

**Д.Ю. Кравченко, Ю.А. Кравченко, В.В. Курейчик, А.Э. Саак**  
**БИОИНСПИРИРОВАННЫЙ МЕТОД ИМИТАЦИОННОГО**  
**МОДЕЛИРОВАНИЯ ДЛЯ ДИСПЕТЧЕРИЗАЦИИ ПОТОКОВ**  
**ПАРАЛЛЕЛЬНЫХ ЗАЯВОК В GRID- СИСТЕМАХ\***

*Статья посвящена решению задачи диспетчеризации потоков параллельных заявок в пространственно-распределённых вычислительных системах. Актуальность задачи обоснована значительным ростом востребованности парадигмы распределённых вычислений в условиях информационного переполнения и неопределённости. В статье рассмотрены проблемы диспетчеризации заявок пользователей, требующих для своего обслуживания несколько процессоров одновременно, что выходит за рамки классической теории расписаний. Проанализированы аспекты эффективности применения эвристических алгоритмов диспетчеризации планарными ресурсами. Определены причины их недостаточности как по результативности, так и по эмпиричности подходов. Предложено решать задачу диспетчеризации параллельных заявок на основе комплексного применения коалиции интеллектуальных агентов и событийной имитационной модели. Классификацию поступающих на вход заявок предлагается проводить на основе применения модифицированного биоинспирированного метода оптимизации поиском кукушки. Совместное использование коалиции интеллектуальных агентов и биоинспирированного метода позволит обеспечить беспрецедентный параллелизм вычислений, а последующее определение путей обработки классифицированных заявок на основе имитационной модели сформирует наборы альтернативных решений, позволяющих ускорить решение задач и оптимизировать распределение имеющихся вычислительных ресурсов в зависимости от наборов поступающих заявок. Для оценки эффективности предложенного подхода разработан программный продукт и проведены эксперименты с разным количеством поступающих на вход заявок. Каждая поступающая на вход заявка имеет определенный набор атрибутов, являющийся вектором признаков заявки. Степень сходства вектора признаков заявки и эталонного вектора признаков вершины в распределяющей имитационной модели является критерием классификации заявки. Для повышения качества процесса диспетчеризации введены новые процедуры дублирования неклассифицированных заявок, позволяющие интенсифицировать поиск совпадений в векторах признаков. Также предусмотрены резервные траектории диспетчеризации необходимые для обработки прецедентов появления на входах заявок с абсолютным приоритетом. Полученные количественные оценки демонстрируют экономию времени при решении задач относительно большой размерности (от 500000 вершин) не менее 10 %. Временная сложность в рассмотренных примерах составила  $O(n^2)$ . Описанные исследования имеют высокий уровень теоретической и практической значимости и напрямую связаны с решением классических задач искусственного интеллекта, направленных на поиск скрытых зависимостей и закономерностей на множестве больших данных.*

*Диспетчеризация; параллельные вычисления; grid- системы; потоки заявок; биоинспирированный поиск; имитационное моделирование; интеллектуальные агенты.*

**D.Yu. Kravchenko, Yu.A. Kravchenko, V.V. Kureichik, A.E. Saak**  
**BIOINSPIRED SIMULATION METHOD FOR SCHEDULING OF PARALLEL**  
**FLows APPLICATIONS IN GRID-SYSTEMS**

*The article is devoted to solving the problem of parallel requests scheduling flows in spatially distributed computing systems. The relevance of the task is justified by a significant increase in the demand for the distributed computing paradigm in the conditions of information overflow and uncertainty. The article discusses the problems of scheduling user requests that require several*

\* Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов №№ 20-01-00148 и 18-01-00658.

