https://doi.org/10.31891/csit-2025-3-10
UDC 004.75

Olena ZHDANOVA, Maksym BUKASOV, Svyatoslav TSYMBAL,
Viacheslav CHYMSHYR, Rostyslav OMELCHENKO
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

INFORMATION SYSTEM FOR SERVICE PORTFOLIO FORMATION FOR INFOCOMMUNICATION PROVIDERS

This paper addresses the challenge of service portfolios formation for infocommunication service providers. A comprehensive analysis of existing approaches enabled the formulation of the problem within the framework of contemporary business models and underscored the necessity of developing new effective methods. These methods must reflect the specifics of interaction between IT companies, service providers, and end users, as well as the inherent characteristics of services and their delivery environments. The paper substantiates the choice of methodological foundations underlying the proposed system for service portfolio formation. The formal problem is categorized as a nonlinear, multicriteria Boolean programming task. Growing demands for alignment between service packages and user needs, coupled with the resource constraints of modern IT infrastructures and the complexity of inter-service dependencies, highlight the need to solve large-scale optimization problems. To address this, a hybrid method is proposed, combining problem decomposition into subproblems, the application of metaheuristic techniques for their resolution, and heuristic procedures for integrating partial solutions. The paper presents experimental results demonstrating the effectiveness of the proposed approach. Additionally, it outlines system-level design solutions that support the development and implementation of the information system for service portfolio formation.

Keywords: service portfolio, infocommunication service provider, service portfolio formation model, multicriteria optimization, metaheuristic methods, information system

Олена ЖДАНОВА, Максим БУКАСОВ, Святослав ЦИМБАЛ,
Вячеслав ЧИМШИР, Ростислав ОМЕЛЬЧЕНКО
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

ІНФОРМАЦІЙНА СИСТЕМА ФОРМУВАННЯ ПАКЕТІВ СЕРВІСІВ ДЛЯ ПРОВАЙДЕРІВ ІНФОКОМУНІКАЦІЙ

Стаття присвячена створенню інформаційної системи формування пакетів сервісів для провайдерів інфокомунікацій. Зростання вимог до узгодженості пакетів сервісів із потребами користувачів і ресурсними обмеженнями сучасної IT-інфраструктури, а також складність міжсервісних залежностей зумовлюють необхідність розробки нових моделей і відповідних методів розв'язання. Проведений аналіз існуючих підходів до формування пакетів сервісів дозволив сформулювати математичну постановку задачу з урахуванням сучасних бізнес-концепцій та обґрунтувати необхідність розроблення нових ефективних методів, які враховують особливості взаємодії IT-компаній, провайдерів і їхніх клієнтів, специфіку сервісів та середовища їх надання. Формальна постановка задачі належить до класу нелінійних багатокритеріальних задач оптимізації. Обґрунтовано вибір методів, що покладені в основу реалізації системи формування пакетів сервісів.. У статті запропоновано комбінований метод формування пакетів сервісів, який поєднує декомпозицію задачі на підзадачі, метаевристичні підходи до їх розв'язання та евристичні процедури інтеграції часткових розв'язків. Наведено результати експериментального дослідження ефективності запропонованого методу. Описано системні рішення, які покладено в основу проєктування та реалізації інформаційної системи формування пакетів сервісів.

Ключові слова: пакети серсвісів, провайдер інфокомунікаційних сервісів, модель формування пакетів сервісів, багатокритеріальна задача оптимізації, метаевристичні методи, інформаційна система

Received / Стаття надійшла до редакції 15.07.2025
Accepted / Прийнята до друку 25.08.2025

Introduction

The current stage of information technology development is characterized by a wide variety of infocommunication services available to end users. At the same time, it imposes increasingly stringent requirements on infocommunication service providers, as well as on IT companies responsible for delivering technological support across all stages of the service life cycle. These challenges necessitate the resolution of a range of complex scientific and applied problems.

The business environment is continuously evolving. On the one hand, this is reflected in the widespread adoption of the service-oriented approach across all areas of human activity [1, 2]; on the other hand, it manifests in the changing nature of interactions between business units and IT departments within companies and organizations [3, 4]. Businesses now expect IT departments not only to support ongoing operations, but also to play an active role in achieving strategic objectives, implementing new working methods, and making effective use of available opportunities.

In this context, new scientific and practical challenges emerge, giving rise to distinct research directions. These challenges are becoming increasingly relevant due to qualitative shifts in the architecture of IT environments, in particular the consolidation of IT infrastructures and the development of global digital platforms. Addressing them requires the development of comprehensive methodologies for the creation, maintenance, and evolution of information systems that integrate business, operational, and technological aspects of modern organizational activity.

The adoption of the service-oriented approach in the infocommunication domain introduces additional dimensions, driven by its growing importance in shaping the technological foundations of the emerging information society.

The service life cycle concept provides a useful framework for decomposing the overall problem into a series of manageable subproblems. However, the complexity and interdependence of business, operational, technological, and social factors in the activities of IT companies, infocommunication providers, and their clients result in a substantial number of such subproblems – each typically related to specific stages of the service life cycle.

Furthermore, solving these subproblems requires a coordinated and integrated decision-making framework that ensures mutual benefits for all stakeholders involved in the delivery and consumption of services.

While the service-oriented approach continues to gain traction in the infocommunication industry, its further advancement depends on the resolution of increasingly complex and multidimensional challenges.

1. The Problem of Service Portfolio Formation

The infocommunication industry is undoubtedly a leader in the adoption of the service-oriented approach. Professional communities within the industry have systematically documented and described the business processes of infocommunication service providers [5]. In a competitive environment, IT companies develop information technologies to support these business processes. As a result, the information systems of infocommunication providers are designed, implemented, and operated based on modern methodologies, whose foundational principle is the integration of diverse solutions that enable the support of more than a hundred distinct business processes.

Given the constant evolution characteristic of the sector, coupled with the ongoing pursuit of enhanced efficiency by service providers, the informatization of enterprises within the infocommunication industry represents a continuous process of improvement and development of information systems and technologies, alongside the implementation of contemporary concepts.

Today, the resolution of complex scientific and practical challenges arising during the development of information systems for infocommunication providers is based on service life cycle concepts [6], End-to-End (E2E) approaches [7], platform-based solutions [8], ISTM [9, 10], and ITIL frameworks [11]. The rapid advancement of artificial intelligence fosters the adoption of large language models (LLMs), retrieval-augmented generation (RAG) systems, and intelligent agents to support providers' business processes. An important factor is also the influence of provider clients, who form a powerful community capable of assessing service quality and providing recommendations for its improvement. Equally significant is the impact of the modern complex environment in which providers, their clients, and IT companies – the developers of information systems, technologies, and tools – operate to ensure the design, implementation, and maintenance of services throughout all stages of their life cycle.

In addition to addressing the complex challenges of organizing interactions among information system components to ensure the desired behavior of IT companies or providers as controllable entities, it is also necessary to solve problems related to the support of each process across all stages of the service life cycle. This requires the formulation of appropriate mathematical problem statements, the development of models and methods, experimental validation of their effectiveness, and the creation of corresponding technologies based on these results.

The aforementioned technologies constitute the foundation of an integrated solution for managing the operations of an infocommunication service provider. At the same time, effective collaboration between IT companies and providers requires the development and implementation of comprehensive solutions that encompass various aspects of service design, implementation, and maintenance. Within this context, the problem of forming service portfolios offered by IT companies to infocommunication providers becomes particularly significant. The need to address this problem arises both at the stage of creating the initial service configuration for the provider and during the final determination of the set of services to be supported by the provider's information system.

Moreover, this problem arises not only within the interaction between the IT company and the infocommunication provider, but also in the interaction between the provider and its clients, and is typically addressed in multiple stages.

