УДК 004.023+004.891.2
DOI: 10.26102/2310-6018/2024.44.1.018

Метод управления вычислительными ресурсами распределенных систем на основе «жадной» стратегии и онтологии эффективных алгоритмов

А.Б. Клименко✉, А.А. Баринов

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

Резюме. В настоящее время управление вычислительными ресурсами в современных системах распределенных вычислений является актуальной проблемой. Эволюция потенциала инфраструктуры привела к тому, что распределенные вычисления могут быть организованы в динамичных гетерогенных и географически распределенных вычислительных средах, примерами которых являются среды «туманные» и «краевые». Динамика как нагрузки, так и топологии подразумевает необходимость смены конфигурации системы, а именно закрепления пользовательских задач за вычислительными устройствами с выделением необходимых ресурсов. Последнее актуализирует вопрос повышения эффективности функционирования планировщика (брокера), обеспечивающего управление ресурсами сети в пределах выделенного фрагмента. Алгоритмическое и программное обеспечение планировщиков основано на моделях и методах теории расписаний и, исходя из постановки задачи, реализует либо простые эвристики, либо методы математического программирования, либо метаэвристики. Однако анализ представленных в открытом доступе постановок задач показал, что они, во-первых, являются частными случаями и реализуют определенные ситуации распределения вычислительных ресурсов, во-вторых, не отражают в полной мере свойств гетерогенности, географической распределенности и динамики вычислительных сред. В рамках данного исследования предложена общая модель задачи распределения вычислительных ресурсов с учетом перечисленных свойств и предложен ее метод решения с использованием предметной онтологии метаэвристических методов. Показана целесообразность построения и применения онтологии на примере анализа эффективности генетических алгоритмов в зависимости от значений параметров решаемой задачи распределения вычислительных ресурсов.

Ключевые слова: онтология, распределение ресурсов, распределенные вычисления, управление распределенными вычислениями, управление ресурсами, оптимизация.

Для цитирования: Клименко А.Б., Баринов А.А. Метод управления вычислительными ресурсами распределенных систем на основе «жадной» стратегии и онтологии эффективных алгоритмов. *Моделирование, оптимизация и информационные технологии.* 2024;12(1). URL: https://moitvivt.ru/ru/journal/pdf?id=1508 DOI: 10.26102/2310-6018/2024.44.1.018

Distributed computing resource management method based on greedy strategy and efficient algorithms ontology

A.B. Klimenko✉, A.A. Баринов

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

Abstract. Currently, managing computing resources in modern distributed computing systems is the relevant problem. As a result of infrastructure capability evolution, distributed computing can be organized in dynamic, heterogeneous and geographically distributed computing environments, examples of which are “fog” and “edge” ones. The dynamics of both load and topology imply the need

to change the system configuration, namely, assigning user tasks to computing devices with the allocation of the necessary resources. The latter raises the issue of increasing the efficiency of the scheduler (broker), which facilitates management of network resources within the allocated fragment. Algorithmic and software schedulers are based on models and methods of scheduling theory and implement either simple heuristics, mathematical programming methods or metaheuristics. However, an analysis of publicly available problem statements has shown that, firstly, they are special cases and implement certain situations of computing resource distribution, and secondly, they do not fully reflect the properties of heterogeneity, geographical distribution and dynamics of computing environments. As part of this study, a general model of computing resource allocation problem is proposed with consideration to the listed properties, and a solution method using the subject ontology of metaheuristic methods is proposed. The feasibility of constructing and applying an ontology is shown using the example of analyzing the effectiveness of genetic algorithms depending on the values of the computing resource allocation problem parameters which is being solved.

Keywords: ontology, resource allocation, distributed computing, distributed computing management, resource management, optimization.

For citation: Klimenko A.B., Barinov A.A. Distributed computing resource management method based on greedy strategy and efficient algorithms ontology. *Modeling, Optimization and Information Technology*. 2024;12(1). URL: https://moitvivt.ru/ru/journal/pdf?id=1508 DOI: 10.26102/2310-6018/2024.44.1.018 (In Russ.).

Введение

Управление вычислительными сетями является комплексной задачей, актуальность которой существенно возросла в последнее десятилетие: появление больших объемов данных, циркулирующих в сети, необходимость в функционирующих распределенных вычислительных системах, появление новых концепций организации вычислений в распределенных средах ставят вопрос о разработке новых моделей и методов управления вычислительными ресурсами таких систем.

Задача управления вычислительными ресурсами имеет достаточно длительную историю: классической постановкой можно считать задачу о составлении расписания для параллельных машин [1] и далее, в направлении распределенных вычислений, модели формировались либо на основании модели задачи об упаковке в контейнеры или полосы [2, 3], либо задача составления расписаний адаптировалась к особенностям распределенных вычислительных сред [4].

Однако в настоящее время происходит интенсификация смещения вычислительной нагрузки к краю сети [5,6], рассматриваются способы размещения нагрузки в группах низколетящих спутников, требуется управление вычислениями в группах мобильных роботов / беспилотных летательных аппаратов (БЛА), что акцентирует внимание на следующих особенностях вычислительных сред:

– относительно высокая динамика нагрузки и / или топологии (низколетящий спутник находится над зоной покрытия от 4 минут [7]);

– становится нежелательным пренебрегать организацией коммуникационной среды в случае информационно связанных задач;

– высокая степень гетерогенности вычислительных устройств и каналов связи требует соответствующего моделирования задачи многокритериальной оптимизации.

Предлагаемая в рамках данного исследования новая общая постановка задачи учитывает перечисленные особенности современных вычислительных сетей, являясь интегральной как в аспекте критериев оптимизации, так и с точки зрения времени изменения конфигурации вычислительной системы. В рамках данного исследования разработан метод повышения эффективности управления вычислительными ресурсами

распределенных гетерогенных сетей с динамикой нагрузки и топологии, который направлен на решение поставленной задачи.

Материалы и методы

Обзор постановок задачи управления вычислительными ресурсами в распределенных системах

В целом, модели задачи управления вычислительными ресурсами в распределенных системах, и, соответственно, методы их решения могут быть классифицированы по следующим признакам (Рисунок 1):

1) наличие информационных связей между задачами, предназначенными к решению, т. е. имеется ли ограничение на следование задач;

2) особенности коммуникационной среды, объединяющей вычислительные устройства;

3) относится ли задача к классу задач параметрической оптимизации или же является задачей структурно-параметрической оптимизации.