*processors at the same time, which goes beyond the classical theory of schedules. The aspects of the efficiency of using heuristic algorithms for scheduling planar resources are analyzed. The reasons for their insufficiency are determined both in terms of effectiveness and empirical approaches. The paper proposes to solve the problem of scheduling parallel applications based on the integrated application of intelligent agents coalition and an event simulation model. It is proposed to classify incoming applications on the basis of using a modified bio-inspired optimization method for cuckoo search. The joint use of a coalition of intelligent agents and a bio-inspired method will allow for unprecedented parallelism of calculations, and the subsequent determination of the processing classified applications ways on the basis of a simulation model will allow us to form sets of alternative solutions to speed up problem solving and optimize the distribution of available computing resources depending on the sets of incoming applications. To evaluate the effectiveness of the proposed approach, a software product was developed and experiments were conducted with a different number of incoming applications. Each incoming application has a certain set of attributes, which is a vector of the application characteristics. The degree of the application similarity feature vector and the vertex reference feature vector in the distributing simulation model is a classification criterion for the application. To improve the quality of the dispatch process, new procedures for duplicating unclassified applications have been introduced, which allow intensifying the search for matches in feature vectors. It also provides backup dispatching trajectories necessary for processing precedents for the appearance of applications with absolute priority at the inputs. The quantitative estimates obtained demonstrate time savings in solving problems of relatively large dimension (from 500,000 vertices) of at least 10%. The time complexity in the considered examples was  $O(n^2)$ . The described studies have a high level of theoretical and practical significance and are directly related to the solution of classical problems of artificial intelligence aimed at finding hidden dependencies and patterns on a large set of big data.*

*Scheduling; parallel computing; grid-systems; application flows; bio-inspired search; simulation; intelligent agents.*

**Введение.** В настоящее время проблема эффективного диспетчирования параллельных информационных потоков с оптимизацией распределения используемых ресурсов до конца не решена. Не предложен формальный аппарат среды диспетчирования, который позволит выявить закономерности множества заявок, провести их классификацию и предложить эвристические алгоритмы распределения данных заявок по информационным процессам, обеспечивающим параллелизм выполнения и оптимальное использование вычислительных ресурсов. Разработка такого формального аппарата является актуальной задачей.

Рассмотрим базовые архитектуры для диспетчирования задач (job scheduling) и управления ресурсами Grid (Grid resource management systems), предложенные в работах [1–3]. В основе существующих классификаций данными архитектурами являются: централизованная (centralized), иерархическая (hierarchical) и распределённая (decentralized, distributed). В работах [4–9] эти архитектуры систем диспетчирования были названы парадигмами. Централизованная, иерархическая и распределённая структуры различаются способами принятия решения об управлении ресурсами и задачами. Так, в централизованной структуре решение принимается центральным диспетчером, обладающим полной информацией о вычислительных ресурсах и задачах. В иерархической структуре решение принимается иерархическим способом: центральный диспетчер распределяет задачи по локальным диспетчерам, которые производят их распределение в информационных процессах. В распределённой структуре решение принимается локальными диспетчерами, которые обмениваются между собой информацией, принимая решение параллельно.

В данной работе использована комбинированная распределённо-централизованная схема поиска решений при диспетчеризации. Авторами использован многоагентный подход, обеспечивающий необходимую связность субъектов принятия решений при достаточной автономности и целеустремленности каждого интеллектуального агента. Задача классификации поступающих на вход заявок

решается на основе применения коалиции агентов обучения классификатора, функционирующих на основе модифицированного биоинспирированного метода оптимизации поиском кукушки, позволяющего обеспечить максимальное количество итераций локального поиска при достаточном количестве – глобального. Предложена имитационная модель, повышающая эффективность решения задачи построения возможных альтернатив распределения вычислительных ресурсов. После чего централизованное распределение данных альтернативных решений по вычислительным ресурсам осуществляет агент-диспетчер комбинирования классификатора.

**1. Постановка задачи.** Представим распределенную вычислительную систему в виде  $P$  вычислительных подсистем, которые состоят из  $R$  элементарных автоматов. Тогда  $r_k$  элементарных автоматов входит в состав подсистемы  $k \in S = \{1, 2, \dots, P\}$  [5–8]. В отличие от известных постановок подобных задач авторы предлагают закрепить за каждой подсистемой вектор ее атрибутов-признаков  $I^S = \{i_1^S, i_2^S, \dots, i_m^S\}$ , ( $m = 1, \dots, M$ ), где  $i$  – атрибут-признак подсистемы,  $m$  – номер атрибута-признака подсистемы;  $M$  – общее количество атрибутов-признаков в подсистеме. Данный вектор несет информацию о классе или классах обрабатываемых данной подсистемой заявок, о предельных возможностях параллельной обработки, о пропускной способности каналов подсистемы и т.д. Применение вектора атрибутов-признаков позволяет упростить процесс диспетчеризации заявок заменив диспетчеров каждой подсистемы одним агентом-диспетчером комбинирования классификатора.