In this paper, the problem of service portfolio formation is considered primarily in the context of the initial configuration of services for the provider and within the scope of interaction between the IT company and the infocommunication service provider. However, with appropriate customization, the information system developed on the basis of the proposed models and methods can also be applied:
at the stage of finalizing the service portfolio to be supported by the provider's information system;
in the course of interaction between the infocommunication provider and its clients.

2. Review of Existing Solutions

The problem of service portfolio formation has attracted the attention of researchers for quite some time. However, the approaches proposed in the literature are not always suitable for application in the infocommunication industry. This is largely due to the specific characteristics of the industry, particularly the presence of technological and quality-related constraints, which are difficult – or even impossible – to fully account for within the models and methods that have proven effective in other industries.

In general, methods for solving the service portfolio formation problem in other domains are based on leveraging accumulated experience in service delivery. Among these, statistical methods [12] and clustering techniques [13] are particularly noteworthy. Statistical data on service usage – especially when user identity and timestamps are recorded and processed correctly – have shown promising results across various applied fields. Such approaches have also proven to be effective in addressing service portfolio formation tasks within the infocommunication industry [14]. The same holds true for clustering methods: they have demonstrated high efficiency in other industries and have been successfully applied in the infocommunication industry as well [15].

However, ongoing transformations in the infocommunication industry – driven by the evolution of business models and the growing emphasis on mutual benefit – necessitate solving the service portfolio formation problem on the basis of well-founded mathematical models and efficient computational methods.

Moreover, the specific nature of the domain requires consideration of additional constraints, including technological limitations and inter-service dependencies. Among such constraints are SLA (Service Level Agreement) indicators, which define the required quality of service delivery and must be respected when forming the final package for each provider. All of these constraints can be effectively incorporated by formalizing the service portfolio formation task as a mathematical programming problem.

Consequently, the focus on creating mutually beneficial service portfolios calls for the development of adequate mathematical models that incorporate relevant objective functions and constraints reflecting the specifics of service-based interaction. Several approaches based on this concept have already been proposed in the literature (e.g., [16]).

Thus, beyond statistical and clustering methods, there is a growing need to apply alternative approaches – specifically, efficient methods of mathematical programming and heuristic algorithms. Notable examples of approaches that successfully take into account the specifics of the service portfolio formation problem in the infocommunication industry, while also expanding the set of applicable methods, can be found in studies such as [17, 18].

The literature also presents other models and methods for solving the service portfolio formation problem in the infocommunication industry (e.g., [19]). However, the ongoing evolution of the industry, along with the contextual variability of the problem in different applied settings, highlights the need for novel mathematical models and more effective solution methods.

This paper explores the potential of decomposing the multicriteria problem into single-criterion subproblems, as well as the use of combinations of metaheuristic algorithms for solving these subproblems and integrating the resulting partial solutions.

3. Formal Statement of the Service Portfolio Formation Problem

Given:

- a set of services $S = \{S_1, S_2, \dots, S_k\}$, where k is the number of services offered by the IT company to providers;
- a set of infocommunication providers $P = \{P_1, P_2, \dots, P_m\}$, where m is the number of providers;
- a set of interdependent services $R_i = \{S_1^i, S_2^i, \dots, S_{n_i}^i\}, i = 1, \dots, k,$ where each service $S_g^i \in S, g = 1, \dots, n_i,$ can only be provided if all services from the corresponding subset R_i are also included;
- a preference price matrix $\mathbf{D} = \| d_{ij} \|$, where d_{ij} is the base price charged by the IT company for providing service S_i to provider $P_j, i = 1, \dots, k; j = 1, \dots, m$;
- a provider revenue matrix $\mathbf{P} = \| p_{ij} \|$, where p_{ij} denotes the revenue obtained by provider P_j from offering service S_i to its clients, $i = 1, \dots, k; j = 1, \dots, m$;
- L – the number of resource types used in provision of services;
- T_l – the total amount of resource l available to the IT company during the planning period, $l = 1, \dots, L$;
- a three-dimensional resource utilization tensor $\boldsymbol{\beta} = \| \beta_{ijl} \|$, where β_{ijl} denotes the amount of resource l required by the IT company to provide service S_i to provider P_j, $i = 1, \dots, k,\ j = 1, \dots, m, l = 1, \dots, L$;
- a provider cost matrix $\mathbf{B} = \| b_{ij} \|$, where b_{ij} denotes the cost incurred by provider P_j for delivering service S_i to its clients, $i = 1, \dots, k, j = 1, \dots, m$;
- a matrix of minimum relative service values $\mathbf{C} = \| c_{ij} \|$, where c_{ij} represents the minimum acceptable relative value of service S_i provided by provider $P_j, i = 1, \dots, k; j = 1, \dots, m$;
- G – the number of SLA indicators used to characterize the services;
- a three-dimensional SLA tensor $\mathbf{A} = \| a_{ijg} \|$, where a_{ijg} is the value of SLA indicator g for service S_i offered to provider P_j as listed in the IT company's service catalog;
- a matrix of required SLA values $\boldsymbol{\alpha} = \| \alpha_{ig} \|$, where α_{ig} is the SLA value for indicator g of service S_i, as specified in the service delivery terms between the provider and its clients.

To find:

- the composition of service portfolios for each provider, represented by a Boolean matrix $\mathbf{V} = \| v_{ij} \|$, where

$$v_{ij} = \begin{cases} 1, \text{if service } S_i \text{ is included in the portfolio of provider } P_j, \\ 0, \text{otherwise}, \end{cases} \quad i = 1, \dots, k, j = 1, \dots, m;$$

- a discount matrix $\mathbf{R} = \| r_{ij} \|$, where $r_{ij} \in [0,1)$ denotes the discount applied to the preference price d_{ij} for providing service S_i to provider P_j, $i = 1, \dots, k$, $j = 1, \dots, m$.

The objective of the problem is to determine such a configuration of service portfolios **V** and discount matrix **R** that maximizes the total benefit for both the IT company and each individual provider.

Given the notation introduced above, the objective functions for the IT company and for each provider $P_j, j = 1, \dots, m$, are defined by formulas (1) and (2), respectively:

$$W = \sum_{j=1}^{k} \sum_{i=1}^{m} d_{ij}(1 - r_{ij})v_{ij}, \quad (1)$$

$$Q_j = \sum_{i=1}^{m} (p_{ij} - d_{ij}(1 - r_{ij}))v_{ij}, j = 1, \dots, m. \quad (2)$$

It is worth noting that expressions (1) and (2) not only quantify the benefits gained by the IT company and the providers as a result of delivering service portfolios; they also support the implementation of a modern business concept aimed at achieving mutual benefit for both parties.

In practice, however, the solutions corresponding to the optimal values of the objective functions are typically constrained. These constraints may be related to available resources, economic factors, quality requirements, or technological limitations.

The resource constraints are defined by expression (3):

$$\sum_{j=1}^{k} \sum_{i=1}^{m} \beta_{ijl} v_{ij} \le T_l, l = 1, \dots, L. \quad (3)$$

Constraints on the relative value of services for the provider are defined by expression (4):

$$c_{ij} v_{ij} \le \frac{p_{ij}}{b_{ij}}, \quad i = 1, \dots, k, j = 1, \dots, m. \quad (4)$$

Constraints on the SLA indicators of the provided services are defined by expression (5):

$$a_{ijg} v_{ij} \le \alpha_{ig}, \quad i = 1, \dots, k, j = 1, \dots, m, g = 1, \dots, G. \quad (5)$$

Constraints on inter-service dependencies for the services provided to the providers are defined by expression (6):

$$v_{ij} v_{tj} = \rho_{it}, \quad i = 1, \dots, k, j = 1, \dots, m, t = 1, \dots, k, \quad (6)$$

where

$$\rho_{it} = \begin{cases} 1, \text{if service } S_i \text{ in the provider's portfolio requires service } S_t, \\ 0, \text{otherwise}. \end{cases}$$

The matrix $\boldsymbol{\rho} = \| \rho_{it} \|$ defines all pairs of interdependent services, as determined by the subsets $R_i, i = 1, \dots, k$. As noted above, the technological constraints (6) arise from the fact that the provision of certain services may require the support of specific functionalities from other services.

