software implementation of a mathematical model of a computer system and its research. Computer simulation of the system is an important and effective stage in the design of the system.

The indicated design procedures for simulation and optimization are inherent in the initial stages of system design. This design method, when design procedures are carried out in close interaction between the designer and the computer, is called computer-aided design. Computer-aided design is characterized by a rational distribution of functions between the human designer and the computer.

7.2 Procedures and features of simulation of computer communication networks

Simulation is a method of scientific knowledge, when using which the studied object is replaced by a simpler object – its mathematical model, and as a result of studying the model new information about the real object arises. Depending on the implementation method of the mathematical model, mathematical, physical (full-scale) and semi-natural simulation are distinguished. Physical simulation is a research method according to which the system is replaced by physically feasible elements, in particular, a system layout. In semi-natural simulation, part of the system is implemented as a physical model, and its other part is in the form of a mathematical model.

During physical (full-scale) simulation, the investigated network is replaced by the corresponding material system, reproduces the properties of the system under study with the preservation of their physical nature. An example of this type of simulation is a pilot network, with the help of which the fundamental possibility of building a network based on various computers, communication devices, and operating systems is studied. However, the possibilities of physical simulation of networks are very limited, which allows to solve individual problems when setting a small number of compounds of the studied system parameters. Indeed, when field simulation a network, it is almost impossible to verify its operation for all options using various types of communication devices – routers, switches, etc. Testing in practice about a dozen different types of routers is associated not only with large time costs, but also with considerable material costs.

However, even in those cases when not the types of devices and operating systems are changed during network optimization, but only their parameters, carrying out experiments in real time for a huge number of various combinations of these parameters is almost impossible in the foreseeable time. Even simply changing the maximum package size in any protocol requires changing the configuration of the operating system in hundreds of network computers, it requires a lot of work from the network administrator.

Mathematical simulation is a research method, according to which the system model is implemented in the form of mathematical relationships characterizing the structure of the system and the conversion of signals and noise in a real system. It is possible to use both analytical and numerical methods of mathematical simulation. When using analytical methods, the necessary solutions and dependencies are obtained from the mathematical model of the system by the consistent application of mathematical rules and transformations. The difficulties in applying analytical methods are connected with the lack of complete

a priori data for carrying out the transformations, as well as the complex nature of these transformations. However, programs of analytical transformations on computers have recently appeared, expanding the capabilities of these methods. The use of numerical methods is reduced to replacing mathematical operations with corresponding computational operations on a mathematical model implemented on a computer. Although numerical methods make it possible to solve a much wider range of problems, they are characterized by a significant complexity of calculations and, in some cases, unstable solutions for approximation and rounding errors.

When optimizing networks in many cases, preference is given to using mathematical simulation on a computer. The mathematical model of the network is a set of relations (formulas, equations, inequalities, logical conditions) that determine the process of changing the state of the system depending on its parameters, input signals, initial conditions and time. A special class of mathematical models are simulation models. Such models are a computer program, step by step reproduces the events occurring in a real system. Regarding networks, their simulation models reproduce the processes of generating messages breaking messages into packages and frames of specific protocols, delays associated with processing messages, packages and frames within the network, the process of gaining access to a shared network environment, the process of processing packages, etc. When simulation the network, it is not necessary to purchase expensive equipment – its work is imitated by programs that accurately reproduce all the main features and parameters of such equipment.

The advantage of simulation models is the ability to replace the process of changing events in the studied network in real time with an accelerated process of changing events at the pace of the program. As a result, in a few minutes it is possible to reproduce the network for several days, which makes it possible to evaluate the network in a wide range of variation parameters. The result of the simulation model is collected during the monitoring of events that flow statistics on the most important characteristics of the network: response times, utilization rates of channels and nodes, package loss probabilities, etc.

Among the methods of researching a computer system, simulation methods based on the implementation and study of a mathematical model in the form of algorithms and programs that reflect both the structure of the system and the processes of its functioning in time are widely used. In some cases, the capabilities of algorithmic languages make it possible to obtain more flexible and accessible means of describing complex systems in comparison with the language of mathematical functional relations. With a probabilistic approach to computer systems simulation, an approximate numerical method of research is used – the method of statistical simulation. In this case, the mathematical model of the

system is implemented in software on a computer, and the necessary characteristics of the system are obtained by conducting statistical tests of the system on samples of real or model signals and noise, as well as processing the results of studies using methods of mathematical statistics. A positive property of this method is its universality, which guarantees the fundamental possibility of analyzing a system of any complexity and with arbitrary detailing. Complexity of the simulation processes and the particular nature of the results obtained for specific operating conditions of the system are negative.

To conduct system research by the method of statistical simulation on a computer, the following procedures are typical:

- formulation of the simulation problem, which includes the totality of information that must be obtained as a result of simulation;
- determination of the boundaries of the system to be simulated, as well as a set of limitations and assumptions, according to which the simulation will be carried out;
- collection and evaluation of a priori information about the system under study, the volume of which should be sufficient to build its mathematical model;
- selection of a criterion for the quantitative assessment of the results of a system study by a computer simulation method;
- formation of a mathematical model of the system, which includes an informal and formal description of the object of simulation;
- software implementation of the mathematical model of the system and its implementation on a computer;
- assessment of the adequacy of the selected model, that is, determining the correct functioning of the model and its compliance with the real system;
- research planning, that is, such an organization of the process of statistical simulation in order to obtain the necessary information about the system with a given reliability in a minimum amount of time;
- carrying out statistical tests of the system on the corresponding samples of signals and interference;
- finding an assessment of the criterion characterizing the quality of the studied system;
- interpretation of system simulation results obtained as a result of simulation;
- decision making based on simulation results.

The information obtained as a result of simulation is compared with the stated goal of simulation. If the comparison is satisfactory, then the simulation results are recorded in the final protocol or document. If the results are unsatisfactory, some procedures are adjusted and the simulation process is repeated.

7.3 Analysis of software implementation tools for mathematical models of communication networks

In the course of the implementation on a computer of a simulation mathematical model of a communication network, it is specified by a plurality of nodes, each of which is connected to at least one other node by means of communication channels. Let's consider the basic properties of a package-switched communication network that need to be reflected in its model:

1. The network structure is determined by the number of nodes, their characteristics, connectivity matrix, characteristics of communication lines between nodes.
2. Principle of package switching.
3. Call servicing algorithm, which is determined by the discipline of servicing calls at network nodes.
4. Call routing, which is determined by the rules for selecting routes for servicing calls and the routing table in each of the network nodes.
5. Subscriber traffic is determined by the properties of call flows, their numerical characteristics.