Входом агента-диспетчера служат выходы обслуживающих аппаратов-переходов  $t_w$  имитационной модели  $Q = \langle C, T, H, O \rangle$ , где  $C$  – множество вершин возможных  $G$  классов входных заявок  $C = \{c_1, c_2, \dots, c_g\}$ , ( $g = 1, \dots, G$ ),  $T$  – множество  $W$  обслуживающих аппаратов-переходов  $T = \{t_1, t_2, \dots, t_w\}$ , ( $w = 1, \dots, W$ ),  $W \leq 2^G$ , задающих множество альтернативных комбинаций параллельной обработки входных заявок для последующего выбора вычислительной подсистемы агентом-диспетчером комбинирования классификатора,  $H$  – входные для аппаратов-переходов ребра связей с вершинами классов заявок,  $O$  – ребро-связь аппарата-перехода и агента-диспетчера. При этом, каждый класс имеет определенный вектор-набор признаков  $K^g = \{k_1^g, k_2^g, \dots, k_z^g\}$ , ( $z = 1, \dots, Z$ ), где  $k$  – атрибут-признак класса,  $g$  – номер класса,  $z$  – номер признака,  $Z$  – общее количество атрибутов-признаков.

Зададим множество входных заявок  $V = \{v_1, v_2, \dots, v_n\}$ , ( $n = 1, \dots, N$ ). Каждая заявка  $v_n$  из  $N$  поступивших на вход системы имеет свой вектор атрибутов-признаков  $J^n = \{j_1^n, j_2^n, \dots, j_e^n\}$ , ( $e = 1, \dots, E$ ), где  $j$  – атрибут-признак заявки,  $n$  – целочисленный номер заявки,  $e$  – номер признака,  $E$  – общее количество атрибутов-признаков заявки. Данный вектор используется для решения задачи классификации заявок с целью определения класса каждой заявки. Параллельную классификацию заявок осуществляет коалиция интеллектуальных агентов обучения классификатора. Кодировка решений и организация механизма поиска реализованы на основе применения модифицированного биоинспирированного метода оптимизации.

Тогда при задании порогового значения  $D_g$  для вхождения заявки в класс  $g$ , целевая зависимость классификации входных заявок примет вид следующего выражения:

$$\forall v_n, c_g: \sum_{e=1}^E \sum_{z=1}^Z j_e^n \cap k_z^g \geq D_g. \quad (1)$$

Прошедшие процедуру классификации заявки заполняют входные вершины имитационной модели, активизируя при этом определенные альтернативные комбинации, распределение которых по вычислительным подсистемам выполняет агент-диспетчер комбинирования классификатора.

Предложенная постановка задачи отличается от существующих применением векторов признаков для классификации входных заявок и биоинспирированной оптимизации, а также распределенно-централизованной схемы поиска, что позволяет с одной стороны, распараллелить процесс классификации, а с другой, уменьшить размерность задачи принятия решений по диспетчеризации на основе применения одного агента-диспетчера комбинирования классификатора.

**2. Разработка метода диспетчеризации потоков параллельных заявок в grid-системах.** Эффективность применения агентных моделей для решения сложных вычислительных задач, отличительной особенностью которых является необходимость построения параллельных информационных процессов, обусловлена наличием необходимых качеств автономности и целенаправленности интеллектуальных агентов одновременно с их высокой связностью.

Интеллектуальный агент является сущностью, получающей информацию о состоянии управляемых ею процессов через систему сенсоров и осуществляющей влияние на них через систему актуаторов [10], при этом поведение агентов рационально, т.к. процессы управляемые ими содействуют достижению определённой цели. Наиболее близким аналогом поведению агентов в живой природе является инстинктивное поведение насекомых. Термин «интеллектуальный» не содержит прямого смысла, а подчеркивает более высокий уровень функций управления у агента по сравнению с примитивными триггерными системами автоматического управления.

Наиболее известными типами интеллектуальных агентов являются следующие:

1. Агенты с простым поведением, которые действуют только на основе текущих знаний и происходящих событий. Их агентская функция поддерживает схему условие-действие.

