

УДК 519.687+004.023

DOI: [10.26102/2310-6018/2025.50.3.045](https://doi.org/10.26102/2310-6018/2025.50.3.045)

Ресурсно-ориентированная технология организации информационного процесса распределения вычислительных ресурсов при интеграции концепций Интернета вещей и краевых вычислений

А.Б. Клименко 

*Российский государственный гуманитарный университет, Москва,
Российская Федерация*

Резюме. Статья посвящена формированию ресурсно-ориентированной технологии организации информационного процесса распределения вычислительных ресурсов в условиях интеграции концепций Интернета вещей и краевых вычислений. В ходе исследования был проведен анализ существующих моделей и методов и выявлены их недостатки, а именно: отсутствие учета ресурсной стоимости транзита данных для вычислительных узлов, участвующих в передаче данных и вычислительном процессе и отсутствие учета ресурсных затрат, которые требуются для операции распределения вычислительных ресурсов. При этом, учитывая ресурсную ограниченность устройств на краю сети, перечисленные недостатки весьма актуальны. Целью данного исследования является минимизация ресурсных затрат на распределение ресурсов и решение вычислительных задач в системах с ограниченными ресурсами устройств. Основу предложенной технологии составляют предложенная общая математическая модель информационного процесса распределения вычислительных ресурсов, представляющая собой задачу оптимизации; предложенные методы решения указанной задачи, основанные на эвристических правилах и метаэвристиках; алгоритмы учета ресурсной стоимости транзита данных и миграции вычислительных задач, носящие вспомогательный характер и используемые в рамках разработанных методов; а также репозиторий метаэвристических алгоритмов, используемый для выбора оптимального алгоритма решения задачи распределения ресурсов. Предлагаемая технология реализует процесс распределения вычислительных ресурсов, минимизируя ресурсные издержки на транзит данных как при решении вычислительной задачи, так и при принятии решения о распределении ресурсов, учитывая ресурсные ограничения устройств и динамику изменения нагрузки и топологии сети. Проведенное экспериментальное моделирование подтвердило целесообразность применения предложенной технологии. Показано значительное снижение ресурсных затрат на распределение вычислительных ресурсов и достижение улучшенных результатов по показателям эффективности распределенных вычислений. Результаты исследования демонстрируют перспективность предложенной технологии для организации распределенных вычислений в системах с ограниченными ресурсами, таких как системы Интернета вещей и краевые вычисления.

Ключевые слова: распределение вычислительных ресурсов, распределенные вычисления, технология, оптимизация ресурсной стоимости, моделирование распределенных вычислений.

Для цитирования: Клименко А.Б. Ресурсно-ориентированная технология организации информационного процесса распределения вычислительных ресурсов при интеграции концепций Интернета вещей и краевых вычислений. *Моделирование, оптимизация и информационные технологии.* 2025;13(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=2038> DOI: 10.26102/2310-6018/2025.50.3.045

Resource-aware technology of resource allocation informational process organizing in integrated Internet of things and edge computing concepts

A.B. Klimenko✉

Russian State University for Humanities, Moscow, the Russian Federation

Abstract. The article is devoted to the development of a resource-oriented technology for organizing an information process of computational resource distribution under conditions of integrating the concepts of the Internet of Things (IoT) and edge computing. During the research, an analysis of existing models and methods was conducted and their shortcomings were identified, namely: the lack of consideration of the resource cost of data transit for computing nodes involved in data transmission and the computing process and the lack of consideration of the resource costs required for the operation of distributing computing resources. Given the limited resources of devices at the network edge, these drawbacks are particularly relevant. The goal of this study is to minimize resource consumption during resource distribution and solving computational tasks within systems constrained by device limitations. The foundation of the proposed technology includes: an overall mathematical model of resource allocation process, formulated as an optimization problem; proposed methods for solving said problem based on heuristic rules and meta-heuristics; algorithms for calculating the resource cost of data transit and migration of computational tasks, which serve auxiliary purposes within the developed methods; a repository of meta-heuristic algorithms used to select the optimal method for solving the resource distribution problem. This technology implements the distribution of computational resources while minimizing resource expenses associated with data transit, taking into account both the computational task itself and decision-making regarding resource allocation. It considers the resource constraints of devices and dynamic changes in load and network topology. Experimental modeling confirmed the effectiveness of applying the proposed technology. Significant reductions in resource expenditure for computational resource distribution have been demonstrated, leading to improved results in terms of distributed computing efficiency metrics. The results of the study demonstrate the potential of the proposed technology for organizing distributed computing in systems with limited resources, such as IoT systems and edge computing.

Keywords: computing resource allocation, distributed computing, technology, resource costs optimization, distributed computing modelling.

For citation: Klimenko A.B. Resource-aware technology of resource allocation informational process organizing in integrated Internet of things and edge computing concepts. *Modeling, Optimization and Information Technology*. 2025;13(3). (In Russ.). URL: <https://moitvvt.ru/ru/journal/pdf?id=2038> DOI: 10.26102/2310-6018/2025.50.3.045

Введение

В настоящее время широкий круг технических систем строится с использованием концепций туманных / краевых вычислений, реализуемых на различных интеллектуальных устройствах, которые, в совокупности с M2M коммуникацией, образуют Интернет вещей. В качестве примеров могут быть приведены: кооперативное решение задачи SLAM в группах автономных роботов [1], частичная обработка и агрегация видео данных в группах беспилотных летательных аппаратов [2], реализация алгоритмов машинного обучения на пользовательских устройствах [3], выполнение пользовательских приложений в MANET, VANET, FANET [4, 5] и другие. Отличительными признаками таких систем являются: динамика нагрузки и топологий, выраженная ограниченность устройств в ресурсах (вычислительных, энергетических, времени доступа и др.), использование устройств края сети при передаче данных.

Решение вычислительных задач (ВЗ), не всегда реализуемо каким-либо одним устройством; следовательно, необходим процесс планирования распределенных вычислений, включающий распределение вычислительных ресурсов (РВР) для решения подзадач ВЗ.

В настоящее время информационный процесс распределения вычислительных ресурсов (ИПРВР) реализуется, в целом, в двух направлениях:

- с применением эвристических правил (Best Fit, FCFS, HEFT, EFT и др.) [6];
- с применением метаэвристик (имитация отжига, алгоритм роя частиц, генетические алгоритмы и др.) [7].

И если реализация ИПРВР посредством первого направления позволяет получить РВР с минимальными ресурсными затратами на ИПРВР, то второе направление, в зависимости от размерности задачи РВР и количества ограничений, становится достаточно ресурсоемкой процедурой.

В то же время, необходимо отметить ресурсную ограниченность вычислительных устройств (ВУ): для сценариев функционирования системы ВУ, где решение задачи РВР возложено на одно из устройств, которое также является участником процесса распределенного решения ВЗ, с учетом динамики системы и необходимости перепланирования, ресурсная стоимость ИПРВР может быть значительной.

Ресурсная ограниченность ВУ на краю сети также диктует необходимость учета дополнительного расходования ресурсов, которое возникает при транзите данных узлами в случае, если подзадачи ВЗ расположены на различных ВУ, а также при миграции подзадач ВЗ на ВУ, где, как результат ИПРВР, предполагается выделение РВР. При этом возникает необходимость учета маршрутов, по которым будут передаваться данные, что невозможно реализовать применением простых эвристик.

Анализ современных источников по рассматриваемой предметной области показал, что, несмотря на активное исследование систем, реализованных посредством интеграции концепций Интернета вещей и краевых вычислений [8, 9], по-прежнему отсутствуют модели ИПРВР и реализующие его методы, которые бы составили ресурсно-ориентированную технологию организации ИПРВР в указанных условиях, на основе использования метаэвристических алгоритмов оптимизации.