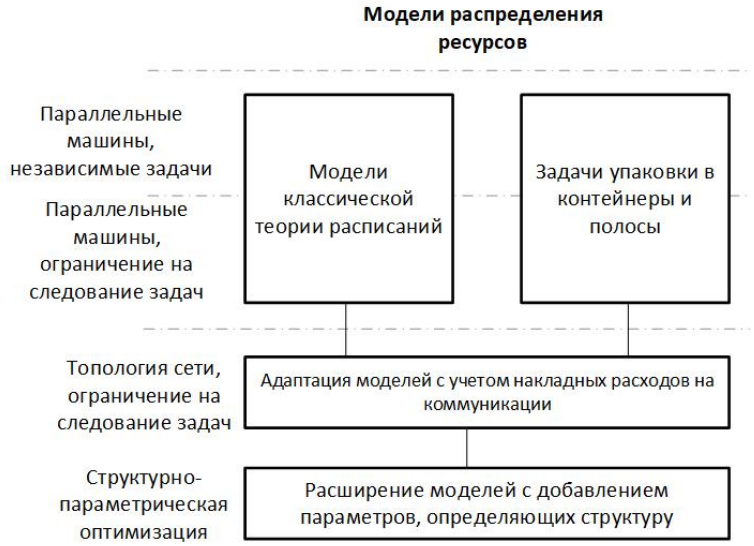

Рисунок 1 – Модели задач планирования / распределения ресурсов
Figure 1 – Scheduling / resource allocation task models

В настоящее время задача управления вычислительными ресурсами, как правило, сводится к задаче планирования / распределения ресурсов / составления расписаний, которая решается либо один раз при распределении ресурсов, если вычислительная среда статична, либо в определенных точках перепланирования с учетом горизонта планирования, либо в рамках политики реактивного перепланирования [8-10].

Среди публикаций в открытом доступе широкий круг работ посвящен задачам, где по независимым параллельным машинам распределяются независимые задачи (ситуация Bag of Tasks), что моделирует ситуацию выполнения множества задач в распределенной среде, принадлежащих разным пользователям и потому независимых друг от друга. Такая ситуация была формализована в рамках теории расписаний [11], и в зависимости от того, допускается ли распределение на машину только одной задачи в любой момент времени, либо нескольких задач, может быть также формализована как пакетная обработка задач (batch jobs), либо как задача упаковки в контейнеры или полосы. Задачи упаковки также весьма многочисленны [12-14], и в некотором роде позволяют реализовать структурно-параметрическую оптимизацию, например,

минимизировать количество заполняемых контейнеров, что напрямую находит применение при консолидации виртуальных машин [15, 16]. Упаковка в полосы подразумевает минимизацию заполнения пространства в фиксированном количестве полос, моделирует выравнивание нагрузки в устройствах и также ориентировано на оптимизацию расходования вычислительных ресурсов в ситуации «множество независимых задач».

Появление ограничений на следование задач также моделируется в рамках классической теории расписаний и задачами упаковки: например, упаковка задач с ограничением на следование [17], распределение вычислительной нагрузки [18], многомерная упаковка, обобщение задач упаковки к гетерогенным ресурсам [19] и др. Структурно-параметрическая оптимизация для описываемого типа входных данных в известных нам работах заключается в подборе архитектуры вычислительной среды минимальной стоимости при заданном ограничении на время выполнения комплекса задач [20].

Если рассматривать ситуации, когда коммуникационной средой нет возможности пренебречь, и необходимо учитывать накладные расходы (время и вычислительные ресурсы) при передаче данных между задачами, то в известных нам работах канал передачи данных обобщается к дополнительному узлу, а задача передачи данных представляется в виде дополнительного узла в графе задач (Рисунок 2).

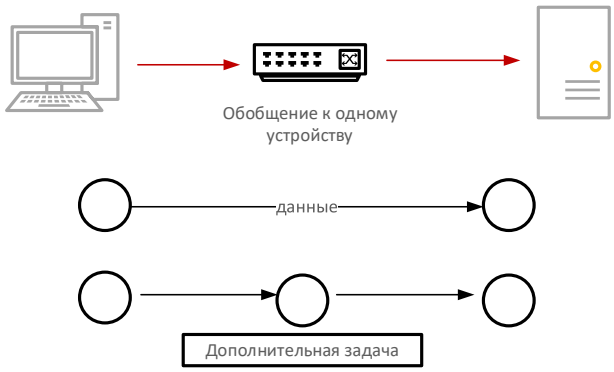

Рисунок 2 – Моделирование транзитной нагрузки
Figure 2 – Transit workload modeling

На Рисунке 1 такие задачи представлены как адаптированные модели, которые перечисленными преобразованиями также могут сводиться к задачам в рамках теории расписаний.

Также следует отметить, что в литературе представлены постановки задач распределения ресурсов с различными критериями оценивания эффективности распределения.

Например, используются следующие критерии [21]:

– время завершения решения комплекса задач;
– стоимость комплекса вычислительных устройств;
– энергопотребление;
– время реакции системы на внешнее воздействие;
– балансировка нагрузки.

Некоторые работы формализуют задачи распределения вычислительных ресурсов как многокритериальные, например, в [22] рассмотрена постановка задачи двухкритериальной оптимизации, где критерии – время выполнения задач и стоимость.

В работе [23] производится оптимизация распределения нагрузки по критериям времени выполнения и энергопотребления.

В [24] рассмотрена постановка задачи с оптимизацией по критериям времени выполнения комплекса задач и балансировки нагрузки.

Проанализируем рассмотренные выше модели с точки зрения таких свойств вычислительной среды, как географическая распределенность, гетерогенность и динамичность.

Географическая распределенность вычислительных сред и участие вычислительных устройств невысокой производительности в передаче данных делают крайне нежелательным пренебрежение транзитной нагрузкой, возникающей при этом, поскольку последняя расходует вычислительные ресурсы. Таким образом, обобщение канала связи к одному узлу становится нецелесообразным.

Гетерогенность вычислительной среды подразумевает наличие многих критериев оценивания качества распределения ресурсов, при этом необходимо допускать ситуацию, когда различные узлы могут иметь собственные критерии и ограничения, что выражается в увеличении количества целевых функций оптимизационной задачи и ограничений.

Динамика вычислительной среды ставит вопрос об оценке процедуры перепланирования, включая физический перенос задач, с точки зрения требующихся для этого ресурсов (процессорных, времени, энергозатраты). Представленные в настоящее время работы не отражают этого важного аспекта, в то время как задача распределения ресурсов, будучи многокритериальной, становится *np*-сложной и может потребовать недопустимо долгое время решения, во-вторых, перенос задач, т. е. пересылка некоторых объемов данных по сети также занимает время и ресурсы. Последнее необходимо учитывать, поскольку в условиях частого перепланирования ресурсы, затрачиваемые на пересылку задач и перепланирование, могут достигать существенных значений.

Таким образом, известные к настоящему моменту модели задач распределения вычислительных ресурсов не учитывают особенностей современных распределенных гетерогенных динамичных сред, что ставит вопрос о формировании новой общей модели задачи распределения вычислительных ресурсов.

Формальная постановка задачи распределения ресурсов

Формализуем задачу распределения ресурсов в общем виде.

1) Имеется граф задач, предназначенных к решению: $G_1 = \{<g_i, r_i, w_i>, R\}$, где g_i – вычислительная сложность задачи, r_i – требования к ресурсам устройства, на котором будет происходить размещение, включая требования к объему памяти, пропускной способности канала, производительности и т. д., w_i – объем данных, который соответствует задаче при передаче ее по сети. Данный граф – ациклический и направленный, где ребра R взвешены объемом передаваемых данных между задачами.