Thus, we arrive at the formal problem statement (1)–(6) for the service portfolio formation task. This problem belongs to the class of multicriteria mixed-integer programming problems. The proposed formulation takes into account the specific features of the business operations of IT companies and infocommunication providers. It aims to support the achievement of individual business goals for the IT company (objective function (1)) and for each provider (objective function (2)). To implement the concept of mutual benefit, discount parameters have been introduced into the model. In this study, we consider service-specific discounts provided to each individual provider.

Now proceed to discuss the methods for solving the formulated service portfolio formation problem.

4. A Combined Method for Solving the Service Portfolio Formation Problem

To effectively address the specific features of the service portfolio formation problem, it is necessary to expand the set of applicable solution methods.

Statistical approaches and clustering-based techniques enable the use of accumulated experience in service portfolio design. However, to reflect current trends aimed at creating mutually beneficial service portfolios, additional approaches must be applied–particularly mathematical programming methods and metaheuristic algorithms.

The formal problem statement possesses several features related to the definition of variables that describe the solution structure. First, the structure of service portfolio is defined by the decision variables v_{ij}, which are incorporated into the objective functions of both the IT company and the providers. The variables r_{ij},, which also appear in both objective functions, determine the discount rates applied to the preference prices. Second, two groups of objective functions have been formulated: (1) aims to maximize the IT company's benefit, and (2) aims to maximize the benefit for each provider. Both groups of functions take into account the impact of discounts on revenue. Third, a system of constraints (3)–(6) has been established, which also accounts for discounting and ensures resource feasibility, economic viability, and technological consistency of the resulting service portfolios.

It is advisable to use combined methods for solving multicriteria problems based on the following approaches:

- decomposition of the multicriteria problem into corresponding single-criterion subproblems – separately for the IT company and for the providers;
- application of metaheuristic algorithms adapted to the specifics of each subproblem;
- identification of a common subset within the solution sets of the IT company's and providers' subproblems;
- heuristic improvement of the combined solution by searching for a compromise solution.

Before proceeding to the description of a specific variant of the combined method for solving the service portfolio formation problem, we first formulate the corresponding $m+1$ single-criterion subproblems.

Subproblem for the IT company:

- criterion: maximization of objective function (1);
- constraints: (3)–(6).

Subproblem j for provider $P_j, j = 1, \dots, m$:

- criterion: maximization of the objective function

$$Q_j = \sum_{i=1}^{m} (p_{ij} - d_{ij}(1 - r_{ij}))v_{ij} ;$$

- constraints: (3)–(6).

Thus, two models are considered: one focused on the benefit of the IT company, and the other – on the benefit of the providers. In each case, the result is the formation of a service portfolio that maximizes the profit for the corresponding party.

Next, based on the solutions to the subproblems for the IT company and the providers, a compromise solution is sought using a heuristic algorithm that simulates mutual concessions between the parties within the defined discount system.

Description of the Combined Method for Solving the Service Portfolio Formation Problem

Stage 1. Solving the subproblem for the IT company

The most beneficial service portfolio for the IT company is formed, taking into account all constraints – including resource limitations, provider-specific conditions, and inter-service dependencies.

Stage 2. Solving the subproblems for providers $j = 1, \dots, m$

For each provider, the most profitable service portfolio is formed while satisfying all imposed constraints.

Stage 3. Searching for a compromise solution

Based on the solutions obtained in Steps 1 and 2, a compromise solution to the overall problem (1)–(6) is found using a heuristic algorithm that models mutual trade-offs between the IT company and the providers.

Now proceed to describe the algorithms and methods used in each stage of the combined approach. To ensure high-quality solutions, it is advisable to develop several algorithms for each stage and select the most appropriate one depending on the specifics of the particular instance of the problem, based on the results of experimental evaluation.

5. Algorithms for Implementing the Stages of the Combined Method for Solving the Service Portfolio Formation Problem

To reduce the complexity of the mathematical model, focus on the key aspects of the service portfolio formation problem, and demonstrate the specific features of the developed solution algorithms, we introduce several simplifications. Henceforth, a "service" will refer to a group of interrelated services. We also assume that only one type of IT company resource is limiting (i.e., in constraint (3), $L = 1$, and the third index in β_{ijl} can be omitted: $\beta_{ij1} = \beta_{ij}$). This simplification is justified, as in most practical cases a single resource type – such as human, computational, or time-based – is critical and constrains the ability to deliver services, while other resources are either overprovisioned or do not reach their upper limits. This allows us to focus on the essential aspects of the service portfolio formation problem without loss of generality in the model. Furthermore, we assume that constraints (4) and (5) are satisfied for all feasible solutions.

5.1. Algorithms for Solving the IT Company's Subproblem (Stage 1)

To implement Stage 1, two approximate algorithms were developed: a probabilistic-greedy heuristic algorithm and an ant colony optimization (ACO) algorithm.

5.1.1. Probabilistic-Greedy Algorithm for Solving the IT Company's Subproblem

The concept of the proposed probabilistic-greedy algorithm is based on a combination of greedy and probabilistic selection principles for constructing solutions by processing fragments related to services, providers, and discounts. At the initial stage, for each pair $S_i - P_j$, a so-called resource value is calculated. This value is defined as the ratio of the preferential price of providing the service to the amount of resource required to support it. Taking the discount into account, the resource value is computed using the following formula:

$$\theta_{ij} \coloneqq \frac{d_{ij}\left(1 - r_{ij}\right)}{\beta_{ij}}.$$

The algorithm combines greedy and probabilistic selection principles in the following way.

First, greedy selection is used to quickly improve the current solution based on the following factors:

- it is grounded in the idea of prioritizing $S_i - P_j$ assignments with high resource utilization efficiency;
- the value of each "service – provider" pair (θ_{ij}) directly depends on the ratio of profit to resource consumption, corresponding to the classical greedy criterion of maximum benefit per unit of cost;
- therefore, more effective assignments have a higher probability of being selected.

Second, unlike deterministic greedy strategies that always choose the best available pair according to the criterion, this algorithm incorporates a probabilistic selection mechanism, which helps avoid local optima by considering the following:

- the selection probability for each admissible assignment is proportional to its value θ_{ij}, which adds an element of exploration to the solution space;
- the algorithm is executed multiple times, enabling the discovery of better combinations through repeated random selections.