Целью данного исследования является формирование ресурсно-ориентированной технологии организации ИПРВР, для РВР в условиях интеграции концепций краевых вычислений и Интернета вещей. Для достижения поставленной цели будут решены следующие задачи:

- сформулирована общая математическая модель ИПРВР в виде задачи оптимизации;
- предложены методы решения сформулированной задачи оптимизации, полученные путем алгоритмизации сформулированных в данной работе эвристических правил;
- сформулированы алгоритмические реализации дополнительных процедур, необходимых для учета ресурсной стоимости транзита данных;
- сформулирована концепция репозитория метаэвристик, что также используется как средство реализации методов решения задачи оптимизации на основе математической модели ИПРВР;
- построена технология организации ИПРВР, объединяющая разработанные модели, методы, алгоритмы и репозиторий метаэвристик.

Приведены выборочные результаты экспериментального моделирования, являющиеся обоснованием использования основных блоков технологии, что позволяет продемонстрировать целесообразность применения предложенной технологии.

Материалы и методы

Математическая модель ИПРВР. Будем полагать, что имеется некоторое лицо, принимающее решение (ЛПР), владеющее устройствами сети. ЛПР имеет возможность выбора критериев, по которым осуществляется принятие решения о РВР для решения ВЗ. Критерии условно поделены на два множества – общие для сети устройств и индивидуальные, относящиеся к РВР отдельно взятого устройства:

- S_0 – множество общих критериев эффективности РВР;
- P_0 – множество индивидуальных критериев эффективности распределения задач.

Множества S_0 и P_0 формируются на основании предварительно сформулированных и нормализованных критериев оптимизации РВР:

$$S_0 = \{mins_{i_norm}\}_{H, Ro}, \quad (1)$$

$$P_0 = \{P_i^0\} = \{\{minp_{i_norm}\}\}_{H, Ro}, i = 1 \dots M, \quad (2)$$

где $H = \|h_{ij}\|$ – матрица назначений ВЗ на ВУ,

$$h_{ij} = \begin{cases} 1, & \text{если задача } j \text{ назначена на устройство } i, \\ 0, & \text{в противном случае.} \end{cases}$$

- Ro – множество маршрутов передачи данных между задачами, сформированных при распределении фрагментов ВЗ по ВУ.

Далее будем представлять совокупности критериев оптимизации РВР в виде кортежей, причем кортеж индивидуальных критериев оптимизации будет получен конкатенацией индивидуальных кортежей критериев:

$$S'_0 = (mins_{i_norm})_{H, Ro}, \quad (3)$$

$$P'_0 = (minp_{1_norm}, \dots, minp_{k_norm})_{H, Ro}, k = M \cdot |P_i^0|. \quad (4)$$

Также заданы: множество графов задач, реализующих одну пользовательскую задачу с возможностью выбора для решения: $G_l = \{\{G_{ls}\}_v\}$, $G_{ls} = \{g_j, R\}$, где G_{ls} – ациклический орграф, g_j – ресурсные требования задачи, включая прием и передачу данных в сеть; R – множество ребер, взвешенных значениями объемов передаваемых данных между задачами. R также может быть описано матрицей смежности $V = |v_{ij}|$, определяющей объемы информационных обменов между фрагментами ВЗ.

В рамках $\{G_{ls}\}_v$ графы отличаются значениями g_i и весами r_i , матрица смежности задач одна. Каждое множество $\{G_{ls}\}_v$ описывает семейство графов, семейства могут иметь различную матрицу смежности.

Граф доступного фрагмента сети представляется неориентированным графом $G_2 = \{Y, C\}$, $Y = \{\gamma_i\}$ – известные характеристики узла, $i = 1 \dots M$; $VC = \{vc_{kl}\}$ – скорости каналов связи, $k = 1 \dots Z$, $l = 1 \dots Z$, где M – количество узлов, подлежащих рассмотрению, Z – количество каналов связи.

$$\gamma_i = \langle p_i, c_i, c_{0i}, e_i^{comp}, \lambda_0^i, \tau_0^i \rangle, \quad (5)$$

где p_i – производительность узла, c_i – стоимость привлечения узла к вычислениям, c_{0i} – стоимость процессорной операции узла, e_i^{comp} – энергозатраты на выполнение одной инструкции, λ_0^i – интенсивность отказов узла в начале эксплуатации, τ_0^i – предполагаемое время эксплуатации узла.

Также заданы: $S_0, P_0; T_0$ – время выполнения ВЗ.

Рассмотрим переменные математической модели ИПРВР.

Вычислительная сложность ИПРВР g_r .

Матрица назначения подзадач ВЗ на устройства, которая уточняет РВР в соответствии с ранее введенной матрицей H :

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}, \quad (6)$$

где

$$a_{ij} = \begin{cases} t_{ij}, & \text{если задача } j \text{ назначена на узел } i, \\ 0, & \text{в противном случае,} \end{cases}$$

t_{ij} – время назначения задачи j на узле i .

Ранее введенная матрица назначений задач на узлы H может быть получена из матрицы A соответственно:

$$h_{ij} = \begin{cases} 1, & \text{если задача } a_{ij} \neq 0, \\ 0, & \text{в противном случае.} \end{cases}$$

Матрица распределения ресурсов ВУ:

$$U = \begin{bmatrix} u_{11} & \cdots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nm} \end{bmatrix}, \quad (7)$$

где u_{ij} – количество ВУ, в долях от полного вычислительного ресурса ВУ, выделяемых устройством i для решения фрагмента ВЗ j , $0 \leq u_{ij} \leq 1$.

Множество Ro маршрутов передачи данных между фрагментами ВЗ, закрепленными за узлами, $Ro = \{Ro_{jk}\}, j \leq N, k \leq N$.

Вводятся дополнительные матрицы, которые также будут использованы в качестве переменных.

A_Φ , которая представляет собой матрицу A , расширенную моментами времени начала передачи данных для всех узлов, принимающих участие в вычислениях как транзитные ВУ. Элементами матрицы соответственно являются: t_{ij}^Φ ,

$$t_{ij}^\Phi = \begin{cases} t_{ij}^\Phi, & \text{если на ВУ } i \text{ назначена задача } J, \\ 0, & \text{в противном случае,} \end{cases}$$

U_Φ , которая представляет собой расширенную матрицу U значениями ВР, выделенными для передачи данных на транзитных ВУ.

$$u_{ij}^\Phi = \begin{cases} u_{ij}^\Phi, & \text{если на ВУ } i \text{ назначена задача } J, \\ 0, & \text{в противном случае.} \end{cases}$$

Введение матриц A_Φ и U_Φ является необходимым для учета ресурсных затрат на транзит данных; количество ВУ остается неизменным $i = 1 \dots M$, количество задач изменяется в зависимости от значения алгоритмического решения задачи построения маршрута передачи данных, $J = 1 \dots X$, где X – количество ВУ, принимающих участие как в решении ВЗ, так и в передаче данных между подзадачами ВЗ. При этом, в зависимости от сформированных маршрутов Ro , исходный граф ВЗ преобразуется следующим образом (Рисунок 1).