Based on the analysis of the processes of information transmission through the communication network, the following functional elements can be distinguished that should be described in the mathematical model of the network: data source, switching node, communication channel, data package, network control module, external network exposure module.

The data source provides simulation of the process of data receipt for transmission over the network, in particular, from the sending node to the receiving node. The data source is directly connected to the corresponding switching node.

The switching node simulates the operation of network nodes, ensuring the direction of data packages directly from the data source to the node through the communication channel or between adjacent nodes according to the selected routing strategy. This takes into account the procedure for processing application queues in switching nodes taking into account priorities.

In the course of simulation the operation of communication channels between adjacent nodes, the following processes are taken into account: package delay associated with limited channel capacity; propagation delay of electromagnetic waves in communication lines; package loss processes during transmission over communication channels and exit from communication channels. Communication channels are actually single-channel or multi-channel communication systems that connect adjacent switching nodes.

A data package is a certain amount of data transmitted over a network.

In the model, it is specified by a header containing the necessary information for its processing in communication nodes, and also by the length of the package models the presence of data in it.

The network control module makes it possible to simulate control processes in the network, including the transmission of service packages with the information necessary for the operation of the network. The network control module is connected to the switching node and is for it a source of packages with service information transmitted to other nodes.

The module of external influence on the network simulates the processes of exiting and restoring the operability of nodes and communication channels.

Thus, the communication network contains a significant number of elements. Processes in different parts of the network can occur independently, sequentially or in parallel. Given the fulfillment of certain conditions, various mathematical models of the network can be built on the basis of the provisions of the theory of random flows, the theory of mass service, and the theory of teletraffic.

When analyzing a specific simulation model of a network, it is possible to investigate various probabilistic-temporal characteristics, in particular, the average delay time of messages, the average busyness of a communication line with a proportional change in the intensity of input streams for all «sender-destination» pairs.

To implement mathematical models of communication networks on computers, both universal programming languages and specialized simulation software packages can be used. It should be noted that on one edge of universality are common programming languages such as *FORTRAN*, and more specialized languages such as *GPSS*. They provide an opportunity, with sufficient qualifications in programming, to create and implement models of systems and networks of any complexity, but at the cost of a significant expenditure of time and effort. Although the *GPSS* software system has graphical tools for manipulating flowcharts, it is possible to use animation and an interface with *C++*, however, it is hardly acceptable for simulation network objects with a large number of connections.

Another way to ensure the versatility of the simulation system is in identification and programmatically implementation of the basic network functions that are most elementary and common for various applications. This approach provides in most cases the maximum efficiency of using the functionality of simulation systems. However, the main part of the work is shifted to the user, which increases the complexity of creating complex information models. Another way is in implementation of the maximum number of different functions, covering the entire range of possible applications. But this complicates the work with such a simulation system. An example is a simulation software system, which contains about ten thousand functions and object-oriented classes.

On the other side of versatility are specialized computer mathematics packages that allow to perform character-numerical simulation of general-purpose systems. Among them, the most widely used software packages are: *Eureka, Mercury, MathCAD, Derive, Mathematica, Mapl.*

Among packages of this type, *MatLab* stands out. This system is now widely distributed in engineering and university circles due to outstanding advantages, which include:

1. Simplicity of comprehension and accessibility of texts of almost all software tools, except the built-in ones.
2. A large library of achievable mathematical programs, which contains almost all modern numerical methods and functions.
3. Ability to create your own software and even adjust existing ones.
4. A very convenient and adapted for the practical needs of engineers and scientists apparatus for graphical presentation of calculation results.

MatLab is a high-level matrix and array language with control of flows, functions, data structures, data input and output, and object-oriented programming features.

MatLab environment is a set of tools and devices that a user or *MatLab* programmer works with. It includes tools for managing variables in the *MatLab* workspace, data input and output, and the creation, control and debugging of *MatLab* files and applications.

MatLab graphics system includes high-level commands for visualizing two- and three-dimensional data, image processing, animation, and illustrated graphics. It also includes low-level commands that allow to completely edit the appearance of graphical information in the same way as when creating a graphical (for the user) interface for *MatLab* applications.

In this package, specialized tasks of expanding the capabilities for simulation telecommunication and infocommunication are solved with the help of additional specialized packages (*Toolbox*), which provide the capabilities of symbolic and analytical calculations, special means of integration with other packages. Integration with the Simulink package, which is intended for simulation of block-defined dynamic systems and devices, provided new properties. *Simulink* package has a wide library of mathematical models of various functional blocks. Generators of various types of signals, virtual measuring instruments, graphic means for displaying the results of system simulation can also be included in the structure of the created model of the system.

Simulink is an interactive tool for simulation, simulating and analyzing dynamic systems, including discrete, continuous and hybrid, non-linear and discontinuous systems. It makes it possible to build graphical block diagrams, simulate dynamic systems, investigate the performance of systems and improve projects.

Key features:
- interactive graphical environment for building block diagrams;
- expandable library of ready-made blocks;
- tools for constructing multi-level hierarchical multicomponent models;
- tavigation and parameter settings for complex models;
- means of integration of ready-made *C/C++, FORTRAN, ADA* and *MatLab* algorithms into the model, interaction with external programs for simulation;
- modern means of solving differential equations for continuous, discrete, linear and nonlinear objects (including with hysteresis and discontinuities);
- simulation of non-stationary systems with the help of solvers with variable and constant step or by the method of batch simulation controlled with MatLab;
- interactive visualization of output signals, settings and tasks of input actions;
- a tool for debugging and analyzing models;
- full integration with MatLab, including numerous methods, visualization, data analysis and graphical interfaces.

A similar *Mathematica* package is Wolfram Research's algebra system. It contains many functions for both analytical transformations and numerical calculations. In addition, the program supports graphics and sound, including the construction of two- and three-dimensional graphs of functions, floods of arbitrary geometric shapes, import-export of images and sound. The difference between this package lies in its use primarily for analytical simulation of systems based on one or another mathematical apparatus.

Key features:
- solution of recurrence equations;
- simplification of expressions;
- finding boundaries;
- integration and differentiation of functions;
- finding finite and infinite sums and products;
- solution of differential equations and partial differential equations;
- Fourier and Laplace transforms, as well as Z-transforms;
- expansion of a function in a Taylor series, operations with Taylor series: addition, multiplication, composition, obtaining an inverse function, etc.
- discrete Fourier transform;
- plotting functions;
- construction of geometric shapes: broken lines, circles, rectangles, etc.;

- reproduction of a sound which graph is set by an analytical function or a set of points;
- import and export of graphics in many raster and vector formats, as well as sound;
- construction and manipulation of graphs.