Scheme of the Probabilistic-Greedy Algorithm for Solving the Subproblem of Stage 1

Input $k, m, \| d_{ij} \|, \| \beta_{ij} \|, \| r_{ij} \|, T, N$
Output v_{best} // best service portfolio found
W_{best} // corresponding best value of objective function
Auxiliary variables
$z_{ij},\ i = 1, \dots, k; j = 1, \dots, m$ // admissibility flag for assignment $S_i - P_j$ (true if sufficient resource is available)
Compute $\theta_{ij} \coloneqq \frac{d_{ij}(1-r_{ij})}{\beta_{ij}}$ for all i, j //"resource value"
$W_{best} \coloneqq 0$ // current value
for $n \coloneqq 1$ **to** N
$v_{ij} \coloneqq 0$ for all i, j // current package
$z_{ij} \coloneqq 1$ for all i, j // all assignments initially admissible
$T_{used} \coloneqq 0$ // current total resource usage
while (there are admissible assignments) //resource for that $\beta_{ij} \le T - T_{used}$
For all admissible pairs i, j **compute** selection probability: $p_{ij} \coloneqq \frac{\theta_{ij}}{\sum_{ij/z_{ij}=1} \theta_{ij}}$
Randomly **select** a pair i^*, j^* according to p_{ij}

$v_{i^*j^*} := 1$
$z_{i^*j^*} := 0$
$T_{used} := T_{used} + \beta_{i^*j^*}$
for all unassigned pairs i, j, who do not have enough current resources, **set** $z_{ij} := 0$
endwhile
Compute W for the obtained portfolio v
if $W > W_{best}$ **then** ($W_{best} := W$ та $v_{best} := v$)
endfor
Return v_{best}, W_{best}

5.1.2. Ant Colony Optimization Algorithm for Solving the IT Company's Subproblem

The probabilistic-greedy algorithm effectively combines the idea of prioritizing service-provider pairs $S_i - P_j$ with high utility (greedy selection) and randomness in choice (probability), allowing the algorithm to avoid local optima. However, it has several limitations: it lacks learning from past experience, it does not incorporate collective memory, it scales poorly, and it has a weak balance between exploration and exploitation – although it selects based on the value θ_{ij}, it does not reinforce the long-term benefit of previously found solutions.

The ant colony optimization algorithm helps overcome the limitations of the probabilistic-greedy algorithm by combining local heuristics (the value of each pair) with pheromone memory, which supports self-organized and adaptive learning. This enables flexible control over the balance between exploring new combinations and reinforcing already successful solutions. As a result, ACO represents a natural evolution of the approach used to solve the problem of service portfolio formation for providers.

The parameters of the algorithm are as follows:

- τ_0 – initial pheromone level, which defines the starting amount of pheromone for each possible $S_i - P_j$ pair;
- α – pheromone weight, which determines how strongly an ant is influenced by pheromone information when making a decision (a higher α value increases the impact of prior experience, i.e., exploitation);
- β – heuristic information weight, which indicates how much an ant relies on the local value of an assignment θ_{ij} (a higher β value enhances greedy behavior, i.e., local effectiveness);
- $\rho \in (0,1]$ – pheromone evaporation coefficient, which controls the rate at which pheromone diminishes over time (regulates forgetting old information and promotes search flexibility);
- A – the number of ants in the population.

The key components of ACO are: probability calculation, estimation of the upper bound of the objective function values, the pheromone update procedure, and the termination condition.

Transition probability calculation. To construct a solution, an ant selects the next $S_i - P_j$ pair with the probability:

$$p_{ij} := \frac{(\tau_{ij})^{\alpha}(\theta_{ij})^{\beta}}{\sum_{(i'j')_{ij}\in E}(\tau_{i'j'})^{\alpha}(\theta_{i'j'})^{\beta}}, \quad (7)$$

where τ_{ij} is the current pheromone level for $S_i - P_j$ pair; θ_{ij} is the value of the assignment $S_i - P_j$; E is the set of feasible assignments, i.e., those assignments for which sufficient resources are available at the current step.

Upper bound estimation of the objective function values can be obtained using the rule "Selecting the most efficient pairs until resources are exhausted." For this purpose, all pairs $S_i - P_j$ are first sorted in non-increasing order of θ_{ij}. Then, as long as the available resource T is not exhausted, pairs are added to the sum according to:

$$W^{max} = \sum_{(i,j)\in F} (1 - r_{ij}) d_{ij},$$

where F is the set of top pairs in the sorted list such that $\sum \beta_{ij} \le T$.

Pheromone update. After each iteration, the pheromone level on each edge is updated according to the formula:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{l=1}^{L} \Delta\tau_{ij}^{l}, \quad (8)$$

where $\Delta\tau_{ij}^{l}$ is the amount of pheromone deposited by ant l on the pair $S_i - P_j$:

$$\Delta\tau_{ij}^{l} = \begin{cases} \frac{W_l}{W_{max}}, \text{if the pair } S_i - P_j \text{ is included in the portfolio,} \\ 0, \text{otherwise,} \end{cases}$$

where W_l is the value of the objective function for the solution generated by ant l, and W^{max} is the upper bound of the objective function values.

Termination condition of the algorithm. The algorithm stops after reaching a fixed number of iterations N. The value of N depends on the problem size, and the exact relationship has been determined experimentally.

Scheme of the Ant Colony Optimization (ACO) Algorithm for the IT Company

Input $k, m, \| d_{ij} \|, \| \beta_{ij} \|, \| r_{ij} \|, T, A, \tau_0, \alpha, \beta, \rho, N$
Output v_{best} , W_{best}
Auxiliary variables
  z_{ij}, $i = 1, \dots, k; j = 1, \dots, m$ // indicator of the feasibility of assigning $S_i - P_j$ (i.e., whether enough resources are available at the current step)
  τ // current pheromone matrix
Compute $\theta_{ij} := \frac{d_{ij}(1-r_{ij})}{\beta_{ij}}$ for all i, j // "value" of a unit of resource
  // for the assignment $S_i - P_j$
$W_{best} := 0$ // current best (record)
$\tau_{ij} := \tau_0$ for all i, j // current pheromone
for n **from** 1 **to** N the termination condition is not met
  for a **from** 1 **to** A // loop over ants
    $v_{ij} := 0$ for all i, j // current portfolio
    $z_{ij} := 1$ for all i, j // all assignments initially admissible
    $T_{used} := 0$ // used resource for the current portfolio v

    while (there are allowed pairs) //for which resource $\beta_{ij} \le T - T_{used}$
      For each allowed pair i, j **compute** selection probability p_{ij} according to formula (7)
      Randomly **select** pair i^*, j^* in accordance with p_{ij}
      $v_{i^*j^*} := 1$
      $z_{i^*j^*} := 0$
      $T_{used} := T_{used} + \beta_{i^*j^*}$
      for all allowed pairs i, j, where $\beta_{ij} > T - T_{used}$, **set** $z_{ij} := 0$
    endwhile
    Compute profit W for current portfolio v
    if $W > W_{best}$ **then** ($W_{best} := W$ та $v_{best} := v$)
  endfor
  Update pheromone matrix according to formula (8)
endfor
Return v_{best} and W_{best}

5.2. Heuristic Algorithms for Solving the Subtask for Providers (Stage 2)

For this stage, two algorithms have also been developed: a probabilistic-greedy algorithm and an ant colony algorithm. These algorithms are similar to those presented in sections 5.1.1 and 5.1.2, but all actions in them are performed from the perspective of profitability specifically for providers. That is, for each pair $S_i - P_j$, the so-called value of a unit of resource is calculated, which is defined as the ratio of the provider's revenue from the service to the amount of resource required by the IT company to support it:

$$\theta_{ij} := \frac{p_{ij} - d_{ij}(1 - r_{ij})}{\beta_{ij}}.$$

5.3. Algorithm for Finding a Compromise Solution (Stage 3)

Let's begin with a formal statement of the problem of stage 3.

Given:

- $v^1 = \{v_{ij}^1\}$ is the solution obtained at Stage 1 according to the criterion of maximizing the IT company's profit;
- $v^2 = \{v_{ij}^2\}$ is the solution obtained at Stage 2 according to the criterion of maximizing the providers' profit.

The task is to find a compromise solution $v = \{v_{ij}\}$, which balances the interests of both parties. To do this, weights $\omega_1, \omega_2 \in (0,1)$ are preassigned to the criteria of the IT company and the providers, respectively, such that $\omega_1 + \omega_2 = 1$

The idea behind the compromise solution algorithm is as follows:

- for each pair (i, j), if it appears in both solutions – it is included in the compromise solution;
- for the remaining pairs – their significance is evaluated using a weighted compromise criterion;
- the best of the feasible pairs are then gradually added until the resource limit is reached.

