

УДК: 004

## Ситуационное распределение ресурсов: обзор технологий решения задач на основе систем знаний

В. Д. Ильин

Федеральный исследовательский центр «Информатика и управление» Российской академии наук,  
Россия, 119333, г. Москва, ул. Вавилова, д. 44, корп. 2

E-mail: vdilyin@yandex.ru

*Получено 15.04.2025, после доработки — 12.06.2025.  
Принято к публикации 23.07.2025.*

В обзоре представлены обновленные технологии решения двух классов линейных задач распределения ресурсов при динамично изменяющихся характеристиках систем ситуационного управления и информированности экспертов (и/или обучаемых роботов), решающих задачи. Поиск решений выполняется в интерактивном режиме вычислительного эксперимента с использованием обновляемых систем знаний о задачах, рассматриваемых как конструктивные объекты (в соответствии с методологией формализации знаний о программируемых задачах, созданной в теории S-символов). Технологии ориентированы на реализацию в виде интернет-сервисов. К первому классу отнесены задачи распределения ресурсов, решаемые методом целевого перемещения решения. Ко второму — задачи распределения одного ресурса в иерархических системах с учетом приоритетов расходных статей, решаемые (в зависимости от заданных обязательных и ориентирующих требований к решению) или методом интервального распределения (при этом входные данные и результат представлены числовыми сегментами), или методом целевого перемещения решения. Постановки задач определяются требованиями к решениям и спецификацией их применимости, которые задает эксперт на основе результатов анализа портретов целевой и достигнутой ситуации. В отличие от известных методов решения задач распределения ресурсов как задач линейного программирования метод целевого перемещения решения нечувствителен к малым изменениям данных и позволяет находить наилучшие приближения к реализуемым решениям при несовместности системы ограничений. В технологиях распределения одного ресурса сегментное представление данных и результатов позволяет более адекватно (по сравнению с точечным представлением) отражать состояние ресурсного пространства системы и повышает практическую применимость решений. Обсуждаемые в статье технологии программно реализованы и применялись для решения задач ресурсного обоснования решений, бюджетного проектирования с учетом приоритетов расходных статей и др. Технология распределения одного ресурса реализована в виде действующего интернет-сервиса планирования расходов. Методологическая состоятельность технологий подтверждена результатами сравнения с известными технологиями решения рассматриваемых задач.

Ключевые слова: линейные задачи распределения ресурсов, технологии решения задач ситуационного распределения ресурсов, пространство ресурсного состояния системы, портреты ситуаций, обязательные и ориентирующие требования к решению, метод целевого перемещения решения, метод интервального распределения, теория S-символов

UDC: 004

## Situational resource allocation: review of technologies for solving problems based on knowledge systems

V. D. Ilyin

Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences,  
44/2 Vavilov st., Moscow, 119333, Russia

E-mail: vdilyin@yandex.ru

*Received 15.04.2025, after completion — 12.06.2025.*

*Accepted for publication 23.07.2025.*

The article presents updated technologies for solving two classes of linear resource allocation problems with dynamically changing characteristics of situational management systems and awareness of experts (and/or trained robots). The search for solutions is carried out in an interactive mode of computational experiment using updatable knowledge systems about problems considered as constructive objects (in accordance with the methodology of formalization of knowledge about programmable problems created in the theory of S-symbols). The technologies are focused on implementation in the form of Internet services. The first class includes resource allocation problems solved by the method of targeted solution movement. The second is the problems of allocating a single resource in hierarchical systems, taking into account the priorities of expense items, which can be solved (depending on the specified mandatory and orienting requirements for the solution) either by the interval method of allocation (with input data and result represented by numerical segments), or by the targeted solution movement method. The problem statements are determined by requirements for solutions and specifications of their applicability, which are set by an expert based on the results of the portraits of the target and achieved situations analysis. Unlike well-known methods for solving resource allocation problems as linear programming problems, the method of targeted solution movement is insensitive to small data changes and allows to find feasible solutions when the constraint system is incompatible. In single-resource allocation technologies, the segmented representation of data and results allows a more adequate (compared to a point representation) reflection of the state of system resource space and increases the practical applicability of solutions. The technologies discussed in the article are programmatically implemented and used to solve the problems of resource basement for decisions, budget design taking into account the priorities of expense items, etc. The technology of allocating a single resource is implemented in the form of an existing online cost planning service. The methodological consistency of the technologies is confirmed by the results of comparison with known technologies for solving the problems under consideration.

**Keywords:** linear resource allocation problems, technologies for solving situational resource allocation problems, states of system’s resource space, profiles of situations, mandatory and orienting requirements for solutions, method of targeted solution movement, interval method of allocation, theory of S-symbols

Citation: *Computer Research and Modeling*, 2025, vol. 17, no. 4, pp. 543–566 (Russian).

## 1. Введение

Проблема методологического обеспечения технологий решения задач *ситуационного управления* [Клыков, 1974; Поспелов, 1986] относится к числу постоянно актуальных.

Методы моделирования *систем ситуационного управления (су-систем)* и решения *в су-системах задач распределения ресурсов (срр-задач)*, реализация методов, способы получения и обработки данных и реализации решений, методы построения обновляемых систем знаний о задачах — все это составляющие *технологий решения срр-задач (срр-технологий)*, рассматриваемых в данной статье.

Живучесть и качество функционирования живых и неживых *су-систем* во многом зависят от уровня совершенства применяемых *срр-технологий*.

В организмах животных ресурсы, полученные из окружающей среды и пищи (вода, белки, жиры и др.), используются как ресурсы для построения, питания и функционирования клеток; клетки — как ресурсы для построения клеточных структур, тканей, органов и так далее — до систем организма (нервной, пищеварительной и др.) и организма в целом. На уровне организма решением *срр-задач* управляет нервная система, используя генетически заданные методы, которые могут быть усовершенствованы или повреждены (нездоровым образом жизни и болезнями) в процессе функционирования организма.

В *су-системах*, создаваемых людьми, добываемые ресурсы (руды, минералы и др.), оборудование для добычи, электроэнергия, трудозатраты и др. используются как ресурсы для производства материалов (металлов, полупроводников и др.); материалы входят в состав ресурсов для производства комплектующих машин, компьютерной техники и др. и так далее — до производственных предприятий, организаций и др. составляющих экономических систем стран. От структурной сложности и изменчивости *су-систем* и сред их функционирования зависят требования к *срр-технологиям*.

В наши дни (апрель 2025 г.) уровни совершенства *срр-технологий* во многом зависят от использования современных *S-машинных средств* и передовых технологий *символьно-кодОВО-сигнальной среды (S-среды)* [Шуйн, 2022; Ильин, 2023]. К ним относятся *DT-технологии (Digital Twins Technologies)* [Tao et al., 2024], *M2M-технологии (Machine-to-Machine Technologies)* [Prasad, Rohokale, 2020], *IoT-технологии (Internet of Things Technologies)* [Suryawanshi et al., 2022], *IoS-технологии (Internet of Services Technologies)* [Cardoso et al., 2009], технологии *Fog Computing* [Ahammad, 2023], *5G-технологии* [Pavan et al., 2023] и ряд других.

### 1.1. О методологическом обеспечении технологий решения *рр-задач*

В обзоре [Katoh et al., 2025] результатов, относящихся к методологическому обеспечению, особое внимание уделено *рр-задачам* с целочисленными переменными. В [Patni et al., 2025] сделан обзор ряда стратегий распределения ресурсов, реализованных в методах, ориентированных на технологии облачных вычислений. Основное внимание уделено эффективности, масштабируемости и адаптируемости методов к требованиям пользователей. Рассмотрены различные тактики распределения (оптимизационная, аукционная и др.). При сопоставлении методов статического и динамического распределения отмечено, что динамическое распределение позволяет более гарантированно обеспечить эффективность и соответствие требованиям пользователей. В [Tang, Wu, 2024] рассматривается решение *рр-задач* в производственных системах с применением *DT-технологий* (см. подпараграф 2.2). В статье [Prashanth et al., 2025] предложен подход к решению влияющих на доверие пользователей проблем (затрат, задержек передачи данных и др.) при реализации облачных технологий решения *рр-задач*. Цель обзора [Nemati, Mansouri, 2025] — дать представление о современных методах решения *рр-задач* на основе технологий

туманных вычислений<sup>1</sup> и направлениях развития таких технологий. В работе [Aditi et al., 2025] рассматривается применение технологий *глубокого обучения (DL)* (на основе *сверточных нейронных сетей (CNN)*) для решения *pp-задач* и проблем «узких мест» в облачных ресурсах.

### 1.2. *S-формализованная запись и семантическая разметка текста*

Для записи формул (не требующей применения редакторов формул) и *семантической разметки текста* (выделения определений, замечаний и примеров) используются средства *TSM-комплекса (TSM: textual S-symbolic modeling)*. Эти средства применяются в работах, связанных с *теорией S-символов* [Ильин, 2023] и *S-символьным моделированием (S-моделированием)* программируемых задач [Ильин, 1989; Ilyin, 2022].

В данной статье применены следующие *TSM-средства семантической разметки текста*:

□ ⟨фрагмент описания⟩ □ ≈ определение (здесь и далее символ ≈ заменяет слово «означает»);

◇ ⟨фрагмент описания⟩ ◇ ≈ замечание;

△ ⟨фрагмент описания⟩ △ ≈ пример;

⟨фрагмент описания⟩ /*текстовый S-символ*/ ≈ обозначение фрагмента описания;

пара символов =: заменяет группу слов «определено как».