These packages make it possible to quickly orientate them towards solving simulation problems in various fields: neural networks and telecommunications, designing event-control systems.

However, these universal software tools do not always meet the requirements of simulation systems in the field of communications. The most critical parameter in this case is most often performance. If it is necessary to conduct a significant amount of research, the presence of a specialized software tool or simulation language can significantly (sometimes by several orders of magnitude) speed up the research process and significantly improve the quality characteristics. This circumstance has led to the emergence of an extremely large number of different specialized simulation tools and languages, focused on specific areas of application.

Moreover, one of the approaches to simulation is creation of specialized software simulation models in which the description of the model of the system under study is performed in terms of this system. In the case of simulation communication networks during the use of this method, the network topology, the composition and parameters of the equipment in the switching nodes, and the characteristics of the incoming information flows are set. The characteristic representatives of this approach are specialized software packages (software systems) *CLASS/ANKLES, REAL, NEST, ANSAN, NS.*

In particular, *CLASS/ANKLES* system is designed for simulation of ATM networks. The peculiarity of its construction is that it consists of two software simulation modules that allow simulation of the network under study with varying degrees of detail. *ANKLES* software simulation model enables network simulation at the call level. At this level of detail, it is possible to explore call blocking probability, routing, and connection control protocols. *CLASS* software simulation model makes it possible to study the probability of loss of a portion of the protocol data, the distribution of the delivery delay time, and the flow control efficiency.

The considered system is designed to study ATM networks and can't be used to study other types of networks. However, when designing in the general case, it becomes necessary to carry out simulation of various types of networks. In this case, it is advisable to use the software environment *NS (Network Simulator)*, which developed from *REAL*. *NS* is designed to simulate a wide range of network architectures using the TCP/IP protocol stack. *NS* system is an object-oriented simulation system created by *C++* with the additional use of *OTcl*. *C++* language is convenient from the point of view of significant performance when implementing

protocol details and when working with large data arrays. *OTcl* language is slower in operation, but requires less time for changes in the program, it is important when studying the influence of the configuration, parameters and control of the network on its quality indicators.

Along with the software packages mentioned above, other packages are also widely used in network simulation.

NetCracker is a software tool for network design and simulation of hardware and software information networks, with which it is possible to create static and dynamic models of networks with elements of visualization of the transmission of data packages in real and model time. *NetCracker* contains databases with a set of various devices that make it possible to simulate networks of various configurations, technologies with different topological structures. *NetCracker* can be used to design local, global, corporate communication networks. On such projects, it is possible tosee the location of the selected equipment, its technical characteristics, according to what protocol it works, what technology is used, etc. Such projects make it possible to automatically perform the optimal equipment location and cost calculation, as well as investigate the main characteristics of the designed networks until the moment of construction (current and average workload, average wait time, the number of transmitted and lost packages for a certain time interval, the number of blocked requests et al.).

Cinderella is a software tool that allows to develop, analyze and model processes in dynamic systems, which is described in the specification and description language *Cinderella* in combination with two other specification languages *ASN.1* and *MSC*. Today, *Cinderella* landuage has evolved into an object-oriented language and is now widely used not only in telecommunications, but also in many other industries. *Cinderella* is designed to simulate interaction processes that describe data transfer protocols, signaling protocols, input and output in-plant connections, and much more that can be represented as interaction processes. *ASN.1* and *MSC* are intended mainly for data specification and are recognized for describing data in protocols that are built in accordance with the open system interaction model.

Package Tracer is a structural-logical design program for computer networks. It allows to perform network simulations based on Cisco equipment, and supports modular network equipment. Due to the presence of the simulation mode and protocol analyzer, one can see both the structure of packages generated by various network protocols and the algorithms of various devices.

BONeS is a general-purpose graphical simulation system for analyzing the architecture of systems, networks and protocols. Describes the models at the transport level and at the application level. It makes it possible to analyze the effects of client-server applications and new technologies on the network.

Netmaker is topology design, planning and analysis tools for a wide class of networks. It consists of various modules for calculation, analysis, design, visualization, planning and analysis of results.

Optimal Performance has the ability to quickly evaluate and accurate simulation, helps optimize distributed software.

Prophesy is a simple system for simulation local and global networks. It allows to evaluate the computer reaction time to the request, the number of «hits» at the WWW server, the number of workstations for servicing active equipment, the network performance margin in case of failure of certain equipment.

CANE family is design and reengineering of a computing system, evaluation of various options, «what if» scenarios. Simulation at various levels of the OSI model. Developed library of devices, which includes the physical, electrical, temperature and other characteristics of objects. It is possible to create your own libraries.

OPNET family is a tool for designing and simulation local and global networks, computer systems, applications, and distributed systems. Ability to import and export topology and network traffic data. Analysis of the effects of client-server applications and new technologies on the network. Simulation hierarchical networks, multi-protocol local and global networks; accounting routing algorithms. Object oriented approach. A comprehensive library of protocols and objects. It includes the following products: *Netbiz* (design and promotion of a computing system), *Modeler* (simulation and analysis of network performance, computer systems, applications and distributed systems), *IT Guru* (performance evaluation of communication networks and distributed systems).

OPNET IT GURU allows:

- create a virtual network consisting of appropriate hardware, protocols, as well as application software;
- is a purely software interface that can work on an individual workplace;
- allows to study and compile useful statistics about a virtual network built with its help;
- create virtual networks in the field of software, and also provides tools for the dynamic collection of network information.

7.4 Software packages for simulation and optimization of communication networks

Let's consider the features of simulation and the possibility of using some software packages on the example of the task of researching performance in the design of communication networks. Protocol analyzers are indispensable for the study of real networks, but they do not allow to obtain quantitative estimates of

the characteristics for networks that are not yet in the design stage. In these cases, designers can use simulation tools with which models are created that reproduce information processes in networks.

There are special simulation languages that facilitate the process of creating a software model compared to using universal programming languages. Examples of simulation languages include languages such as *SIMULA, GPSS, SIMDIS*. There are also simulation systems that focus on a narrow class of systems under study and allow to build models without programming. Similar systems for computer networks are discussed below.