2) Граф сети представляется произвольным направленным мультиграфом, где вершины взвешены характеристиками узлов сети (производительность, объем памяти, энергопотребление и т. д.), ребра взвешены скоростями передачи данных по каналам связи. То есть: $G_2=\{M,C\}$, $M = \{m_j\}$ – ресурсы, которыми располагает узел, $C =\{c_j\}$ – каналы связи.

3) Имеются общие критерии оценивания качества распределения задач $S_0 = \{s_k\}$.

4) Гетерогенность сети и специфика используемых устройств продуцирует индивидуальные критерии качества распределения, специфичные отдельным узлам, и составляет множество: $P_0=\{p_l\}$.

5) Имеются общие ограничения *constr = {constr$_k$}* и ограничения индивидуальные, которые может иметь каждое устройство *constr_ind = {constr_ind$_l$}*.

6) Процедура перепланирования характеризуется параметрами <g_r, r_r, t_r>, где

g_r – вычислительная сложность задачи перепланирования, r_r – требования к ресурсам узла, где выполняется расчет нового закрепления, t_r – время выполнения перепланирования (перемещение данных).

7) Остаточный вычислительный ресурс транзитных узлов после включения их в маршруты передачи данных определяется как:

8) $\forall i, k, L\ m_{i\_\text{ост}} = m_i - \sum_{k=1}^{L} r_k$, где L – количество задач, для которых узел i включен в маршрут передачи данных.

Таким образом, управление системой РВ будет заключаться в решении последовательности экземпляров следующих задач: необходимо для графов G_1 и G_2 найти такое закрепление задач за устройствами, чтобы при имеющихся ограничениях: $r_i <= m_j$, $constr_{,j}$, *constr_ind*, $r_r <= m_j$ обеспечить $S_0 \to max, P_0 \to max, m_{i\_\text{ост}} \to max, \forall i \in \{Ro_\alpha\}$, где $\{Ro_\alpha\}$ – множество маршрутов.

Общий метод решения

Проанализируем возможность решения поставленной задачи. В зависимости от моделируемой ситуации, имеющихся критериев и ограничений она может быть сведена к хорошо известным частным задачам (например, упаковки в *n* контейнеров), для которых существует достаточно широкий круг простых эвристик с низкой сложностью выполнения.

Для общего случая использование классических методов математического программирования представляется проблематичным: во-первых, получение оптимального решения может занять недопустимо долгое время, во-вторых, формирование маршрутов является отдельной и нетривиальной задачей в условиях наличия многих критериев, в-третьих, необходимо таким образом произвести поиск решения, чтобы в итоге получить наилучшее распределение при минимально возможных затратах в аспекте целевых ресурсов на решение задачи планирования и перенос данных.

Рассмотрим процесс смены конфигурации системы с точки зрения потребляемых ресурсов.

Смена конфигурации – это дополнительная задача, которая может возникать в малопредсказуемые моменты времени. Будем считать, что ее решение производится непосредственно на узле, где расположен планировщик, и точно также будем полагать, что на нем всегда достаточно ресурсов для решения задачи планирования в каком-либо виде. Смена конфигурации включает расчет новой конфигурации и пересылку данных, последнее совмещает в себе потребление ресурсов каналов связи и вычислительных ресурсов. Также следует отметить, что задача смены конфигурации имеет трудоемкость, которая является варьируемым параметром и зависит как от выбранного метода расчета новой конфигурации, так и от удаленности тех вычислительных узлов, на которые будут рассылаться задачи. Сделаем допущение о том, что каналы связи имеют неограниченную емкость и в рамках данного исследования будем рассматривать задачу смены конфигурации как задачу с ненулевой трудоемкостью. Анализируя возможные критерии оценивания качества распределения исходных задач, можно утверждать, что трудоемкость задачи, как параметр, влияет на следующие критерии: загруженность устройств, энергопотребление и время выполнения комплекса задач, включая реконфигурацию. Соответственно, варьирование трудоемкостью реконфигурации оказывает влияние на общие критерии эффективности процедуры управления и, уменьшая трудоемкость реконфигурации, может быть гарантировано отсутствие

ухудшения целевых критериев оценивания эффективности распределения ресурсов. Однако вместе с этим уменьшение трудоемкости решения задачи планирования может привести к ухудшению качества решения этой задачи, в чем состоит противоречие (Рисунок 3).

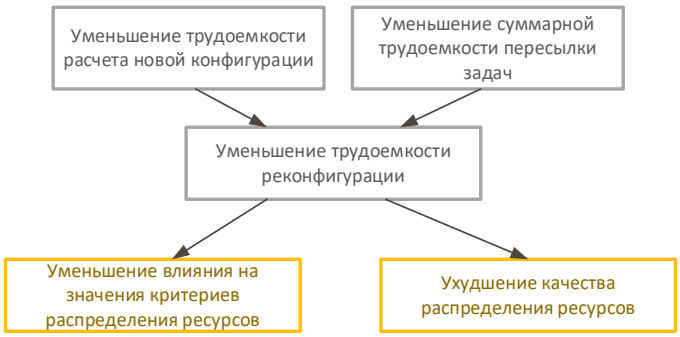

Рисунок 3 – Зависимость качества распределения ресурсов от времени, выделенного на планирование
Figure 3 – Resource allocation quality dependence on planning time

В аспекте предсказуемости соотношения качества получаемых решений и временных затрат на получение этих решений достаточно эффективны разного рода метаэвристики – эволюционные алгоритмы, семейство алгоритмов имитации отжига, оптимизация роем частиц и т. п. Все эти алгоритмы улучшают получаемые решения в зависимости от затрачиваемого времени, однако известен также и факт, что для определенных исходных данных наилучшим образом работают заранее подобранные параметры алгоритмов. Таким образом, оценивая трудоемкость одной итерации такого алгоритма, и имея знания о его эффективности, можно подобрать алгоритм, который позволит получить наилучшее из возможных решений за достаточно короткий интервал времени (Рисунок 4).

Рисунок 4 – Уменьшение противоречия между минимизацией трудоемкости вычислительной процедуры и ухудшением качества получаемого решения
Figure 4 – Contradiction decrease between the computational complexity minimization and the decision quality degradation

Задача распределения ресурсов может быть разбита на ряд подзадач и, руководствуясь «жадной» стратегией, сформируем следующий метод (учитывая допущение о том, что брокер ресурсов обладает информацией о состоянии узлов-кандидатов на размещение задач):

1) Уменьшение количества переносов задач достигается за счет добавления нового критерия оценивания распределения ресурсов: минимум изменений в матрице распределения задач.

Пусть имеется матрица исходного распределения

$A = \begin{vmatrix} a_{ij} & \dots \\ \dots & a_{ij} \end{vmatrix}$, также имеется матрица результирующего распределения $A' = \begin{vmatrix} a'_{ij} & \dots \\ \dots & a'_{ij} \end{vmatrix}$, где $a_{ij} = 1$, если задача i назначена на ресурс j, и $a_{ij} = 0$ в противном случае.