*Курсивом* (и **полужирным**) выделены первые вхождения названий понятий и фрагменты описания, к которым автор хочет привлечь внимание.

◇ *S-формализованная запись* и *семантическая разметка текста* способствуют правильности его интерпретации (△ позволяют однозначно определить начала и окончания *определений, замечаний* и *примеров* △). Рассчитаны на *автоматизированную обработку* текста как *S-сообщения* [Ильин, 2023; Ilyin, 2022]. ◇

## 2. Обоснование методологического подхода

Основным препятствием для решения практически значимых *линейных задач распределения ресурсов*<sup>2</sup> как *задач линейного программирования (ЛП-задач)* [Dantzig, 1963; Schrijver, 1998] является частая *несовместность системы ограничений*. *ЛП-задачи* обычно решаются *симплекс-методом* [Юдин, Гольштейн, 2022; Vanderbei, 2020], реже — *методом внутренних точек* [Дикин, 1998]. Но и в редких случаях совместности системы ограничений трудно получить реализуемые решения (из-за *чувствительности ЛП-алгоритмов к малым изменениям данных* [Tikhonov, Arsenin, 1977]).

При несовместности системы ограничений может быть решена задача *поиска чебышёвской точки* [Коллатц, Крабе, 1978]. Однако решения, найденные как *чебышёвская точка*, практически неприменимы, так как нарушают *обязательные ограничения*.

Но и при совместности системы ограничений практическая применимость методов решения *ЛП-задач* существенно ограничена *предположением о совпадении прогнозируемых значений входных данных, используемых для поиска решения, с фактическими при реализации*. Известно, что *ЛП-задачи* относятся к *математически некорректным* (из-за неустойчивости решения к малым изменениям входных данных) [Tikhonov, Arsenin, 1977]. В практически значимых задачах такие изменения обычно существуют, так как расчеты выполняются с некоторым упреждением

<sup>1</sup> *Туманные вычисления (Fog Computing)* — модель децентрализованных вычислений, в соответствии с которой вычислительные ресурсы, хранилища данных и сетевые сервисы размещаются возможно ближе к конечным устройствам и источникам данных. В *Fog Computing* множество связанных между собой узлов используют локальные вычислительные ресурсы.

<sup>2</sup> Постановка и решение задачи (как *задачи планирования производства*) были впервые опубликованы в [Канторович, 1939].

и при реализации решений значения переменных состояния системы неизбежно отличаются от прогнозируемых значений, используемых при поиске решений.

Кроме того, практика решения задач распределения ресурсов показывает, что с момента получения решения до момента начала его реализации *может измениться и множество основных переменных, и система ограничений*.

Для создания *технологии решения практически значимых линейных задач распределения ресурсов* был предложен подход, основанный на идее их неформальной постановки [Ильин, 1996]. В такой постановке *задача решается в интерактивном режиме вычислительного эксперимента*, на каждом шаге которого система ограничений формируется экспертом (или обучаемым роботом) из *обязательных и ориентирующих требований к решению*, в которых учтены и реально достижимая точность входных данных, и реализуемость решения, и его эффективность. Идея реализована А. В. Ильиным в *методе целевого перемещения решения* [Ильин, 1999; Пуин, Пуин, 2013]. Применение этого метода позволяет найти *наилучшее приближение к реализуемому решению и при несовместности системы ограничений*.

## 2.1. Создание и развитие методологического подхода

Подход формировался при выполнении конкретных научно-исследовательских проектов: *автоматизации управления энергосистемами (1970-е годы), автоматизации программирования (с 1980-х гг. по наст. время [Ильин, 1989; Пуин, 1995; Пуин, Пуин, 2021]), информатизации системы государственного управления (1990-е гг. [Ильин, 1996]) и экономической системы (с 2000-х гг. по наст. время [Пуин, Пуин, 2013; Пуин, Пуин, 2016])*. Созданные методы были программно реализованы при разработке технологий и прошли проверку применением<sup>1</sup>.

◇ В 2024 г. начаты *исследования биосистем, изучаемых как су-системы*. Есть основания полагать, что результаты этих исследований помогут усовершенствовать методологический арсенал *срр-технологий*. ◇

В данной статье рассматриваются обновления *системы знаний об S-задачных объектах* (в *S-модель системы знаний* добавлено множество *Pres ≈ P-объектные ресурсы* =: каталогизированные спецификации *S-задачных объектов* (элементов библиотек алгоритмов, программ и др.) и *технологии формирования и анализа портретов ситуаций с использованием цифровых двойников*. Приведены уточненные описания *срр-технологий*, постановок и методов решения *срр-задач*.

## 2.2. Использование цифровых двойников (Digital Twins)

□ *Цифровой двойник /DT/* =: обновляющаяся *S-символьная модель (S-модель)* [Пуин, 2022] некоторого объекта (Δ деятельности, технологии или др. Δ), предназначенная для анализа его состояния и управления поведением. □

Для построения *DT* используются значения параметров, характеризующих поведение *S-моделируемого объекта* (Δ для объектов с измеряемыми переменными состояниями и управления — данные, поступающие от соответствующих датчиков в форме *S-сигналов* [Пуин, 2022] Δ).

*DT* (как *S-образ объекта*) накапливает знания о своем *S-прообразе*, обмениваясь с ним *S-сообщениями* и обновляясь [Ильин, 2023].

Различают *DT-прототипы /DTP/*, *DT-экземпляры /DTI/* и *агрегированные DT /DTA/*. *DTP* используется при создании физической версии модели объекта. *DTI* =: *S-модель некоторого объекта, с которым DTI предназначены взаимодействовать*; *DTA* =: *комплекс DT-средств обработки данных, предназначенный для взаимодействия с заданной совокупностью DTI*.

<sup>1</sup> Список работ, в которых опубликованы результаты, полученные при выполнении указанных проектов, представлен на ORCID-странице автора: <https://orcid.org/0000-0002-9761-082X> (дата обращения: 14.04.2025).

◇ В *spp-технологиях*, представленных в данном обзоре, используются *DTI* и *DTA*. Данные для их построения и обновления поступают от информационных источников, реализующих мониторинг данных *spp-задач*. ◇

### 3. Портреты ситуаций в *spp-технологии*: формирование и анализ

□ *Целевой* называем ситуацию, в которую требуется перевести *су-систему* /*ссу*/; *отправной* — ситуацию, в которой ищется управляющее воздействие; *достигнутой* — полученную в результате управляющего воздействия.

*Портрет ссу-ситуации* — спецификация состояния *ссу*, включающая описание пространств *ссу-состояний*, располагаемые варианты управляющих воздействий и ресурсной *ссу-обеспеченности* [Ильин, 1996]. □

Процессы формирования и первичного анализа портретов *ссу-ситуаций* реализуются (с использованием *DTI* и *DTA*) в соответствии с □ *системой правил*  $=: \{R, S\}$ , где  $R$  — множество правил,  $S \subseteq R \times R$  — бинарное отношение на  $R$ . *Правило* представлено в виде  $D_1 \Rightarrow D_2$  или  $D \Rightarrow I$  (где  $D, D_1, D_2$  — специфицированные описания, а  $I$  — инструкция). □

Порядок срабатывания правил определен с помощью отношения *специализации*  $S$ . Если для любой пары правил  $r_1, r_2$  указано, что  $r_2$  — *специализация*  $r_1$ , то  $r_1$  является *обобщением*  $r_2$ .

*Правила-специализации* предназначены для срабатывания в более частных *ссу-ситуациях*, чем те, в которых предусмотрено срабатывание соответствующих *правил-обобщений*. *Правило-специализация* имеет приоритет в срабатывании перед соответствующим *правилом-обобщением*.

*Механизм интерпретации* запросов на  $R$  определяет состав и порядок срабатывания правил. Запросы представлены спецификациями *портретов ссу-ситуаций*.

△ В результате интерпретации запроса выдана пара правил, одно из которых определяет необходимые действия при нарушении графика поставок ресурса, а второе — при критическом нарушении поставок ресурса (которое может сорвать выполнение заказов). Второе правило служит специализацией первого. △

Для формирования обновляемых *портретов ссу-ситуаций* (включающих данные об источниках и потребителях распределяемых ресурсов, удельных расходах ресурсов и др.) используются цифровые двойники типа *DTI*, получающие данные о состоянии своих прообразов.

Анализ *портретов ссу-ситуаций* реализуется экспертами с использованием результатов первичного анализа, выполненного цифровыми двойниками типа *DTA*, связанными с двойниками типа *DTI*, предназначенными для формирования портретов *ссу-ситуаций*.