Such software systems themselves generate a network model based on input data on its topology and the protocols used, on the intensity of the flow of requests between network computers, the length of communication lines, and on the types of equipment used. Software simulation systems can be narrowly specialized and quite universal, allowing to simulate networks of various types. The quality of the simulation results largely depends on the accuracy of the initial network data transmitted to the simulation system.

Network simulation software systems are a tool that any network administrator may need, especially when designing a new network or making fundamental changes to an existing one. Software products of this category allow to check the consequences of the implementation of various design decisions even before the payment for the equipment is purchased. Of course, most of these software packages are quite expensive, but the potential savings can also be very tangible.

Network simulation programs use in their work information about the spatial location of the network, the number of nodes, communication configurations, data transfer rates, protocols used and the type of equipment. Of course, the simulation model is not built from scratch. There are ready-made simulation models of the main elements of networks: the most common types of routers, communication channels, access methods, protocols, etc. These models of individual network elements are created on the basis of various data: the results of test tests of real devices, analysis of the principles of their operation, and analytical relationships. As a result, a library of typical network elements is created, which can be configured using parameters predefined in the models.

Simulation systems usually also include a set of tools for preparing the source data for the investigated network – preprocessing data on the network topology and measured traffic. These tools can be useful if the model network is a variant of an existing network and it is possible to measure traffic and other parameters necessary for simulation in it. In addition, the system is provided with means for statistical processing of the obtained simulation results.

Table 7.1 shows the characteristics of some software systems for simulation of a different class – from simple programs to powerful systems, including libraries

of most of the communication devices available on the market and provide a significant degree of automation of the study of the designed communication network.

Table 7.1

Software packages for simulation and optimization and communication networks

Program	Purpose, brief description of the program
COMNET III	Simulation of X.25 network, ATM, Frame Relay, LAN-WAN, SNA, DECnet, OSPF, RIP. Access CSMA / CD, FDDI, etc. A library of routers 3COM, Cisco, DEC, HP, Wellfleat has been built
NetMaker	Building network models using an extensive library of network devices. Checking network topology data; Import traffic information in real time
StressMagik	Support for standard tests of measuring network performance; simulation of peak load on the file server
MIND	Communication network optimization, contains data on the cost of typical configurations with the ability to accurately evaluate performance
AutoNet/Designer	Determining the optimal location of hubs in the global communications network, the ability to assess cost savings by reducing the tariff, changing the service provider and restoring equipment; comparison of communication options through the nearest and optimal access point, as well as through the local telephone network
AutoNet/ MeshNET	Capacity simulation and cost optimization for the organization of a global communications network by simulating damaged lines, support for the tariff network of AT & T, Sprint, WiTel, Bell
AutoNet/Performance-1	Simulation the performance of hierarchical communication networks by analyzing sensitivity to the duration of the delay, response time, and also bottlenecks in the network structure
AutoNet/Performance-3	Performance simulation of multi-protocol associations of local and global communication networks; estimating queue delays, predicting response times, as well as bottlenecks in the network structure; accounting for real traffic data coming from network analyzers
GNS3	Network simulation with performance assessment based on real operating systems of routers and managed switches.
Arena	It allows to build simulation models from a large number of basic structural elements, track the simulation process, analyze the simulation results and optimize the response of the model

COMNETBaseline allows to create a variety of filters with which it is possible to get the information necessary to simulate networks from imported data. Using *COMNETBaseline* it is possible to:

- enter information about the network topology, in particular, in a hierarchical form;
- combine information from several traffic registration files that can be imported from different monitoring tools into a single traffic model;
- provide the received traffic model for a preliminary cursory inspection;
- view a graphical representation of inter-nodal interactions in which the traffic of each pair of nodes is displayed by a line of a certain color.

COMNETIII. COMNETIII network simulation system allows to accurately predict the performance of local, global and corporate communications networks. *COMNETIII* offers a simple and intuitive way to construct a network model, based on the use of ready-made base units that correspond to well-known network devices such as computers, routers, switches, multiplexers and communication channels. The user applies the drag-and-drop technique for graphical depiction of the network, which is modeled from library elements. Then the *COMNETIII* system performs a detailed simulation of the resulting network, dynamically displaying the results in the form of a visual animation of the resulting traffic. Another option for setting the modeled network topology is the import of topological information from network management and monitoring systems.

After the end of the simulation, the user receives at its disposal the following characteristics of network performance:

- predicted delays between end and intermediate network nodes, capacity of communication channels, utilization rates of segments, buffers and processors;
- peaks and drops of traffic as a function of time, and not as averaged values;
- source of delays and network bottlenecks.

COMNETIII system operates with three types of nodes – processor nodes, router nodes, and switches. Nodes can be connected using ports to communication channels of any type, from channels of local networks to satellite communication lines. The nodes and communication channels can be characterized by the mean time between failures and the average recovery time for simulation network reliability.

COMNETIII models not only the interaction of computers over a network, but also the process of dividing the processor of each computer between its applications. The application is modeled using several types of commands, including data processing, sending and reading messages, reading and

writing data to a file, established sessions, and termination of the program before receiving messages. For each application, a so-called command repertoire is specified.

Router nodes can simulate the operation of routers, switches, bridges, hubs, and any devices that have an internal bus resolution through which packages are transmitted between ports. The bus is characterized by capacity and the number of independent channels. The router node also has all the characteristics of a processor node, so that it can run applications that, for example, restore routing tables or send routing information from the network. Non-blocking switching nodes can be modeled by setting the number of independent channels equal to the number of switch modules. *COMNETIII* library includes a large number of descriptions of specific router models with parameters based on the test results in the *Harvard NetworkDeviceTestLab* system.

The switch node simulates the operation of switches, as well as routers, hubs, and other devices that transmit packages from the input port to the output port with a slight delay.http://www.cityholding.dp.ua/list/htm/12/4423

Communication channels are modeled by setting their type, as well as two parameters – capacity and introduced propagation delay in the channel. The unit of data transmitted over the channel is a frame. Packages when transmitted over channels are segmented into frames. Each channel is characterized by: minimum and maximum frame size, overhead per frame and error rate in frames.

COMNETIII includes tools for simulation global communications networks at the highest level of abstraction. Such a representation of global networks is advisable when specifying accurate information about the topology of physical connections and the full traffic of the global network is impossible or impractical. For example, it makes no sense to accurately simulate the operation of *INTERNET* during the study of traffic transmission between two local networks connected to *INTERNET*.

During the simulation of global networks, package splitting into frames is simulated, and each type of global service is characterized by minimum and maximum frame sizes and overhead for service information.