2. Целенаправленные агенты имеют возможность хранить информацию о прецедентах, которые для них желательны. Это позволяет агенту осуществлять выбор альтернативных траекторий поведения для достижения нужной цели. Целенаправленные агенты стремятся к достижению состояния достигнутой цели.

3. Практичные агенты также способны различать, насколько желанно для них текущее состояние. Такая оценка достигается на основе применения «функции полезности», которая проецирует множество состояний на множество критериев полезности состояний.

4. Обучающиеся агенты независимы и способны к обучению и адаптации к изменяющимся условиям.

В данной работе использован целенаправленный тип интеллектуальных агентов. Коалиция агентов моделирует процесс рассуждений референтной группы в процессе решения задачи классификации потоков параллельных заявок при диспетчеризации в grid-системах. В исследовании решена задача создания классификаторов в виде агентов с заданными условными атрибутами.

Зададим два типа агентов: для организации распределенной классификации входных потоков заявок будут использованы агенты обучения классификатора (АОК); для централизованного распределения вычислительных ресурсов между альтернативными наборами заявок – агент-диспетчер комбинирования классификатора (АДКК). Предлагаемый метод диспетчеризации потоков параллельных заявок включает в себя три основных этапа (рис. 1).

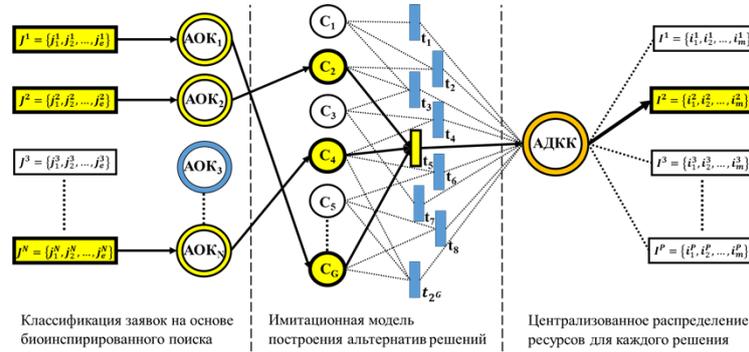


Рис. 1. Структура метода диспетчеризации потоков параллельных заявок

На этапе классификации потоков входных заявок реализацию модифицированного биоинспирированного метода оптимизации поиском кукушки осуществляет коалиция агентов обучения классификатора. Опишем разработку алгоритма классификации потоков входных заявок. Данный алгоритм составляет 1-ый этап реализации метода диспетчеризации потоков параллельных заявок.

В первую очередь формируется коалиция интеллектуальных АОК, размер которой соответствует максимально возможному числу параллельных заявок на входе системы. Также создается агент-диспетчер комбинирования классификатора. Агент обучения классификатора проверяет построенную модель классификации на основе встроенного биоинспирированного метода и определяет наличие элементов, принадлежащих разным классам. Агент-диспетчер комбинирования классификатора объединяет результаты проверок, проведенных агентами обучения классификатора, вносит коррективы в распределение вычислительных ресурсов на основе полученных имитационной моделью альтернатив решений.

Каждый АОК в рамках назначенной ему части пространства решений создает независимую модель классификации с учетом заданных системно значимых признаков, чем предсказывает наличие проблемной области. На этом заканчивается этап предсказания классификации, после чего АДКК производит разбиение обозначенных областей, которые предсказаны агентами обучения классификатора. АОК анализируют характеристики и присваивают определенную метку экземплярам, несоответствующим предварительной классификации, а также отправляют результаты полученной оценки с меткой класса АДКК. АДКК меняет статус класса и проводит необходимые разбиения проблемных областей.

Атрибуты каждой поступившей на вход заявки  $v_n$  присваиваются одному из коалиции АОК. В дальнейшем происходит запуск модифицированного биоинспирированного метода оптимизации поиском кукушки, позволяющего проверить атрибуты поступающих заявок на эквивалентность атрибутам имеющихся классов. По результатам работы биоинспирированного метода полученные заявки распределяются по классам.