### 4. Задача ситуационного распределения ресурсов /*spp-задача*/

□ В постановке *spp-задачи* будем использовать следующие понятия:

- *удельный расход  $i$ -го ресурса для  $j$ -го средства использования ресурса* —  $a_{ij}$  ( $i = 1, \dots, m$ ;  $j = 1, \dots, n$ ),  $a_{ij} \neq 0$ ;
- *имеющееся количество  $i$ -го ресурса* —  $b_i$  ( $i = 1, \dots, m$ ),  $b_i \geq 0$ ;
- *решение (искомая величина активности  $j$ -го средства)* —  $x_j$  ( $j = 1, \dots, n$ ),  $x_j \geq 0$ ;
- *ресурсная функция (суммарный расход  $i$ -го ресурса)* —  $a_{i1}x_1 + \dots + a_{in}x_n$ ;
- *составное требование к искомому решению* — набор  $\{a_{i1}x_1 + \dots + a_{in}x_n \leq b_i\}$  ( $i = 1, \dots, m$ );

- набор *оценочных функций* (показателей эффективности) решения —  $\{c_{i1}x_1 + \dots + c_{in}x_n\}$  ( $i = 1, \dots, k$ ), где  $c_{ij}$  — величина  $i$ -го показателя эффективности, приходящаяся на единичное использование ресурса  $j$ -м средством ( $c_{ij} \neq 0$ );
- могут быть заданы *приоритеты требований* к решению —  $p_i$  ( $0 < p_i \leq \infty, i \in \{1, \dots, m+k\}$ );
- для каждой линейной формы  $(a_{i1}x_1 + \dots + a_{in}x_n, c_{i1}x_1 + \dots + c_{in}x_n)$  может быть задано двустороннее ограничение (конъюнкция двух *простых требований*), тогда система требований будет иметь следующий вид:

$$\{[b_i \leq] a_{i1}x_1 + \dots + a_{in}x_n [ \leq B_i] [p_i]; x_j \geq 0\} \quad (i = 1, \dots, m+k, j = 1, \dots, n)$$

(здесь все коэффициенты ресурсных и оценочных функций обозначены через  $a_{ij}$ ; квадратные скобки обозначают необязательность задания ресурсного ограничения, показателя эффективности, приоритета);

- *простое требование* к искомому решению представлено одной из трех форм:  $F_i(\mathbf{x}) = c_i[p_i]$ ,  $F_i(\mathbf{x}) \leq c_i[p_i]$ ,  $F_i(\mathbf{x}) \geq c_i[p_i]$ , где  $F_i(\mathbf{x})$  —  $i$ -я функция (ресурсная или оценочная),  $\mathbf{x} = (x_1, \dots, x_n)$  — вектор решения (векторы и матрицы выделены полужирным шрифтом),  $c_i$  — константа,  $p_i$  — *приоритет* требования ( $0 < p_i \leq \infty$ );
- *составное требование* — логическая комбинация простых требований.

*Обязательные требования* не должны быть нарушены (имеют *абсолютный приоритет*  $p_i = \infty$ ).  $\triangle$  *Обязательными* являются *требования*, относящиеся к системам требований ресурсной обеспеченности *управляющих ссу-воздействий* и функционирования *Решающей стороны* [Ильин, 1996].  $\triangle$

*Ориентирующие требования* задают предпочтительные значения ресурсных и оценочных функций ( $0 < p_i < \infty$ ).  $\triangle$  К *ориентирующим* могут быть отнесены некоторые *требования* в системах требований, определяющих ресурсную обеспеченность функционирования *обеспечивающей стороны* [Ильин, 1996], и показатели эффективности.  $\triangle$

Задача заключается в отыскании вектора распределения ресурсов  $\mathbf{x}$ , доставляющего такую совокупность значений ресурсных и оценочных функций, какую *эксперт* (группа экспертов или обучаемый робот) сочтет наилучшей в текущей ситуации.  $\square$

## 5. Метод целевого перемещения решения (*цпр-метод*)

Идея создания *цпр-метода* сформировалась во время участия автора в проекте *информатизации системы государственного управления* [Ильин, 1996]. Поскольку при обычной для практически значимых задач несовместности системы ограничений методы линейного программирования неприменимы, необходимо было создать удобный инструмент, позволяющий эксперту находить *наилучшие приближения к реализуемым решениям*, чтобы, проанализировав их, можно было увидеть требующиеся изменения в системах ограничений (пример см. в подпараграфе 6.1).

### 5.1. Технология решения

Поиск решения *срр-задачи* выполняется в интерактивном режиме вычислительного эксперимента. На каждом шаге поиска экспертом вводится обновленное описание задачи, постановка и входные данные которой определяются на основе результатов *сопоставления портретов текущей, целевой и достигнутой ситуации*.

*Портрет текущей ситуации* система формирует на основе *DT-данных* о состоянии ресурсного пространства *су-системы*. *Портрет достигнутой ситуации* формируется на основе решения, полученного на последнем выполненном шаге.

**Начальный вектор решения (начальная точка)** может быть выбран экспертом произвольно. По умолчанию система предлагает *компромиссное решение (чебышёвскую точку* [Коллатц, Крабе, 1978]).  $\diamond$  В случае существования симплекса *поиск чебышёвской точки* дает одинаковые запасы по образующим симплекс ограничениям, а в случае, когда симплекс не существует, — минимизацию максимального дефицита:

$$\min(\max(a_{i1}x_1 + \dots + a_{in}x_n - b_i) \mid i = 1, \dots, s) \mid x_j \geq 0, j = 1, \dots, n,$$

где  $s$  — количество ресурсных функций, на которые наложены ограничения.  $\diamond$

$\diamond$  На любом шаге поиска решения эксперт может изменить любые *требования* (в частности, может задать *ориентирующее требование фиксации* значений некоторых функций). Влияние *ориентирующих требований* на *целевое перемещение решения* определено их *приоритетами* (чем выше приоритет, тем значительнее влияние требования и выше уровень его выполнения). Таким способом реализуется пошаговое приближение к решению, сочетающему удовлетворяющие эксперта свойства эффективности и реализуемости.

Найденные решения могут быть занесены в *базу возможных решений* для последующего анализа.  $\diamond$

**Шаг целевого перемещения решения.** Пусть  $\mathbf{x}' = (x'_1, \dots, x'_n)$  ( $x'_j \geq 0, j = 1, \dots, n$ ) — решение на предыдущем шаге или начальная точка и сформированные требования, определяющие перемещение из  $\mathbf{x}'$  в *целевую точку*  $\mathbf{x} = (x_1, \dots, x_n)$  ( $x_j \geq 0, j = 1, \dots, n$ ), определены как

$$\{F_i(\mathbf{x}) = F_i(\mathbf{x}') + h_i[p_i]\}, \quad \text{где } F_i(\mathbf{x}) = a_{i1}x_1 + \dots + a_{in}x_n, \quad i = 1, \dots, l, \quad 0 < p_i \leq \infty.$$

Чтобы изменить на  $h_i$  значения функций  $F_i(\mathbf{x}')$ , эксперт вводит значения  $h_i$  и  $p_i$ . Приоритеты  $p_i$  функций и требования  $h_i$  к изменению их значений эксперт определяет на основе результатов подготовленного системой *сопоставительного анализа значений ресурсных и оценочных функций, соответствующих портретам текущей, достигнутой и целевой ситуации.*

Формально заданная система требований  $\{F_i(\mathbf{x}) = F_i(\mathbf{x}') + h_i[p_i]\}$  может быть несовместной. Поэтому требования рассматриваются как *ориентирующие*, а  $h_i$  — как *задающий шаг* (нередко отличающийся от *реального шага*, который может быть получен для заданного набора требований). Для фиксации значения  $i$ -й функции эксперт задает  $h_i = 0$ .

*Если реальные и задающие шаги имеют один и тот же знак для всех требований, новая точка в любом случае повышает эффективность решения.*

Точка  $\mathbf{x}$  ищется следующим образом.

1. Вычисляется проекция точки  $\mathbf{x}'$  на гиперплоскость  $a_{i1}x_1 + \dots + a_{in}x_n = F_i(\mathbf{x}') + h_i$   $\forall p_i < \infty$ . *Направляющий вектор нормали к гиперплоскости* равен  $(a_{i1}, \dots, a_{in})$  [Воеводин, 1982]. Для вычисления проекции необходимо дать компонентам направляющего вектора нормали приращения  $ha_{i1}, \dots, ha_{in}$  (где  $h$  — неизвестное число). Перемещение по нормали дает приращение функции  $h(a_{i1}^2 + \dots + a_{in}^2)$ , которое должно быть равно  $h$ , следовательно,  $h = \frac{h_i}{a_{i1}^2 + \dots + a_{in}^2}$  ( $a_{i1}^2 + \dots + a_{in}^2 \neq 0$ ). В итоге получаем проекцию  $(x'_1 + \frac{a_{i1}h_i}{a_{i1}^2 + \dots + a_{in}^2}), \dots, (x'_n + \frac{a_{in}h_i}{a_{i1}^2 + \dots + a_{in}^2})$ .

2. Когда проекции найдены для всех  $p_i < \infty$ , выполняется поиск требуемых приращений по всем переменным:  $\Delta x_{ji} = \frac{a_{ij}h_i}{a_{i1}^2 + \dots + a_{in}^2}$  ( $j = 1, \dots, n$ , индексы функций пробегают значения от 1 до  $s, 1 \leq s \leq l$ ).

3. Затем вычисляется конец *среднедействующей* векторов-нормалей  $y_j = x'_j + \frac{p_1 \Delta x_{j1} + \dots + p_s \Delta x_{js}}{p_1 + \dots + p_s}$  ( $j = 1, \dots, n$ ).  $\diamond$  Такая точка ближе к гиперплоскостям, соответствующим требованиям с более высокими приоритетами.  $\diamond$

4. Если приоритеты не были установлены, они полагаются равными 1, и получаются формулы  $y_j = x'_j + \frac{\Delta x_{j1} + \dots + \Delta x_{js}}{s}$  ( $j = 1, \dots, n$ ).

5. Далее, если  $\exists k: p_k = \infty$  ( $1 \leq k \leq l$ ), конец среднедействующей  $y$  проецируется на гиперплоскости  $a_{k1}x_1 + \dots + a_{kn}x_n = F_k(x') + h_k$  и, аналогично пп. 1–4, вычисляется конец среднедействующей этих проекций  $x$ ; в противном случае на роль искомой точки  $x$  претендует  $y$ .