Communication with the global network is simulated using an access channel, has a certain propagation delay and capacity. The global network itself is characterized by a delay in the delivery of information from one access channel to another, the probability of frame loss or its forced removal from the network (in case of violation of the CIR type traffic parameters agreement). These parameters depend on the degree of congestion of the global network; it can be set as normal, moderate and high. It is possible to simulate virtual channels in a network.

In a *COMNETIII* system, a workload is created by traffic sources. Each node can be connected to several different types of traffic sources. Application sources generate applications executed by nodes such as processors or routers. The node executes command by command, simulating the operation of applications on the network. Sources can generate complex non-standard applications, as well as simple ones, mainly engaged in sending and receiving messages over the network.

Call sources generate connection requests in circuit-switched networks (networks with virtual dial-up ISDN, POTS). Sources of the planned load generate data, the original time-dependent schedule. In this case, the source generates data periodically, using a specific distribution of the time interval between pieces of data. It is possible to simulate the dependence of the intensity of data generation on the time of day.

The communication protocols of the physical and link layers are taken into account in the *COMNETIII* system in network elements such as channels. Network layer protocols are reflected in the operation of the model nodes; they decide on the choice of the package route in the network. The network backbone and each of the subnets can operate on the basis of different and independent routing algorithms. The routing algorithms used by *COMNETIII* make decisions based on the calculation of the shortest path. Various variations of this principle are used, differing in the metric used and the means of restoring routing tables. Static algorithms are applied, in which the table is updated only once at the beginning of the simulation, and dynamic algorithms periodically update the table. It is possible to simulate multi-way routing, in which traffic balance is achieved along several alternative routes.

The protocols that perform transport and message delivery functions between end nodes are represented in the *COMNETIII* system by a large set of protocols: ATP, NCP, NCPBurstMode, TCP, UDP, NetBIOS, SNA. When using these protocols, the user selects them from the system library and sets specific parameters, for example, message size, window size, etc. http://www.cityholding.dp.ua/list/htm/12/4426

COMNETIII allows to specify the form of a report on the results of the simulation for each individual element of the model. There are various ways to obtain statistical results of a model run, in particular, collecting statistics for each type of model element – nodes, channels, traffic sources, routers, switches, etc. The statistics monitor of each element can be set up to collect only basic statistical parameters (minimum, maximum, average value and variance) or to collect data on a temporary scale for plotting. If the results of the observations are saved in a file for subsequent graphing and analysis, then it is also possible to build histograms and percentages. It is possible to build graphs during the simulation.

The menu options allow to change the speed of the simulation steps and the speed of the tokens – graphic symbols corresponding to frames and packages. In the animated mode, *COMNETIII* system shows the arrival of packages to the communication channels and their output from the channels, the current number of packages in the nodes, the number of sessions installed with this node, percentage of use, and much more.

COMNETIII includes an integrated set of tools for statistical analysis of source data and simulation results. With their help, it is possible to choose the appropriate probability distribution for the experimentally obtained data. Results analysis tools provide the ability to calculate confidence intervals, perform a regression analysis, and evaluate variation estimates obtained over several spans of the model.

COMNETPredictor. This software product is intended for those cases when it is necessary to assess the consequences of changes in the network, but without its detailed simulation. *COMNETPredictor* works as follows.

Data on the operation of an existing network option is downloaded from a network management or monitoring system and an assumption is made about changing network parameters: the number of users or applications, channel capacity, routing algorithms, node performance, etc.

COMNETPredictor then evaluates the impact of the proposed changes and provides graphical results and charts that show delays, utilization rates, and estimated network bottlenecks.

COMNETPredictor complements the *COMNETIII* system, which can then be used to more thoroughly analyze critical network options.

HTZ-Simulation computer-aided design system is intended for planning and comprehensive simulation of radio communication networks in the HF, UHF and microwave wave bands. The system provides a solution to a wide class of design problems, ranging from the selection of construction sites on the ground to the optimization of the entire radio network as a whole.

The main tasks that can be successfully solved with the help of this software design system are: optimization of existing radio networks at low financial cost; spatial planning of new radio networks; assessment of electromagnetic compatibility of new and existing radio networks; design of microwave transmission lines for fixed subscribers; frequency-territorial planning of cellular radio networks for mobile subscribers; joint design and research of a cellular network for mobile subscribers and microwave lines; assessment of the correlation between the results of calculations and measurements; preparation of calculation results for inclusion in research reports.

Construction of pilot projects of designed communication networks. If it is not necessary to have a real network to set information on the network

topology, then measurements on pilot networks may be required to collect initial data on the intensity of network traffic sources, which are full-scale models of the designed network. These measurements can be performed by various means, including using protocol analyzers.

In addition to obtaining initial data for simulation, the pilot network can be used to solve important independent tasks. It can provide answers to questions regarding the basic operability of a particular technical solution or equipment compatibility. Field experiments may require significant material costs, but they are compensated by the high probability of the results.

The pilot network should be as close as possible to the network that is being created, for the selection of the parameters of which the pilot network is being created. To do this, it is necessary first of all to highlight those features of the network being created that can affect its operability and productivity. If there are doubts about the compatibility of products from different manufacturers, for example, switches that support virtual networks or others, the capabilities have not yet been standardized, then these devices should be checked for compatibility in the pilot network and in those modes that cause the greatest doubt.

As for the use of a pilot network to predict the capacity of a real network, here the possibilities of this type of simulation are very limited. The pilot network itself is unlikely to provide a good estimate of the performance of a real network, which includes many more subnet nodes and users. It is not clear how to extrapolate the results obtained in a small network; there are many large sizes on the network. Therefore, it is advisable to use the pilot network together with the simulation model of the network, which uses the values of traffic characteristics, delays and capacity of devices received in the pilot network.

GNS3 is a graphical network simulator that allows to simulate and analyze the functioning of networks with complex topology when using existing modern network protocols. It should be noted that *GNS3* is an open source project, that is, a free program that can be used on many operating systems.

A feature of this software product is that it can fully and completely reproduce the operation of Cisco and Juniper network devices, emulating their network operating systems on your own computer. Thus, the configuration and operation of these devices is fully consistent with the work of existing routers and switches, allow to simulate and analyze the operation of any networks. It should also be noted that *GNS3* has the built-in tools that allow to organize the connection of the designed topology with a real network.

GNS3 program includes an integrated Virtual PC program, with the help of which end devices acting as sources of traffic flows transmitted over the network are emulated, as well as the Wireshark software product, which is a traffic analyzer in computer networks. Wireshark has libraries of structures of most network

protocols, and therefore allows to parse a network package, displaying the value of each protocol field at any level.