Представим критерий, оценивающий количество переносов задач следующим образом:

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

то есть, к исходному описанию задачи распределения вычислительных ресурсов добавляется следующее: $Tr_{ij} \to \min$.

2) Выбрать алгоритм решения, позволяющий получить лучшее решение за минимальное время в соответствии с описанием задачи распределения.

3) Решить задачу планирования с учетом добавленной ЦФ путем применения выбранного алгоритма за предварительно оцененный интервал времени.

Оценка допустимого для расчетов интервала осуществляется следующим образом:

– имеется возможность оценить максимальное время, за которое могут быть решены все задачи, предназначенные к планированию, а именно: при пессимистичном сценарии все задачи будут решаться последовательно самым низкопроизводительным узлом;

– имеется возможность оценить максимально возможное время пересылки данных при размещении задач: им будет отношение суммарного объема данных, составляющих задачи, к минимальной скорости канала, суммированное с временем, которое будут тратить узлы на выполнение дополнительной транзитной нагрузки самого длинного маршрута к самому слабому узлу-кандидату.

Имея эти расчетные данные, по минимальному времени, которое может быть выделено для решения задачи распределения ресурсов, осуществляется выбор наиболее эффективного алгоритма.

В случае, если сумма пессимистичных оценок времени выполнения двух этапов смены конфигурации превышает ограничение на время выполнения комплекса задач, предлагается выбор алгоритма, который быстрее всего предоставляет какое-либо решение (выбор самого быстрого). При этом вполне очевидно, что самый быстрый алгоритм может предоставить и самое плохое решение.

Результаты

Эффективность различных типов алгоритмов оценивается на основе многоразовых запусков алгоритмов на одних и тех же тестовых задачах. Учитывая стохастический характер большинства метаэвристик, многократные запуски позволяют получить некоторые усредненные значения, которые отражают зависимость качества результата вычислений от трудозатрат. Трудозатраты, в свою очередь, оцениваются количеством вызовов целевых функций.

С целью получения оценочных значений эффективности алгоритмов, были проведены тестовые запуски генетического алгоритма и имитации отжига в различных комбинациях параметров.

Для ГА использовались следующие описания задач (Таблица 1).

Таблица 1 – Описание экспериментальных данных для выбора эффективного алгоритма решения задачи распределения ресурсов
Table 1 – Description of the experiment data for choosing the efficient algorithm for solving the resource allocation problem

Эксперимент 1	Эксперимент 2	Эксперимент 3
(ГА) 300 поколений и 50 популяции (ГА) 300 поколений и 100 популяции (ГА) 300 поколений и 200 популяции **Целевые функции:** Энергопотребление на всех узлах -> min Коэффициент вариации нагрузки на всех узлах -> min Нагрузка на 3, 10 узлах -> min Ограничения: Времени на решение комплекса задач не более 150 Загруженность не более 70 % узла номер 4 Загруженность не более 70 % узла номер 11 Загруженность не более 50 % узла номер 5 Загруженность не более 50 % узла номер 10	(ГА) 300 поколений и 50 популяции (ГА) 300 поколений и 100 популяции (ГА) 300 поколений и 200 популяции **Целевые функции:** Энергопотребление на всех узлах Коэффициент вариации нагрузки на всех узлах Ограничения: Времени на решение комплекса задач не более 150 Загруженность не более 70 % узла номер 4 Загруженность не более 70 % узла номер 11 Загруженность не более 50 % узла номер 5 Загруженность не более 50 % узла номер 10	(ГА) 300 поколений и 50 популяции (ГА) 300 поколений и 100 популяции (ГА) 300 поколений и 200 популяции **Целевые функции:** Нагрузка на 0,3,4,7,8,9,10,11 узлах -> min Передача данных на 1,2,5,6 узлах -> min Ограничения: Времени на решение комплекса задач не более 150 Загруженность не более 70 % узла номер 4 Загруженность не более 70 % узла номер 11 Загруженность не более 50 % узла номер 5 Загруженность не более 50 % узла номер 10

Исходные данные о состоянии фрагмента сети (узлов-кандидатов) показаны на Рисунках 5 и 6.

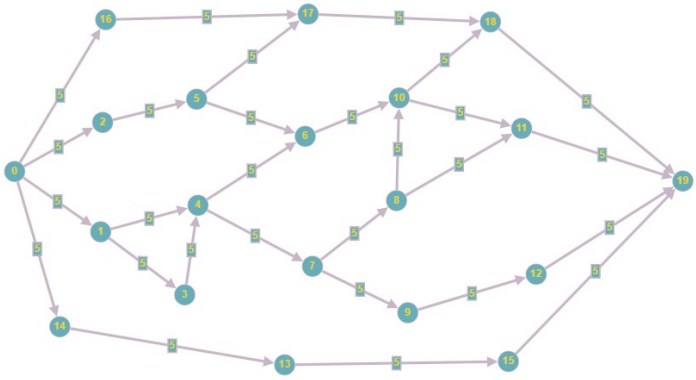

Рисунок 5 – Граф задачи, подлежащей распределению
Figure 5 – Task graph

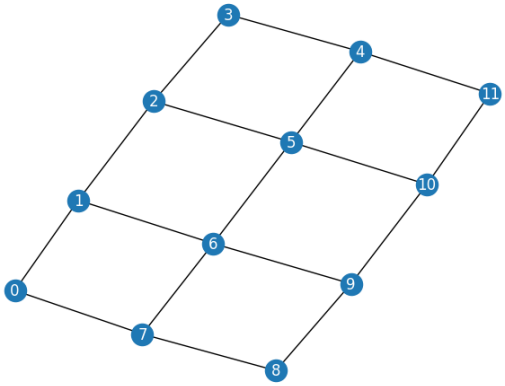

Рисунок 6 – Граф сети
Figure 6 – Network graph

По результатам тестовых прогонов были получены следующие зависимости эффективности алгоритмов от трудозатрат (трудозатраты определены пропорциональными количеству вызовов целевой функции) (Рисунки 7, 8).