6. Затем проверяется неотрицательность переменных ( $x_j \geq 0$ ,  $j = 1, \dots, n$ ) и отрицательные обнуляются. Наконец, проверяется совпадение знаков реальных и задающих шагов. Если они совпадают для всех требований, вычисленная точка считается целевой точкой  $x$ . При несовпадении система предлагает эксперту скорректировать систему требований.

## 6. Применение и сравнительный анализ *цпр-метода*

Рассматриваемый *цпр-метод* реализован в программной системе «РЕСУРС-комплекс» [Ильин, 1999], предназначенной для решения различных (по назначению и размеру) *срр-задач*.

◊ В настоящее время реализуется версия *РЕСУРС-комплекса*, в которой учитываются рассмотренные в данной статье обновления (*системы знаний об S-задачных объектах* и технологии формирования и анализа портретов ситуаций с использованием цифровых двойников типа *DTI* и *DTA*). ◊

### 6.1. Планирование ресурсного обеспечения мобильных групп патрулирования (*мгп*) экологически опасных объектов

Один из примеров применения: планирование ресурсного обеспечения *мгп*. Для ликвидации чрезвычайных ситуаций (ЧС) в зоне экологически опасных объектов применяются *мгп* типа 1 и типа 2.

*МГП* имеют средства пожаротушения, борьбы с радиационным, биологическим и химическим заражением.

Единицей измерения работы *мгп* служит *экипаж-час* (час работы одного экипажа).

Параметры *отправной ситуации*:

- база патрулирования объекта располагает десятью экипажами *мгп* типа 1 и четырьмя экипажами *мгп* типа 2;
- расходные коэффициенты ресурсов (топлива, спец. средств и т. д.) при работе экипажей *мгп* типа 1 и типа 2 (коэффициент  $a_{ij}$  означает расход  $i$ -го ресурса за час работы экипажа  $j$ -го типа) и ограничения на расход ресурсов;
- стоимость часа работы экипажей *мгп* типа 1 и типа 2 и ограничение на стоимость планируемой операции;
- площади территорий, которые обрабатывают в течение часа *мгп* типа 1 и типа 2, и общая площадь (5000 км<sup>2</sup>), которую необходимо обработать;
- ограничения на количество *мгп* типа 1 и типа 2 и время проведения операции.

*Цель*: за 3 часа обработать 5000 км<sup>2</sup> территории, используя десять *мгп* типа 1 и четыре *мгп* типа 2.

Пусть  $x_1$  и  $x_2$  — компоненты искомого плана применения *мгп* типа 1 и типа 2.

Ограничения на применение *мгп*:  $x_1 \leq 30$  (3 часа работы десяти *мгп* типа 1) и  $x_2 \leq 12$  (3 часа работы четырех *мгп* типа 2).

Экипаж *мгп* типа 1 обрабатывает 25 км<sup>2</sup> за 1 час, а экипаж *мгп* типа 2 — 400 км<sup>2</sup>. Так как необходимо обработать 5000 км<sup>2</sup>, имеем ограничение  $25x_1 + 400x_2 = 5000$ .

*МПП* типа 1 расходует за 1 час 0,01 т горючего, а *мгп* типа 2 – 0,03 т. Поскольку база располагает 4 т горючего, имеем  $0,01x_1 + 0,03x_2 \leq 4$ .

Аналогично имеем ограничения по средствам:

- пожаротушения –  $1,2x_1 + 3,5x_2 \leq 20$ ;
- дезактивации –  $0,08x_1 + 0,2x_2 \leq 5$ ;
- дезинфекции –  $0,3x_1 + 0,6x_2 \leq 3$ ;
- дегазации –  $0,9x_1 + 1,55x_2 \leq 12$ .

Стоимость 1 часа работы *мгп* типа 1 50 тыс. руб., а *мгп* типа 2 – 100 тыс. руб. Минимизируемая функция суммарных расходов:  $\min(5x_1 + 10x_2)$ .

**Поиск решения.** При попытке решить задачу минимизации суммарных расходов эксперт получает от системы «РЕСУРС-комплекс» сообщение о несовместности системы ограничений.

Тогда вместо минимизации суммарных расходов эксперт задает величину максимального расход на операцию (3000 тыс. руб.) и вводит ограничение:  $5x_1 + 10x_2 \leq 3000$ .

Вычисляет начальную точку:  $x_1 = 0$ ,  $x_2 = 12,44$  (см. табл. 1).

Таблица 1. Решение в режиме «КОМПРОМИСС» (поиск чебышёвской точки)

Параметры ситуации	Расход	Остаток
Топливо (т)	0,37	3,63
Средства пожаротушения (т)	43,54	-23,54
Средства дезактивации (т)	3,49	2,51
Средства дезинфекции (т)	7,46	-4,46
Средства дегазации (т)	19,28	-7,28
Площадь обработки (км <sup>2</sup> )	4976,00	-24,00
Общая стоимость работы (тыс. руб)	1244,00	1756,00
Применение <i>мгп</i> типа 1 (экипаж-час)	0,00	30,00
Применение <i>мгп</i> типа 2 (экипаж-час)	12,44	-0,44

Компромиссный вариант непригоден, так как нарушены критические ограничения: обрабатывается не вся территория (4976 км<sup>2</sup>) и превышена реальная применимость *мгп* типа 2.

Эксперт устанавливает требование «Увеличить» для функции «Площадь обработки» и делает шаг к необходимому значению функции (см. табл. 2).

Таблица 2. Шаг при требовании «Увеличить» («Площадь обработки»)

Параметры ситуации	Расход	Остаток
Топливо (т)	0,38	3,62
Средства пожаротушения (т)	43,75	-23,75
Средства дезактивации (т)	2,50	2,50
Средства дезинфекции (т)	7,50	-4,50
Средства дегазации (т)	19,38	-7,38
Площадь обработки (км <sup>2</sup> )	5000,00	0,00
Общая стоимость работы (тыс. руб)	1250,00	1750,00
Применение <i>мгп</i> типа 1 (экипаж-час)	0,00	30,00
Применение <i>мгп</i> типа 2 (экипаж-час)	12,50	-0,50

На следующем шаге, зафиксировав значение функции «Площадь обработки», эксперт устанавливает требование «Увеличить» для функции «Применение мги типа 1», так как, изменяя только мги типа 2, операцию выполнить не удастся.

В итоге пошагового перемещения эксперт приближается к реализуемому решению:  $x_1 = 29,92$ ,  $x_2 = 10,63$  (см. табл. 3).

Таблица 3. Приближение к реализуемому решению

Параметры ситуации	Расход	Остаток
Топливо (т)	0,62	3,38
Средства пожаротушения (т)	73,11	-53,11
Средства дезактивации (т)	4,52	0,48
Средства дезинфекции (т)	15,35	-12,35
Средства дегазации (т)	43,40	-31,40
Площадь обработки (км <sup>2</sup> )	5000,00	0,00
Общая стоимость работы (тыс. руб)	2559,00	441,00
Применение мги типа 1 (экипаж-час)	29,92	0,08
Применение мги типа 2 (экипаж-час)	10,63	1,37

Для реализации решения  $x_1 = 29,92$ ,  $x_2 = 10,63$  базе необходимо приобрести 53,11 т средств пожаротушения, 12,35 т средств дезинфекции и 31,4 т средств дегазации.

## 6.2. Отличительные особенности цпр-метода

Поскольку при несовместности системы ограничений методы линейного программирования (ЛП-методы) не имеют средств поиска приближений к реализуемым решениям, сравнение ЛП-методов с цпр-методом имеет смысл только для решений, полученных при совместности системы ограничений.

При несовместности далее рассмотрим только решения, полученные с помощью цпр-метода.

Для получения сравниваемых решений применены программные системы:

- РЕСУРС-комплекс (реализующий цпр-метод, симплекс-метод [Юдин, Гольштейн, 2022; Vanderbei, 2020] и метод поиска чебышёвской точки [Коллатц, Крабе, 1978]);
- LINGO 20.0<sup>1</sup> (для решения линейных задач распределения ресурсов применяются прямой и двойственный симплекс-методы или метод внутренних точек [Дикин, 1998]).

Учитывая требование аналитической обзримости, рассмотрим простую задачу производственного планирования [Al-Mashari, 2002; Umble et al., 2003].

Пусть фирма CompuQuick выпускает два типа продуктов: Standard и Turbo. Прибыль от продажи единицы Standard составляет 100, а от продажи Turbo — 150. Standard может быть произведено не более 100 единиц в день, а Turbo — не более 120. Суммарные дневные трудозатраты не должны превышать 160 человеко-часов. Производство единицы Standard требует 1 человеко-часа, а производство Turbo — 2 человеко-часа. Требуется определить дневные объемы производства Standard и Turbo, максимизирующие суммарную прибыль при заданных ограничениях по производственным мощностям и трудозатратам.

Рассмотрение начнем с примеров, в которых система ограничений совместна.

**При совместности системы ограничений.** Основные переменные: STANDARD и TURBO; требование максимизации прибыли:  $MAX = 100 \cdot STANDARD + 150 \cdot TURBO$ ;

<sup>1</sup> Современная версия известной программной системы LINGO 2.0 [Schrage, 1998].