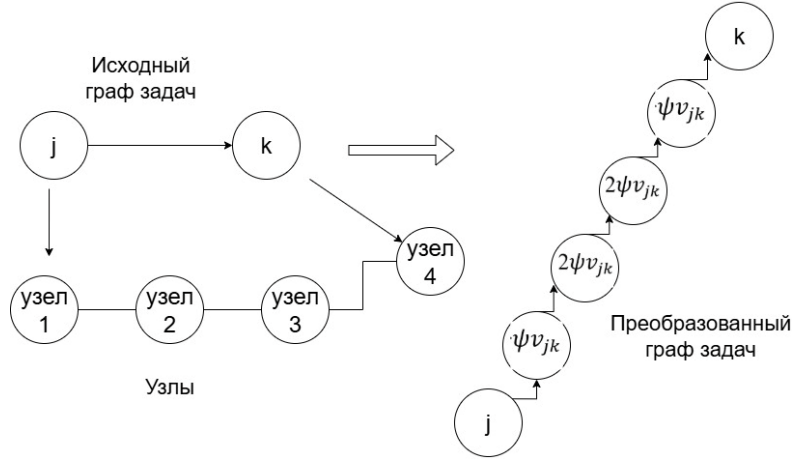


Рисунок 1 – Преобразование исходного графа задач
Figure 1 – Initial tasks graph transformation

Формируется некоторое количество новых задач, являющихся реализацией ВЗ приема-передачи данных ВУ. Такой преобразованный граф будем называть далее Θ .

В общем случае, имеется множество ограничений $constr = \{constr_k\}$, которые формируются на основании особенностей используемых вычислительных устройств (например, ограничение на время использования ВУ, на загруженность ВУ и т. п.).

При этом обязательными будем считать следующие ограничения:

1. Время начала выполнения j -й задачи на i -м устройстве не может быть меньшим, чем время окончания предшествующих задач и время передачи данных по сформированному маршруту Ro_{kj} :

$$\forall t_{ij}^{\Phi}, t_{lk}^{\Phi} \text{ из } A_{\Phi} \max \left(\left(t_{lk}^{\Phi} + \frac{g_k^{\Theta}}{u_{lk}^{\Phi} p_l} \right) + \tau(Ro_{kj}) \right) \leq t_{ij}^{\Phi}, \quad (8)$$

где $\tau(Ro_{kj})$ – время передачи данных по маршруту от задачи k к задаче J , $i, l \in [1 \dots M]$, g_k^{Θ} – вычислительная сложность задачи, предшествующей задаче J в графе Θ .

2. Каждый вычислительный узел сети не должен быть перегружен, т. е. загруженность его процессора должна быть менее 100 %. При этом, если вычислительный узел участвует в транзите данных, такая нагрузка тоже учитывается и суммируется с нагрузкой по решению ВЗ:

$$\forall u_{ij}^{\Phi} \text{ из } U_{\Phi} \sum_{j=1}^X u_{ij}^{\Phi} \leq 1. \quad (9)$$

3. Ограничение на время выполнения ВЗ:

$$\forall i, J, \max \left(\left(t_{ij}^{\Phi} + \frac{g_J^{\Theta}}{u_{ij}^{\Phi} p_j} \right) \right) \leq T_0, \quad (10)$$

где g_J^{Θ} – вычислительная сложность подзадачи ВЗ J в графе Θ .

Введем штрафную функцию, отражающую влияние ресурсной стоимости ИПРВР на результирующее распределение ВР: $\alpha(g_r)$, полагая ее априори неизвестной, но получаемые при этом значения достаточными, чтобы оказывать влияние на показатели РВР.

Также введем штрафную функцию, отражающую влияние необходимости миграции фрагментов ВЗ к узлам, на которых по результатам распределения ВР они должны располагаться. Будем описывать влияние переноса вычислительной нагрузки, которое происходит при миграции посредством штрафной функции $\beta(A)$.

Обобщенный критерий оценивания эффективности ИПРВР имеет вид:

$$I = \min_{A,U,Ro,A\Phi,U\Phi,g_r} \prod_{z=1}^{|S_0|} s_0 \cdot \prod_{x=1}^{|P_i^0|} \prod_{l=0}^{M-1} p_{0[x+l \cdot |P_i^0|]} + \alpha(g_r) + \beta(A). \quad (11)$$

Соответственно, необходимо найти:

$$\operatorname{argmin}_{A,U,Ro,A\Phi,U\Phi,g_r} \prod_{z=1}^{|S_0|} s_0 \cdot \prod_{x=1}^{|P_i^0|} \prod_{l=0}^{M-1} p_{0[x+l \cdot |P_i^0|]} + \alpha(g_r) + \beta(A)$$

при заданных ограничениях и исходных данных.

Репозиторий метаэвристик. Репозиторий представляет собой базу данных с описанием производительности используемых для решения задачи РВР метаэвристических алгоритмов. При этом полагается ограниченность множества решаемых задач и возможность разбиения множества на непересекающиеся классы.

1. Пусть имеется множество задач РВР, характерных для рассматриваемой технической системы $C = \{c_i\}$.

2. Также пусть имеется разбиение множество задач на классы C_j , такое, что:

$$C = C_1 \cup C_2 \dots \cup C_k, C = C_1 \cap C_2 \dots \cap C_k = \emptyset.$$

3. Для каждого класса C_j существует такой алгоритм и определены его параметры, $a_j \in A$, для которого справедливо следующее: на оси отсчета вызовов фитнес-функции (что определяет ресурсные траты на выполнение алгоритма), значения $F_{kj}(x)$ обобщенных критериев задач c_j , объединенных в множество C_j , подчиняются следующему неравенству:

$$F_{low_j}(x) \leq F_{kj}(x) \leq F_{up_j}(x),$$

где $F_{low_j}(x)$, $F_{up_j}(x)$ – определяют некоторый интервал значений для вызова фитнес-функции x для алгоритма a_j , т. е. производительность алгоритма будет сопоставимой для класса задач.

4. Для каждого класса задач C_j может быть представлено более одного алгоритма a_j , но для каждого алгоритма a_k не более одного подкласса задач C_j .

5. Для каждого a_j также имеется составленный на основе статистического анализа эксперимента интервал значений вызовов целевой функции (ЦФ), где для алгоритма a_j решающего задачу из подкласса C_j достигается удовлетворение ограничений задачи c_j : $[t_min_csp, t_max_csp]$.

Таким образом, описания метаэвристик в репозитории в соответствии с описанным выше разбиением на классы позволяет производить выбор метаэвристик, которые:

- 1) позволяют решить задачу РВР наиболее точным образом в рамках зафиксированного бюджета ресурсов;
- 2) позволяет решить задачу РВР с наименьшими ресурсными затратами в рамках зафиксированных требований к точности решения;
- 3) позволяет получить решения, удовлетворяющие ограничениям, с наименьшими затратами бюджета ресурсов.

Базовые методы решения задачи оптимизации ИПРВР. Решение задачи оптимизации, сформулированной выше, не является тривиальным и не может быть решено при помощи классических методов исследований операций или математического программирования, во-первых, в силу зависимости значения критериальной функции от вычислительной сложности процедуры РВР, во-вторых, в силу алгоритмического вычисления некоторых переменных и зависимости их значений от результатов работы используемых алгоритмов. Поэтому для решения сформулированной оптимизационной

задачи предлагаются следующие эвристические правила, направленные на снижение ресурсной стоимости ИПРВР:

– **Правило 1:** $\beta(A)$ удаляется из обобщенного критерия (9), при этом перемещение вычислительной нагрузки / миграция фрагментов ВЗ осуществляется таким образом, чтобы минимизировать длину маршрута перемещения фрагмента ВЗ и количество мигрирующих фрагментов ВЗ.

– **Правило 2:** $\alpha(g_r)$ удаляется из состава обобщенного критерия (9), при этом из заранее подготовленного репозитория метаэвристических алгоритмов оптимизации следует выбрать такой, который при фиксированном времени окончания работы позволяет получить наиболее точный результат, либо при незаданном времени окончания работы позволяет получить решение задачи РВР, удовлетворяющее ограничениям, за минимальное время.

Предложенные эвристические правила алгоритмируются, преобразуясь в соответствующие методы.