Scheme of the Compromise Solution Finding Algorithm

Input $k, m, T, v^1 = \{v_{ij}^1\}, v^2 = \{v_{ij}^2\}, \omega_1, \omega_2$
Output v // compromised service portfolio
W, Q // objective function values
$v_{ij} \coloneqq 0$ for all i, j // current portfolio
$T_{used} \coloneqq 0$ // used resource for the current service portfolio v
// Adding assignments that are common to both solutions to the portfolio
for $i \coloneqq 1$ **to** k та $j \coloneqq 1$ **to** m
  if ($v_{ij}^1 = 1$ і $v_{ij}^2 = 1$ і $T_{used} + \beta_{ij} \le T$) **then**
    $v_{ij} \coloneqq 1$
    $T_{used} \coloneqq T_{used} + \beta_{ij}$
  endif
endfor
// Computation of compromise value for the remaining assignments
$C \coloneqq \emptyset$ // list of candidates for inclusion in the portfolio
for $i \coloneqq 1$ **to** k та $j \coloneqq 1$ **to** m
  if ($v_{ij}^1 + v_{ij}^2 = 1$) **then**
    $\theta_{ij} \coloneqq \dfrac{\omega_1 d_{ij}(1 - r_{ij}) + \omega_2 (p_{ij} - d_{ij}(1 - r_{ij}))}{\beta_{ij}}$
    $C \coloneqq C \cup (i, j, \theta_{ij})$
  endif
endfor
Sort list C in non-increasing order of θ_{ij}
// Adding assignments until the resource is exhausted
for $c \coloneqq 1$ **to** $|C|$
  if ($T_{used} + \beta_{ij} \le T$) **then**
    $v_{ij} \coloneqq 1$
    $T_{used} \coloneqq T_{used} + \beta_{ij}$
  endif
endfor
For portfolio v **compute** W and Q
Return v, W, Q

6. Results of Experimental Study of the Developed Algorithms

6.1 Comparative Analysis of Algorithms for Solving the Subtask for the IT Company (Stage 1)

In the first three experiments, problems were generated where $k = m = 10$, $d_{ij} \in [20; 100]$, $\beta_{ij} \in [10; 50]$, $T \in [0.2 \sum_i \sum_j \beta_{ij}; 0.8 \sum_i \sum_j \beta_{ij}]$.

Experiment 1. The goal of the experiment is to study the influence of the parameter β (the weight of heuristic information) on the performance of the ant colony algorithm. Other algorithm parameters were set as follows: $A = 10, \alpha = 1,\ \rho = 0.2,\ \tau_0 = 1$. During the experiment, for each value of $\beta \in \{0.25, 0.5, 0.75, 1, 2, 3, 4\}$, the problem was solved 5 times, after which the average value of the objective function was calculated. Figure 1 shows the graph of the objective function value depending on the parameter β. As seen from the graph, increasing β initially leads to a significant improvement in the result, but after $\beta = 2$, a saturation effect occurs – the quality improvement of the solution slows down. Thus, it is advisable to use values of $\beta \approx 2 \div 3$, as further increase of this parameter does not yield noticeable enhancement.

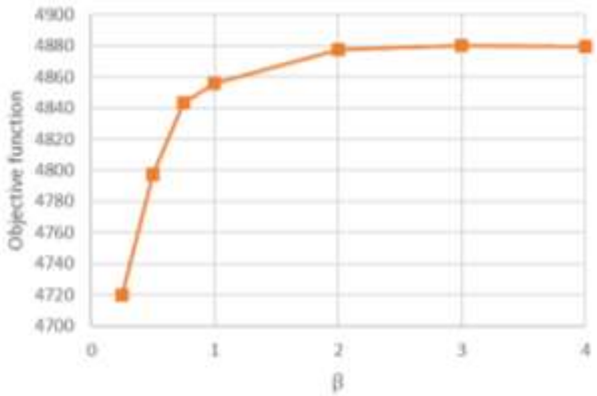

Fig.1. Dependence of the objective function value on the parameter β

Experiment 2. The goal of this experiment is to determine the impact of the termination parameter n on the convergence of both algorithms. The parameter n corresponds to the number of iterations for both algorithms. To preliminarily assess the convergence behavior, an experiment was conducted in which both algorithms were run in parallel. Figure 2 shows the dynamics of the best objective function value (record) obtained at each iteration for both methods. The results demonstrated that the ant colony algorithm consistently achieves better objective function values than the probabilistic-greedy algorithm, especially during the early iterations.

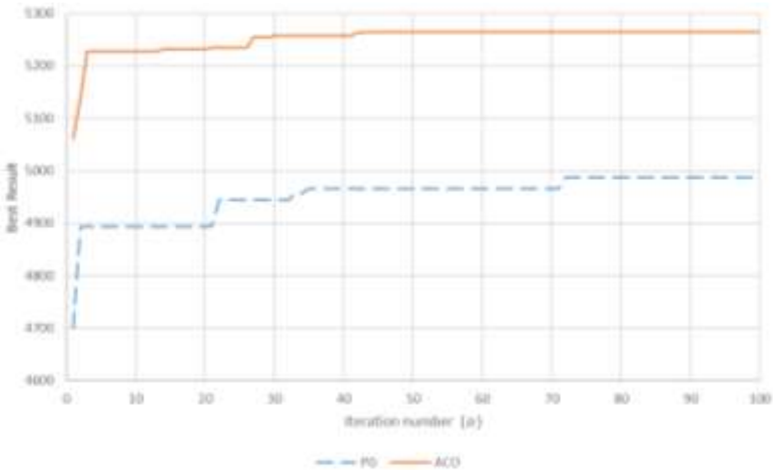

Fig. 2. Dynamics of the best objective function value (record)

Experiment 3. The goal of this experiment is to determine the recommended value of parameter N, i.e., to study how this parameter depends on the problem size. The procedure was as follows: for each value of $N \in \{10, 100, 200, 300, 500, 1000, 2000, 3000\}$, 5 problems were generated, and for each of them, the average execution time and the average value of the objective function (OF) were computed.

Figures 3 and 4 illustrate, for the probabilistic-greedy algorithm (PG) and the ant colony optimization algorithm (ACO), respectively:

- the dependency of average execution time on the parameter N;
- the dependency of the average objective function value on N.

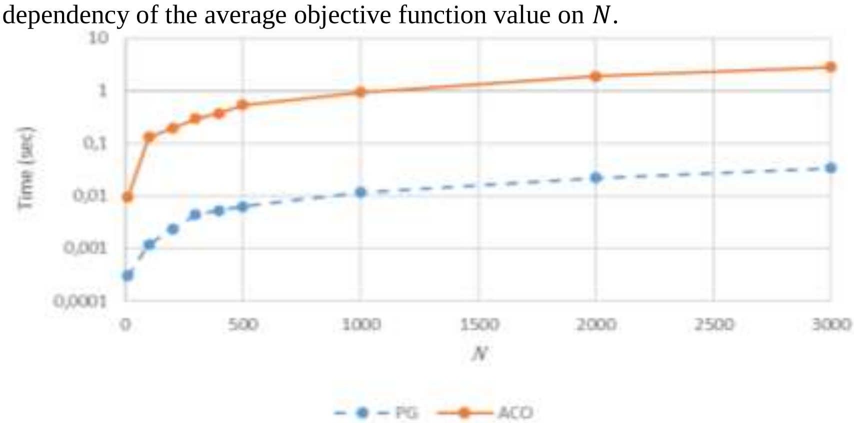

Fig. 3. Dependence of algorithm execution time on the number of iterations *N*

It should be noted that the probabilistic-greedy algorithm implements parallelization of computations, which significantly reduces the total execution time even with a large number of iterations. In contrast, in the ant colony optimization algorithm, only the actions of individual ants within a single iteration can be executed in parallel. However, the iterations themselves are performed sequentially, as pheromone trail updates are required after each iteration. This limits the degree of parallelism and leads to a significant increase in execution time as the value of N grows.