Таблица 4. Основные преимущества подхода, реализованного в *цпр-методе*

Подход, реализованный в <i>цпр-методе</i>	Традиционный подход
Существование решения	
Наилучшее приближение к реализуемому решению может быть найдено и в случае <i>несовместности системы ограничений</i>	При <i>несовместности системы ограничений</i> решение не существует. Если решение отыскивается как <i>чебышёвская точка</i> , то не гарантирована его реализуемость
Реализуемость решения	
Реализуемость решения может быть увеличена перемещением от границы области допустимых решений	И при совместности системы ограничений решение не всегда может быть реализовано, так как находится на границе области допустимых решений
Показатели эффективности решения	
Может одновременно контролироваться несколько показателей эффективности	Оптимизируется значение одной целевой функции
Прикладная точность решения	
Реализуемое решение может быть найдено для любой прикладной точности	Не для любой прикладной точности может быть найдено реализуемое решение

## 7. Задача ситуационного распределения одного ресурса

К этому классу относятся задачи *бюджетного проектирования, планирования расходов на проекты* ( $\Delta$  федеральные, региональные и др.  $\Delta$ ) и др. [Ильин, 2015; Pyin, Pyin, 2016].

**Постановка.** Для числового отрезка  $[a^{\min}, a^{\max}]$  ( $a^{\min} \geq 0$ ,  $a^{\max} > 0$ ), задающего величину распределяемого ресурса ( $\Delta$  сумму денег  $\Delta$ ), отрезков  $[b_i^{\min}, b_i^{\max}]$  ( $b_i^{\min} \geq 0$ ,  $b_i^{\max} > 0$ ,  $i = 1, \dots, n$ ), задающих запросы получателей ресурса, и весовых коэффициентов (приоритетов) получателей  $p_i > 0$  ( $i = 1, \dots, n$ ) требуется найти распределение

$$[x_i^{\min}, x_i^{\max}]: \{0 \leq x_i^{\min} \leq b_i^{\min}, x_i^{\max} \leq b_i^{\max}, \sum x_i^{\min} \leq a^{\min}, \sum x_i^{\max} \leq a^{\max}, i = 1, \dots, n\}. \quad (1)$$

Для совокупного вектора искомого распределения  $\mathbf{x} = (x_1^{\min}, \dots, x_n^{\min}, x_1^{\max}, \dots, x_n^{\max})$  может быть также задан набор требований

$$C\mathbf{x} \leq \mathbf{d} \leftarrow \mathbf{q}, \quad (2)$$

где  $C$  — матрица вещественных коэффициентов размера  $k \times 2n$  ( $k \geq 1$ ),  $\mathbf{d}$  — вектор-столбец вещественных констант размером  $k$ ,  $\mathbf{q}$  — вектор-столбец приоритетов требований ( $0 < q_i \leq \infty$ ,  $i = 1, \dots, k$ ).

Требования (1) относятся к обязательным, а требования (2) могут быть как обязательными, так и ориентирующими.

**Методы решения.** В трех следующих случаях задача решается итеративным *методом приоритетного интервального распределения* [Ильин, 2015], реализованным в действующем *интернет-сервисе планирования расходов*<sup>1</sup>:

<sup>1</sup> Ильин А. В. Интернет-сервис планирования расходов // <https://avi1975.wixsite.com/res-plan/services-ru> (дата обращения: 14.04.2025).

1)  $\sum b_i^{\min} > a^{\min}$ ,  $\sum b_i^{\max} > a^{\max}$  ( $i = 1, \dots, n$ ), и набор требований (2) имеет вид

$$\left\{ \sum x_i^{\min} = a^{\min} \leftarrow \infty, \quad \sum x_i^{\max} = a^{\max} \leftarrow \infty \quad (i = 1, \dots, n), \right.$$

$$p_j b_j^{\min} x_i^{\min} - p_i b_i^{\min} x_j^{\min} = 0 \leftarrow 1$$

для каждой пары  $i, j \{1 \leq i \leq n, 1 \leq j \leq n\}$ ,

$$p_j (b_j^{\max} - b_j^{\min}) (x_i^{\max} - x_i^{\min}) - p_i (b_i^{\max} - b_i^{\min}) (x_j^{\max} - x_j^{\min}) = 0 \leftarrow 1 \quad (3)$$

для каждой пары  $i, j \{b_i^{\max} > b_i^{\min}, b_j^{\max} > b_j^{\min}, 1 \leq i \leq n, 1 \leq j \leq n\}$ ,

$$p_j b_j^{\max} x_i^{\max} - p_i b_i^{\max} x_j^{\max} = 0 \leftarrow 1$$

для каждой пары  $i, j \{b_i^{\max} = b_i^{\min}, b_j^{\max} = b_j^{\min}, 1 \leq i \leq n, 1 \leq j \leq n\}$ };

2)  $\sum b_i^{\min} \leq a^{\min}$ ,  $\sum b_i^{\max} > a^{\max}$  ( $i = 1, \dots, n$ ), и набор требований (2) имеет вид

$$\left\{ x_i^{\min} = b_i^{\min} \leftarrow \infty, \quad \sum x_i^{\max} = a^{\max} \leftarrow \infty \quad (i = 1, \dots, n), \right.$$

$$p_j (b_j^{\max} - b_j^{\min}) (x_i^{\max} - x_i^{\min}) - p_i (b_i^{\max} - b_i^{\min}) (x_j^{\max} - x_j^{\min}) = 0 \leftarrow 1$$

для каждой пары  $i, j \{b_i^{\max} > b_i^{\min}, b_j^{\max} > b_j^{\min}, 1 \leq i \leq n, 1 \leq j \leq n\}$ ,

$$p_j b_j^{\max} x_i^{\max} - p_i b_i^{\max} x_j^{\max} = 0 \leftarrow 1$$

для каждой пары  $i, j \{b_i^{\max} = b_i^{\min}, b_j^{\max} = b_j^{\min}, 1 \leq i \leq n, 1 \leq j \leq n\}$ };

3)  $\sum b_i^{\min} > a^{\min}$ ,  $\sum b_i^{\max} \leq a^{\max}$  ( $i = 1, \dots, n$ ), и набор требований (2) имеет вид

$$\left\{ \sum x_i^{\min} = a^{\min} \leftarrow \infty, \quad x_i^{\max} = b_i^{\max} \leftarrow \infty \quad (i = 1, \dots, n), \right.$$

$$p_j b_j^{\min} x_i^{\min} - p_i b_i^{\min} x_j^{\min} = 0 \leftarrow 1$$

для каждой пары  $i, j \{1 \leq i \leq n, 1 \leq j \leq n\}$ .

◇ *Прикладная точность*, задаваемая экспертом, определяет в выбранных единицах измерения минимальную значимую величину ресурса ( $\Delta$  в *интернет-сервисе планирования расходов* — целые степени 10 от  $-4$  до  $9$   $\Delta$ ). Данные округляются в соответствии с заданной *прикладной точностью*. Результаты расчетов округляются по специальному алгоритму — с сохранением требуемой суммы, учетом запросов и приоритетов. ◇

В случаях, когда набор требований (2) отличается от вариантов (3)–(5), задача решается методом целевого перемещения решения.

Схематическое представление *технологии ситуационного распределения одного ресурса* приведено на рис. 3.

## 8. Интернет-сервис планирования расходов (*пр-сервис*)

*Пр-сервис* позволяет эффективно планировать распределение произвольного дефицитного ресурса [Ильин, 2015]. Планирование бюджетов — наиболее широкая область его применения [Шуйн, Шуйн, 2016].

Алгоритмы вычислений *пр-сервиса* реализуются в серверном приложении, работающем на надежном и высокопроизводительном сервере 24 часа в сутки 7 дней в неделю. Клиентское приложение предназначено для ввода пользовательских данных и всех манипуляций с ними, за исключением вычислений.

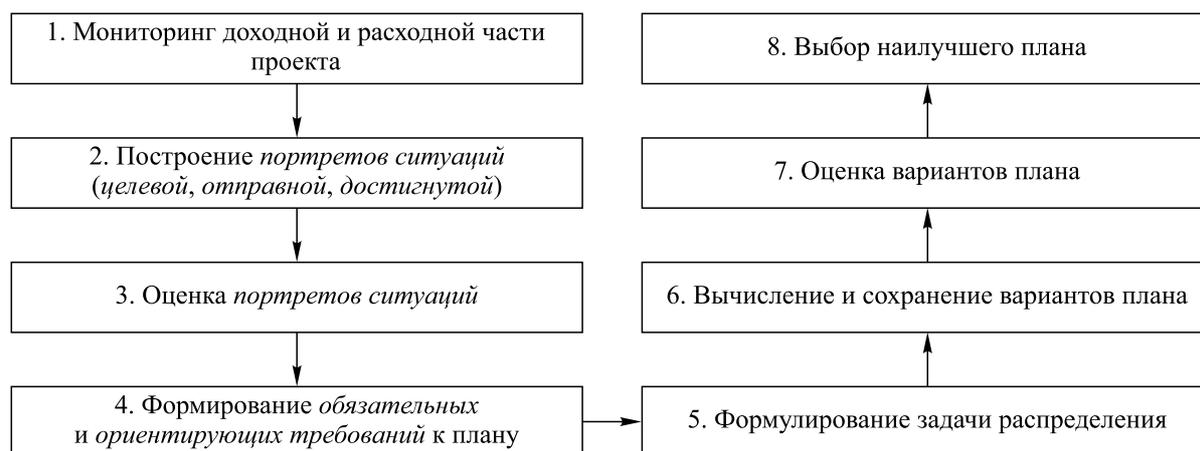


Рис. 3. Схема технологии ситуационного распределения одного ресурса (1–3 реализуются с использованием *DTI* и *DTA*)

Клиентское приложение скачивается пользователем через интернет и устанавливается на его рабочем месте. При регистрации пользователь задает уникальное имя и пароль, адрес электронной почты, страну и тип пользователя (физическое или юридическое лицо).