Метод сокращения накладных расходов при переносе нагрузки (Правило 1). Сформируем два показателя эффективности размещения фрагментов задач за ВР:

- 1) количество пересылаемых фрагментов ВЗ;
- 2) средняя длина маршрута пересылаемых данных.

Метод сокращения ресурсопотребления процедуры переноса нагрузки сводится к следующей последовательности этапов:

- определить требуемый вид свертки критериев для эффективности расположения фрагментов ВЗ;
- сформировать обобщенный критерий количества и расстояний миграций фрагментов ВЗ.

Далее представлено формализованное представление обобщенного критерия количества и расстояний миграций фрагментов ВЗ.

Пусть имеется матрица исходного распределения вычислительных задач

$$H = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{bmatrix} \quad (12)$$

и матрица результирующего распределения

$$H' = \begin{bmatrix} h'_{11} & \cdots & h'_{1n} \\ \vdots & \ddots & \vdots \\ h'_{m1} & \cdots & h'_{mn} \end{bmatrix}, \quad (13)$$

где $h_{ij} = 1, h'_{ij} = 1$ если вычислительная задача j назначена на ресурс i , и $h_{ij} = 0, h'_{ij} = 0$ в противном случае.

Представим оценку количества переносов задач следующим образом:

$$Tr_{ij} = |h_{ij} - h'_{ij}| = \begin{cases} 0, & \text{если размещения совпали,} \\ 1, & \text{в противном случае.} \end{cases}$$

Количество перемещений задач будет иметь вид:

$Tr = \sum_{i=1}^M \sum_{j=1}^N Tr_{ij}$, и соответственно, критерий уменьшения перемещений задач имеет вид:

$$I = \min_{H'} Tr. \quad (14)$$

Максимальное значение, достигаемое $Tr = N$, т. е. в случае, когда перемещаются все N задач. Минимальное значение, соответственно, равно 0 – в случае, когда ни одна из вычислительных задач не изменяет своего расположения.

Следовательно, нормализованный критерий имеет вид:

$$I = \min_{H'} \frac{Tr}{N}. \quad (15)$$

Предположим, что сформированы маршруты перемещения для N задач с длинами $Sp_{h_{ij}}^{h'_{kj}}$, где k – номер узла, на котором размещена j -я задача соответственно матрице H' . Значения $Sp_{h_{ij}}^{h'_{kj}}$ получены как результат выполнения алгоритма построения маршрута переноса задачи из узла i , для которого $h_{ij} = 1$, в узел k , для которого $h_{kj} = 1$.

Тогда оценка средней длины маршрутов перемещенных задач для заданных матриц H и H' будет иметь вид:

$$ASp = \sum_{j=1}^N Sp_{h_{ij}}^{h'_{kj}} / Tr, \quad (16)$$

т. е. отношение суммы всех длин маршрутов при переносе задач к количеству переносов.

Верхняя граница значений, приобретаемых ASp , может быть выражена как сумма диаметров сети по количеству задач N (то есть, все задачи переносятся по самому длинному маршруту) поделенная на количество переносов задач:

$$ASp_{max} = \sum_{j=1}^M \vartheta / Tr, \quad (17)$$

где ϑ – диаметр сети.

Соответственно, нормализованная оценка средней длины маршрута перемещения задачи будет иметь вид:

$$ASp^n = \frac{\sum_{j=1}^N Sp_{h_{ij}}^{h'_{kj}}}{\sum_{j=1}^N \vartheta}. \quad (18)$$

Интегральный критерий эффективности при этом, в виде мультипликативной свертки будет иметь вид:

$$\min_{H'} Tc = \min_{H'} Tr_{H,H'}^n \cdot ASp^n = \min_{H'} \frac{Tr_{H,H'}}{N} \cdot \frac{\sum_{j=1}^N Sp_{h_{ij}}^{h'_{kj}}}{\sum_{j=1}^N \vartheta}. \quad (19)$$

Если допустить, что уменьшение количества пересылок возмещает потери при удлинении маршрутов, и использовать аддитивную свертку, то, соответственно:

$$\min_{H'} Tc = \min_{H'} (Tr_{H,H'}^n + ASp^n) = \min_{H'} \left(\frac{Tr_{H,H'}}{N} + \frac{\sum_{j=1}^N Sp_{h_{ij}}^{h'_{kj}}}{\sum_{j=1}^N \vartheta} \right). \quad (20)$$

Добавление сформулированного критерия к обобщенному критерию (11) позволяет получать решения, где распределение ВР преимущественно происходит поблизости от узла-точки входа данных.

Метод снижения ресурсной стоимости распределения вычислительных ресурсов (Правило 2). Время ожидания результатов решения ВЗ T_0 задается такое, что временные затраты на распределение ВР и решение ВЗ включены в него, и ситуация, когда время решения ВЗ не может быть меньше T_0 , исключена.

Определение значения T , если не задано время начала выполнения ВЗ T_e . T здесь определяет время, которое может быть потрачено на процесс распределения ВР, т. е. определяет g_r . Для выбора алгоритма оптимизации в соответствии с эвристикой производится классификация задачи распределения ВР и далее выбирается подходящий алгоритм из репозитория алгоритмов.

- Классификация ЗРВР производится по имеющимся признакам, выбирается алгоритм a_{ik} , где i – индекс алгоритма, k – индекс класса задачи РВР, для которого $\min(max_t_csp)$ – алгоритм, который гарантированно позволяет получить решение, удовлетворяющее ограничениям, с минимальными ресурсными затратами. T устанавливается $\min(max_t_csp)$.
- Алгоритм a_{ik} применяется к решению на протяжении времени T .
- Результатом является некоторое решение задачи РВР $solution(a_{ik}, T) = \langle A, U, Ro, A_\phi, U_\phi \rangle$ и время решения ВЗ $makespan(a_{ik}, T)$, полученное в результате решения.
- Если $makespan(a_{ik}, T) < T_0 - T - T(PBR)$: $T = T_0 - makespan(a_{ik}, T) - T(PBR)$, где $T(PBR)$ – время решения задачи РВР.
- Произвести классификацию ЗРВР для нового T .
- Выбрать a_{ik} , обеспечивающий $\min solution(a_{ik}, T)$ при условии, что $max_t_csp(a_{ik}) \leq T_0 - T_e$ в соответствии с данными репозитория.
- Решение – $solution(a_{ik}, T)$.
- В случае, если задано T_e (ожидаемое время начала решения ВЗ), общая схема решения принимает следующий вид.
- Классификация задачи.
- Для $(T_0 - T_e)$ выбирается алгоритм, для которого выполняется $\min solution(a_{ik}, T_0 - T_e)$ при условии, что $max_t_csp(a_{ik}) \leq T_0 - T_e$, либо может выбираться алгоритм, достигающий заданной точности ε с минимальными ресурсными затратами.
- Произвести решение ВЗ выбранным алгоритмом a_{ik} , получая решение $solution(a_{ik}, T)$.
- Решение – $solution(a_{ik}, T_0 - T_e)$.
- Предлагаемый здесь метод можно условно разбить на две фазы:
- Выбор алгоритма, который позволяет получить первое допустимое решение с минимальной ресурсной стоимостью.
- Если после получения допустимого решения полученное значение времени выполнения ВЗ оставляет некоторый интервал времени, то он может быть использован для повышения точности решения задачи РВР с выбором такого алгоритма, который в заданных ограничениях на время позволяет получить наиболее точный результат при решении задачи РВР либо достигает заданной точности с наименьшими ресурсными затратами.

Комплексный метод организации ИПРВР. Объединим полученные эвристические методы в комплексный метод организации ИПРВР, который будет включать следующие этапы (Рисунок 2):

1. Определение типа модели ИПРВР в гетерогенных динамических вычислительных средах (ГГДВС) (вариативность структуры и вычислительной сложности ВЗ или вариативность ресурсов, либо и то, и то одновременно). Этот этап используется для выбора соответствующего репозитория с целью сокращения времени поиска по базе данных.