Метод, инспирированный поведением кукушек в процессе гнездового паразитизма, отличается от известных аналогов более эффективным способом определения траектории передвижения агента в пространстве поиска. Кукушка свои передвижения осуществляет в виде полетов Леви (Levy flights), реализуемых по следующей канонической формуле (2) с тензорным произведением вектора размера шагов и случайного вектора независимых вещественных чисел.

$$X' = X + U \otimes L_{|X|}(\lambda), \quad (2)$$

где  $\mathbf{U} = (\mathbf{u}_a, a \in [1: |X|])$  – вектор размера шагов по соответствующим компонентам вектора координат местонахождения агента  $X$ ;  $L_{|X|}(\lambda) = (|X| \times \mathbf{1})$  – вектор независимых вещественных случайных чисел, распределенных по закону Леви [11]. Данный способ обеспечивает высокую вероятность малых и достаточную вероятность больших перелетов агента-кукушки (рис. 2), что позволяет интенсифицировать поиск в локальных областях, с одной стороны, и реализовать выход из локальных оптимумов, с другой.

В отличие от канонического метода в данном модифицированном авторы предлагают смоделировать ситуацию, когда в каждом гнезде, куда кукушка прилетает помимо оригинального яйца хозяйки гнезда (анализируемого класса), уже есть чужое подложенное (случайным образом присвоенная входная заявка), которое имеет определенную степень сходства с оригинальным. Задача кукушки – принести такое «яйцо» (не прошедшую процедуру классификации заявку) в случайно выбранную вершину множества классов, степень сходства которого (атрибутов заявки) с оригиналом (атрибутами класса) будет выше, чем у ранее подброшенного.

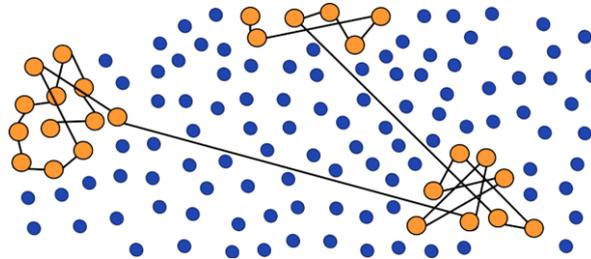


Рис. 2. Иллюстрация способа определения траектории передвижения агента в пространстве поиска

Таким образом, в процессе классификации происходит сравнение значений выражения (1) для случайных и сформированных в процессе поиска пар атрибутов заявка-класс. В процессе решения с заданной вероятностью (свободный параметр метода) периодически удаляется определенное число наихудших полученных решений. При исследовании большого пространства поиска, по мнению авторов, подобные (рис. 2) случайные блуждания (2) являются наиболее эффективными по сравнению с другими более упорядоченными передвижениями агентов, как например, в методе бактериальной оптимизации [12–24].

Полученные результаты классификации на 2-ом этапе выполнения метода диспетчеризации потоков параллельных заявок поступают на вход имитационной модели, построенной на основе применения аппарата сетей Петри (рис. 3). Данная модель на основе комбинирования своих входов  $G$  создает полный спектр  $2^G$  возможных альтернативных решений распределения потоков параллельных заявок между имеющимися ресурсами – вычислительными подсистемами. За соблюдение условий выбора той или иной альтернативы отвечает множество переходов  $T$  имитационной модели. Заполнение необходимой для открытия перехода  $t_n$  комбинации классифицированных заявок происходит по результатам работы модифицированного биоинспирированного метода на 1-ом этапе решения задачи диспетчеризации.

Для повышения качества процесса диспетчеризации предусмотрены процедуры дублирования неклассифицированных заявок, позволяющие интенсифицировать поиск совпадений в векторах атрибутов-признаков. Также предусмотрены резервные траектории диспетчеризации необходимые для обработки прецедентов появления на входах заявок с абсолютным приоритетом. Описанные две дополни-

тельные возможности метода реализованы на основе применения отдельных агентов-диспетчеров комбинирования классификатора  $АДКК_2$  и  $АДКК_3$  (рис. 3). Каждое очередное срабатывание одного из агентов-диспетчеров через определенный для него переход активизирует работу модуля поддержки принятия решений (МПП).