При первом запуске клиентского приложения проходит процесс регистрации рабочего места на сервере (рабочему месту присваивается уникальный идентификатор).

◇ *PP-сервис* предоставляет возможность ознакомиться с методологическим обоснованием (включая обновляемый список публикаций) и описанием технологии решения задач (одна из графических составляющих которого представлена в черно-белом варианте на рис. 4)<sup>1</sup>. ◇

◇ Клиентское приложение содержит *примеры решения типовых задач планирования расходов для государственного, корпоративного и семейного бюджета*. ◇

Пример расчета *бюджета фермерского хозяйства* приведен в [Пуин, Пуин, 2016].

В клиентском приложении *пр-сервиса* пользователю предоставлена возможность задать произвольное название распределяемого ресурса, выбрать единицы его измерения (△ тыс. руб, тонны и т. д. △) и *прикладную точность планирования* (минимальное значимое количество ресурса). Пользователь задает минимальное количество распределяемого ресурса (как сумму запаса и ожидаемых поступлений при наихудшем раскладе) и максимальное (как сумму запаса и поступлений при наилучшем раскладе). Заполняет таблицу запросов по расходным статьям, для каждой из которых можно ввести наименьший и наибольший предполагаемый расход (или одно значение). Предусмотрена возможность задать *приоритеты* (весовые коэффициенты) *расходных статей*. Некоторые запросы можно пометить как *обязательные* (для каждого из них распределяемое значение должно быть равно запрашиваемому).

Для каждой расходной статьи может быть задана *таблица детализации* (количество ресурса, выделяемое расходной статье, распределяется по детализирующим ее расходным статьям). Число уровней детализации не ограничено: таким образом, предоставлена возможность задать *иерархическую систему распределения ресурса*.

Для каждой таблицы расходных статей можно установить признак «Использовать вес. коэф-ты и прикладную точность». Команда «Распределить» может быть дана как для всей совокупности таблиц, так и для любой части этой совокупности.

*Все данные задач хранятся только у пользователя. На пр-сервере сохраняются только регистрационные данные рабочих мест и данные о реализованных запросах.*

<sup>1</sup> Сервис планирования расходов // <https://avi1975.wixsite.com/res-plan/services-ru>



Рис. 4. Интернет-сервис планирования расходов: схема реализованного подхода

Каждая задача планирования сохраняется в отдельном файле, что позволяет пользователю копировать ее с одного рабочего места на другое. В одну задачу он может объединить планирование расходов нескольких ресурсов. Предусмотрена возможность копирования данных между таблицами задач планирования расходов и листами Microsoft Excel, OpenOffice Calc, LibreOffice Calc и др. Специальные операции импорта/экспорта данных предусмотрены для файлов формата MS Excel (XLS, XLSX).

Получив команду пользователя «*Распределить*», клиентское приложение связывается (по протоколу TCP) с серверным приложением и отправляет запрос в специальном текстовом формате. В тело запроса включаются только необходимые для расчетов числовые данные. Конфиденциальные данные, такие как названия ресурсов, единицы их измерения, названия потребителей ресурсов и др., не передаются. Заголовок запроса состоит из закодированных данных о типе запроса и рабочем месте пользователя. Серверное приложение, получив запрос, производит вычисления и отправляет клиентскому приложению результаты расчета (также в специальном текстовом формате, известном клиентскому приложению). Полученные результаты клиентское приложение предъявляет пользователю: для каждой расходной статьи — значения «*Выделить мин.*» и «*Выделить макс.*». Дополнительно выводятся значения *Выделить средн.* и диаграммы удовлетворения запросов.

По ходу выполнения плана, получив некоторые поступления распределяемого ресурса, сделав часть расходов, уточнив предстоящие поступления и/или расходы, пользователь вводит соответствующие обновленные данные, снова дает команду «*Распределить*» и получает уточненные результаты. Он также имеет возможность смоделировать любой потенциальный расход:  $\Delta$  задать минимальный запрос равным максимальному, пометить его как *обязательный*, дать команду «*Распределить*» и увидеть, как изменятся границы затрат по остальным расходным

статьям  $\Delta$ . Пользователю предоставлена возможность корректировать полученные результаты расчетов. При этом *пр-сервис* уведомит его, если сделанные изменения некорректны.

**Некоторые преимущества *пр-сервиса*.** Накоплен значительный опыт планирования (включая семейные и корпоративные бюджеты с иерархической детализацией статей), подтверждающий методологические преимущества *интервального распределения расходов*.

- Использование *пр-сервиса* значительно дешевле, чем приобретение автономного приложения. Техническая поддержка и получение новых версий клиентского приложения включены в стоимость пользования *пр-сервисом*.
- *ПР-сервис* (по договору) может использоваться сторонними разработчиками: для этого предлагаются специальные интерфейсы программирования приложений (API), позволяющие взаимодействовать с *пр-сервисом*.
- Вычислительные алгоритмы реализованы только на сервере, что позволяет исключить их несанкционированное использование.
- При осмотрительном задании границ для распределяемого ресурса и запросов по расходным статьям и следовании плану, рассчитанному *пр-сервисом*, вероятность его невыполнения радикально снижается.
- При задании ожидаемых доходов и расходов *числовыми сегментами* нет необходимости просчитывать различные сценарии в отдельных задачах. На любом этапе выполнения плана можно смоделировать любой планируемый расход: задать минимальный запрос равным максимальному, пометить его как «*обязательный*», выполнить команду «*Распределить*» и, получив результат, посмотреть, как изменились границы плана по остальным расходным статьям. Любую расходную статью можно временно исключить из рассмотрения, поставив галочку в соответствующей клетке таблицы.
- Число уровней детализации не ограничено (таким способом может быть задана *произвольная иерархическая система распределения*). Для любой таблицы расходных статей предусмотрена возможность задания своей *прикладной точности* (не ниже точности детализируемой статьи) и признака «*Использовать вес. коэф-ты*» (приоритеты).

**Обратная связь.** Реализуется с учетом *политики конфиденциальности*<sup>1</sup>. В полученных отзывах преобладают предложения по совершенствованию *пр-сервиса*. На первом этапе значительное число конструктивных предложений по улучшению интерфейса клиентского приложения и информационного обслуживания пользователей было получено от студентов РТУ МИРЭА (базовая кафедра ФИЦ ИУ РАН), выполнявших курсовые проекты с использованием *пр-сервиса*.

В настоящее время доминируют отзывы с просьбой разработать веб-серверный вариант *пр-сервиса*.

## 9. Система знаний об S-задачных объектах в *сpp-технологии*

Понятие *S-задача* было введено с целью изучения программируемых задач (определенных формально или неформально) и технологий их автоматизированного конструирования и решения в *S-среде* на основе систем знаний об *S-задачных объектах*.

<sup>1</sup> Интернет-сервисы планирования ресурсов. Политика конфиденциальности // <https://avi1975.wixsite.com/res-plan/policy-ru>

***S*-задачный объект** представлен как *конструктивный объект* определенной (в спецификации) степени сложности (*S*-задача / *P* / *простая* или *составная*, *S*-задачный граф /  $G_P$  / одного из трех типов) [Ильин, 2023].

□ *S*-модель задачи  $P =: Stm, Rsr, Alg, Prg, Us$ , где

- $Stm \approx$  постановка  $P =: Mem, Rel$ , где  $Mem$  — множество понятий *S*-задачи, на котором определено разбиение  $Mem = Inp \cup Out$  ( $Inp \cap Out = \emptyset$ );  $Rel$  — семейство бинарных отношений на  $Mem$  ( $Rel \subset Inp \times Out$ );  $Mem$  называем памятью  $P$ , а  $Inp$  и  $Out$  — ее входом и выходом соответственно;
- $Rsr \approx$  множество систем *обязательных* и *ориентирующих требований* к решению (обязательные требования не могут быть нарушены, а выполнение ориентирующих требований не обязательно и зависит от их приоритетов);
- $Alg \approx$  объединение множеств *S*-алгоритмов, каждый из которых соответствует одному элементу из  $Rsr$ ;
- $Prg \approx$  объединение множеств *S*-программ, каждое из которых соответствует одному элементу из  $Alg$ ;
- $Us \approx$  спецификация применимости *S*-задачи (объединение множеств спецификаций применимости  $Ap^{Rsr}$ ,  $Ap^{Alg}$  и  $Ap^{Prg}$ , где  $Ap^{Rsr} \approx$  множество спецификаций *типов решателей S-задачи* ( $\Delta$  однопроцессорный S-сервер, многопроцессорный S-сервер или др.  $\Delta$ ), требования к точности решения, информационной безопасности и др.;  $Ap^{Alg} \approx$  множество спецификаций режимов функционирования каждого решателя S-задачи ( $\Delta$  интерактивный локальный, интерактивный распределенный или др.  $\Delta$ ), ограничения на время получения решения и др.;  $Ap^{Prg} \approx$  множество спецификаций элементов множества  $Prg$  ( $\Delta$  применимые языки программирования, операционные системы, тестовые примеры и др.  $\Delta$ ). □

□ *S*-алгоритм  $=:$  система правил решения  $P$ , позволяющая за конечное число шагов означенному входу  $Inp$  *S*-задачи поставить во взаимно однозначное соответствие ее означенный выход  $Out$ . □

□ *S*-программа  $=:$  *S*-алгоритм, реализованный на языке высокого уровня, машинно-ориентированном языке или в системе S-машинных команд и представленный в виде *S*-сообщения, определяющего поведение *S*-машинного решателя в *S*-среде [Ильин, 2023]. □

◇ В отправной ситуации  $Alg$  и  $Prg$  могут быть пустыми. ◇

□ ***S*-модель системы знаний об S-задачных объектах /  $Skp$**   $=: Pa, Lnp, Intr, Pres$ , где  $Pa \approx$  *S*-задачная область;  $Lnp \approx$  множество языков спецификации объектов, относящихся к  $Pa$ ;  $Intr \approx$  множество интерпретаторов спецификаций конструируемых  $P$ -объектов на  $Pa$ ;  $Pres \approx$  объектные ресурсы  $=:$  каталогизированные спецификации объектов, относящихся к  $Pa$  (включая библиотеки алгоритмов, программ и др.). □

$Pa$  может быть представлена *S*-задачными графами  $G_{P_u}$ ,  $G_{P_c}$  или  $G_{P_G}$ , где  $G_{P_u} =: G_P$ , в множество вершин которого / *G*-вершин / входят только *простые S-задачи* /  $P_u$  /;  $G_{P_c} =: G_P$ , в множество *G*-вершин которого входят *простые* и *составные S-задачи* /  $P_c$  /;  $G_{P_G} =: G_P$ , в множество *G*-вершин которого входит не менее одного *S*-задачного графа ( $G_{P_u}$ ,  $G_{P_c}$  или  $G_{P_G}$ ).