2. Сбор данных: $S'_0, P'_0, s^0, p^0, G_1, G_2, T_0, T_e$ (опционально), $constr$.

3. Добавить критерий, полученный в результате применения метода сокращения ресурсопотребления переноса нагрузки, к обобщенному критерию (13). Получается модифицированный обобщенный критерий.

4. Применить метод сокращения ресурсопотребления ИПРВР и решения ВЗ, в зависимости от исходного типа модели ИПРВР, используя соответствующий репозиторий.

Результатом применения метода организации ИПРВР в ГГДВС являются значения переменных A, U, Ro, A_Φ, U_Φ что определяет РВР, в том числе, на транзит данных между подзадачами.

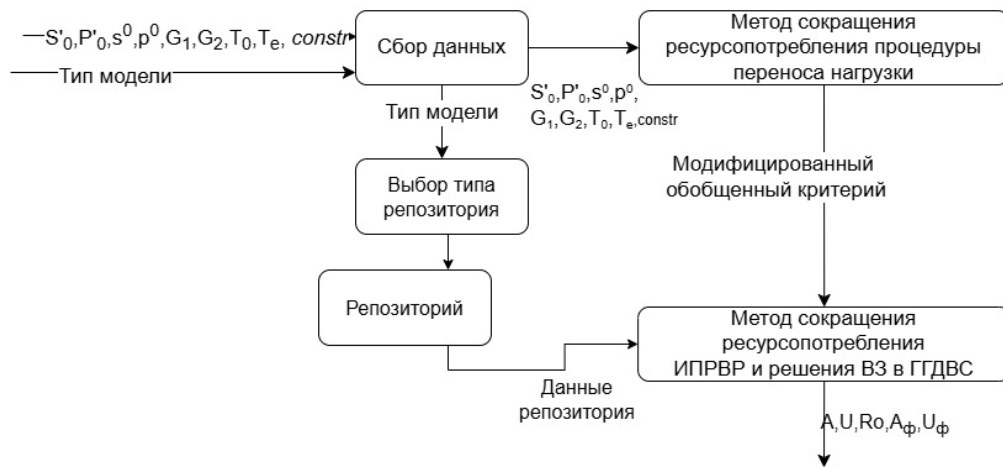


Рисунок 2 – Схема метода организации ИПРВР в ГГДВС

Figure 2 – Scheme of the method of organizing the information process of distributing computing resources in heterogeneous dynamic computing environments

Вспомогательные алгоритмы технологии организации ИПРВР.

Алгоритм учета ресурсной стоимости транзита данных. Как было сказано ранее, переменные A_Φ, U_Φ формируются с вовлечением алгоритмического вычисления маршрутов передачи данных между подзадачами. Для формирования значений в A_Φ, U_Φ используется алгоритм для вычисления ресурсной стоимости транзита данных, представленных ниже.

Обозначим ресурсную стоимость транзита данных для узла x следующее:

$$\Phi(x) = F(Ro, V),$$

где Ro – множество маршрутов передачи данных между задачами, сформированных при распределении ВЗ по ВУ. $V = |v_{jz}|$, где

$$v_{jz} = \begin{cases} v_{jz}, & \text{если подзадача } j \text{ передает данные подзадаче } z, \\ 0, & \text{в противном случае.} \end{cases}$$

Матрица V является матрицей смежности заданного графа G_{ls} .

Приведем алгоритм оценивания ресурсной стоимости транзита данных для ВУ x при исходных данных Ro, V .

1. $L_x = []$ – массив объемов данных, передаваемых узлом с номером x . $K = []$ – инициализация вспомогательного массива.
2. Для множества Ro найти все $\langle j, k \rangle$, для которых x является частью маршрута и принадлежит множеству ВУ, составивших маршрут $\{Ro_{jk}\}$.
3. Двойки $\langle j, k \rangle$ сохраняются в массив K . $Y = \text{размер массива } K$.
4. Если $x = i \parallel x = z$, то $L_x[y] = v_{jk}$; в противном случае $L_x[y] = 2v_{jk}$.
5. Для $y = 1 \dots Y$: $\Phi(x)[y] = \psi L_x[y]$, где ψ – коэффициент, связующий объем передаваемых данных и ресурсную стоимость передачи единицы данных.

Алгоритм преобразования исходного графа задач к графу Θ производится для использования при вычислении соответствия получаемого решения ограничениям на следование подзадач, что также включает и новые подзадачи, образующиеся в результате транзита данных по маршрутам.

Алгоритм преобразования графа задач. Исходными данными являются массив Ro , граф $B3 G_{Is}$.

Ранее было описано содержимое $Ro_{jk} = |i \dots z|$, где j, k – индексы задач, между которыми осуществляется передача данных, $i, \dots, z$ – индексы узлов, на которых размещены задачи j, k .

Как было предложено в работах [10, 11], исходный граф задач преобразуется с учетом добавления вычислительных задач приема-передачи данных, однако, в данной работе предлагается выполнить преобразование графа задач с учетом всех узлов, принимающих участие в транзите, в то время как ранее использовался либо виртуальный канал, либо шина. Преобразование графа G_{Is} к графу Θ выполняется следующим образом.

1. Для всех $j = 1 \dots N$:
2. Если находится для j некоторый маршрут Ro_{jk} , то:
 $c = j$;
 для $x = 1; x \leq |Ro_{jk}|$
 {добавить в граф G_{Is} вершину, смежную с вершиной c и взвешенную:
 Если $x=1$ или $x=k$, то вес вершины ψv_{jk} , в противном случае – $2\psi v_{jk}$.
 Ребро направлено от c к x и взвешено v_{jk} ;
 $c = x; x = x + 1$;
 }
- 3. Ввести ребро, объединяющее вершины c и k .

Алгоритм формирования решений и проверки ограничений на следование задач при использовании метаэвристик с учетом маршрутов передачи данных.

1. Случайным образом задать A, U, Ro .
2. Для каждого узла x производится оценка $\Phi = |\Phi(x)|$ с использованием алгоритма оценивания ресурсной стоимости транзита данных
3. На основании значений Ro формируется граф задачи Θ .
4. A_Φ создается на основе A и Φ путем добавления случайных значений времен начала выполнения задач транзита данных.
5. U_Φ создается на основе U и Φ путем добавления случайных значений ресурсов, выделенных под задачи транзита данных.
6. Решение A_Φ, U_Φ проверяется на предмет удовлетворения ограничениям с использованием графа задач транзита данных Θ (ограничение на следование задач и ограничение на время выполнения). Если не достигается условие останова алгоритма, перейти к шагу 1.

Алгоритм оценки передачи данных по маршруту. Имея следующие данные: $Ro_{jk} = |1 \dots Z|, v_{jk}, VC$, оценка времени передачи данных по маршруту имеет вид:

$$\begin{aligned} \tau(jk) = & \frac{\psi v_{jk}}{u_{Ro_{jk}[1]j} p_{Ro_{jk}[1]}} + \frac{v_{jk}}{v_{c_{Ro_{jk}[1]Ro_{jk}[2]}}} + \sum_{i=2}^{Z-1} \frac{2\psi v_{jk}}{u_{Ro_{jk}[i]j} p_{Ro_{jk}[i]}} + \sum_{i=2}^{Z-1} \frac{v_{jk}}{v_{c_{Ro_{jk}[i]Ro_{jk}[i+1]}}} + \\ & + \frac{\psi v_{jk}}{u_{Ro_{jk}[Z]j} p_{Ro_{jk}[Z]}} + \frac{v_{jk}}{v_{c_{Ro_{jk}[Z-1]Ro_{jk}[Z]}}} \end{aligned} \quad (21)$$

Следовательно, чтобы получить значение $\tau(jk)$, необходимо:

1. Получить входные данные в виде: Ro_{jk}, v_{jk}, VC .
2. Вычислить $\tau(jk)$ в соответствии с выражением (21).