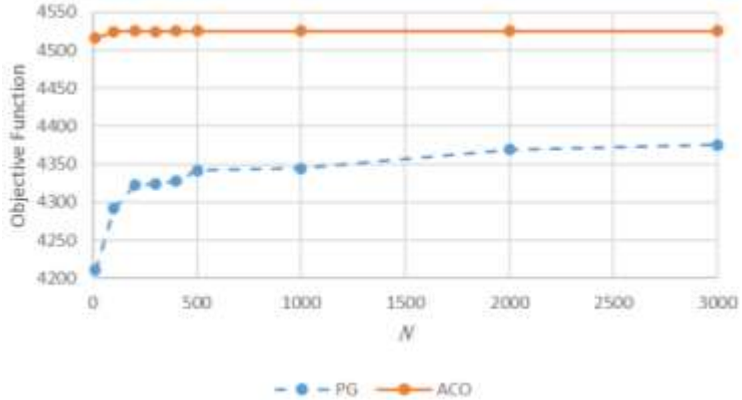

Fig 4. Dependence of the objective function value on the number of iterations N

Both algorithms demonstrate an improvement in solution quality as the number of iterations N increases; however, the rate of this improvement gradually decreases, indicating a saturation effect. According to the results of numerous experiments on problems of various sizes, the recommended value for the parameter is $N = 20m \cdot k$ for the probabilistic-greedy algorithm and $N = 0.5m \cdot k$ for the ant colony algorithm. Further increases in this parameter have little effect on solution quality but significantly increase computational costs.

Experiment 4. The goal of this experiment is to compare the execution time and accuracy of the two developed approximate algorithms. During the experiment, for each problem size $m \cdot k \in \{5 \cdot 5, 10 \cdot 10, 20 \cdot 20, 30 \cdot 30\}$, five problem instances were generated, and the objective function values obtained by each algorithm were recorded. Figure 5 shows the graph of the objective function values produced by both algorithms depending on the problem size.

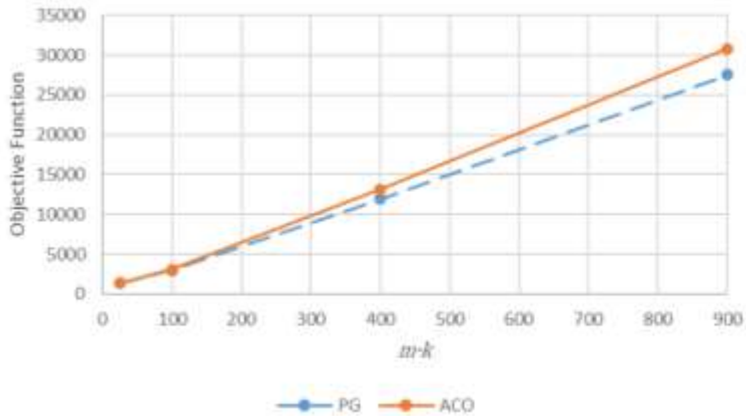

Fig 5. Graph of the dependence of the objective function value on the problem size

As the problem size increases, the ant colony algorithm demonstrates an increasingly pronounced advantage in accuracy. Its solutions consistently achieve higher objective function values, with a relative difference of up to 12% compared to the probabilistic-greedy algorithm.

Figure 6 shows the graph of algorithm runtime dependence on problem size.

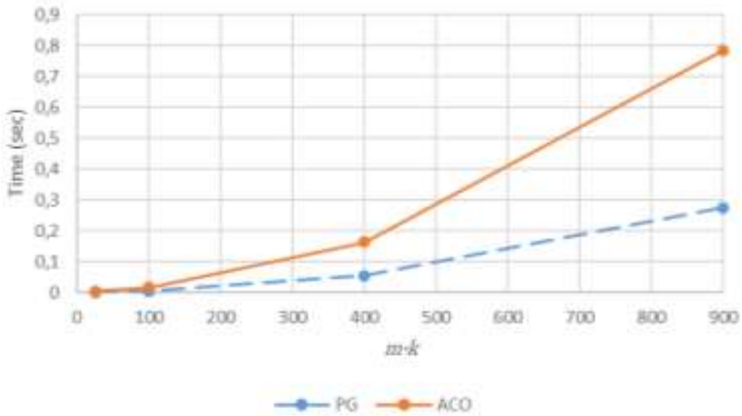

Fig 6. Graph of execution time dependence on problem size

Thus, when solving the subtask for the IT company (Stage 1), it is advisable to use the ant colony algorithm, since its execution time remains acceptable even for large-scale problems, and the accuracy of the solutions found significantly surpasses the results of the other algorithm.

Similar results were obtained for the subtask for providers (Stage 2).

6.2 Verification of the Developed Combined Method's Effectiveness

Experimental validation of the developed combined method was carried out within the framework of the service portfolio formation system. For comparison of solution quality, a guided genetic algorithm [18] was used, while the accuracy was evaluated by the exhaustive search method. A series of experiments were conducted using both the proposed method and the compared approaches. Due to space limitations, this article presents results from only a part of these studies.

Table 1 shows the results of the comparative analysis of the efficiency of the combined method, the guided genetic algorithm, and the exhaustive search method on small-scale problems. The metric used was the average deviation of the objective function value – the sum of revenues of the IT company and the providers – from the optimal value found by the exhaustive search method.

Table 1

Average deviation of the objective function from the optimal value (in %)

Problem size $m \cdot k$	Combined method	Guided Genetic Algorithm (GGA)
$3 \cdot 3$	2,1	2,3
$4 \cdot 4$	3,0	3,7
$5 \cdot 5$	3,8	4,2
$6 \cdot 6$	4,2	5,8
$7 \cdot 7$	5,4	6,9
$8 \cdot 8$	6,2	8,3

Exhaustive search method has exponential complexity, which makes its application to larger-sized problems extremely resource-intensive.

The experimental results confirm the feasibility of the combined method within the problem formulation presented in Section 3.

First, it has been demonstrated that the proposed approach ensures sufficiently high solution quality, approaching optimal values, as evidenced by the small average deviation from the benchmark results obtained via exhaustive search method.

Second, the combined method exhibits robustness to increasing problem size – the quality of solutions declines gradually, indicating its suitability for medium- and large-scale problems. In this respect, the combined method performs on par with the guided genetic algorithm, maintaining competitive accuracy and stability of results.

Overall, the experimental study confirms the practical viability of the method developed in this paper for solving the service portfolio formation problem. The results obtained highlight the potential of its integration into a service portfolio formation information system for future implementation in the operations of infocommunication service providers.

7. Creation of a Service Portfolio Formation System

The development of the service portfolio formation system was carried out in accordance with the software system life cycle. An analysis of the activities of IT companies and providers, the application of requirements engineering techniques, and the assessment of constraints and risks enabled the construction of the system architecture. Based on the selected appropriate tools, the system was implemented. This made it possible to conduct experimental research on the system and the algorithms embedded in it for solving the service portfolio formation problem.

A microservice architecture was used for building the system. This division into microservices ensures, on the one hand, the required system behavior through microservices interaction, and on the other hand, allows each component to be developed independently of the others.

To comprehensively address the architecture from all key perspectives, traditional architectural patterns were applied – namely, multi-tier and component-based models. In addition, the deployment-oriented view was supported by the use of a container diagram and system architecture targeting deployment in the Microsoft Azure cloud environment, which was chosen as the resource access platform.