*Ребро G* / *G*-ребро  $=:$  пара вершин с непустым пересечением по памяти. *Нагрузка ребра G*  $=:$  множество всех пар элементов памяти, входящих в пересечение по памяти. *Память G-вершины*  $=:$  память объекта, представляющего *G*-вершину.

Представление *задачной области Pa* рассчитано на *формализацию знаний об S-задачных объектах* и их *автоматизированное конструирование с использованием системы знаний*, построенной в соответствии с *S*-моделью  $Skp$  [Ильин, 2007; Ильин, 2023].

$P_a$  служит для интерпретации спецификаций (составленных на языках из множества  $Lnp$ ) конструируемых  $S$ -задачных объектов (с заданными свойствами и областями применимости).

**Связи по памяти.**  $S$ -задачные конструкции создаются посредством связей по памяти  $S$ -задачных объектов, входящих в конструкцию.

Определены три типа функций связи по памяти, каждая из которых позволяет поставить в соответствие паре  $S$ -задачных объектов некоторый третий объект [Ильин, 1989; Ильин, 2023].

□  $S$ -задачный объект  $a$  связан с объектом  $b$  по памяти, если существует хотя бы одна пара элементов  $\{elem\ Mem^a, elem\ Mem^b\}$ , принадлежащих памяти  $Mem^a$  объекта  $a$  и памяти  $Mem^b$  объекта  $b$ , относительно которой определено общее означивание (элементы имеют одно и то же множество значений).

Если  $S$  и  $H$  — множества объектов и  $D \subseteq S \times S$ , а каждой паре  $(s_i, s_j)$  элементов из  $D$  ставится в соответствие определенный элемент из  $H$ , то задана функция связи по памяти  $h =: conn(s_i, s_j)$ . При этом  $D$  называем областью определения функции  $conn$  и обозначаем как  $D^{conn}$ . Множество  $R =: \{h: elem\ H; h =: conn(s_i, s_j); s_i: elem\ D^{conn}, s_j: elem\ D^{conn}\}$  называем областью значений функции  $conn$ . □

Тип связи по памяти зависит от содержимого пересечения по памяти: составлена ли связь из элементов выхода одной и входа другой  $S$ -задачи или из элементов выходов  $S$ -задач или из элементов их входов или же связь получена путем комбинации предыдущих способов.

△ Часть вершин  $S$ -задачного графа, представляющего  $S$ -задачную область «Планирование ресурсов», связана по памяти посредством матрицы коэффициентов. △

**Конструирование.** Элементарная  $S$ -задачная конструкция —  $S$ -задачная пара. Любая  $S$ -задачная конструкция, в свою очередь, может быть использована как составляющая более сложной конструкции.

Искомая конструкция задается спецификацией, содержащей описание памяти конструируемого  $S$ -задачного объекта, ограничений на число вершин  $S$ -задачного графа, используемых в искомой разрешающей  $S$ -структуре, и спецификацию применимости объекта.

Спецификация искомой конструкции интерпретируется на  $S$ -задачном графе, который служит представлением выбранной  $S$ -задачной области  $P_a$ . Средством интерпретации спецификации служит механизм конструирования разрешающих  $S$ -структур на  $S$ -задачном графе, представляющем  $P_a$  [Ильин, 2007].

Интерпретация заключается в постановке в соответствие некоторому подмножеству (или паре подмножеств) памяти  $P_a$  некоторой подобласти  $pA$  (названной разрешающей  $S$ -структурой).

**Применение.** При проектировании и развитии архитектуры системы «РЕСУРС-комплекс» и серверного приложения  $pr$ -сервиса применена методология автоматизированного конструирования  $S$ -задачных объектов.

Вычислительное ядро РЕСУРС-комплекса представлено моделью  $P_a$   $S$ -задачной области «Планирование ресурсов». Автоматизированное конструирование разрешающих структур на  $S$ -задачном графе, представляющем  $P_a$ , — это процесс построения  $S$ -задачных объектов из уже имеющихся (в системе  $Skp$  знаний об объектах, относящихся к области «Планирование ресурсов»). Результатами служат схемы редукции рассматриваемых объектов к конструкциям, построенным из программно реализованных объектов, содержащихся в  $Pres$  [Ильин, 2023; Пуин, 2022].

## 10. Заключение

В обзоре представлены обновленные версии технологий решения двух классов линейных задач ситуационного распределения ресурсов (*методом целевого перемещения решения и методом интервального распределения*).

Технологии позволяют экспертам (и/или обучаемым роботам) на основе анализа *портретов ситуаций* (с использованием цифровых двойников типа *DTI* и *DTA*) пошагово отыскивать решения, соответствующие их представлениям о реализуемости и эффективности решений.

Представление данных и результата планирования расходов в виде *числовых сегментов* позволяет учесть точность прогнозирования распределяемого ресурса и ожидаемых расходов, способствуя увеличению вероятности реализации расчетного плана.

Представленные версии технологий решения задач прошли проверку на задачах различного назначения и размера (*ресурсного обеспечения управляющих воздействий в системах ситуационного управления, производственного планирования* и др. [Ильин, 1996; Ильин, 2015; Пуин, Пуин, 2016]).

Приведены основные результаты сравнения *метода целевого перемещения решения* (реализованного в *программной системе «РЕСУРС-комплекс»*) с *ЛП-методами* (реализованными в современной версии известной *программной системе LINGO 20.0* [Schrage, 1998]). Наиболее важные из них: *цпр-метод* (в отличие от *ЛП-методов*) предоставляет возможность эксперту (и/или обучаемому роботу) в интерактивном режиме *получать наилучшие приближения к реализуемым решениям и при несовместности системы ограничений*; решения, полученные с использованием *цпр-метода*, всегда реализуемы при малых изменениях данных (в отличие от решений, полученных с помощью *ЛП-методов*).

Приведено описание действующего *интернет-сервиса планирования расходов* (решение задач распределения одного ресурса в иерархической системе с возможностью задания приоритетов расходных статей)<sup>1</sup> [Ильин, 2015].

Рассмотренные в статье обновления *системы знаний об S-задачных объектах* [Пуин, 2022; Ильин, 2023] и *технологии формирования и анализа портретов ситуаций с использованием цифровых двойников* в настоящее время реализуются в версии *РЕСУРС-комплекса*, который будет функционировать как интернет-сервис.

## Список литературы (References)

- Воеводин В. В. Линейная алгебра. — М.: Мир, 1982. — 424 с.  
Voevodin V. V. Lineinaya algebra [Linear algebra]. — Moscow: Mir, 1982. — 424 p. (in Russian).
- Дикин И. И. Определение допустимых и оптимальных решений методом внутренних точек. — Новосибирск: Наука, 1998. — 110 с.  
Dikin I. I. Opredelenie dopustimyykh i optimal'nykh reshenii metodom vnutrennikh tochek [Determination of acceptable and optimal solutions by the method of internal points]. — Novosibirsk: Nauka, 1998. — 110 p. (in Russian).
- Ильин А. В. Интернет-сервис планирования расходов // Системы и средства информатики. — 2015. — Т. 25, № 2. — С. 111–122.  
Ilyin A. V. Internet-servis planirovaniya raskhodov [The online service for cost planning] // Systems and Means of Informatics. — 2015. — Vol. 25, No. 2. — P. 111–122 (in Russian). — DOI: 10.14357/08696527150207
- Ильин А. В. Конструирование разрешающих структур на задачных графах системы знаний о программируемых задачах // Информационные технологии и вычислительные системы. — 2007. — № 3. — С. 30–36.  
Ilyin A. V. Konstruirovaniye razreshayushchikh struktur na zadachnykh grafakh sistemy znaniy o programmiruemyykh zadachakh [Constructing resolving structures on problem graphs of a knowledge system about programmable problems] // Journal of Information Technologies and Computing Systems. — 2007. — No. 3. — P. 30–36 (in Russian).

<sup>1</sup> Ильин А. В. Интернет-сервис планирования расходов // <https://avi1975.wixsite.com/res-plan/services-ru> (дата обращения: 14.04.2025).