Технология организации ИПРВР. Модель ИПРВР, методы, алгоритмы и репозиторий метаэвристик объединяются в технологию организации ИПРВР при

интеграции концепций Интернета вещей и краевых вычислений и имеет вид, как показано на схеме Рисунка 3.

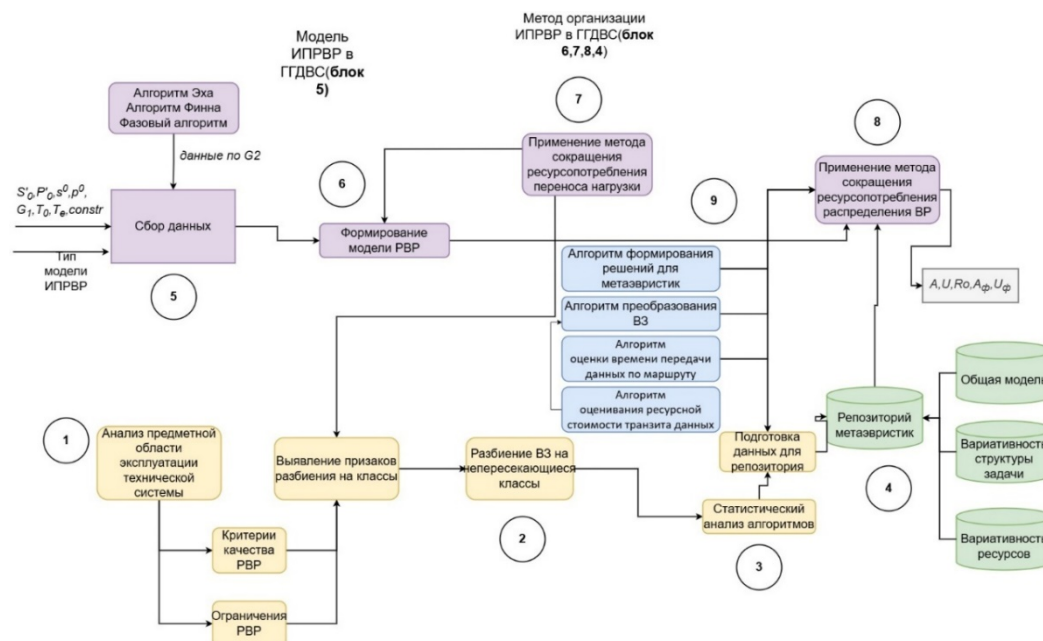


Рисунок 3 – Схема технологии организации ИПРВР

Figure 3 – Scheme of technology of organization of information process of distribution of computing resources

Также следует отметить использование дополнительных алгоритмов обхода сети при сборе данных о свободных ресурсах узлов (алгоритмы Эха, Финна, фазовый алгоритм). На схеме Рисунка 3 показано взаимодействие между элементами предлагаемой технологии, результатом которой является RVP для решения ВЗ на основе использования метаэвристических алгоритмов.

Результаты и обсуждение

Учет ресурсной стоимости транзита данных. Эксперимент проводился с использованием Particle Swarm Optimization алгоритма при заданных графе сети и графе ВЗ (Рисунки 4, 5). На Рисунке 4 указаны идентификаторы устройств и производительность. На Рисунке 5 указаны идентификаторы подзадач, их вычислительные сложности и объемы передаваемых данных между подзадачами.

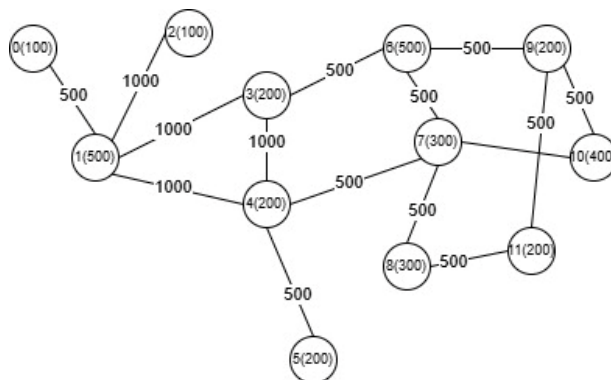


Рисунок 4 – Графическое представление фрагмента сети

Figure 4 – Network fragment graph representation

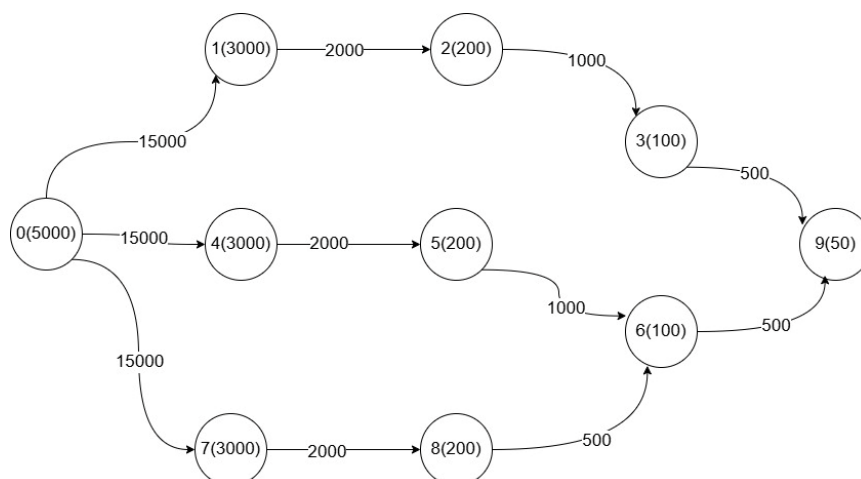


Рисунок 5 – Графическое представление решаемой ВЗ
Figure 5 – Computational task graph representation

В Таблице 1 содержится сравнение ресурсной стоимости решения ВЗ в рамках следующих моделей: классической модели теории расписаний (без учета стоимости транзита), с учетом стоимости приема-передачи данных узлом, с учетом полной ресурсной стоимости транзита данных.

Таблица 1 – Учет ресурсных издержек при решении ВЗ для различных моделей
Table 1 – Accounting for resource costs when solving a computational tasks for various models

Тип распределения задач	Распределение задач (указаны номера ВУ)	Энергозатраты на решение ВЗ	Энергозатраты на прием/передачу данных
Без учета ресурсной стоимости транзита данных	10 8 8 7 10 7 0 10 11 5	8,283	0
С учетом ресурсной стоимости приема-передачи	6 5 11 9 6 4 6 6 2 2	28,91	183
С учетом полной стоимости транзита	7 7 10 9 3 0 7 7 7 0	16,98333	232
Лучшее решение для модели с учетом ресурсной стоимости приема-передачи (фактическое)	6 5 11 9 6 4 6 6 2 2	28,9	387,95

Результаты в Таблице 1 иллюстрируют возрастание неучтенных ресурсных издержек при использовании различных моделей, начиная от классической модели в рамках теории расписаний и заканчивая моделью с учетом приема-передачи данных, как в работе А.Б. Барского [10]. При этом неучтенные ресурсные траты на транзит данных составляют по результатам серии экспериментов до 52 %.

Учет ресурсной стоимости ИПРВР и ее влияние на значение критерия эффективности РВР. На графиках Рисунка 6 представлено суммарное расходование энергетических ресурсов ВУ с учетом и без затрат на решение задачи планирования, исходя из того, что вычислительные сложности подзадач планируемой ВЗ и вычислительная сложность задачи планирования одного порядка.