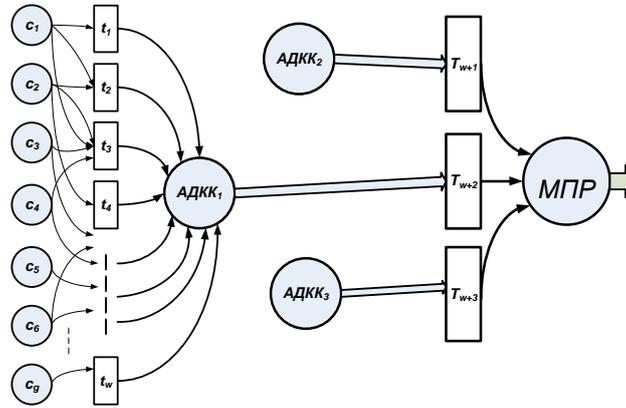


Рис. 3. Имитационная модель построения альтернативных решений

На 3-ем этапе разработанного метода диспетчеризации происходит централизованное распределение  $АДКК_1$  прошедших имитационную модель альтернативных комбинаций решений между вычислительными подсистемами  $P$  соответствующими их атрибутам.

**3. Экспериментальные исследования.** Для подтверждения эффективности разработанного метода диспетчеризации потоков параллельных заявок был создан программный модуль, включающий в себя функции поэтапного решения задач классификации входных заявок на основе биоинспирированного поиска, имитации построения альтернативных вариантов решений и централизованного распределения вычислительных ресурсов. В качестве значений атрибутов применены усредненные абстрактные показатели, полученные случайным образом с заданной вероятностью.

Необходимо отметить, что наибольшее влияние на качество и скорость работы представленного метода оказывает процедура классификации потоков входных заявок. В результате проведенных комплексных исследований получена временная сложность предложенного метода (рис. 4). Временная сложность в рассмотренном примере составила  $O(n^2)$ , где  $n$  – количество анализируемых методом входных данных, которым в данном случае является число входных заявок. В представленной модификации метода биоинспирированной оптимизации поиском кукушки предусмотрено два вложенных цикла прохождения по множествам атрибутов-признаков входных заявок  $J^n$ : первый, для предварительного распределения заявок по классам, а второй, для увеличения значений фитнес-функций эквивалентности и перераспределения заявок. Таким образом, максимально возможное количество операций зависит от числа вершин входных заявок  $N$  как  $n * n$ .

Экспериментальные исследования разработанного метода проводились на вычислительной системе с производительностью порядка  $10^9$  (гигафлопс) операций в секунду. В этом случае, например, при  $n = 50000$  вершин, время выполнения алгоритма не превысит 2,5 секунды, при  $n = 200000$  вершин – 40 секунд, при  $n = 500000$  вершин – 250 секунд, что наглядно иллюстрирует представленная на рисунке 4 полученная зависимость времени выполнения предложенного метода от числа входных вершин.

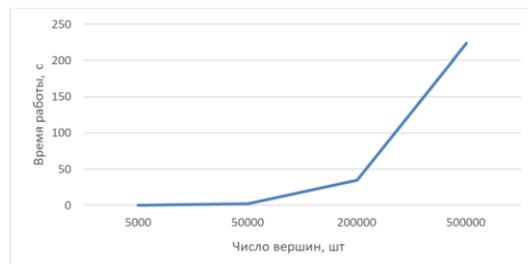


Рис. 4. Временная сложность в зависимости от числа входных вершин

Исследование скорости работы метода биоинспирированной оптимизации поиском кукушки для решения задачи классификации входных заявок в процессе диспетчеризации проводилось в сравнении с каноническим бактериальным методом и методом роя частиц (табл. 1).

Таблица 1

**Сравнение времени работы алгоритмов**

Размерность задачи (количество вершин, шт)	Модифицированный метод поиска кукушки (время работы, с)	Канонический бактериальный метод (время работы, с)	Метод роя частиц (время работы, с)
50000	1.91	2.28	2.37
200000	38.54	47.05	44.87
500000	229.18	313.20	259.33

Из анализа приведенных графика временной сложности и таблицы сравнения времени работы методов видно, что предложенный модифицированный метод показал лучший результат по сравнению с каноническим бактериальным методом и методом роя частиц. Отметим, что при относительно небольших размерностях задачи (от 50000 до 200000 вершин) время работы оцениваемых методов соизмеримо: от 1,91–38,54 секунды в предлагаемой модификации метода кукушкиного поиска до 2,28–47,05 и 2,37–44,87 секунды соответственно в бактериальном методе и методе роя частиц. Наибольший прирост скорости предложенные процедуры интенсификации локального поиска при достаточном количестве глобальных перемещений для выхода из локальных оптимумов позволили достичь при решении задач с размерностью не менее 500000 вершин. Из табл. 1 видно, что в этом случае, если принять за 100 % время работы наиболее медленного последовательного бактериального метода, тогда метод роя частиц затратил на решение этой же задачи 83 % времени от максимального, а предложенный модифицированный – 73 %. Эффективность предложенного модифицированного метода повышена за счет улучшения способа определения траектории передвижения агента в пространстве поиска, позволяющего обеспечить максимальное количество итераций локального поиска при достаточном количестве – глобального.