The basis of *Arena* technology is the SIMAN simulation language and the Cinema Animation system. SIMAN language, which was first implemented in 1982, is an extremely flexible and expressive simulation language. It is constantly improving by adding new features. To display the simulation results used animation system Cinema animation.

The relevant application includes:

- window of the working field;
- module parameters window;
- project window;
- basic process (panel of basic processes) – contains modules that are used for simulation;
- reports (report panel) – notification panel: contains a message reflecting the results of simulation;
- navigate (navigation panel) – the control panel allows to display all types of models, including control through hierarchical submodels.

Let's also note the need to sometimes use cloud technologies for simulation processes in communication networks, especially in the case of analytical simulation of heterogeneous networks with a combination of quality indicators.

The US National Institute of Standards and Technology has established the following mandatory cloud computing features:

1. Self-service on demand, the consumer independently determines and changes computing needs, such as server time, access and data processing speeds, the amount of stored data without interacting with a representative of the service provider.
2. Universal network access, services available to consumers through a data network regardless of the terminal device.
3. Resource pooling, a service provider combines resources for servicing a large number of consumers into a single pool for the dynamic redistribution of capacities between consumers in the face of constant changes in demand for capacities; at the same time, consumers control only the main parameters of the service (for example, the amount of data, access speed), but the actual distribution of resources provided to the consumer is carried out by the supplier (in some cases, consumers can still manage some physical redistribution parameters, for example, specify the desired data center from proximity considerations).
4. Elasticity, services can be provided, expanded, narrowed at any time, without additional costs for interaction with the supplier, as a rule, in an automatic mode.

5. Consumption accounting, the service provider automatically calculates the consumed resources at a certain level of abstraction (for example, the amount of stored data, capacity, number of users, number of transactions), and based on this data estimates the amount of services provided to consumers.

From the point of view of the simulation specialist, due to the pooling of resources and the inconsistent nature of consumption, cloud computing allows economizing on scale using less hardware resources than when allocating hardware capacities for each consumer, and by automating the procedures for modifying the allocation of resources, the cost of subscription services is significantly reduced.

7.5 Software packages for simulation and optimization of communication systems

Mathematical models of a communication system should include models of various types of message sources, encoders and decoders, modulators and demodulators, communication channels. For the software implementation of mathematical models of various variants of the communication system and the procedures for their study by the method of static simulation on a computer, universal languages such as *Pascal and C++* can be used. In these languages, models of all components of a communication system can be described. To organize a convenient interface when working with the implemented mathematical model of the system, it is possible to use the capabilities of *Delphi* and *Builder* software environments. As an example, one can cite software packages created by the indicated software for simulation and comparative studies of various types of communication systems.

Today there are a number of standard software packages that allow to simulate and explore communication systems in their automated design. Let's consider some of them.

System Wiew package is a «constructor», with the help of which, from standard blocks, a given functional diagram of any communication system can be implemented. There is an extensive library in the package, from the catalog of which the necessary functional module is selected, which is transferred to the circuit. After connecting all the functional modules and connecting all the measuring devices, the system parameters are set: the duration of the observation interval, the sampling frequency, the parameters of the fast Fourier transform, and then the simulation is performed. Fourier transforms are calculated at various points in the diagram, correlation and mutual correlation functions, arithmetic and

trigonometric operations are performed, statistical processing of simulation data is carried out, and much more.

Despite the powerful analysis tools in the System Wiew package, it is more convenient to process the data obtained as a result of simulation using the *LabView* package. This is a program for functional simulation of systems and analysis of research results using an extensive library of programs for statistical analysis, evaluation of various characteristics of signals, regression analysis, time-frequency analysis, digital and other types of signal processing.

Microwave Office 2001 package is intended for simulation communication systems at the level of structural and functional diagrams, since the *System View package*. A feature of this package is the ability to design high-frequency devices, as well as simulation processes for processing complex signals, which is typical for real communication systems. Non-linear analysis in this package is performed by the method of harmonious balance and Voltaire series.

HyperSignal Block Diagram package is a program for simulation analog and digital devices defined by functional diagrams.

SPT and IPT packages perform digital processing of signals and two-dimensional images in accordance. The package includes signal processing functions for analyzing and converting time sequences, as well as two-dimensional raster images. This package includes more than 130 functions that solve the problems of digital signal processing. In particular, the functions perform Fourier and Hilbert transforms, as well as digital filtering using digital filters with different frequency characteristics. The package allows to calculate the correlation functions, the spectral power density of the signals, evaluate the filter parameters using the measured samples of the input and output signals.

APLA package is designed for the design and simulation of electrical circuits and systems in the time and frequency domains. The structure of simulation systems can include both digital and analog components, including high-frequency devices. Calculations are performed: frequency characteristics, spectral density and noise figure, transients, signal spectra, parametric optimization, statistical analysis by statistical tests with random input influences. An important feature of the package is the presence of a library of a significant number of circuit elements and individual blocks used in analog and digital communication systems.

DesignLab package is an integrated software package for automating the design of analog, digital, as well as mixed (analog-to-digital) devices, synthesis of programmable logic devices and analog filters. In this design package, it begins with the introduction of the device's schematic diagram, its simulation and optimization, and ends with the creation of control files for programmers, as well as for the photoplotter and drilling machines to create printed circuit boards.

Package Electronics Workbench is a system for circuit simulation and analysis of analog and digital-analog systems of great complexity. The package includes a large library of widely used circuit components, the parameters of which can vary over a wide range. A wide range of measuring instruments (oscilloscopes, spectrum analyzers) allows to measure various quantities and characteristics of the circuit.

Or CAD package is a simulation and end-to-end design program for analog-to-digital electronic devices and the SPECTRA auto-tracer.

Circuit Maker program refers to simple computer-aided design systems. The program has its own circuit editor, it is easy to configure and adapt to specific system design tasks. The capabilities of this program are equivalent to the *Elektronics Workbench package*.

Micro-Cap package is a circuit simulation package. With its help, a graphical introduction of the designed circuits and analysis of the characteristics of analog, digital and analog-digital devices is performed, an analysis of nonlinear DC circuits, calculation of transient processes and frequency characteristics is carried out. The package includes a large library of components, which includes the most popular digital integrated circuits of discrete logic and analog components such as diodes, transistors, lossy transmission lines, quartz resonators, etc.