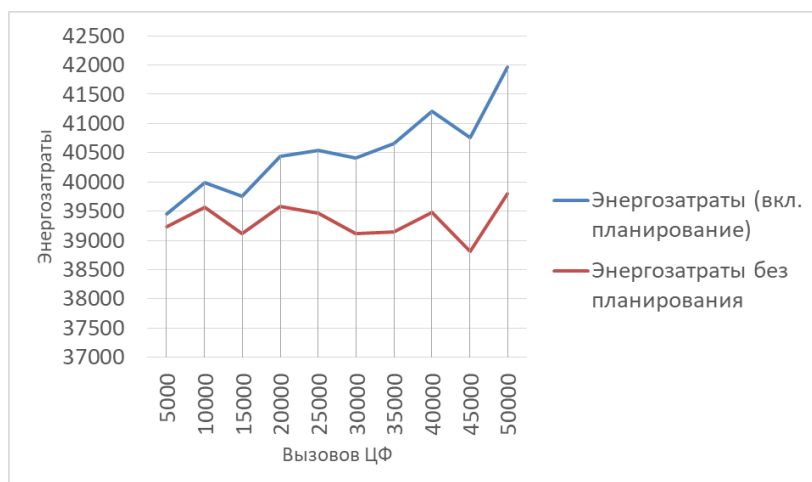


Рисунок 6 – Сравнение энергозатрат с учетом операции планирования и без
Figure 6 – Energy costs comparison with/without computations planning costs

Видно, что с возрастанием объема вычислений для планирования улучшение ЦФ не компенсирует ресурсные затраты на планирование. Следовательно, учет ресурсной стоимости планирования целесообразен.

Эффект применения комплексного метода организации ИПРВР.

Перенос вычислительной нагрузки. В Таблице 2 представлены результаты РВР по критерию энергозатрат и по двум критериям, с добавлением критерия минимизации количества и расстояний миграций фрагментов ВЗ. Добавление соответствующего критерия приводит к существенному сокращению объемов миграции подзадач ВЗ, что снижает ресурсные траты на перенос вычислительной нагрузки, при этом реализуется такое распределение подзадач ВЗ по ВУ, что не увеличивается вычислительная нагрузка на ВУ (Рисунки 7, 8).

Таблица 2 – Применение метода сокращения накладных расходов при переносе нагрузки при исходном критерии энергозатрат на решение ВЗ

Table 2 – Application of the method of reducing overhead costs when transferring the load under the initial criterion of energy costs for solving a computational task

	Без учета миграции подзадач (по критерию энергозатрат)	С учетом миграции подзадач (дополнительно – критерий энергозатрат)
Распределение подзадач	7, 7, 10, 9, 3, 0, 7, 7, 7, 0	0, 0, 5, 7, 11, 5, 11, 0, 8, 6
Число перенесенных подзадач	8	7
Максимальный маршрут переноса	4	5
Средняя длина маршрута	2,875	3,5
Ресурсная стоимость	38000	17000
Суммарная вычислительная нагрузка при выполнении ВЗ	14850	11750

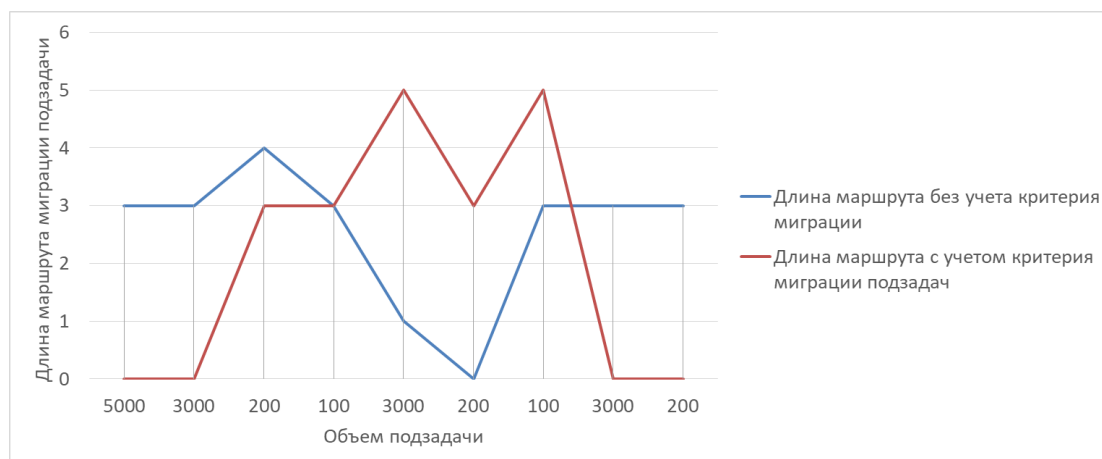


Рисунок 7 – Изменение маршрутов миграции подзадачи ВЗ с введением критерия минимизации количества и расстояний миграций фрагментов ВЗ

Figure 7 – Changing the migration routes of subtasks of a computational task with the introduction of a criterion for minimizing the number and distances of migrations of fragments of a computational task

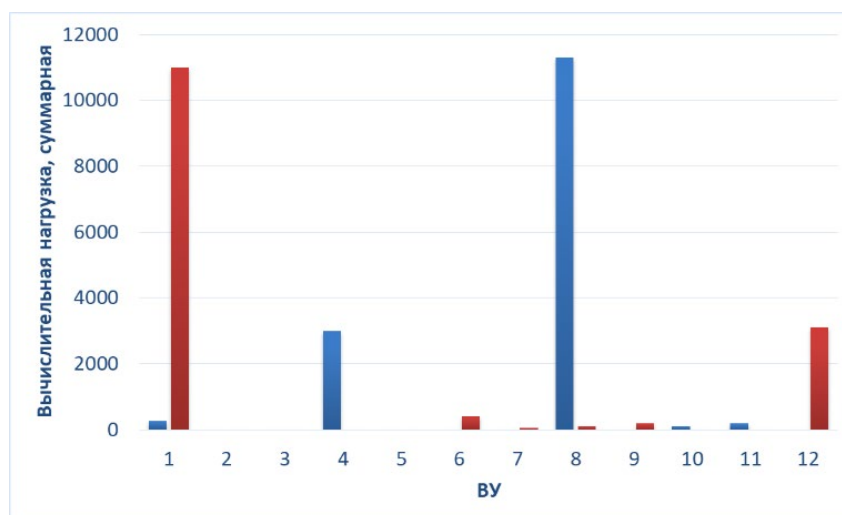


Рисунок 8 – Распределение вычислительной нагрузки с применением критерия минимизации количества и расстояний миграций фрагментов ВЗ

Figure 8 – Distribution of computational load using the criterion of minimizing the number and distances of migrations of fragments of a computational task

Таким образом, применение метода сокращения накладных расходов при переносе нагрузки приводит к снижению ресурсной стоимости переноса подзадачи ВЗ на ВУ, без ухудшения загруженности ВУ при решении ВЗ, что также подтверждает целесообразность применения данного метода.

Снижение ресурсной стоимости PBP. Сформируем репозиторий метаэвристик для задачи PBP 10 машин на 10 подзадач на основе анализа производительности следующих алгоритмов: генетический алгоритм (GA), генетический алгоритм с ламарковскими улучшениями (LGA), имитация отжига (SA), дифференциальная эволюция (DE), алгоритм роя частиц (PSO). По средним значениям получены описания алгоритмов, такие, что имеются значения $\min(\max_t_csp)$, а также описание скорости снижения значений ЦФ (Таблица 3). Задается $\varepsilon = 5\%$ от минимального значения ЦФ в репозитории (что для заданных условий составляет пороговое значение ЦФ, равное

2214). В Таблице 3 представлен пример описания производительности метаэвристик на основе полученных средних значений для заданного класса задач. Видно, что имитация отжига позволяет получать допустимые решения на более ранних сроках, чем, например, генетический алгоритм с ламарковскими улучшениями. При этом на больших интервалах ресурсного бюджета наилучший результат достигается применением генетического алгоритма.