**Заключение.** В данной работе представлена разработка метода диспетчеризации потоков параллельных заявок в grid-системах. Задача диспетчеризации решена на основе комплексного применения коалиции интеллектуальных агентов и событийной имитационной модели. Использована комбинированная распределенно-централизованная схема поиска решений при диспетчеризации, что позволило уменьшить размерность задачи. Классификация поступающих на вход заявок проводится на основе применения модифицированного биоинспирированного метода

оптимизации поиском кукушки. Совместное использование коалиции интеллектуальных агентов и биоинспирированного метода обеспечивает беспрецедентный параллелизм вычислений, а последующее определение путей обработки классифицированных заявок на основе имитационной модели позволяет формировать наборы альтернативных решений.

Для оценки эффективности предложенного метода разработан программный продукт и проведены эксперименты с разным количеством поступающих на вход заявок. Проведенные экспериментальные исследования подтвердили эффективность разработки в сравнении с каноническим бактериальным методом и методом роя частиц. Полученные количественные оценки подтверждают снижение времени решения задачи диспетчеризации потоков параллельных заявок в grid-системах при помощи разработанного модифицированного метода не менее чем на 10 % по отношению в результатам работы канонических методов роя частиц и бактериальной оптимизации. При относительно небольших размерностях задачи (от 50000 до 200000 вершин) время работы методов соизмеримо, наибольший прирост скорости решения достигается в задачах с размерностью не менее 500000 вершин. Временная сложность в рассмотренных примерах составила  $O(n^2)$ , где  $n$  – количество анализируемых методом входных данных, которым в данном случае является число входных заявок.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Magoulès F., Nguyen T., Yu L. Grid resource management: toward virtual and services compliant grid computing, *Numerical analysis and scientific computing*. CRC Press, UK, 2009.
2. Magoulès F. (ed.). Fundamentals of grid computing: theory, algorithms and technologies, *Numerical analysis and scientific computing*. CRC Press, UK, 2010.
3. Patel S. Survey Report of Job Scheduler on Grids, *International Journal of Emerging Research in Management & Technology*, 2013, No. 2 (4), pp. 115-125.
4. Li M., Baker M. The grid: core technologies. John Wiley & Sons Ltd, England, 2005, 452 p.
5. Saak A.E., Kureichik V.V., Kravchenko Y.A. Scheduling quality of precise form sets which consist of tasks of circular type in GRID systems, *Journal of Physics: Conference Series*, 2018, 1015 (4).
6. Saak A.E., Kureichik V.V., Lezhebokov A.A. Scheduling of parabolic-type tasks arrays in GRID systems, *Advances in Intelligent Systems and Computing*, 2017, pp. 292-298.
7. Saak A., Kureichik V., Kravchenko Y. To scheduling quality of sets of precise form which consist of tasks of circular and hyperbolic type in grid systems, *Advances in Intelligent Systems and Computing*, 2016, pp. 157-166.
8. Saak A.E., Kureichik V.V., Kuliev E.V. Ring algorithms for scheduling in grid systems, *Advances in Intelligent Systems and Computing*, 2015, pp. 201-209.
9. Kravchenko Y.A., Kravchenko D.Y., Kursitys I.O. Architecture and method of integrating information and knowledge on the basis of the ontological structure, *Advances in Intelligent Systems and Computing. 1st International Conference of Artificial Intelligence, Medical Engineering, and Education, AIMEE 2017*. Moscow: 2018, Vol. 658, pp. 93-103.
10. Kravchenko Y.A., Kureichik V.V. Knowledge management based on multi-agent simulation in informational systems, *Conference proceedings. 8<sup>th</sup> IEEE International Conference "Application of Information and Communication Technologies – AICT 2014". – 15-17 October 2014, Astana, Kazakhstan*, pp. 264-267.
11. Payne R.B., Sorenson M.D., and Klitz K. The Cuckoos, Oxford University Press, 2005.
12. Shukran M.A.M., Chung Y.Y., Yeh W.C., Wahid N., and Zaidi A.M.A. Artificial Bee Colony based Data Mining Algorithms for Classification Tasks // *Mod. Appl. Sci.*, 2011, Vol. 5, pp. 217-231.
13. Martens D., De Backer M., Haesen R., Vanthienen J., Snoeck M. and Baesens B. Classification With Ant Colony Optimization // *IEEE Transactions on Evolutionary Computation*, 2007, Vol. 11, No. 5, pp. 651-665.
14. Falco I.D., Cioppa A.D., and Tarantino E. Evaluation of particle swarm optimization effectiveness in classification // *LNAI3849*, 2006, pp. 164-171.
15. Soliman O.S. and Adly A. Bio-inspired algorithm for classification association rules, *8th International Conference on Informatics and Systems (INFOS)*, Cairo, 2012, pp. 154-160.