PSPICE A/D program is best suited for more complex circuit simulation tasks. Previously, she was part of the *DisineLab package*. Then it entered the *OrCAD package*. This program can perform various types of circuit analysis and has a number of functions for viewing simulation results. Supplemented by a special module *PSPICE Optimizier*, it makes it possible not only to simulate, but also to optimize the circuit according to various criteria.

SIM 99 SE software module makes it possible to carry out all types of parametric circuit analysis, changing two components at the same time. This module is part of the *P-CAD 2002* and *Protel 99 SE packages*.

View Analog software module has a standard set of simulation functions for mixed analog-to-digital devices. It also allows to simulate the behavior of programming logic circuits. This module is included in the Product Designer package.

Programmable logic integrated circuit design packages. A separate task in the automation of system design is the synthesis of predetermined logic circuits that determine the operation of complex devices and are implemented on programmable logic integrated circuits. For these purposes, the following software products can be used: *Peak FPGA program, PLD 99 SE module, FPGA Studio program, System View package, Fusion/Speed Wave, Fusion/ViewSim, View PLD programs*.

It should be noted that a number of these software packages make it possible not only to simulate and analyze the given circuits of devices and systems, but also to design the topology of the corresponding large integrated circuits (LIC), perform their electromagnetic compatibility analysis and thermal analysis, and also design the corresponding printed circuit boards.

In conclusion, it should be noted that the programming languages, program modules and software packages mentioned in this section are far from a complete list of existing software tools that can be used to simulate and optimize communication systems and networks.

AFTERWORD

This monograph addresses the optimization and mathematical modeling of technical systems, in particular, communication networks. These issues are especially important at the initial stages of design, when it is necessary to choose the best options for constructing communication systems and networks. In this case, it is necessary to build an adequate mathematical model of the system, within the framework of this mathematical model at a formalized level, set the optimality criterion for the system, using which, by solving the corresponding optimization problem, select the optimal version of the system. The use of various mathematical models of systems and various types of optimality criteria leads to the need to solve various types of optimization problems.

In the monograph, the theoretical foundations of scalar and multicriteria system optimization and methods for solving various types of optimization problems, which are concretized taking into account mathematical models of communication networks, are considered in a rather concise form. Examples of solving some problems of optimizing communication networks are given. Information is given about standard software packages for modeling and optimization on computer networks and communication systems that can be used in their computer-aided design.

It should be noted that not all issues that arise during the solution of problems of mathematical modeling and system optimization are presented in sufficient detail due to the limited volume of the monograph. If necessary, an in-depth study of certain issues of mathematical modeling and optimization of systems, it is possible to refer to the corresponding specialized books of a scientific nature, which are listed in the list of references.

REFERENCES

1. Bezruk, V. M., Bidnyi, Yu. M., Omelchenko, A. V. (2011). Informatsiini merezhi zviazku. Part 1. Matematychni osnovy informatsiinykh merezh zviazku. Kharkiv: KhNURE, 292.
2. Zakharchenko, M. V., Horokhov, S. M., Balan, M. M., Hadzhyev, M. M., Korchynskyi, V. V., Lozhkovskyi, A. H. (2010). Matematychni osnovy optymizatsii telekomunikatsiinykh system. Odesa: ONAZ, 240.
3. Hloba, L. S. (2007). Matematychni osnovy pobudovy informatsiino-telekomunikatsiinykh system. Kyiv: Norita-plius, 360.
4. Zaichenko, Yu. P. (2006). Doslidzhennia operatsii. Kyiv: Vydavnychyi dim «Slovo», 816.
5. Popovskyi, V. V., Saburova, S. O., Oliinyk, V. F., Losiev, Yu. I., Aheiev, D. V., Kaliekina, T. H. et. al.; Popovskyi, V. V. (Ed.) (2006). Matematychni osnovy teorii telekomunikatsiinykh system. Kharkiv: Kompaniia SMIT, 563.
6. Larionov, Yu. I., Levykin, V. M., Khazhmuradov, M. A. (2005). Doslidzhennia operatsii v informatsiinykh systemakh. Kharkiv: Kompaniia SMIT, 364.
7. Steklov, V. K., Berkman, L. N., Kilchytskyi, Ye. V. (2004). Optymizatsiia ta modeliuvannia prystroiv i system zviazku. Kyiv: Tekhnika, 576.
8. Liamets, V. Y., Teviashev, A. D. (2004). Systemnii analiz. Kharkiv: KhNURE, 448.
9. Chernoruckii, I. G. (2004). Metody optimizacii v teorii upravleniia. Saint Petersburg: Nitev, 256.
10. Bezruk, V. M., Bukhanko, A. N., Chebotareva, D. V. (2013). Priniatie optimalnykh reshenii v infokommunikaciiakh s uchetom sovokupnosti pokazatelei kachestva. Naukoemkie tekhnologii v infokommunikaciiakh: obrabotka i zaschita informacii. Kharkiv: KHNURE, 104–125.
11. Chebotareva, D. V., Bezruk, V. M. (2013). Mnogokriterialnaia optimizaciia proektnykh reshenii pri planirovanii sotovykh setei mobilnoi sviazi. Kharkiv: Kompaniia SMIT, 148.
12. Klymash, M. M., Burachok, R. A., Andrusiv, T. V. (2009). Metody vyznachennia pokaznyka yakosti posluh v telekomunikatsiinykh merezhakh. Lviv: NU «Lvivska politekhnika», 285.
13. Zaichenko, O. Yu., Zaichenko, Yu. P. (2007). Doslidzhennia operatsii. Kyiv: Vydavnychyi dim «Slovo», 472.
14. Semenov, Iu. V. (2005). Proektirovanie setei sviazi sleduiuschego pokoleniia. Saint Petersburg: Nauka i tekhnika, 240.
15. Vyshnevskyi, V. M. (2003). Teoretycheskye osnovi proektyrovanyia kompiuternikh setei. Moscow: Tekhnosfera, 512.