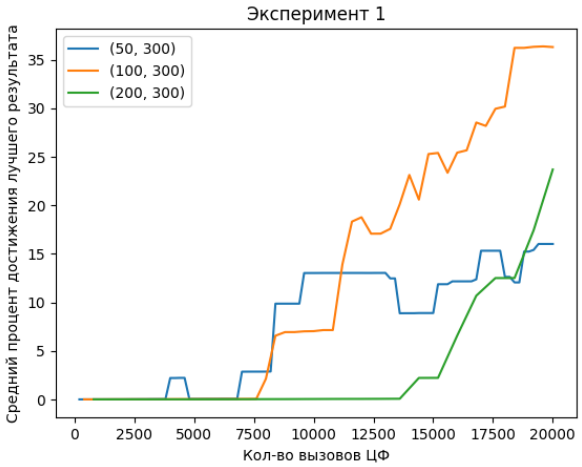

Рисунок 7 – Зависимость эффективности алгоритмов от количества вызовов ЦФ для эксперимента 1
Figure 7 – Simulation 1 algorithm quality dependence on the number of objective function calls

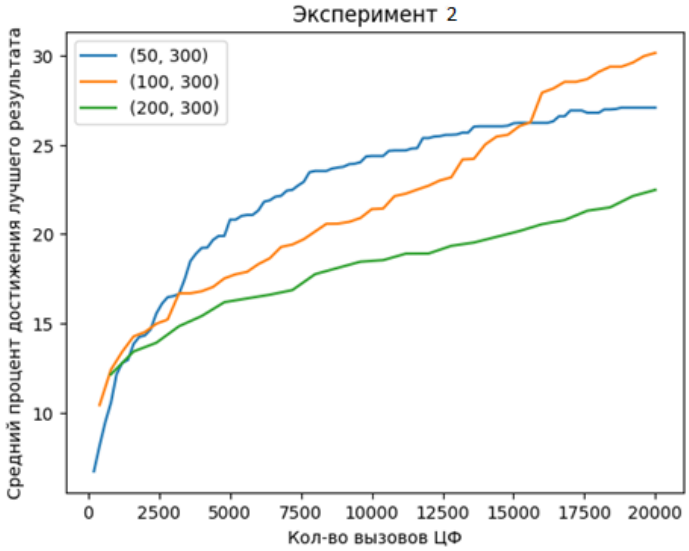

Рисунок 8 – Зависимость эффективности алгоритмов от количества вызовов ЦФ для эксперимента 2
Figure 8 – Simulation 1 algorithm quality dependence on the number of objective function calls

Для эксперимента № 3, где предполагается наличие определенного множества только индивидуальных критериев, улучшение качества результата с каждым новым поколением незначительно, поэтому для задачи с индивидуальными критериями вычислительных устройств использование ГА нецелесообразно (Рисунок 9).

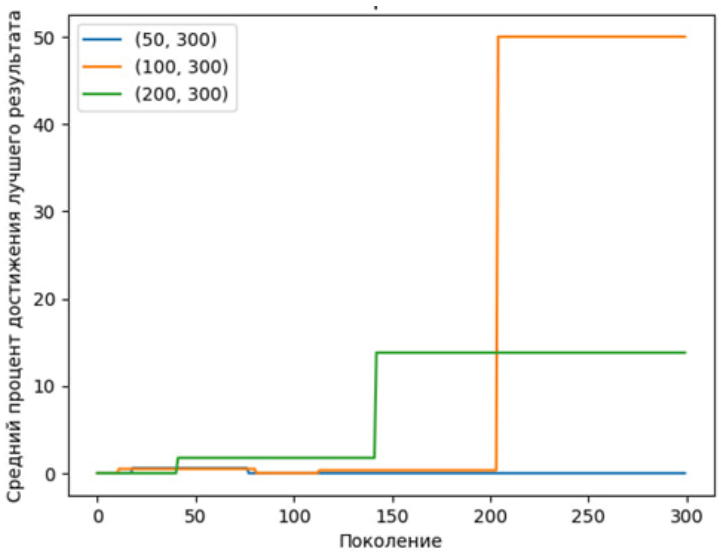

Рисунок 9 – Пример неэффективной работы алгоритмов в заданных условиях
Figure 9 – Inefficient algorithm functioning example

Далее, имея информацию о том, сколько итераций (вызовов ЦФ) может реализовать метаэвристика в условиях пессимистичной оценки имеющегося для расчетов времени, имеется возможность организовать выбор наилучшего в этом смысле алгоритма.

Онтология эффективных алгоритмов решения задач распределения вычислительных ресурсов

Вполне очевидно, что данные об эффективности алгоритмов должны быть каким-то образом организованы с возможностью полной автоматизации выбора в зависимости от исходных данных задачи. Для этой цели может быть использована онтология, которая, во-первых, позволяет формализовать признаки ситуации, для которой осуществляется выбор эффективного алгоритма решения, во-вторых – на основе отношений между экземплярами классов реализовать собственно выбор на основе продукций. Также онтология, являясь основой для базы знаний в предметной области алгоритмов решения задач распределения ресурсов, может включать множественные частные случаи, для которых, как уже говорилось ранее, существуют обширные наборы простых эвристик, которые позволяют реализовать процедуру распределения наиболее быстрым способом.

Основными классами онтологии предметной области решения задач распределения ресурсов является класс «Алгоритмы» и класс «Данные задачи реконфигурации». Внутри каждого класса-алгоритма определены подклассы – признаки, в рамках которых могут быть классифицированы алгоритмы данного типа. Например, алгоритмы имитации отжига различают, в основном, по следующим признакам: закон изменения температуры, порождающее семейство распределений вероятностей, функция принятия нового состояния. Также подклассом класса «Имитация отжига» является «количество вычислений ЦФ», что является мерой вычислительной сложности применяемой реализации метода. На Рисунке 10 показан экземпляр алгоритма Имитации отжига с его отношениями с экземплярами подклассов, описывающих реализацию алгоритма имитации отжига.

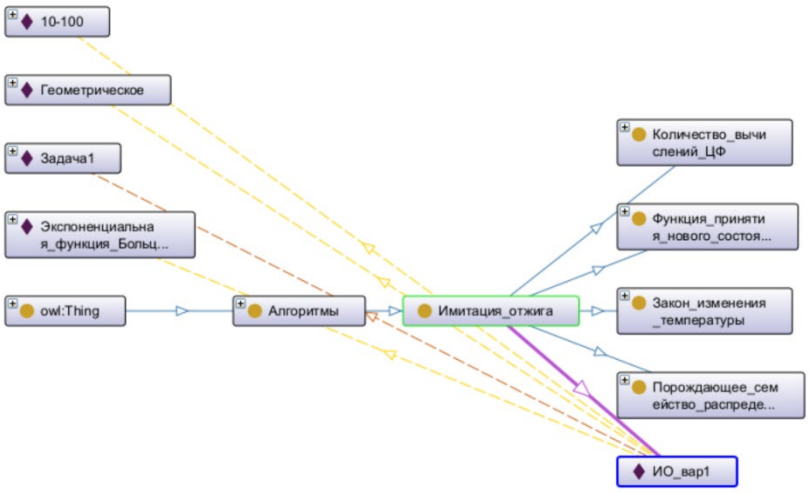

Рисунок 10 – Экземпляр класса алгоритмов имитации отжига
Figure 10 – Simulated Annealing class sample

Класс «Данные задачи реконфигурации» содержит подклассы:
– описание пользовательской задачи,
– описание фрагмента сети,
– число индивидуальных ограничений для устройств,
– число индивидуальных ЦФ,
– число общих ЦФ,
– число общих ограничений.

Подклассы класса «Описание пользовательской задачи» состоят из:
– амплитуды вычислительной сложности,
– амплитуды значений передаваемых объемов данных,
– диаметра графа,
– доминирующая вычислительная сложность,
– доминирующий объем передаваемых данных,
– количество подзадач в пакете,
– наличие информационных связей между задачами,
– наличие приоритетов для несвязанных задач,
– плотность связности графа.