16. Bova V., Zaporozhets D., and Kureichik V. Integration and processing of problem-oriented knowledge based on evolutionary procedures, *Advances in Intelligent Systems and Computing*, 2016, Vol. 450, pp. 239-249.
17. Semenova A.V. and Kureichik V.M. Ensemble of classifiers for ontology enrichment, *Journal of Physics: Conference Series*, 2018, Vol. 1015, Issue 3, article id. 032123.
18. Kureychik V.M. Overview and problem state of ontology models development, *9th International Conference on Application of Information and Communication Technologies, AICT 2015 - Proceedings 9*, 2015, pp. 558-564.
19. Semenova A.V. and Kureychik V.M. Application of swarm intelligence for domain ontology alignment, *Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry" (ITI'16)*, 2016, Vol. 1, pp. 261-270.
20. Bova V., Kureichik V. and Zaruba D. Heuristic approach to model of corporate knowledge construction in information and analytical systems, *2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, 2016*, pp. 1-5.
21. Kureichik V., Zaporozhets D., and Zaruba D. Generation of bioinspired search procedures for optimization problems, *Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings*, 2016, Vol. 10.
22. Kar A.K. Bio inspired computing - A review of algorithms and scope of applications, *Expert Systems with Applications*, 2016, Vol. 59, pp. 20-32.
23. Zaporozhets D., Zaruba D., and Kulieva N. Parallel approach for bioinspired algorithms, *Journal of Physics: Conference Series Ser. "International Conference Information Technologies in Business and Industry 2018 - Enterprise Information Systems"*, 2018.
24. Bova V.V., Nuzhnov E.V., Kureichik V.V. The combined method of semantic similarity estimation of problem oriented knowledge on the basis of evolutionary procedures, *Advances in Intelligent Systems and Computing*, 2017, Vol. 573, pp. 74-83.

Статью рекомендовал к опубликованию к.т.н. С.Г. Буланов.

**Кравченко Даниил Юрьевич** – Южный федеральный университет; e-mail: kravchenkodaniil122@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; студент.

**Кравченко Юрий Алексеевич** – e-mail: yakravchenko@sfedu.ru; кафедра систем автоматизированного проектирования; доцент.

**Курейчик Владимир Викторович** – e-mail: vkur@sfedu.ru; кафедра систем автоматизированного проектирования; зав. кафедрой.

**Саак Андрей Эрнестович** – e-mail: aesaak@sfedu.ru; 347922, г. Таганрог, ул. Чехова, 22; тел.: +78634393373; кафедра государственного и муниципального управления зав. кафедрой.

**Kravchenko Daniil Yurievich** – Southern Federal University; e-mail: kravchenkodaniil122@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; student.

**Kravchenko Yury Alekseevich** – e-mail: yakravchenko@sfedu.ru; the department of computer aided design; associate professor.

**Kureichik Vladimir Viktorovich** – e-mail: vkur@sfedu.ru; the department of computer aided design; head of department.

**Saak Andrey Ernestovich** – e-mail: aesaak@sfedu.ru; 22, Chehov street, Taganrog, 347922, Russia; phone: +78634393373; the department of state and municipal management; head of department.