16. Dymarskii, Ia. S., Krutiakova, N. P., Ianovskii, G. G. (2003). Upravlenie setiami sviazi: Principy, protokoly, prikladnye zadachi. Moscow: ITC «Mobilnye telekommunikacii», 384.
17. Tymchenko, A. A.; Bykov, V. I. (Ed.) (2003). Osnovy systemnoho proektuvannia ta systemnoho analizu skladnykh obiektiv: Osnovy SAPR ta systemnoho proektuvannia skladnykh obiektiv. Kyiv: Lybid, 272.
18. Bezruk, V. M. (2002). Vektorna optymizatsiia ta statystychne modeliuvannia v avtomatyzovanomu proektuvanni system zviazku. Kharkiv: KhNURE, 164.
19. Vinnickii, V. P., Khilenko, V. V. (2002). Metody sistemnogo analiza i avtomatizacii proektirovaniia telekommunikacionnykh setei. Kyiv: Interlink, 192.
20. Nogin, V. D. (2002). Priniatie reshenii v mnogokriterialnoi srede: kolichestvennii podkhod. Moscow: FIZMATLIT, 176.
21. Steklov, V. K., Berkman, L. N. (2002). Proektuvannia telekomunikatsiinykh merezh. Kyiv: Tekhnika, 792.
22. Nazarov, A. N. (2002). Modeli i metody rascheta strukturno-setevykh parametrov ATM setei. Moscow: Goriachaia liniia-Telekom, 256.
23. Ventcel, E. S. (2001). Issledovanie operacii. Zadachi, principy, metodologiia. Moscow: Vysshaia shkola, 208.
24. Chernoruckii, I. G. (2001). Metody optimizacii i priniatiia reshenii. Saint Petersburg: Iz-vo «Lan», 384.
25. Zakharchenko, M. V., Steklov, V. K., Kniazeva, N. O. et. al. (1996). Avtomatyzatsiia proektuvannia prystroiv, system ta merezh zviazku. Kyiv: Radioamator, 268.
26. Shvarc, M. (1992). Seti sviazi: protokoly, modelirovanie i analiz. Vol 1. Moscow: Nauka, 336.
27. Shtoier, R. (1992). Mnogokriterialnaia optimizaciia. Teoriia, vychisleniia i prilozheniia. Moscow: Radio i sviaz, 504.
28. Ashmalov, A. L., Tikhonov, V. A. (1991). Teoriia optimizacii v zadachakh i uprazhneniiakh. Moscow: Nauka, 488.
29. Dmitriev, A. N., Ekupov, N. D., Shestopalov, A. M., Moiseev, Iu. G. (1990). Mashinnye metody rascheta i proektirovaniia sistem elektrosviazi i upravleniia. Moscow: Radio i sviaz, 271.
30. Zaichenko, Iu. P., Shumilova, S. A. (1990). Issledovanie operacii. Kyiv: Vischa shkola, 237.
31. Bersekas, D., Galager, Ch. (1989). Seti peredachi dannykh. Moscow: Mir, 544.
32. Perepelica, V. A. (1989). Mnogokriterialnye zadachi teorii grafov. Algoritmicheskii podkhod. Kyiv: UMK VO, 68.
33. Zaichenko, Iu. P. (1988). Issledovanie operacii. Kyiv: Vysshaia shkola, 549.
34. Morozov, V. K., Doganov, A. V. (1987). Osnovy teorii informacionnykh setei. Moscow: Vysshaia shkola, 269.

35. Shvarc, M. (1987). Seti EVM: Analiz i proektirovanie. Moscow: Mir.
36. Nogin, V. D., Protodiakonov, I. O., Evlampiev, I. I. (1986). Osnovy teorii optimizacii. Moscow: Vysshaia shkola, 379.
37. Zaichenko, Iu. P., Gonta, Iu. V. (1986). Strukturnaia optimizaciia setei EVM. Kyiv: Tekhnika, 168.
38. Dubov, Iu. A., Travkin, S. I., Iakimec, V. N. (1986). Mnogokriterialnye modeli formirovaniia i vybora variantov sistem. Moscow: Nauka, 324.
39. Berezovskii, B. A., Baryshnikov, Iu. M., Borzenko, V. I., Kepner, L. M. (1986). Mnogokriterialnaia optimizaciia. Matematicheskie aspekty. Moscow: Nauka, 296.
40. Morozov, V. V., Sukharev, A. G., Fedorov, V. V. (1986). Issledovanie operacii v zadachakh i uprazhneniiakh. Moscow: Vysshaia shkola, 285.
41. Sukharev, A. G., Timokhov, V. A., Fedorov, V. V. (1986). Kurs metodov optimizacii. Moscow: Nauka, 328.
42. Brakhtman, T. R. (1984). Mnogokriterialnost i vybor alternativ v tekhnike. Moscow: Sov. Radio, 326.
43. Poliak, B. T. (1983). Vvedenie v optimizaciiu. Moscow: Nauka, 384.
44. Podinovskii, V. V., Nogin, V. D. (1982). Pareto-optimalnye resheniia mnogokriterialnykh zadach. Moscow: Nauka, 256.
45. Kleinrok, L. (1979). Vychislitelnye sistemy s ocherediami. Moscow: Mir, 600.
46. Moiseev, N. N., Ivanilov, N. N., Stoliarova, E. M. (1978). Metody optimizacii. Moscow: Nauka, 352.
47. Pashkeev, S. D., Minizov, R. I., Mogilevskaia, V. D. (1976). Mashinnye metody optimizacii v tekhnike sviazi. Moscow: Sviaz, 272.
48. Martin, Dzh. (1975). Sistemnii analiz peredachi dannykh. Vol. 2. Proektirovanie sistem peredachi dannykh. Moscow: Mir, 431.
49. Gutkin, L. S. (1975). Optimizaciia radioelektronnykh ustroistv po sovokupnosti pokazatelei kachestva. Moscow: Sov. radio, 358.
50. Semenets V., Grebennik I., Listrovoy S., Minuhin S., Ovezgeldyev A. (2019). Modeli i metody kombinatornoy optimizatsii v proyektirovanii i upravlenii. Kyiv: Naukova dumka, 256.
51. Bezruk V. M., Chebotaryova D. V., Skorik Yu. V. (2017). Multi-criteria analysis and selection of telecommunications facilities. Kharkov: SMIT Company, 268.

Scientific publication

Bezruk V. M., Semenets V. V., Chebotarova D. V., Kaliuzhniy N. M.,
Guo Qiang, Zheng Yu

OPTIMIZATION AND MATHEMATICAL MODELING OF
COMMUNICATION NETWORKS

MONOGRAPH. 2nd EDITION

Signed in print 28.07.2019. Format 60×84/16. Offset paper.
Printed on risograph. Typeface Times New Roman. Conventional printing sheets 12
Circulation of 300 copies. Order No. 03m-2019. Negotiated price

---

PC «Technology Center»
Enlisting the subject of publishing No. 4452 – 10.12.2012
Address: Shatilova dacha str., 4, Kharkiv, Ukraine, 61145