На Рисунке 11 показан фрагмент онтологии, описывающий признаки генетического алгоритма NGSAII и экземпляры алгоритмов с выбранными параметрами, которые тестировались в ходе эксперимента. На Рисунках 12-13 показаны фрагменты онтологии, описывающие отношение между экземпляром задачи и экземпляром эффективного алгоритма.

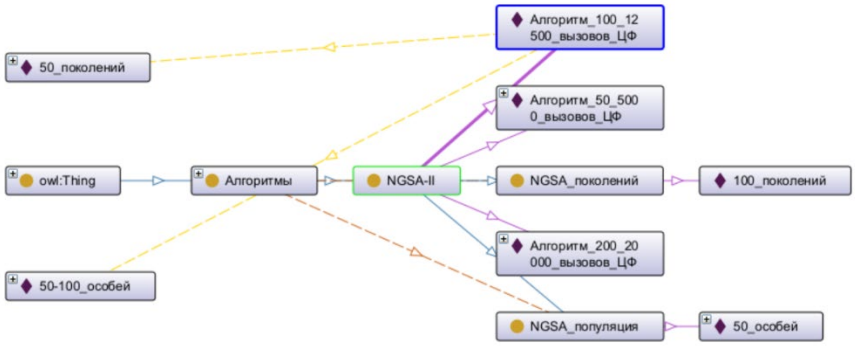

Рисунок 11 – Фрагмент онтологии описания генетического алгоритма NGSAII
Figure 11 – Ontology fragment of the NGSAII description

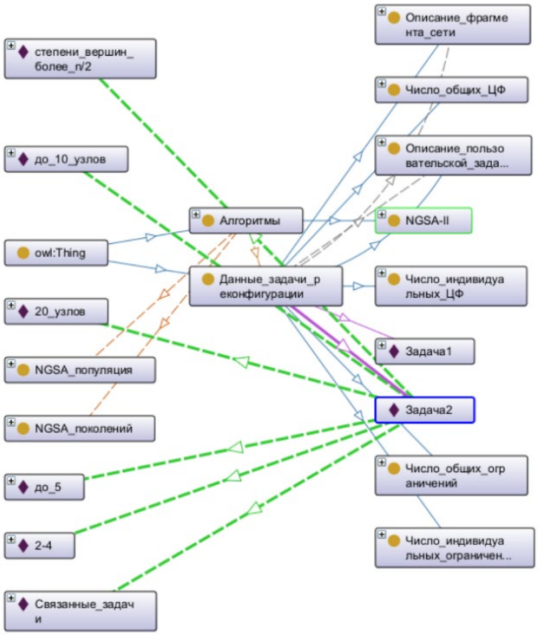

Рисунок 12 – Описание задачи реконфигурации (экземпляр «Задача 2»)
Figure 12 – Reconfiguration task description (Task 2 sample)

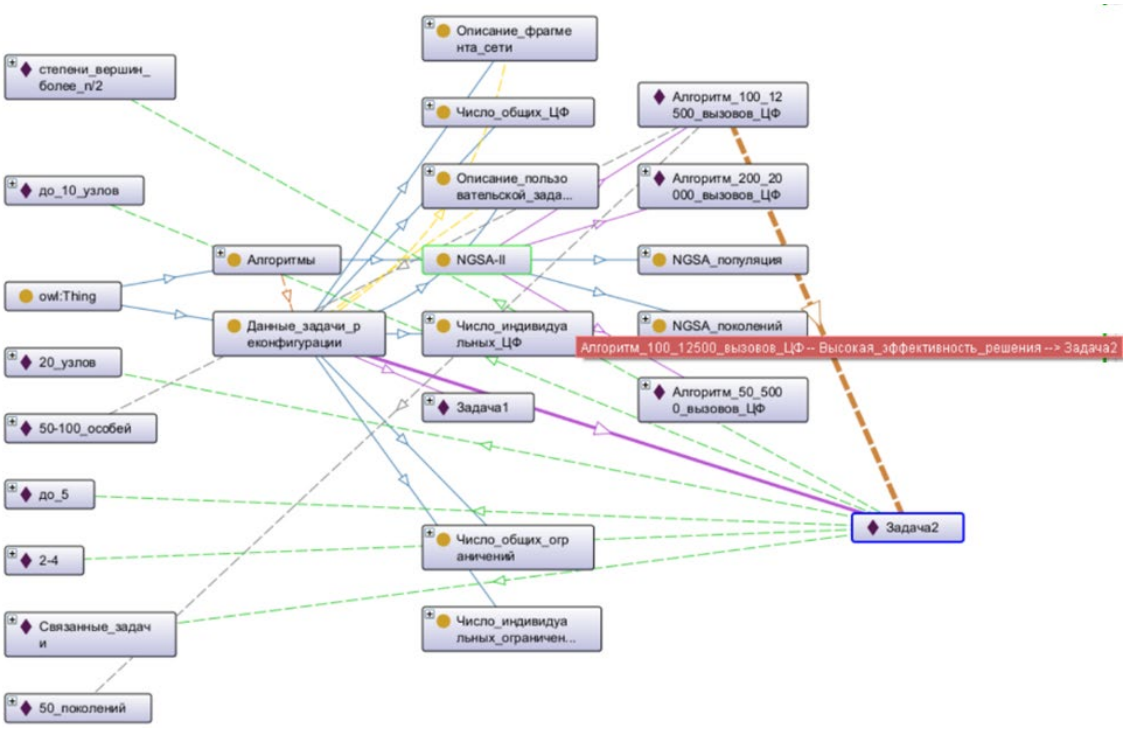

Рисунок 13 – Отношение «Высокая эффективность решения»
Figure 13 – “High efficiency of problem solving” relation.

Обсуждение

Результаты проведенных экспериментов по запускам генетического алгоритма для задач, различных по числу ЦФ и их типам, наглядно иллюстрируют ситуацию, когда для различных входных данных требуется настройка параметров метаэвристик, но в то же время, имея заранее определенное множество типов задач к распределению, могут быть сформированы базы знаний / базы данных, содержащие ограниченное количество алгоритмов, эффективных для определенных исходных данных задачи и на определенном интервале времени. На основе онтологии возможна разработка методов классификации исходных данных с целью выбора наиболее эффективного метода, однако здесь возникает вопрос о целесообразности какого бы то ни было выбора алгоритма, потому что и классификация типа исходных данных, и поиск по базе данных также влечет определенные затраты ресурсов. Этот вопрос требует дополнительного и углубленного исследования. Очевидно, что ответ на него будет зависеть от предметной области применения предложенного в данной статье метода, поскольку целесообразность поиска наиболее эффективного алгоритма оптимизации прежде всего определяется высокой вычислительной сложностью целевых функций.

Заключение

В статье рассматривается проблема управления ресурсами в распределенных гетерогенных и динамичных вычислительных средах, примерами которых являются туманные и краевые вычисления, в том числе, группы роботов, группы БЛА и др.

В настоящее время управление сводится к последовательности задач реконфигурирования, которые, в свою очередь, включают расчет новой конфигурации и пересылку / запуск задач на целевых узлах.