Таблица 3 – Описание производительности метаэвристических алгоритмов

Table 3 – The description of metaheuristic algorithms performances

Вызовов ЦФ	GA	LGA	SA	DE	PSO
50			260334,7		
100			2122,126	56412,25	2373,048
500	3213,887		2122,126	2376,598	2089,451
1000	3213,887	2109,823	2122,126	2202,566	2089,451
1500	3213,887	2109,823	2122,126	2202,566	2089,451
2000	3213,887	2109,823	2122,126	2152,511	2089,451
2500	3213,887	2044,151	2122,126	2124,092	2089,451
3000	2013,585	2044,151	2122,126	2116,356	2089,451
3500	2013,585	2039,546	2122,126	2067,202	2089,451
4000	2013,585	2039,546	2122,126	2067,202	2089,451
4500	2013,585	2039,546	2122,126	2022,596	2089,451
5000	2013,585	2026,373	2122,126	2022,596	2089,451

В Таблице 4 приведены результаты применения метода снижения ресурсной стоимости РВР. Следует отметить, что при желаемой точности 5 % решения задачи РВР, выбор алгоритма PSO позволяет снизить ресурсные траты на 2000 вызовов ЦФ относительно алгоритма LGA, который достигает заданного порога точности следующим после PSO. При сравнении с потенциальным заданным ресурсным бюджетом в 5000 вызовов ЦФ применение предложенного метода существенно снижает ресурсную стоимость РВР (более, чем в 8 раз).

Таблица 4 – Результаты применения метода снижения ресурсной стоимости РВР

Table 4 – Results of applying the method of reducing the resource cost of RVR

	T ₀ , с	Алгоритм	Вызовов ЦФ/время	Время завершения решения ВЗ по результатам моделирования	Значение ЦФ
Шаг 1	100	SA	100 (1,47)	32,61	2122
Остаточное время = 100–32,61–1,47 = 65,92; классификация задачи при T ₀ = 32,61; Поиск алгоритма, который с минимальными ресурсными затратами достигает порогового значения ЦФ 2113 с учетом ограничения на время выполнения задачи T ₀ < 32,61					
Шаг 2	32,61	PSO	500 (13)	32,61	2089
Остаточное время – 65,92–13 = 52,92, получено распределение задач [2, 9, 6, 7, 8, 0, 3, 1, 4, 5] которое дает значение ЦФ равное 2089, что удовлетворяет заданному условию точности. На поиск решения потрачено 600 вызовов с ухудшением точности решения до 5%, снижение 8,33 раза.					

Заключение

Распределение вычислительных ресурсов в динамичных системах с ресурсно-ограниченными устройствами является актуальной современной задачей. Ограниченность в ресурсах устройств на краю сети, реализующих концепцию Интернета вещей, ставит вопрос о снижении расходования ресурсов как на решение ВЗ, так и на решение задачи РВР. Проведенное исследование показало недостаточность моделей и методов, которые бы учитывали ресурсную стоимость транзита данных маломощными устройствами, равно как и ресурсную стоимость решения задачи РВР, что явилось обоснованием создания ресурсно-ориентированной технологии организации информационного процесса распределения вычислительных ресурсов при интеграции концепций Интернета вещей и краевых вычислений. Основу предлагаемой технологии составляют методы, полученные алгоритмизацией эвристических правил, совместно с использованием репозитория описаний метаэвристических алгоритмов и алгоритмов для расчета ресурсной стоимости транзита данных. Проведенные экспериментальные исследования элементов технологии позволяют сделать вывод о перспективности ее применения для организации распределенных вычислений в ресурсно-ограниченных вычислительных средах.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Ou W., Feng D., Luo K., Chen X. Eco-SLAM: Resource-Efficient Edge-Assisted Collaborative Visual SLAM System. In: *Algorithms and Architectures for Parallel Processing: 23rd International Conference, ICA3PP 2023: Proceedings: Part IV, 20–22 October 2023, Tianjin, China*. Singapore: Springer; 2024. P. 307–324. https://doi.org/10.1007/978-981-97-0859-8_19
2. Li B., Shan H. Offloading Revenue Maximization in Multi-UAV-Assisted Mobile Edge Computing for Video Stream. *IEEE Internet of Things Journal*. 2025;12(8):10866–10875. <https://doi.org/10.1109/JIOT.2024.3513409>
3. Irlon I., Muryanto T., Alvionita A. IoT Security Model with Machine Learning and Edge Computing for Smart Poultry Farm. *Interdisciplinary Social Studies*. 2025;4(3):334–340. <https://doi.org/10.55324/iss.v4i3.876>
4. Sekaran R., Patan R., Al-Turjman F. A Novel Approach for Efficient Packet Transmission in Volunteered Computing MANET. *ACM Transactions on Internet Technology*. 2021;21(4). <https://doi.org/10.1145/3418203>
5. Bartolini N., Coletta A., Maselli G., Prata M. TaMaRA: A Task Management and Routing Algorithm for FANETs. *IEEE Transactions on Mobile Computing*. 2024;23(5):4930–4942. <https://doi.org/10.1109/TMC.2023.3300746>
6. Goel S., Mikek B., Aly J., Arun V., Saeed A., Akella A. A Performance Verification Methodology for Resource Allocation Heuristics. arXiv. URL: <https://arxiv.org/abs/2301.04205> [Accessed 5th June 2025].
7. Buss G., Lee K., Veit D. Scalable Grid Resource Allocation for Scientific Workflows Using Hybrid Metaheuristics. In: *Advances in Grid and Pervasive Computing: 5th International Conference, CPC 2010: Proceedings, 10–13 May 2010, Hualien, Taiwan*. Berlin, Heidelberg: Springer; 2010. P. 256–267. https://doi.org/10.1007/978-3-642-13067-0_29
8. Taleb I., Guillaume J.-L., Duthil B. A Survey on Services Placement Algorithms in Integrated Cloud-Fog / Edge Computing. *ACM Computing Surveys*. 2025;57(11). <https://doi.org/10.1145/3729214>
9. Harjula E., Artemenko A., Forsström S. Edge Computing for Industrial IoT: Challenges and Solutions. In: *Wireless Networks and Industrial IoT: Applications, Challenges and*

- Enablers*. Cham: Springer; 2021. P. 225–240. https://doi.org/10.1007/978-3-030-51473-0_12
10. Барский А.Б. *Параллельные процессы в вычислительных системах. Планирование и организация*. Москва: Радио и связь; 1990. 256 с.
 11. Топорков В.В. *Модели распределенных вычислений*. Москва: Физматлит; 2004. 320 с.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Клименко Анна Борисовна, кандидат технических наук, доцент кафедры фундаментальной и прикладной математики, Российский государственный гуманитарный университет, Москва, Российская Федерация.
e-mail: anna.klimenko@mail.ru
ORCID: [0000-0001-6527-8108](https://orcid.org/0000-0001-6527-8108)

Anna B. Klimenko, Candidate in Engineering Sciences, Associate Professor at the Fundamental and Applied Mathematics department, Russian State University for Humanities, Moscow, the Russian Federation.

Статья поступила в редакцию 28.07.2025; одобрена после рецензирования 26.08.2025; принята к публикации 08.09.2025.

The article was submitted 28.07.2025; approved after reviewing 26.08.2025; accepted for publication 08.09.2025.