

УДК 004.023+004.891.2

DOI: [10.26102/2310-6018/2024.47.4.011](https://doi.org/10.26102/2310-6018/2024.47.4.011)

## Эффективное распределение вычислительных ресурсов в геораспределенных гетерогенных динамичных вычислительных средах

А.Б. Клименко✉

*Институт информационных наук и технологий безопасности Российского государственного гуманитарного университета, Москва, Российская Федерация*

**Резюме.** В настоящее время управление вычислительными ресурсами в геораспределенных гетерогенных динамичных вычислительных средах представляет собой нетривиальную научную задачу. В силу сложности таких систем распределение вычислительных ресурсов становится вычислительно сложной задачей, как правило, многокритериальной, с нелинейными ограничениями, целочисленной или смешанно-целочисленной. Решение подобных задач составляет дополнительные ресурсные издержки систем. Кроме того, свойство геораспределенности также приносит дополнительные ресурсные издержки, которые возникают при транзите данных между вычислительными подзадачами в случае, когда при этом задействованы транзитные участки сети и длина маршрута более одного участка. Целью настоящего исследования является реализация эффективного управления вычислительными ресурсами по критерию использования вычислительных ресурсов – как в процессе их распределения, так и при решении вычислительной задачи в вычислительной среде. Для достижения поставленной цели разработана новая постановка задачи распределения вычислительных ресурсов, которая учитывает свойства гетерогенности, динамики и геораспределенности вычислительной среды и отличается наличием управляемых параметров, определяющих затраты ресурсов как на передачу данных по сети, так и на решение задачи распределения вычислительных ресурсов. Разработан метод, позволяющий решить поставленную задачу, который включает этапы разработки репозитория метаэвристик и его использования. Результаты проведенного моделирования позволяют сделать вывод о перспективности разработанного метода – трудоемкость распределения вычислительных ресурсов снизилась в 28 раз при потерях качества полученного решения до 10 %.

**Ключевые слова:** распределение вычислительных ресурсов, распределенные вычисления, управление распределенными вычислениями, динамичная вычислительная среда, оптимизация, метаэвристики.

**Для цитирования:** Клименко А.Б. Эффективное распределение вычислительных ресурсов в геораспределенных гетерогенных динамичных вычислительных средах. *Моделирование, оптимизация и информационные технологии*. 2024;12(4). URL: <https://moitvvt.ru/ru/journal/pdf?id=1704> DOI: 10.26102/2310-6018/2024.47.7.011

## Efficient resource allocation in geodistributed heterogeneous dynamic computing environments

A.B. Klimenko✉

*Institute of IT and Security Technologies, Russian State University for Humanities, Moscow, the Russian Federation*

**Abstract.** Currently, the management of computing resources in geo-distributed heterogeneous dynamic computing environments is a non-trivial scientific problem. Due to the complexity of such systems, the distribution of computing resources becomes a computationally hard problem, usually multi-criteria,

with nonlinear constraints, integer or mixed-integer. The solution of such problems produces some additional costs of system exploitation. In addition, the property of geo-distribution also introduces additional resource costs that arise during data transit between computing subtasks in the case when transit sections of the network are involved and the route length is more than one section. The purpose of this study is to implement effective management of computing resources based on the criterion of using computing resources – both in the process of their distribution and in solving a computational task in a computing environment. To achieve the goal of the study, a new formulation of the computational resource distribution problem has been developed, which takes into account the properties of heterogeneity, dynamics and geo-distribution of the computing environment and is distinguished by the presence of controlled parameters that determine the resource costs both for data transmission over the network and for solving the computational resource distribution problem. A method has been developed that allows solving the formulated problem, which includes the stages of developing a metaheuristic repository and its use. The results of the conducted modeling allow us to conclude that the developed method is promising – the computing resource usage for resources distribution has decreased by 28 times with a loss in the quality of the resulting solution of up to 10%.

**Keywords:** resource allocation, distributed computing, distributed computing management, dynamic computing environment, optimization, metaheuristics.

**For citation:** Klimenko A.B. Efficient resource allocation in geodistributed heterogeneous dynamic computing environments. *Modeling, Optimization and Information Technology*. 2024;12(4). URL: <https://moitvvt.ru/ru/journal/pdf?id=1704> DOI: 10.26102/2310-6018/2024.47.4.011 (In Russ.).

## Введение

В настоящее время проблема выделения и распределения вычислительных ресурсов (ВР) в геораспределенных гетерогенных динамичных вычислительных средах (ГГДВС) [1] является актуальной задачей. В рамках направления, где задача распределения ВР формализуется как задача оптимизации, сложность современных ГГДВС порождает необходимость введения многих критериев и ограничений, решаемые задачи относятся к задачам комбинаторной оптимизации и к классу пр-сложных и, кроме того, должны решаться с некоторой периодичностью. К настоящему времени сложилось два основных направления распределения ВР в ГГДВС: использование простых эвристик (FCFS, MAX\_MIN, HEFT и др.) [2], в том числе и других алгоритмов, реализующих жадный подход к поиску решения [3], а также использование метаэвристик (имитация отжига, генетический алгоритм, алгоритм роя частиц и др.), которые реализуют итерационный стохастический подход [4].