Обзор представленных в открытой печати публикаций показал, что в настоящее время не представлено моделей задачи для общего случая распределения

вычислительных ресурсов, которая бы учитывала распределенный гетерогенный и динамичный характер современных распределенных вычислений.

Модель, представленная в данной статье, обобщает частные постановки задачи распределения ресурсов, и включает новые параметры: суммарную трудоемкость и время реконфигурации, которые зависят от выбора метода расчета новой конфигурации системы и от количества переносимых задач, а также ресурсы устройств, привлекаемых при формировании маршрутов передачи данных. Выбирая «жадную» стратегию, реализуется следующая последовательность решения общей задачи:

– добавление к ЦФ задачи распределения ресурсов ЦФ минимизирующей количество переносов задач относительно исходного их расположения;

– выбор наиболее эффективного алгоритма решения задачи получения нового распределения ресурсов;

– решение.

Экспериментально, на основании ряда тестовых запусков, показано, что для генетического алгоритма NGSA-II существует зависимость качества получаемых решений от трудоемкости и параметров алгоритма. Также представлена структура онтологии, позволяющей автоматизировать процесс выбора наиболее эффективного алгоритма на основании описания задачи распределения ресурсов. Онтологическая модель, составляя основу базы знаний, позволяет в дальнейшем автоматическое создание обучающих выборок для данной предметной области, либо осуществление выбора алгоритмов на основании продукционных правил.

СПИСОК ИСТОЧНИКОВ

1. Pinedo M.L. *Planning and Scheduling in Manufacturing and Services.* 2nd ed. New York, NY: Springer; 2014. 536 p.
2. Han X., Iwama K., Ye D., Zhang G. Strip packing vs. Bin packing. В сборнике: *Third International Conference on Algorithmic Aspects in Information and Management (AAIM'07), 6–8 July 2007, Portland, Oregon, USA.* Heidelberg: Springer; 2007. p. 358–367.
3. Burcea M., Wong P.W.H., Yung F.C.C. Online Multi-dimensional Dynamic Bin Packing of Unit-Fraction Items. In: *CIAC 2013: 8th International Conference on Algorithms and Complexity, 22–24 May 2013, Barcelona, Spain.* Heidelberg: Springer; 2013. p. 85–96.
4. Топорков В.В. *Модели распределенных вычислений.* М.: ФИЗМАТЛИТ; 2004. 320 с.
5. Shaji George A., Hovan George A.S., Baskar T. Edge computing and the future of cloud computing: a survey of industry perspectives and predictions. *Partners Universal International Research Journal (PUIRJ).* 2023;2(2):19–44. DOI: 10.5281/zenodo.8020101.
6. Shaji George A., Hovan George A.S., Baskar T. Unshackled by servers: embracing the serverless revolution in modern computing. *Partners Universal International Research Journal (PUIRJ).* 2023;2(2):229–240. DOI: 10.5281/zenodo.8051052.
7. Brando V., Lovell J., King E., Boadle D., Scott R., Schroeder T. The potential of autonomous ship-borne hyperspectral radiometers for the validation of ocean color radiometry data. *Remote Sensing.* 2016;8(2). URL: https://www.mdpi.com/2072-4292/8/2/150. DOI: 10.3390/rs8020150 (дата обращения: 19.12.2023).
8. Liang Z., Zhong P., Zhang C., Yang W., Xiong W., Yang S., et al. A genetic algorithm-based approach for flexible job shop rescheduling problem with machine failure interference. *Eksploatacja i Niezawodność – Maintenance and Reliability.* 2023;25(4). URL: https://ein.org.pl/A-genetic-algorithm-based-approach-for-flexible-job-shop-

rescheduling-problem-with,171784,0,2.html. DOI: 10.17531/ein/171784 (дата обращения: 19.12.2023).
9. Espinaco F., Henning G.P. Industrial rescheduling approaches: where are we and what is missing? In: *ICPR Americas 2022: International Conference on Production Research – Americas 2022, 23–25 November 2022, Curitiba, Brazil.* Cham: Springer; 2023. p. 461–467.
10. Nair B., Bhanu S.M.S. A reinforcement learning algorithm for rescheduling preempted tasks in fog nodes. *Journal of Scheduling.* 2022;25(5):547–565. DOI: 10.1007/s10951-022-00725-x.
11. Конвей Р.В., Максвелл В.Л., Миллер Л.В. *Теория расписаний.* М.: Наука; 1975. 359 с.
12. Khan A., Lonkar A., Maiti A., Sharma A., Wiese A. Tight approximation algorithms for two dimensional Guillotine strip packing. *arXiv.* 2022.
13. Henrik I.C., Arindam K., Pokutta S., Tetali P. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review.* 2017;24:63–79. DOI: 10.1016/j.cosrev.2016.12.001.
14. Seiden S.S., Woeginger G.J. The two-dimensional cutting stock problem revisited. *Mathematical Programming.* 2005;102(3):519–530. DOI: 10.1007/s10107-004-0548-1.
15. Dow E.M. Decomposed multi-objective bin-packing for virtual machine consolidation. *PeerJ Computer Science.* 2016;2(e47):e47. DOI: 10.7717/peerj-cs.47.
16. Telenyk S., Zharikov E., Rolik O. Consolidation of virtual machines using stochastic local search. In: *CSIT 2017: The International Conference on Computer Science and Information Technologies, 5–8 September 2017.* Cham: Springer; 2018. p. 523–537.
17. Augustine J., Banerjee S., Irani S. Strip packing with precedence constraints and strip packing with release times. *Theoretical Computer Science.* 2009;410(38–40):3792–3803. DOI: 10.1016/j.tcs.2009.05.024.
18. Deppert M.A., Jansen K., Khan A., Rau M., Tutas M. Peak demand minimization via sliced strip packing. *Algorithmica*. 2023;85(12):3649–3679. DOI: 10.1007/s00453-023-01152-w.
19. Bódis A., Csirik J. The variable-width strip packing problem. *Central European Journal of Operations Research.* 2022;30(4):1337–1351. DOI: 10.1007/s10100-021-00772-3.
20. Барский А.Б. *Параллельные информационные технологии.* М.: БИНОМ. Лаборатория знаний; 2007. 502 с.
21. Singh R.M., Awasthi L.K., Sikka G. Techniques for task scheduling in cloud and fog environment: A survey. In: *Second International Conference on Futuristic Trends in Networks and Computing Technologies (FTNCT-2019), 22–23 November 2019, Chandigarh, India.* Singapore: Springer; 2020. p. 673–685.
22. Nguyen B.M., Binh H.T.T., Anh T.T., Son D.B. Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing Environment. *Applied Sciences.* 2019;9(9). URL: https://www.mdpi.com/2076-3417/9/9/1730. DOI: 10.3390/app9091730 (дата обращения: 19.12.2023).
23. Natesan G., Chokkalingam A. Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express.* 2019;5(2):110–114. DOI: 10.1016/j.icte.2018.07.002.
24. Narendrababu Reddy G., Phani Kumar S. Modified ant colony optimization algorithm for task scheduling in cloud computing systems. In: *SCI2018: 2nd International Conference on Smart Computing and Informatics, 27–28 January 2018, Vijayawada, India.* Singapore: Springer; 2019. p. 357–365.