The component-connector architectural pattern of the service portfolio formation system, shown in Figure 7, provides a general view that reflects:

- the interaction between users and the system, as well as between the system and external software;
- the structural functional elements and the way they interact;
- the interconnections between system components.

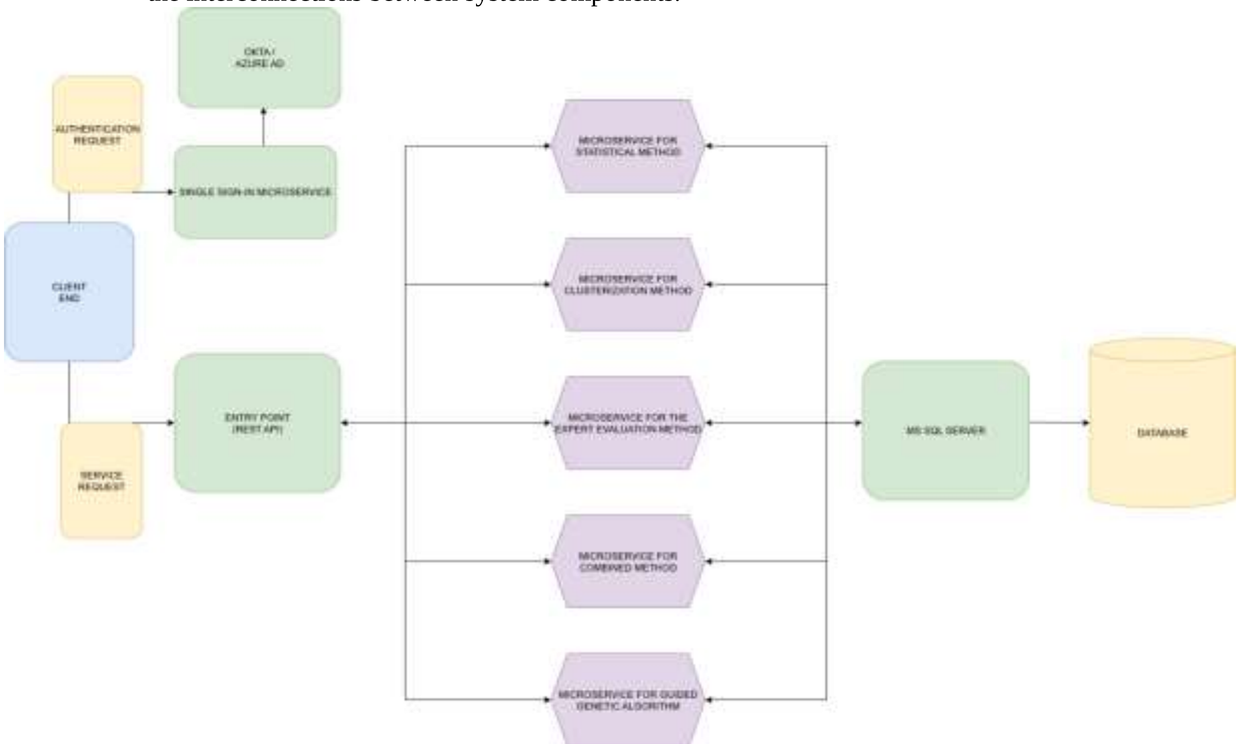

Fig.7. Component-connector pattern of the microservice architecture for the service portfolio formation system

This model presents a technology-agnostic view of the system architecture, illustrating its components and their relationships without binding to specific technologies, protocols, or system-level software.

Before moving on to the visualization of the microservice architecture, it is important to highlight some of its challenges.

First, the complexity of management, configuration, and maintenance increases significantly in a microservice-based system. Since each system function is implemented as a separate microservice, their interaction requires reliable communication between components. This introduces risks and challenges related to network reliability, including connection interruptions, packet loss, or response timeouts. Moreover, the deployment process becomes more complex, as compatibility between versions must be ensured.

Second, data consistency becomes a key issue, as microservices often access different data sources and may hold different states simultaneously. In some cases, maintaining consistency turns into a complex challenge that requires the use of transactional systems to ensure reliable coordination.

Third, in a microservice environment, a single entry point is required to route client requests to the appropriate microservices that implement the desired functionality. Typically, this is handled by an API Gateway – a design pattern that serves as a central interface receiving all client requests and routing them to the relevant microservices. The API Gateway can be compared to a reverse proxy or router that consolidates client communication and returns responses in a way that makes the system appear unified.

In the context of the developed system, the API Gateway is a crucial component with the following responsibilities:

- acts as the external entry point to the system, positioned at the boundary of a specially created private network;

- handles user authentication, resource access control, response caching, rate limiting, event logging, and header injection;
- manages load balancing. Since the API Gateway represents a single point of entry and therefore a single point of failure (SPOF), redundancy mechanisms must be implemented to ensure the system's reliability and operational efficiency [20].

Client or Frontend is the client side of the system that displays the user interface and interacts with the server part of the system by sending HTTP requests. The client is responsible for interacting with the end user and initiating such processes as forming service portfolios for given IT companies and their providers or for given providers and their clients based on the implemented methods in the system (statistical, clustering, combined, guided genetic), viewing service portfolios formed for IT companies and their providers or for providers and their clients, evaluating and approving portfolios, selecting the most effective methods by various parameters, editing decisions and data of decision-makers, and viewing statistics and trends regarding decision-makers and their decisions. The client communicates with the system by sending HTTP requests. In the current version of the system, the client is a web browser. The React framework is used to build the frontend, providing structure and components for creating user interfaces.

For authorization, authentication, new user registration, login for registered users, session control, and access management, a single sign-on (SSO) microservice is used. This microservice, in turn, utilizes the capabilities of external services such as OKTA or MS AzureAD. Together, they serve as the security server for the entire system, based on the SSO concept. The system's resources are protected. To initiate any of the above-described processes, the user must first go through authentication and receive an access token. This token is then used to authenticate client requests to the system's server side.

The central entry point to the system is the API gateway, which receives requests from the client and forwards them to the corresponding microservices. The API gateway, together with the SSO microservice, is responsible for user authentication based on the access token. All endpoints require authentication; the gateway also verifies the user's role, ensuring a high level of security.

Once access is granted, interaction and traffic are maintained within a Kubernetes cluster in the internal network.

The system's functional architectural elements are organized into separate microservices, each implementing a specific set of functional requirements. These functional microservices include:

1) a statistical method microservice that allows users to form a service portfolio for a selected provider in the IT company's information system or for a selected client in the provider's system based on accumulated statistical data;
2) a clustering method microservice that enables users to form service portfolio for a selected provider or client using clustering techniques;
3) a combined method microservice that forms service portfolio based on the combined method for a given provider or client;
4) a guided genetic algorithm microservice that supports service portfolio formation for selected providers or clients using a guided genetic algorithm.
5) an expert evaluation microservice, which allows users to evaluate the performance of the statistical method, clustering method, combined method, and guided genetic algorithm, compare their metrics, and draw conclusions about which method is most suitable, with the final decision left to the human decision-maker.

Now let’s describe the other components of the architecture. The MS SQL Server component is a database management system. It was selected based on the requirements of the service portfolio formation system related to data storage, retrieval, and modification.

The Database component is the actual database of the service portfolio formation system.

An important component of the service portfolio formation system is the content from third-party resources. This includes third-party components used in the functioning of the system, such as OKTA or MS Azure AD.

To deploy the service portfolio formation system in a cloud environment and provide it with the necessary resources, a container diagram was developed.

For the service portfolio formation system, an architecture was developed in accordance with the functional and non-functional requirements, intended for deployment in the Microsoft Azure cloud environment.

Conclusions

The article discusses the mathematical model and solution method for the problem of service portfolio formation, presents the system architecture, and justifies key design decisions related to its implementation in the context of creating an information system for service portfolio formation.