Динамика современных ГГДВС делает необходимым периодическое решение задачи распределения ВР в ГГДВС: причинами тому является как динамика топологий (группы роботов / БЛА), так и динамика нагрузки (сети связи, различные системы, организованные на основе туманных или краевых вычислений). Таким образом, задача распределения ВР в ГГДВС становится дополнительной вычислительной задачей, многократно выполняемой на одном или нескольких узлах фрагмента сети, составляющем вычислительную среду (ВС), увеличивающей количество различных ресурсов, используемых ГГДВС и, соответственно, стоимость ее эксплуатации.

Географическая распределенность ГГДВС также формирует компоненту эксплуатационной стоимости ГГДВС, поскольку, в отличие от централизованно размещенных ВР, генерирует дополнительную нагрузку на узлы, производящие транзит данных.

Современные постановки задачи распределения ВР в ГГДВС в подавляющем большинстве ориентированы на решение задачи в следующих аспектах:

- учет потребления различных ресурсов и разнообразие комбинаций критериев распределения ВР [5–7];

– поиск эффективных алгоритмов решения отдельных классов задач распределения ВР [8–10].

Работы [11, 12] рассматривают издержки передачи данных между вычислительными задачами, однако ориентированы на отсутствие промежуточных транзитных участков сети. Следует отметить отдельное направление исследований, где рассматриваются только независимые вычислительные задачи (задачи разных пользователей), которые необходимо распределить по вычислительному кластеру / процессорам суперкомпьютера. В этом случае задача формализуется как задача в общем случае  $n$ -мерной упаковки и, соответственно, решается с применением методов, используемых для решения этого класса задач.

Анализ публикаций в области распределения ВР показал, что в современных постановках задачи распределения ВР не учитываются ни издержки на транзит данных, ни выделение ресурсов на собственно решение задачи распределения ВР, в то время как транзит данных порождает дополнительное потребление как времени, так и вычислительных ресурсов, а решение задачи распределения ресурсов порождает ресурсные издержки, а также – в случае слишком длительного времени решения – приводит к необходимости изменения временных ограничений на решение основной вычислительной задачи (ВЗ) и потенциально к потере решения. Таким образом, постановка задачи распределения ВР с учетом перечисленных аспектов в виде управляемых параметров, позволит учесть соответствующие ресурсные издержки, при этом поднимая вопрос о разработке метода решения такой задачи.

## Материалы и методы

### 1. Модель ГГДВС и постановка задачи распределения ВР в ГГДВС

ВС существует в виде последовательности состояний  $S_z$ ,  $z < x$ ,  $z, x = 1, \dots, N$  для дискретных интервалов времени  $t_1, \dots, t_x$ . Переход из состояния  $S_{z-1}$  в состояние  $S_z$  осуществляется посредством реализации управляющего воздействия.

Состояние ГГДВС описывается следующим неориентированным графом:  $G(t_i) = (P(t_i), E(t_i))$ , где  $P(t_i)$  – множество вершин, взвешенных производительностями, в момент времени  $t_i$ ,  $P(t_i) = (p_{1t_i}, \dots, p_{mt_i})$ ,  $E(t_i)$  – матрица смежности, элементы которой равны скоростям передачи данных, в момент времени  $t_i$ ;

$$E(t_i) = \begin{bmatrix} e_{(11)t_i} & \dots & e_{(1m)t_i} \\ \dots & \dots & \dots \\ e_{(m1)t_i} & \dots & e_{(nm)t_i} \end{bmatrix}. \quad (1)$$

В соответствии с условием гетерогенности  $P(t_i)$  и  $E(t_i)$  могут быть разбиты на подмножества по критериям производительности и скорости передачи данных:

$$\{p_{1t_i}, \dots, p_{1t_i}\} \cap \{p_{lt_i}, \dots, p_{mt_i}\} \cap \dots \cap \{p_{zt_i}, \dots, p_{yt_i}\} = \emptyset, \quad (2)$$

$$\{e_{1t_i}, \dots, e_{1t_i}\} \cap \{e_{lt_i}, \dots, e_{mt_i}\} \cap \dots \cap \{e_{zt_i}, \dots, e_{yt_i}\} = \emptyset. \quad (3)$$

Также граф  $\forall i, G(t_i)$  содержит хотя бы один маршрут  $((p_{kt_i}, p_{mt_i}), \dots, (p_{m+1t_i}, p_{zt_i}))$  с количеством ребер  $> 1$ , что реализует свойство геораспределенности.

Далее будем рассматривать зафиксированный момент времени  $t_i$ , когда происходит переход вычислительной среды из состояния  $S_{i-1}$  в  $S_i$ . На одном из вычислительных устройств функционирует программное обеспечение, реализующее планирование ресурсов по принципу реактивного перепланирования. Устройство, на котором функционирует планировщик ресурсов, также может принимать участие в

решении ВЗ, при этом предполагается, что с целью обеспечения отказоустойчивости планировщик может располагаться на любом функционирующем узле (как это реализовано в протоколах ViewStamped Replication, RAFT). Таким образом, процедура распределения ВР является дополнительной ВЗ, потребляющей вычислительные ресурсы и неявно влияющей на значения критериев их распределения – тогда как это не отражено в современных моделях распределения ВР. Графическая модель функционирования планировщика представлена на Рисунке 1.

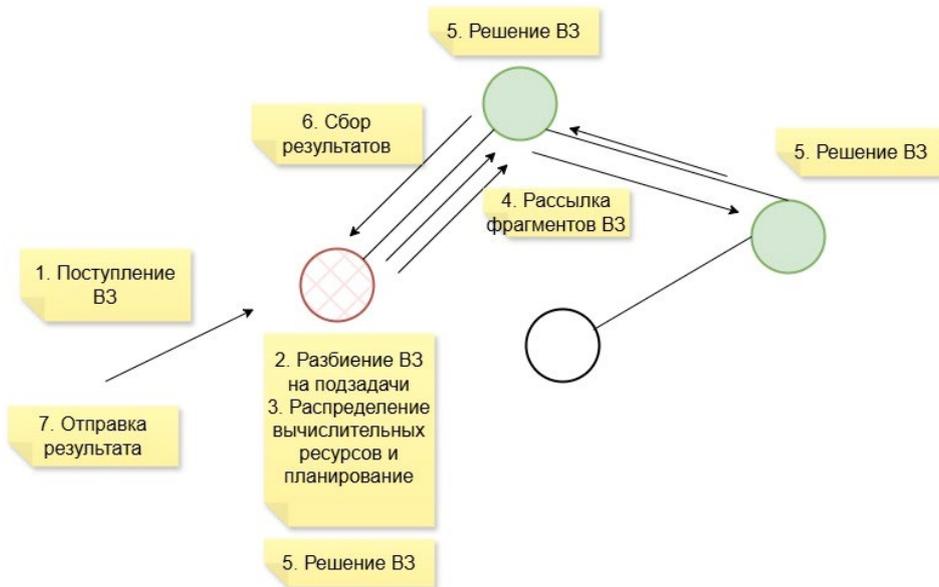


Рисунок 1 – Модель функционирования планировщика вычислительных ресурсов  
 Figure 1 – Computing resource scheduler functioning model

Сформулируем общую постановку задачи распределения ВР в ГГДВС (Постановка 1).

**Неконтролируемые параметры:**

– Множество графов задач, реализующих одну пользовательскую задачу с возможностью выбора для решения:  $G_l = \{G_{ls}\}$ ,  $G_{ls} = \{g_i, R\}$ , где  $G_{ls}$  – ациклический орграф,  $g_i$  – вычислительная сложность задачи;  $R$  – множество ребер, взвешенных значениями объемов передаваемых данных между задачами.

– Граф доступного фрагмента сети представляется неориентированным графом  $G_2 = \{P, C\}$ ,  $P = \{p_j\}$  – вычислительные ресурсы, которыми располагает узел;  $C = \{c_j\}$  – скорости каналов связи.

**Критерии оптимизации:**

– Общие критерии оценивания качества распределения задач  $S_0 = \{s_k\}$ , например, энергопотребление фрагмента сети, для которого производится планирование, или выравнивание нагрузки по вычислительным узлам.

– Критерии оптимизации отдельных узлов  $P_0 = \{p_l\}$ .

– Интегральный критерий оптимизации имеет вид:  $\prod S_k P_l - \alpha(g_r) \rightarrow \max$ , где  $\alpha(g_r)$  представляет штрафную функцию, некоторое ухудшение, привносимое процедурой распределения нагрузки, которые пропорциональны вычислительной сложности алгоритма оптимизации, с неизвестным значением, но достаточно большим, чтобы оказать влияние на  $\prod S_k P_l m_{j\_ост}$ .

**Управляемые параметры:**

– Вычислительная сложность задачи распределения ресурсов  $g_r$ .

– Матрица назначения задач на устройства:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}, \quad (4)$$

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 начала выполнения задачи  $j$  на узле  $i$ .

– Матрица распределения ресурсов вычислительных устройств:

$$U = \begin{bmatrix} u_{11} & \cdots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{n1} & \cdots & u_{nm} \end{bmatrix}, \quad (5)$$

где  $u_{ij}$  – количество вычислительных ресурсов, выделяемых устройством  $i$  для решения задачи  $j$ .

– множество  $Ro$  маршрутов передачи данных между задачами, закрепленными за узлами,  $Ro = \{Ro_j\} = \{\{i\}_{j,j+1}\}, \forall i \leq m, \forall j \leq n$ .

### Ограничения:

Множество ограничений на переменные  $constr = \{constr_k\}$ , включающие:

$$t_{ij} + \frac{g_j}{u_{ij}p_i} + \tau(Ro_j) \leq t_{lj+1},$$

где  $\tau(Ro_j)$  – время передачи данных по маршруту от  $j$  к  $j+1$  задаче;  $i, l$  – узлы, на которых расположены задачи  $j, j+1$ .

$$\forall i: \frac{\sum_{j=1}^n \frac{g_j}{u_{ij}p_i} + \sum_{k=1, k \neq j}^n \frac{r_k}{u_{ij}p_i}}{T} \leq 1,$$

где  $i \in Ro_k$ .

$$m(g_r) + \zeta(g_r) \leq T,$$

где  $m(g_r)$  – время решения ВЗ при вычислительной сложности  $g_r$ ,  $\zeta(g_r)$  – время распределения ВР.

Необходимо: для графов  $G_1$  и  $G_2$  найти такие  $g_r, A, U, Ro$ , чтобы при имеющихся ограничениях:  $constr$ , обеспечить  $\prod S_k P_l m_{j\_ост} - \alpha(g_r) \rightarrow max$ .

## 2. Метод снижения ресурсопотребления распределения ВР в ГГДВС

Эффективное управление ВР в ГГДВС осуществляется посредством применения метода снижения ресурсопотребления распределения ВР в ГГДВС. В рамках проведенного исследования предлагается использование «жадной стратегии», на основании которой, независимо от характеристик узла, где будет производиться принятие решения о распределении ВР, выбирается такая метаэвристика, которая наиболее эффективно решит задачу распределения ВР при известных исходных данных. Таким образом, метод снижения ресурсопотребления распределения ВР в ГГДВС включает два основных этапа: формирование репозитория метаэвристик и его использование.

### Формирование репозитория метаэвристик

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

2. Также пусть имеется разбиение множества задач на классы  $C_j$  по ряду критериев такие, 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  $C_j$  существует такой алгоритм  $a_j \in A$ , для которого справедливо следующее: на оси отсчета вызовов целевой функции (ЦФ), значения  $F_j$  интегральных критериев задач  $c_j$ , объединенных в множество  $C_j$ , подчиняются следующему неравенству:

$$F_{c_x} - \varepsilon \leq F_{c_i} \leq F_{c_y} - \varepsilon,$$

где  $\varepsilon$  – достаточно малая величина,  $x, i, y$  – индексы постановок задач.

4. Для каждого  $C_j$  может быть представлено более одного алгоритма  $a_k$ , 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 ЦФ, где для алгоритма  $a_j$  решающего задачу из подкласса  $C_j$  достигается удовлетворение ограничений задачи  $c_j$ :  $[t\_min\_csp, t\_max\_csp]$ .

На Рисунке 2 представлена базовая диаграмма структуры репозитария алгоритмов.

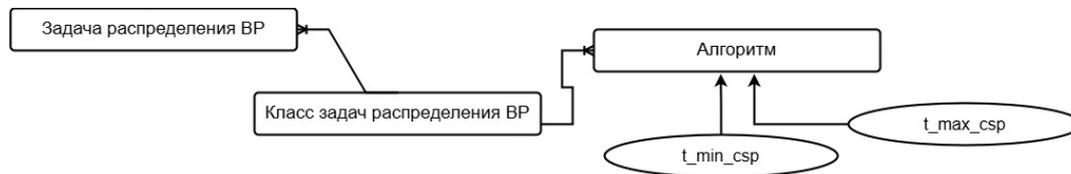


Рисунок 2 – Атрибуты, обозначающие интервал удовлетворения ограничениям задач класса  
 Figure 2 – Attributes describing constraints satisfaction interval for tasks class

На диаграмме Рисунка 3 представлен пример формирования  $t\_min\_csp, t\_max\_csp$  для направленного случайного поиска, при распределении ВР для 15 ВЗ, информационно связанных, по 9 вычислительным узлам. То есть, на основании анализа 100 запусков алгоритма можно ограничить значение  $t\_max\_csp$  150 вызовами ЦФ, тогда как более 75 % результатов получены в пределах 50 первых вызовов ЦФ. Таким образом, полагаем, что 150 вызовов ЦФ в 100 % даст результат, удовлетворяющий ограничениям задачи распределения ВР.

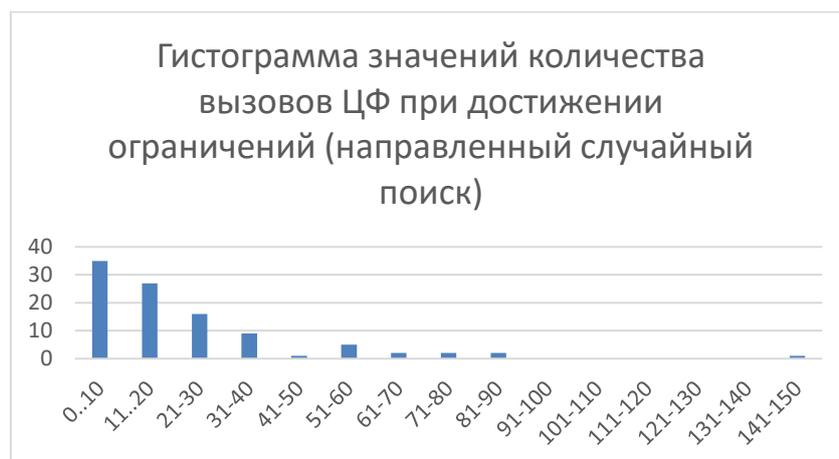


Рисунок 3 – Пример получения  $[t\_min\_csp, t\_max\_csp]$ :  $t\_min\_csp=1, t\_max\_csp=150$  вызовов ЦФ

Figure 3 – The example of  $[t\_min\_csp, t\_max\_csp]$ :  $t\_min\_csp=1, t\_max\_csp=150$  objective function calls

С учетом этого предлагается следующее представление репозитория метаэвристик в виде реляционной базы данных (Рисунок 4).

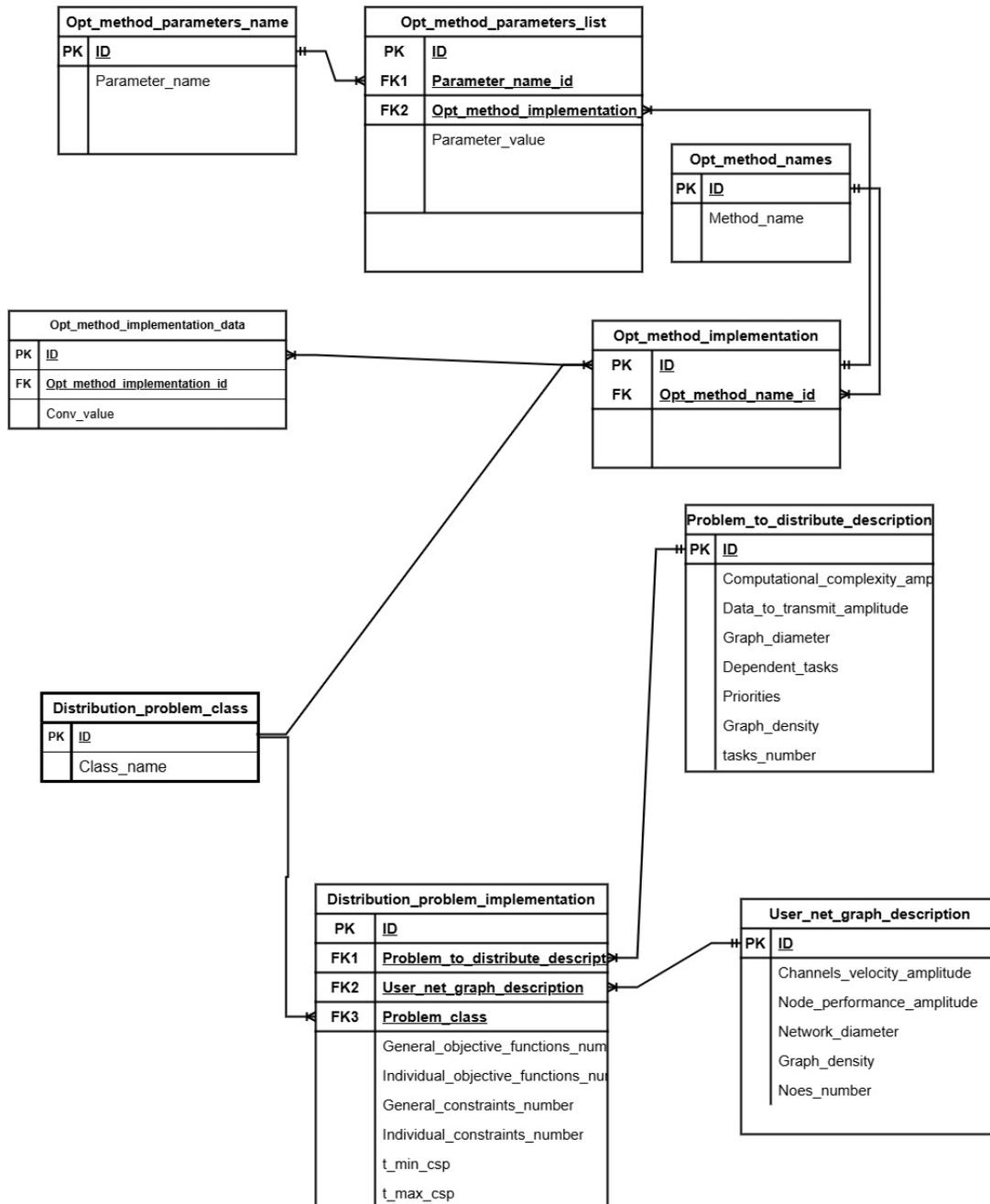


Рисунок 4 – Реализация репозитория метаэвристик в виде реляционной базы данных  
 Figure 4 – The implementation of metaheuristics database as relation one

В приведенной структуре разбиение задач распределения ВР на подмножества предлагается на основании значений таких критериев, как: число ЦФ, число ограничений, принадлежность скорости передачи данных по каналам связи, производительность узлов, диаметр сети, плотность связности графа, число узлов в сети. Подобные критерии использовались в работах [13–15].

### Описание метода сокращения ресурсопотребления распределения ВР в ГГДВС

1. Выбрать  $t_{max\_csp} = \max_{a_i}(t_{max\_csp})$  из репозитория (максимальную верхнюю границу получения решений для любой задачи из  $C$ , решаемой любым алгоритмом, что означает максимальный резерв времени для решения ВР (Рисунок 5).

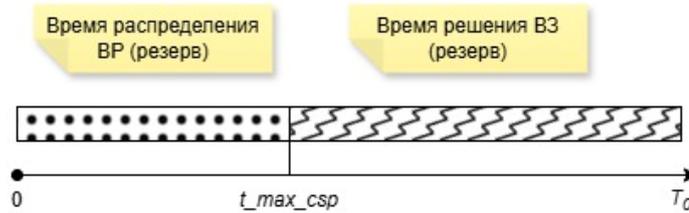


Рисунок 5 – Определение максимально возможной длительности распределения ВР  
 Figure 5 – The statement of maximum of resource allocation procedure

2. Внести  $t_{max\_csp}$  в виде ограничений в постановку задачи в виде:  $\forall i, j t_{ij} \geq t_{max\_csp}$ .
3. Произвести классификацию постановки задачи, относя ее к классу  $C_k, c \in C_k$ .
4. Найти в репозитории алгоритм  $a_i$ , соотнесенного с классом  $C_k$ , для которого значение  $t_{max\_csp}' = \min_{a_i}(t_{max\_csp})$  (Рисунок 6).

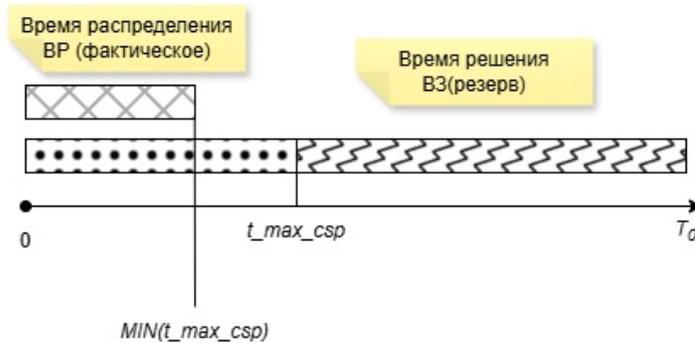


Рисунок 6 – Выбор алгоритма с минимальным значением  $t_{max\_csp}$   
 Figure 6 – The choice of the algorithm with minimum of  $t_{max\_csp}$

5. Получить решение (S, I), применяя алгоритм  $a_i$  на временном интервале  $[0; t_{max\_csp}']$ . При этом происходит сдвиг временной оси: по завершении решения задачи  $t_{max\_csp}'$  становится новым началом оси (Рисунок 7).

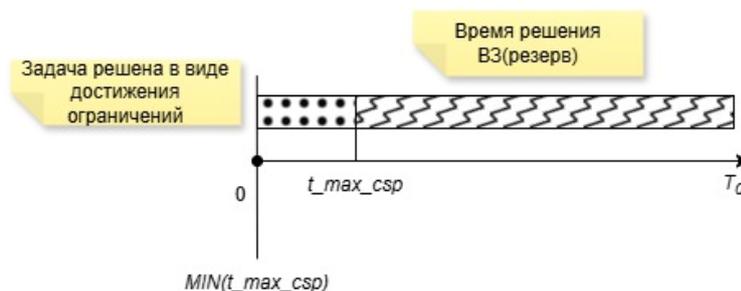


Рисунок 7 – Сдвиг начала временной оси  
 Figure 7 – The shift of time axe

6. Для отрезка времени  $[0; t\_max\_csp]$  выбрать из репозитория такой  $a_i$ , соответствующий классу задачи, установленному в п. 3, для которого на заданном интервале  $[0; t\_max\_csp]$  достигается максимальный прирост интегрального критерия  $I$  и для которого  $t\_min\_csp(a_j) < t\_max\_csp$ . Получить решение  $(S', I')$ .

7. Если  $I' > I$ , то значения параметров  $S'$  являются результатом распределения ВР. В противном случае решением остается  $I$ .

При формировании решения в рамках используемой метаэвристики действует следующий алгоритм:

1. Сформировать случайные значения (4), (5).
2. В соответствии с подлежащим рассмотрению  $G_{Is}$  сформировать случайные  $Ro$ .
3. Для всех  $Ro$  произвести модификацию  $A, U$ : если  $a_{ij} = 0, j \in Ro_k$ , то  $a_{ij} = t_{ij}$ , где  $t_{ij}$  время начала решения вычислительной задачи, порожденной передачей данных по маршруту  $Ro_k$ ;  $u_{ij}$  назначить случайным образом в интервале  $[0...1]$ .

При этом решения, не удовлетворяющие ограничениям задачи распределения ВР в ГГДВС, не будут приняты.

### Результаты

Рассмотрим следующий пример применения метода сокращения ресурсопотребления распределения ВР в ГГДВС в рамках реализации эффективного управления вычислительными ресурсами. На Рисунках 8, 9 представлены граф ВЗ и сети соответственно.

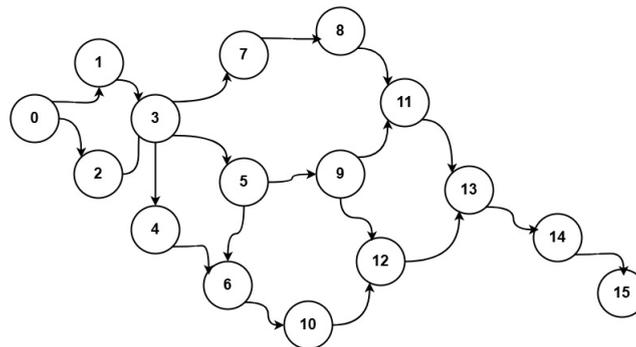


Рисунок 8 – Граф вычислительной задачи  
 Figure 8 – Computing task graph

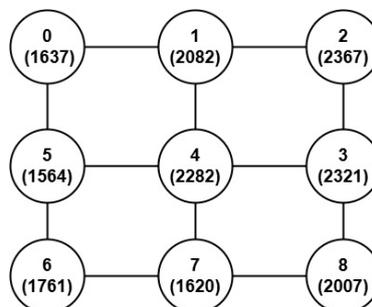


Рисунок 9 – Граф сети  
 Figure 9 – The network graph

Критерии распределения подзадач по сети заданы следующие: суммарное энергопотребление узлов, индивидуальная нагрузка узла (каждый узел пытается создать себе минимальную нагрузку). Ограничения: время выполнения пользовательской задачи

не более 20 с, загруженность узлов в интервале [0,5; 0,9]. За идеальную точность решения примем значение интегрального критерия, равное 0,0001, при размещении задач [6 6 6 6 7 6 6 5 5 6 6 6 6 6 6], которое получено при решении данной задачи за 6000 вызовов ЦФ использования алгоритма PSO.

Репозиторий алгоритмов для решения задачи распределения ресурсов составлен из следующих: направленный локальный поиск (DLS), случайный поиск (RS), оптимизация роем частиц (PSO). По данным репозитория распределение ресурсов реализуется посредством двух этапов разработанного метода (Таблица 1).

Таблица 1 – Результаты применения комплексного метода распределения ВР в ГГДВС  
Table 1 – The results of complex method of resource allocation in geodistributed heterogeneous dynamic computing environment application

	t_csp_max, вызовов ЦФ	Резерв времени	Потрачено времени на шаг	t_csp_max'/ алгоритм	Значение I	Распределение
1	220	16,7	2,1	140/алгоритм роя частиц	$6,02 \times 10^{-5}$	[5 5 5 5 5 5 5 5 5 5 5 5 5 5 5]
2	-	-	1,05	70/алгоритм роя частиц	$9,01 \times 10^{-5}$	[5 5 5 5 5 5 5 5 5 5 5 6 5 5 5]

Итого, 210 вызовов ЦФ в рамках оптимизации роем частиц позволили получить результат на 10 % хуже по точности по сравнению с результатом, полученным за 6000 вызовов ЦФ, вычислительная сложность получения решения уменьшилась в 28 раз. При этом значения, получаемые другими алгоритмами из репозитория, следующие: случайный поиск –  $6,02 \times 10^{-5}$ , направленный случайный поиск –  $5,96 \times 10^{-5}$ .

### Обсуждение

Предлагаемый в данной статье метод ориентирован на отсутствие информации о том, когда должно быть начато выполнение вычислительных задач, подлежащих распределению. По этой причине в репозитории метаэвристик выбирается максимальное значение времени, за которое любой выбранный алгоритм способен достичь решения, удовлетворяющего ограничениям. Затем полученное таким образом ограничение на начало выполнения ВЗ вносится в постановку задачи и таким образом определяется класс ВЗ, для которого в репозитории имеются данные об эффективности метаэвристик. Далее в репозитории выбирается алгоритм, который для ВЗ, соотношенной с некоторым классом, гарантированно находит решение, удовлетворяющее ограничениям, быстрее прочих – таким образом реализуется жадная стратегия с целью минимизации ресурсных затрат на распределения ВР. Оставшееся время используется для возможного улучшения распределения ВР, при этом выбирается тот алгоритм, который за этот отрезок времени достигает наилучшего результата. Таким образом, в условиях, когда неизвестно заранее, сколько времени может быть отведено на распределение ВР, а сколько – соответственно на решение ВЗ, это время устанавливается на основании анализа репозитория метаэвристик. Если отказаться от предварительной установки такого ограничения, то получаем следующее. Пусть  $t_{ij} \geq 0$ . В соответствии с предложенным методом задача будет отнесена к некоторому классу  $C_j$ . Для класса  $C_j$  выбирается  $t_{\max\_csp'} = \min_{a_i}(t_{\max\_csp})$ . При этом делается необходимым изменение ограничения, наложенного на  $t_{ij}$ , а именно  $t_{ij} \geq t_{\max\_csp'}$ , и необходимо снова определить класс  $C_k$ , к которому относится задача. Если  $C_j \neq C_k$ , то осуществляется

поиск в репозитории на предмет нового  $t_{\max\_csp}$ , что снова может привести изменения в ограничение на время начала выполнения ВЗ, и так до тех пор, пока классы задач не совпадут. Очевидно, что это приводит к крайне нежелательному зацикливанию, а в итоге – в случае многократного обращения к репозитарию и, соответственно, сдвигу ограничения на начало выполнения ВЗ таким образом, что будет получено решение низкой точности.

В случае, когда имеется заданное  $t_{ij} \geq T$ , а также желаемая точность результата, из репозитария может быть выбран тот алгоритм, который заданной точности достигает за минимальное количество вызовов ЦФ.

### Заключение

В данной статье с целью реализации эффективного распределения вычислительных ресурсов ГГДВС представлен новый метод снижения ресурсопотребления распределения ВР в ГГДВС, который отличается от существующих использованием репозитария метаэвристик и выбором наиболее эффективной из них, с точки зрения ресурсных затрат на получение решений. Метод включает процедуру создания репозитария метаэвристик и описание способа его использования, позволяя получить существенный (28 раз) эффект снижения трудоемкости процедуры распределения ВР при снижении точности решения на 10 %. Полученные результаты моделирования делают применение разработанного метода перспективным, равно как и дальнейшие исследования в этой области.

### СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Клименко А.Б. Постановка задачи и методы распределения вычислительных ресурсов в геораспределенных гетерогенных вычислительных средах с динамикой и ограничением на время выполнения задач. *Прикладная информатика*. 2024;19(4):48–67. <https://doi.org/10.37791/2687-0649-2024-19-4-48-67>  
 Klimenko A.B. Problem statement and methods for the computing resources distribution in geo-distributed heterogeneous computing environments with dynamics and restrictions on the execution time of tasks. *Journal of Applied Informatics*. 2024;19(4):48–67. (In Russ.). <https://doi.org/10.37791/2687-0649-2024-19-4-48-67>
2. Sukhoroslov O., Gorokhovskii M. Benchmarking DAG Scheduling Algorithms on Scientific Workflow Instances. In: *Supercomputing: Revised Selected Papers: Part II: 9th Russian Supercomputing Days, RuSCDays 2023, 25–26 September 2023, Moscow, Russia*. Cham: Springer; 2023. pp. 3–20. [https://doi.org/10.1007/978-3-031-49435-2\\_1](https://doi.org/10.1007/978-3-031-49435-2_1)
3. Каляев И.А., Каляев А.И. Метод и алгоритмы адаптивного мультиагентного диспетчирования ресурсов в гетерогенных распределенных вычислительных средах. *Автоматика и телемеханика*. 2022;(8):100–122. <https://doi.org/10.31857/S0005231022080062>  
 Kalyaev I.A., Kalyaev A.I. Method and Algorithms for Adaptive Multiagent Resource Scheduling in Heterogeneous Distributed Computing Environments. *Automation and Remote Control*. 2022;83(8):1228–1245. <https://doi.org/10.1134/s0005117922080069>
4. Heba M.F. Optimizing Task Scheduling and Resource Allocation in Computing Environments using Metaheuristic Methods. *Fusion: Practice and Applications*. 2024;15(1):157–179. <https://doi.org/10.54216/FPA.150113>
5. Narwal A. Resource Utilization Based on Hybrid WOA-LOA Optimization with Credit Based Resource Aware Load Balancing and Scheduling Algorithm for Cloud Computing. *Journal of Grid Computing*. 2024;22(3). <https://doi.org/10.1007/s10723-024-09776-0>

6. Hussain M., Nabi S., Hussain M. RAPTS: resource aware prioritized task scheduling technique in heterogeneous fog computing environment. *Cluster Computing*. 2024;27:13353–13377. <https://doi.org/10.1007/s10586-024-04612-2>
7. Behera S.R., Panigrahi N., Bhoi S.K., Sahoo K.S., Jhanjhi N.Z., Ghoniem R.M. Time Series-Based Edge Resource Prediction and Parallel Optimal Task Allocation in Mobile Edge Computing Environment. *Processes*. 2023;11(4). <https://doi.org/10.3390/pr11041017>
8. Dankolo N.M.D., Radzi N.H.M., Mustaffa N.H., Talib M.Sh., Yunus Z.M., Gabi D. Efficient Task Scheduling Approach in Edge-Cloud Continuum Based on Flower Pollination and Improved Shuffled Frog Leaping Algorithm. *Baghdad Science Journal*. 2024;21(2). <https://doi.org/10.21123/bsj.2024.10084>
9. Abdel-Basset M., Mohamed R., Abd Elkhaliq W., Sharawi M., Sallam K.M. Task Scheduling Approach in Cloud Computing Environment Using Hybrid Differential Evolution. *Mathematics*. 2022;10(21). <https://doi.org/10.3390/math10214049>
10. Mishra A.K., Mohapatra S., Sahu P.K. Adaptive Tasmanian Devil Optimization algorithm based efficient task scheduling for big data application in a cloud computing environment. *Multimedia Tools and Applications*. 2024. <https://doi.org/10.1007/s11042-024-19887-1>
11. Барский А.Б. *Параллельное программирование*. Москва: Национальный Открытый Университет «ИНТУИТ»; 2016. 345 с.
12. Топорков В.В. *Модели распределенных вычислений*. Москва: Физматлит; 2004. 320 с.
13. Sadeg S., Hamdad L., Kada O., Benatchba K., Habbas Z. Meta-learning to Select the Best Metaheuristic for the MaxSAT Problem. In: *Modelling and Implementation of Complex Systems: Proceedings of the 6th International Symposium, MISC 2020, 24–26 October 2020, Batna, Algeria*. Cham: Springer; 2020. pp.122–135. [https://doi.org/10.1007/978-3-030-58861-8\\_9](https://doi.org/10.1007/978-3-030-58861-8_9)
14. Kärcher J., Meyr H. A machine learning approach for predicting the best solution heuristic for a large scaled Capacitated Lotsizing Problem. *Research Square*. 2023. <https://doi.org/10.21203/rs.3.rs-3709286/v1>
15. Aksoy İ.C., Mutlu M.M. Comparing the performance of metaheuristics on the Transit Network Frequency Setting Problem. *Journal of Intelligent Transportation Systems*. 2024. <https://doi.org/10.1080/15472450.2024.2392722>

## ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Клименко Анна Борисовна**, кандидат технических наук, доцент, Институт информационных наук и технологий безопасности Российского государственного гуманитарного университета, Москва, Российская Федерация.

*e-mail*: [anna.klimenko@mail.ru](mailto:anna.klimenko@mail.ru)

ORCID: [0000-0001-6527-8108](https://orcid.org/0000-0001-6527-8108)

**Anna V. Klimenko**, Ph.D, Associate Professor of Fundamental and Applied Mathematics Department, Institute of IT and Security Technologies of Russian State University for Humanities, Moscow, the Russian Federation.

*Статья поступила в редакцию 01.10.2024; одобрена после рецензирования 17.10.2024; принята к публикации 21.10.2024.*

*The article was submitted 01.10.2024; approved after reviewing 17.10.2024; accepted for publication 21.10.2024.*