- Ильин А. В.* Математическое обеспечение процессов преобразования ресурсов // Системы и средства информатики. — 1999. — Вып. 9. — С. 159–177.  
*Ilyin A. V.* Matematicheskoe obespechenie protsessov preobrazovaniya resursov [Mathematical support of resource transformation processes] // Systems and Means of Informatics. — 1999. — No. 9. — P. 159–177 (in Russian).
- Ильин В. Д.* Основания ситуационной информатизации. — М.: Наука, Физматлит, 1996. — 180 с.  
*Ilyin V. D.* Osnovaniya situatsionnoi informatizatsii [Foundations of situational informatization]. — Moscow: Nauka, Fizmatlit, 1996. — 180 p. (in Russian).
- Ильин В. Д.* Система порождения программ. — М.: Наука, 1989. — 264 с.  
*Ilyin V. D.* Sistema porozhdeniya programm [Program generating system]. — Moscow: Nauka, 1989. — 264 p. (in Russian).
- Ильин В. Д.* Теория S-символов: формализация знаний об S-задачах // Системы и средства информатики. — 2023. — Т. 33, № 2. — С. 124–131.  
*Ilyin V. D.* Teoriya S-simvolov: formalizatsiya znanii ob S-zadachakh [Theory of S-symbols: formalization of knowledge about S-problems] // Systems and Means of Informatics. — 2023. — Vol. 33, No. 2. — P. 124–131 (in Russian). — DOI: 10.14357/08696527230212
- Канторович Л. В.* Математические методы организации и планирования производства. — Л.: Ленингр. гос. ун-т, 1939. — 68 с.  
*Kantorovich L. V.* Matematicheskie metody organizatsii i planirovaniya proizvodstva [Mathematical methods of production organization and planning]. — Leningrad: Leningr. State University, 1939. — 68 p. (in Russian).
- Клыков Ю. И.* Ситуационное управление большими системами. — М.: Энергия, 1974. — 241 с.  
*Klykov Yu. I.* Situatsionnoe upravlenie bol'shimi sistemami [Situational management of large systems]. — Moscow: Energiya, 1974. — 241 p. (in Russian).
- Коллатц А., Крабе Б.* Теория приближений. Чебышёвские приближения. — М.: Наука, 1978. — 271 с.  
*Collatz A., Krabe B.* Teoriya priblizhenij. Chebyshevskie priblizheniya [Theory of approximations. Chebyshev approximations]. — Moscow: Nauka, 1978. — 271 p. (in Russian).
- Поспелов Д. А.* Ситуационное управление: теория и практика. — М.: Наука, 1986. — 288 с.  
*Pospelov D. A.* Situatsionnoe upravlenie: teoriya i praktika [Situational management: theory and practice]. — Moscow: Nauka, 1986. — 288 p. (in Russian).
- Юдин Д. Б., Гольштейн Е. Г.* Линейное программирование. Теория, методы и приложения. — URSS, 2022. — 424 с.  
*Yudin D. B., Gol'shtein E. G.* Lineinoe programmirovaniye. Teoriya, metody i prilozheniya [Linear programming. Theory, methods and application]. — URSS, 2022. — 424 p. (in Russian).
- Aditi, Prasad V. K., Gerogiannis V. C., Kanavos A., Dansana D., Acharya B.* Utilizing convolutional neural networks for resource allocation bottleneck analysis in cloud ecosystems // Cluster Comput. — 2025. — Vol. 28, No. 22. — DOI: 10.1007/s10586-024-04720-z
- Ahammad I.* Fog computing complete review: concepts, trends, architectures, technologies, simulators, security issues, applications, and open research fields // Sn. Comput. Sci. — 2023. — Vol. 4, No. 765. — DOI: 10.1007/s42979-023-02235-9
- Al-Mashari M.* Enterprise resource planning (ERP) systems: A research agenda // Industrial Management & Data Systems. — 2002. — Vol. 103, No. 1. — P. 22–27.
- Cardoso J., Voigt K., Winkler M.* Service engineering for the internet of services // Filipe J., Cordeiro J. (eds.) Enterprise information systems. — ICEIS 2008. — Lecture Notes in Business Information Processing. — Vol. 19. — Berlin, Heidelberg: Springer, 2009. — DOI: 10.1007/978-3-642-00670-8\_2
- Dantzig G. B.* Linear programming and extensions. — Princeton: Princeton University Press and the RAND Corporation, 1963. — 656 p.
- Ghasemi A. A.* New blockchain-based auction method for resource allocation in fog computing environment // J. Netw. Syst. Managem. — 2025. — Vol. 33, No. 9. — DOI: 10.1007/s10922-024-09883-z
- Ilyin A. V., Ilyin V. D.* The technology of interactive resource allocation in accordance with the customizable system of rules // Applied Mathematical Sciences. — 2013. — Vol. 7, No. 143. — P. 7105–7111. — DOI: 10.12988/ams.2013.311649

- Ilyin A. V., Ilyin V. D.* Updated methodology for task knowledge based development of parallel programs // Silhavy R., Silhavy P., Prokopova Z. (eds.) Data science and intelligent systems. — Cham: Springer, 2021. — P. 319–328. — DOI: 10.1007/978-3-030-90321-3\_25
- Ilyin A. V., Ilyin V. D.* Variational online budgeting taking into account the priorities of expense items // AGRIS on-line papers in economics and informatics. — 2016. — Vol. 8, No. 3. — P. 51–56. — DOI: 10.7160/aol.2016.080305
- Ilyin V. D.* A methodology for knowledge based engineering of parallel program systems // Industrial and engineering applications of artificial intelligence and expert systems. Proceedings of the Eighth International Conference. Melbourne, Australia, June 6–8, 1995. — Melbourne, Australia: Gordon and Breach Publishers, 1995. — P. 805–809.
- Ilyin V. D.* Symbolic modeling (S-modeling): an introduction to theory // Silhavy R. (ed.) Artificial intelligence trends in systems. — CSOC 2022. — Lecture Notes in Networks and Systems. — Vol. 502. — Cham: Springer, 2022. — P. 585–591. — DOI: 10.1007/978-3-031-09076-9\_54
- Jamil S., Qureshi R., Ahmed S., Rehman O.* Impact of resource distribution on performance of fog computing infrastructure // Wireless Pers. Commun. — 2024. — No. 137. — P. 1355–1374. — DOI: 10.1007/s11277-024-11272-3
- Katoh N., Shioura A., Ibaraki T.* Resource allocation problems // Pardalos P. M., Du D.-Z., Thai M. T. (eds.) Handbook of combinatorial optimization. — New York, NY: Springer, 2025. — P. 1–93. — DOI: 10.1007/978-1-4614-6624-6\_44-1
- Marnewick C., Labuschagne L.* A conceptual model for enterprise resource planning (ERP) // Information Management & Computer Security. — 2005. — Vol. 13, No. 2. — P. 144–155.
- Nemati A. M., Mansouri N.* Resource allocation in fog computing: a survey on current state and research challenges // Knowl. Inf. Syst. — 2025. — No. 67. — P. 2091–2170. — DOI: 10.1007/s10115-024-02274-5
- Patni S., Saxena D., Singh A.* Resource allocation methods // Resource management in cloud computing. — Cham: Springer, 2025. — P. 49–66. — DOI: 10.1007/978-3-031-83053-2\_3
- Pavan G., Reddy V. K. R. N., Prasad A. R., Chintala R. R., Sai N. R.* A comprehensive survey on recent advances in 5G networks and mobile systems // 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India. — IEEE, 2023. — P. 949–953. — DOI: 10.1109/ICCMC56507.2023.10083901
- Prasad R., Rohokale V.* Internet of things (IoT) and machine to machine (M2M) communication // Cyber security: the lifeline of information and communication technology. — Springer Series in Wireless Technology. — Cham: Springer, 2020. — P. 125–141. — DOI: 10.1007/978-3-030-31703-4\_9
- Prashanth M. V., Praveen K. N. R., Sharma R., Jain R., Vaishnav J., Goyal M. K.* Understanding multi-cloud platform: innovative AI-assisted trust-aware resource allocation technique // Int. J. Syst. Assur. Eng. Manag. — 2025. — DOI: 10.1007/s13198-024-02688-y
- Schrage L.* Optimization modeling with LINGO. — Lindo Systems Inc., 1998. — 550 p.
- Schrijver A.* Theory of linear and integer programming. — John Wiley & Sons, Ltd., 1998. — 484 p.
- Suryawanshi A., Powar S., Attar H., Patil S., Rode K.* Internet of things (IOT): research challenges and future applications // IJRASET. — 2022. — Vol. 9, No. 6. — P. 3372–3376. — DOI: 10.22214/ijraset.2022.44630
- Tang Q., Wu B.* A digital twin-driven multi-resource constrained location system for resource allocation // Int. J. Adv. Manuf. Technol. — 2024. — No. 130. — P. 4359–4385. — DOI: 10.1007/s00170-023-12886-w.
- Tao F., Zhang H., Zhang C.* Advancements and challenges of digital twins in industry // Nat. Comput. Sci. — 2024. — No. 4. — P. 169–177. — DOI: 10.1038/s43588-024-00603-w

- Tikhonov A. N., Arsenin V. Y.* Solutions of ill-posed problems. — Washington, Winston, New York: distributed solely by Halsted Press, 1977. — 258 p.
- Tomar R., Singh N.* Fog computing: concepts, frameworks, and applications. — Chapman and Hall/CRC, 2022. — 234 p.
- Umble E. J., Haft R. R., Umble M. M.* Enterprise resource planning: implementation procedures and critical success factors // European Journal of Operational Research. — 2003. — Vol. 146, No. 2. — P. 241–257.
- Vanderbei R. J.* Linear programming. Foundations and extentions. — Cham: Springer, 2020. — DOI: 10.1007/978-3-030-39415-8