REFERENCES

1. Pinedo M.L. *Planning and Scheduling in Manufacturing and Services.* 2nd ed. New York, NY: Springer; 2014. 536 p.
2. Han X., Iwama K., Ye D., Zhang G. Strip packing vs. Bin packing. В сборнике: *Third International Conference on Algorithmic Aspects in Information and Management (AAIM'07), 6–8 July 2007, Portland, Oregon, USA.* Heidelberg: Springer; 2007. p. 358–367.
3. Burcea M., Wong P.W.H., Yung F.C.C. Online Multi-dimensional Dynamic Bin Packing of Unit-Fraction Items. In: *CIAC 2013: 8th International Conference on Algorithms and Complexity, 22–24 May 2013, Barcelona, Spain.* Heidelberg: Springer; 2013. p. 85–96.
4. Toporkov V.V. *Modely rspredelennih vychisleniy.* Moscow, FIZMATLIT; 2004. 320 p. (In Russ.).
5. Shaji George A., Hovan George A.S., Baskar T. Edge computing and the future of cloud computing: a survey of industry perspectives and predictions. *Partners Universal International Research Journal (PUIRJ).* 2023;2(2):19–44. DOI: 10.5281/zenodo.8020101.
6. Shaji George A., Hovan George A.S., Baskar T. Unshackled by servers: embracing the serverless revolution in modern computing. *Partners Universal International Research Journal (PUIRJ).* 2023;2(2):229–240. DOI: 10.5281/zenodo.8051052.
7. Brando V., Lovell J., King E., Boadle D., Scott R., Schroeder T. The potential of autonomous ship-borne hyperspectral radiometers for the validation of ocean color radiometry data. *Remote Sensing.* 2016;8(2). URL: https://www.mdpi.com/2072-4292/8/2/150. DOI: 10.3390/rs8020150 (accessed on 19.12.2023).
8. Liang Z., Zhong P., Zhang C., Yang W., Xiong W., Yang S., et al. A genetic algorithm-based approach for flexible job shop rescheduling problem with machine failure interference. *Eksploatacja i Niezawodność – Maintenance and Reliability.* 2023;25(4). URL: https://ein.org.pl/A-genetic-algorithm-based-approach-for-flexible-job-shop-rescheduling-problem-with,171784,0,2.html. DOI: 10.17531/ein/171784 (accessed on 19.12.2023).
9. Espinaco F., Henning G.P. Industrial rescheduling approaches: where are we and what is missing? In: *ICPR Americas 2022: International Conference on Production Research – Americas 2022, 23–25 November 2022, Curitiba, Brazil.* Cham: Springer; 2023. p. 461–467.
10. Nair B., Bhanu S.M.S. A reinforcement learning algorithm for rescheduling preempted tasks in fog nodes. *Journal of Scheduling.* 2022;25(5):547–565. DOI: 10.1007/s10951-022-00725-x.
11. Konvej R.V., Maksvell V.L., Miller L.V. *Teoriya raspisaniy*. Moscow, Nauka; 1975. 359 p. (In Russ.).
12. Khan A., Lonkar A., Maiti A., Sharma A., Wiese A. Tight approximation algorithms for two dimensional Guillotine strip packing. *arXiv.* 2022.
13. Henrik I.C., Arindam K., Pokutta S., Tetali P. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review.* 2017;24:63–79. DOI: 10.1016/j.cosrev.2016.12.001.
14. Seiden S.S., Woeginger G.J. The two-dimensional cutting stock problem revisited. *Mathematical Programming.* 2005;102(3):519–530. DOI: 10.1007/s10107-004-0548-1.
15. Dow E.M. Decomposed multi-objective bin-packing for virtual machine consolidation. *PeerJ Computer Science.* 2016;2(e47):e47. DOI: 10.7717/peerj-cs.47.

16. Telenyk S., Zharikov E., Rolik O. Consolidation of virtual machines using stochastic local search. In: *CSIT 2017: The International Conference on Computer Science and Information Technologies, 5–8 September 2017.* Cham: Springer; 2018. p. 523–537.
17. Augustine J., Banerjee S., Irani S. Strip packing with precedence constraints and strip packing with release times. *Theoretical Computer Science.* 2009;410(38–40):3792–3803. DOI: 10.1016/j.tcs.2009.05.024.
18. Deppert M.A., Jansen K., Khan A., Rau M., Tutas M. Peak demand minimization via sliced strip packing. *Algorithmica*. 2023;85(12):3649–3679. DOI: 10.1007/s00453-023-01152-w.
19. Bódis A., Csirik J. The variable-width strip packing problem. *Central European Journal of Operations Research.* 2022;30(4):1337–1351. DOI: 10.1007/s10100-021-00772-3.
20. Barskiy A.B. *Parallel'nye informatsionnye tekhnologii.* Moscow, BINOM. Laboratoriya znaniy; 2007. 502 p. (In Russ.).
21. Singh R.M., Awasthi L.K., Sikka G. Techniques for task scheduling in cloud and fog environment: A survey. In: *Second International Conference on Futuristic Trends in Networks and Computing Technologies (FTNCT-2019), 22–23 November 2019, Chandigarh, India.* Singapore: Springer; 2020. p. 673–685.
22. Nguyen B.M., Binh H.T.T., Anh T.T., Son D.B. Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing Environment. *Applied Sciences.* 2019;9(9). URL: https://www.mdpi.com/2076-3417/9/9/1730. DOI: 10.3390/app9091730 (accessed on 19.12.2023).
23. Natesan G., Chokkalingam A. Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express.* 2019;5(2):110–114. DOI: 10.1016/j.icte.2018.07.002.
24. Narendrababu Reddy G., Phani Kumar S. Modified ant colony optimization algorithm for task scheduling in cloud computing systems. In: *SCI2018: 2nd International Conference on Smart Computing and Informatics, 27–28 January 2018, Vijayawada, India.* Singapore: Springer; 2019. p. 357–365.

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
ORCID: 0000-0001-6527-8108

Anna B. Klimenko, Candidate of Engineering Sciences, Associate Professor at Fundamental and Applied Mathematics Department, Institute of IT and Security Technologies of Russian State University for the Humanities, Moscow, the Russian Federation.

Баринов Арсений Алексеевич, студент, Институт информационных наук и технологий безопасности Российского государственного гуманитарного университета, Москва, Российская Федерация.
e-mail: BArseniyy@yandex.ru

Arseniy A. Barinov, Undergraduate Student, Institute of IT and Security Technologies of Russian State University for Humanities, Moscow, the Russian Federation.

Статья поступила в редакцию 19.01.2024; одобрена после рецензирования 12.02.2024; принята к публикации 05.03.2024.

The article was submitted 19.01.2024; approved after reviewing 12.02.2024; accepted for publication 05.03.2024.