The proposed model and method for forming service portfolios aim to enhance the efficiency of IT companies and providers, taking into account the specifics of modern business practices in the field of infocommunications.

The proposed version of the combined method for solving the multi-criteria problem of service portfolio formation integrates the decomposition into single-criterion subproblems for IT companies and providers, solving these subproblems using metaheuristic methods and specialized heuristic procedures, and constructing a compromise solution that aligns the interests of both parties.

The developed metaheuristic algorithms for solving the single-criterion subproblems allow for obtaining near-optimal service portfolios within an acceptable computational time. The compromise solution algorithm implements the concept of mutual benefit, enabling the formation of high-quality balanced solutions acceptable to both IT companies and providers.

The results of the conducted experimental research confirmed the feasibility and effectiveness of the developed combined method for solving the service portfolio formation problem. This method, along with other validated approaches, formed the methodological foundation of the developed information system for service portfolio formation.

The article also describes the architecture of the information system for service portfolio formation and substantiates the key technical and algorithmic decisions related to its implementation.

References

1. Field, J. M., Fotheringham, D., Subramony, M., Gustafsson, A., Ostrom, A. L., Lemon, K. N., & McColl-Kennedy, J. R. (2021). Service research priorities: designing sustainable service ecosystems // Journal of Service Research, 24(4), 462-479.
2. Favoretto, C., Mendes, G. H., Oliveira, M. G., Cauchick-Miguel, P. A., & Coreynen, W. (2022). From servitization to digital servitization: How digitalization transforms companies' transition towards services // Industrial Marketing Management, 102, 104-121.
3. Serrano, J., Faustino, J., Adriano, D., Pereira, R., & da Silva, M. M. (2021). An IT service management literature review: challenges, benefits, opportunities and implementation practices // Information, 12(3), 111.
4. Richter, H., & Lantow, B. (2021, June). IT-service value modeling: a systematic literature analysis. In International Conference on Business Information Systems (pp. 267-278). Cham: Springer International Publishing.
5. TM forum. https://dtw.tmforum.org/?utm_medium=Referral&utm_ source=tmforum.org&utm_campaign=DTW-2025&utm_term=TM-Forum-Homepage-Sliding-Banner. Accessed 17.05.2025.
6. Tuunanen, T., Lumivalo, J., Vartiainen, T., Zhang, Y., & Myers, M. D. (2023). Micro-Level Mechanisms to Support Value Co-Creation for Design of Digital Services. Journal of Service Research, 10946705231173116.
7. Maddern, H., Smart, P. A., Maull, R. S., & Childe, S. (2014). End-to-end process management: implications for theory and practice. Production Planning & Control, 25(16), 1303-1321.
8. В. Чимшир, С. Теленик, О. Ролік, Е. Жаріков. Платформа підтримки життєвого циклу сервісів в інформаційних системах провайдерів інформаційно-комунікаційних послуг // Адаптивні системи автоматичного управління. - Том 1.- № 42 DOI: https://doi.org/10.20535/1560-8956.42.2023.279172.
9. Serrano, J., Faustino, J., Adriano, D., Pereira, R., & da Silva, M. M. (2021). An IT service management literature review: challenges, benefits, opportunities and implementation practices // Information, 12(3), 111.
10. Mora, M., Marx-Gomez, J., Wang, F., Diaz, O. (2021). Agile IT Service Management Frameworks and Standards: A Review. In: Arabnia, H.R., Deligiannidis, L., Tinetti, F.G., Tran, QN. (eds) Advances in Software Engineering, Education, and e-Learning. Transactions on Computational Science and Computational Intelligence. Springer, Cham.
11. Agutter, C. (2020). ITIL Foundation Essentials ITIL 4 Edition-The ultimate revision guide. IT Governance Publishing Ltd.
12. Gavrilenko, O., Zhurakovska, O., Kohan, A., Matviichuk, R., Piskun, A., Khavikova, Y. and Khalus, O. (2022), "The principle for forming a portfolio of public services based on the analysis of statistical information", Eastern-European Journal of Enterprise Technologies, No 3 (3 (117)), pp. 57–64, doi: https://doi.org/10.15587/1729-4061.2022.260136.
13. Gavrilenko O., Zhurakovska O., Khomenko O.. ANALYSIS OF THE FEASIBILITY OF IMPLEMENTING SERVICE PORTFOLIOS. The XXVIII International Scientific and Practical Conference «Unusual methods of development of science and thoughts», July 17 – 19, Madrid, Spain. 182 p. Available at: https://eu-conf.com/ua/events/unusual-methods-ofdevelopment-of-science-and-thoughts/.
14. О. Гавриленко, В. Чимшир, Е. Жаріков, С. Теленик, О.Амонс. Метод формування пакетів сервісів за допомогою алгоритмів кластеризації / Адаптивні системи автоматичного управління. - Том 1.- № 44.- 2024. С:182-191. DOI: https://doi.org/10.20535/1560-8956.44.2024.302437.
15. О. Гавриленко, В. Чимшир, Е. Жаріков, С. Теленик, Р. Омельченко. Вирішення задачі впровадження пакетів сервісів за допомогою статистичної інформації / Адаптивні системи автоматичного управління. - Том 2.- № 43 (2023). DOI: https://doi.org/10.20535/1560-8956.43.2023.292257
16. A. Başar, "IT Service Level Optimization: A Case Study," in Proc. Int. Conf. on Industrial Engineering and Operations Management, Istanbul, Turkey, Mar. 2022, pp. 421–427. https://www.researchgate.net/publication/ 362061884_IT_Service_ Level _Optimization_A_Case_Study.
17. Peppard, J. (2003). Managing IT as a Portfolio of Services. European Management Journal, 21(4), 467-483.
18. V. Chymshyr, O. Zhdanova, O. Havrylenko, G. Nowakowski, and S. Telenyk, "Models and methods for forming service packages for solving of the problem of designing services in information systems of providers", Inf. Comput. and Intell. syst. j., no. 5, pp. 29–54, Dec. 2024. DOI: https://doi.org/10.20535/2786-8729.5.2024.316432.
19. S. Michael, B. Michael and S. Thomas, "IT Service Management Frameworks Compared- Simplifying Service Portfolio Management," 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, Apr. 2019, pp. 421-427.
20. Gough J., Bryant D., Auburn M., Mastering API Architecture: Design, Operate, and Evolve API-Based Systems, O'Reilly Media, 2022.

Olena Zhdanova **Олена Жданова**	PhD, Associate professor, Associate professor of the Department of Information Systems and Technologies, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, https://orcid.org/0000-0002-8787-846X e-mail: zhdanova.elena@hotmail.com	кандидат технічних наук, доцент, доцент кафедри інформаційних систем та технологій, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

Maksym Bukasov **Максим Букасов**	PhD, Associate professor of the Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine, https://orcid.org/0000-0003-1482-3107 e-mail: bukasov@gmail.com	кандидат технічних наук, доцент кафедри інформаційних систем та технологій, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна
Svyatoslav Tsymbal **Святослав Цимбал**	Assistant of the Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine, https://orcid.org/0000-0002-3073-8028 e-mail: tsymbal.svyatoslav@lll.kpi.ua	асистент кафедри інформаційних систем та технологій, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна
Viacheslav Chymshyr **Вячеслав Чимшир**	PhD Student of the Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine, https://orcid.org/0009-0005-2555-5373 e-mail: vchimshir@gmail.com	аспірант кафедри інформаційних систем та технологій, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна
Rostyslav Omelchenko **Ростислав Омельченко**	Master’s graduate of the Department of Information Systems and Technologies, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine, e-mail: omelchenkorv@gmail.com	магістр (випускник) кафедри інформаційних систем та технологій, Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна