

УДК 004.021

DOI: [10.26102/2310-6018/2024.46.3.005](https://doi.org/10.26102/2310-6018/2024.46.3.005)

## Алгоритм перераспределения виртуализированных вычислительных и коммуникационных ресурсов центра обработки данных на основе метаэвристики ACS

Н.Ю. Бумажкина 

*Академия Федеральной службы охраны Российской Федерации, Орёл,  
Российская Федерация*

**Резюме.** Перераспределение виртуализированных вычислительных и коммуникационных ресурсов в центрах обработки данных представляет собой значительную проблему в контексте облачных технологий, осложняя обеспечение стабильного функционирования сервисов. Эти сервисы должны соответствовать критериям качества обслуживания, оценки производительности и условиям контрактов на обслуживание, которые предъявляют поставщики облачных услуг. Главная цель перераспределения виртуализированных вычислительных и коммуникационных ресурсов – оптимальное размещение подмножества активных виртуальных машин на минимальном количестве физических машин с учетом их многомерных потребностей в вычислительных и коммуникационных ресурсах. Что значительно улучшит эффективность работы виртуализированного центра обработки данных. Проблема перераспределения вычислительных и коммуникационных ресурсов центра обработки данных попадает под класс проблем, определяемых как «NP-трудные» проблемы, так как предполагает обширное пространство решений. Поэтому необходимо больше времени для нахождения оптимального варианта. В предыдущих исследованиях ряда таких проблем было доказано, что метаэвристические стратегии позволяют находить приемлемые решения за пригодное время. В статье предлагается использовать модифицированный вариант метаэвристического алгоритма муравьиной колонии для решения задачи перераспределения вычислительных и коммуникационных ресурсов между виртуальными машинами центра обработки данных, рассматриваемой в рамках задачи многомерной векторной упаковки.

**Ключевые слова:** виртуализированные вычислительные и коммуникационные ресурсы, метаэвристические методы, многомерная векторная упаковка, алгоритм оптимизации муравьиной колонии, центр обработки данных.

**Для цитирования:** Бумажкина Н.Ю. Алгоритм перераспределения виртуализированных вычислительных и коммуникационных ресурсов центра обработки данных на основе метаэвристики ACS. *Моделирование, оптимизация и информационные технологии.* 2024;12(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=1620> DOI: 10.26102/2310-6018/2024.46.3.005

## An algorithm for redistributing virtualized computing and communication resources of a data center based on ACS metaheuristics

N.Y. Bumazhkina

*Russian Federation Security Guard Service Federal Academy, Oryol, the Russian Federation*

**Abstract:** The redistribution of virtualized computing and communication resources in data centers is a significant problem in the context of cloud technologies, making it difficult to ensure the stable functioning of services. These services must meet the criteria for quality of service, performance evaluation, and terms of service contracts imposed by cloud service providers. The main goal of the

redistribution of virtualized computing and communication resources is the optimal placement of a subset of active virtual machines on a minimum number of physical machines, taking into account their multidimensional needs for computing and communication resources. Which will significantly improve the efficiency of a virtualized data center. The problem of redistributing computing and communication resources of a data processing center falls under the class of problems defined as "NP-hard" problems, since it involves a vast space of solutions. Therefore, more time is needed to find the optimal option. In previous studies of a number of such problems, it has been proven that metaheuristic strategies make it possible to find acceptable solutions in a suitable time. The article proposes to use a modified version of the ant colony metaheuristic algorithm to solve the problem of redistributing computing and communication resources between virtual machines of a data processing center, considered within the framework of the multidimensional vector packaging problem.

**Keywords:** virtualized computing and communication resources, metaheuristic methods, multidimensional vector packaging, ant colony optimization algorithm, data processing center.

**For citation:** Bumazhkina N.Y. An algorithm for redistributing virtualized computing and communication resources of a data center based on ACS metaheuristics. *Modeling, optimization and information technology*. 2024;12(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1620> DOI: 10.26102/2310-6018/2024.46.3.005 (In Russ.).

## Введение

Одна из основных задач облачных вычислений (ОВ) – предоставление пользователям услуг в соответствии с условиями соглашения о качестве их обслуживания SLA (Service Level Agreement) и требованиям, предъявляемым к нему. Технической основой функционирования ОВ является центр обработки данных (ЦОД) с виртуализированными вычислительными и коммуникационными ресурсами (ВКР). Виртуализация ВКР помогает сбалансировать нагрузку на узлы распределенной схемы ЦОД, учитывая интенсивность пользовательских запросов в разное время и на разных участках. Виртуализация аппаратных ресурсов позволяет эффективно использовать серверные мощности, составляющие технологическую основу ЦОД.

Однако в процессе управления виртуализированными ВКР ЦОД возникают проблемы, связанные с эффективностью формирования и предоставления услуг пользователям. Действия пользователей по запуску, приостановке и завершению различного программного обеспечения (ПО) могут значительно влиять на использование виртуализированных ресурсов, что, в свою очередь, меняет требования виртуальных машин (ВМ) к аппаратным ВКР. Поэтому важны исследования, направленные на динамическое перераспределение виртуализированных ВКР распределенных узлов ЦОД в зависимости от текущих или прогнозируемых пользовательских запросов. Для решения подобных задач широко используется технология живой миграции (ЖМ) ВМ (VM Live Migration). Особенности и типы ЖМ ВМ рассмотрены в работе [1].

Важно отметить, что под ЖМ ВМ понимается перенос состояния ее виртуализированных ВКР между аппаратными ВКР серверных платформ (представленных далее, как множество физических машин (ФМ)), расположенных либо в пределах одного ЦОД, либо на территориально распределенных узлах ЦОД [2].

Разные алгоритмы ЖМ отличаются временными характеристиками и затратами ВКР, что влияет на эффективность функционирования ЦОД. В крупных распределенных ЦОД решение задачи автоматической балансировки нагрузки и динамического потребления ВКР ВМ в процессе их функционирования могут привести к фрагментации множества ВМ по множеству ВКР ФМ. Эта проблема со временем усугубляется в зависимости от интенсивности и характера задач, решаемых пользователями ВМ.

Наиболее известным методом решения проблемы фрагментации ВМ является динамическая консолидация ВМ (VM Consolidation) [3, 4], применение которой позволяет повысить общую утилизацию ВКР ЦОД за счет переноса в реальном времени подмножества активных ВМ между множеством ФМ таким образом, чтобы множество ФМ было разделено на подмножество активных ФМ –  $\Phi_{M_A} \subset \Phi_M$  и подмножество простаивающих ФМ –  $\Phi_{M_D} \subset \Phi_M$ .

Подмножество  $\Phi_{M_A}$  обладает следующими свойствами: каждый элемент подмножества имеет максимальное значение показателя утилизации UR, являющегося отношением значения выделенных ВКР для выполняющихся на нем ВМ к общему значению имеющихся ВКР. Мощность подмножества  $\Phi_{M_A}$  должна быть минимизирована при условиях максимизации показателя  $UR_{\Phi_{M_A}}$  и максимизации числа элементов множества ВМ, размещаемых в элементах подмножества  $\Phi_{M_A}$ , что можно выразить следующим образом:

$$\begin{cases} \min |\Phi_{M_A}| \\ \max UR_{\Phi_{M_A}} \\ \max VM \end{cases} \quad (1)$$

Таким образом, задача консолидации ВМ может быть представлена, как оптимизационная задача перераспределения множества виртуализированных ВКР ЦОД по множеству ФМ, с целью уменьшения количества  $\Phi_{M_A}$ , увеличения задействованных в них ВКР.

В общем случае решение такой задачи можно отнести к задачам, решаемым методом глобальной условной оптимизации. В работах [3, 4] формализацию указанной задачи относят к классу задач комбинаторной оптимизации, для решения которых рассматривается совокупность методов, решающих вариант задачи упаковки контейнеров (англ. bin packing problem) [5] для множеств ВМ и ФМ, решением которой является некоторое оптимальное (в глобальном смысле) размещение подмножества активных ВМ на минимальном количестве ФМ с учетом их многомерных потребностей в ВКР.

Исследование публикаций по тематике решения задачи перераспределения виртуализированных ВКР [6–8] показывает, что предлагаемые решения расширяют рассмотренную выше обобщенную постановку оптимизационной задачи (выражение 1) дополнительным условием, связанным с минимизацией количества их миграций  $N_{M_{VM}}^n$ .

$$\begin{cases} \min |\Phi_{M_A}| \\ \max UR_{\Phi_{M_A}} \\ \max VM \\ \min N_{M_{VM}}^n \end{cases} \quad (2)$$

Очевидно, что подобная постановка задачи не учитывает вариации накладных расходов, возникающих в процессе ЖМ ВМ, которые варьируются в зависимости от:

- текущих ВКР, используемых мигрируемой ВМ;
- текущей пропускной способности канала телекоммуникационной структуры ЦОД, выделяемого для реализации процесса миграции.

Поскольку в классической постановке задача упаковки является однопараметрической (параметр – вес упаковываемого «объекта»), то для учета вариации накладных расходов следует рассматривать многомерную (векторную) постановку задачи упаковки, в которой каждый «объект» (элемент множества ВМ) и «контейнер» (элемент множества ФМ) имеют несколько параметров [9].

## Материалы и методы

Для решения NP-сложных задач дискретной оптимизации, таких как задача векторной упаковки, помимо «жадных» стратегий, находят широкое применение методы, которые не гарантируют точного или оптимального решения, но обеспечивают приемлемый результат. Эти методы эффективны для задач с экспоненциально растущими деревьями поиска, например, задач перераспределения и объединения множества объектов. К подобным задачам относятся задачи, связанные с перераспределением и/или объединением множества VM по множеству FM.

Такие методы относят к метаэвристическим – они выполняют итеративный прямой случайный поиск для получения субоптимального результата, используя простые правила, называемые эвристиками [10]. Основная цель этих методов – эффективное исследование пространства поиска для нахождения субоптимальных решений, по какому-либо показателю (часто оптимизируя время). Метаэвристические методы включают как простейшие алгоритмы локального поиска, так и более сложные процессы, основанные на обучении.

В [11] представлено обобщение операций метаэвристических методов. К таким операциям относятся:

1. Инициализация – выбор метода нахождения некоторого начального решения  $s_0 \in S$ , где  $S$  – пространство решений.

2. Определение окрестности каждого решения  $s \in S$  и множества  $N(s)$  связанных с ними переходов  $\{N^s_1, N^s_2, \dots, N^s_q\}$ .

3. Определение подмножества переходов  $C(s) \subseteq N(s)$  для каждой итерации поиска – кандидатов поиска. В зависимости от выбранного метода подмножество  $C(s)$  может быть фиксированным для всех итераций, или обновляемым при каждой новой итерации поиска.

4. Определение критерия принятия наилучшего решения  $\tilde{s} = \operatorname{argopt}\{f(s, N^s_j); N^s_j \in C(s)\}$ .

5. Определение критерия останова итераций. Обычно показателями для выбора этого критерия являются: время выполнения алгоритма, максимальное количество итераций или темп улучшения решения  $\tilde{s}$  на каждой итерации.

В работе [12] указывается на то, что метаэвристические методы объединяют некоторое подмножество эвристик в рамках более высокоуровневых алгоритмов. К наиболее известным метаэвристическим методам [12] относят:

- локальный поиск с итерациями;
- алгоритм табу-поиска (или вероятностного поиска с запретами);
- эволюционные вычисления, например, генетические алгоритмы;
- метод имитации отжига;
- алгоритмы оптимизации муравьиной колонии и др.

Использование того или иного метаэвристического метода зависит от множества факторов, к основным из которых следует отнести:

- класс оптимизационных задач, для которых может быть адаптирован метаэвристический алгоритм;
- необходимая (достаточная) точность получаемого решения  $\tilde{s}$ ;
- временные характеристики реализации алгоритма;
- вычислительная мощность, требуемая для реализации алгоритма.

Метаэвристические методы успешно применяются для решения задач комбинаторной оптимизации на графах, таких как поиск кратчайшего пути (дерева кратчайших путей), задача коммивояжера (TSP – travelling salesman problem), задача о

назначениях и задача максимального потока [13–15]. Они также хорошо работают для задач упаковки [16, 17]. Исследования показывают, что алгоритмы муравьиной колонии (ant colony optimization – далее АСО) эффективны для двумерной и многомерной упаковки контейнеров [18].

### Основы метаэвристического алгоритма АСО

Методы АСО обобщают эвристику взаимодействия особей некоторых видов муравьев при определении наиболее коротких путей от колонии к источникам пищи [19]. Особенностью такой эвристики является отсутствие централизованных методов обмена информацией о маршрутах движения и организация децентрализованного взаимодействия муравьев между собой с использованием химического соединения – феромона, который каждый муравей оставляет по пути своего следования. Использование феромона позволяет заменить случайное блуждание муравьев вероятностно упорядоченным вдоль путей, отмеченных феромоном. Особенностью феромона, как маркера путей получения решения является возможность его постепенного испарения на малоиспользуемых маршрутах и, наоборот, усиления на маршрутах, являющихся субоптимальным решением.

В общем виде метаэвристика АСО, применительно к решению задач комбинаторной оптимизации, представлена на Рисунке 1.

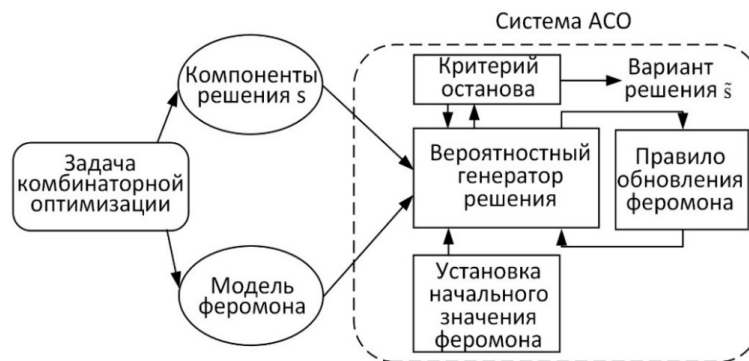


Рисунок 1 – Обобщенное представление метаэвристики Ant Colony Optimization  
Figure 1 – Generalized representation of Ant Colony Optimization metaheuristics

В системе АСО поиск субоптимального решения производится параллельно некоторым множеством  $ИМ = \{ИМ_1, ИМ_2, \dots, ИМ_m\}$  искусственных муравьев (ИМ) – программных агентов, реализующих следующую эвристику поведения реального муравья:

1.  $k$ -й ИМ, находящийся в некотором  $i$ -м узле имеет память  $J_{ik}$  – список узлов по маршруту его следования, которые он может пройти. Выполнив переход в новый узел, ИМ удаляет его из  $J_{ik}$ . По завершении итерации поиска список  $J_{ik}$  очищается для новой итерации. В ряде случаев рассматривается инверсный список узлов, которые ИМ не должен выбирать на новом шаге итерации – *taboo list*.

2. Правило выбора  $k$ -м ИМ нового узла  $j$  для перехода из узла  $i$  на некотором шаге  $t$  является вероятностным и зависит от двух параметров:

- $\eta_{ij}$  – величины значимости ребра, являющейся обратной величине веса ребра  $\omega_{ij}$  (расстояния между узлами  $i$  и  $j$ ) –  $\eta_{ij} = \frac{1}{\omega_{ij}}$ ;
- $\tau_{ij}$  – уровня феромона ребра между узлами  $i$  и  $j$ .



В общем виде вероятность перехода  $a$ -го ИМ из узла  $i$  в узел  $j$  определяется выражением:

$$p_{ij,a}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}(t)]^\beta}{\sum_{l \in J_{ia}} [\tau_{il}(t)]^\alpha \times [\eta_{il}(t)]^\beta} & \text{при } j \in J_{ia}, \\ 0 & \text{в противном случае} \end{cases} \quad (3)$$

где  $\alpha$  и  $\beta$  – эмпирические параметры, определяющие степень влияния уровня феромона. Так при  $\alpha=0$  ИМ стремится выбирать кратчайшее ребро, то есть алгоритм вырождается в «жадную» стратегию, а при  $\beta=0$  – ИМ выбирает ребро, имеющее наибольший уровень феромона  $\tau_{ij}$ .

В начале каждой итерации вероятности перехода ИМ из  $i$ -го узла в соседние узлы равны. Однако с течением времени, вероятность  $p_{ij}$  выбора наиболее короткого пути между узлами  $i$  и  $j$  возрастает, поскольку прирост уровня феромона  $\Delta\tau_{ij,a}$  происходит обратно пропорционально длине  $L_a$  некоторого маршрута  $T_a$  и определяется выражением:

$$\Delta\tau_{ij,a}(t) = \begin{cases} \frac{Q}{L_a} & \text{при } (i,j) \in T_a \\ 0 & \text{в противном случае} \end{cases}, \quad (4)$$

где  $Q$  – эмпирический параметр, определяемый исследователем и имеющий смысл порядка длины оптимального пути.

С целью соблюдения правил эвристики, для путей, не являющихся кратчайшими и вероятность выбора которых стремится к 0, требуется определить правило уменьшения уровня феромона (функция испарения) на шаге  $t+1$ . Оно задается следующим выражением:

$$\tau_{ij}(t+1) = (1-\delta) \times \tau_{ij}(t) + \sum_{a=1}^m \Delta\tau_{ij,a}(t), \quad (5)$$

где параметр  $0 \leq \delta \leq 1$  является коэффициентом испарения, определяющим величину уровня феромона, оставшегося после каждой итерации.

### Решение задачи многомерной упаковки виртуализированных вычислительных и коммуникационных ресурсов с помощью алгоритма ACS

На базе рассмотренных выше и обобщенных этапов метаэвристики АСО был разработан модифицированный алгоритм перераспределения виртуализированных ВКР с целью уменьшения мощности подмножества  $\Phi M_A \subset \Phi M$ . Его схема представлена на рисунке 2. В качестве основы разработанного алгоритма использовался известный вариант АСО, именуемый ACS [20]. Его особенностями являются:

- обновление уровня феромона на каждом ребре выполняется не только в конце итерации алгоритма, но и при каждом переходе ИМ на новый узел. При этом повышение уровня феромона выполняется только на кратчайшем из найденных путей в соответствии с выражением  $\tau_{ij}(t+1) = \delta \times \tau_{ij}(t) + (1-\delta) \times \Delta\tau_{ij}^{opt}(t)$ ;

- правило перехода ИМ на новый узел усложняется за счет выбора в соответствии с однородным распределением и носит название псевдослучайного пропорционального правила, где выбор  $j$ -го узла происходит, либо с вероятностью, определенной в классическом алгоритме AS, либо ИМ безусловно выбирает лучшее ребро, с точки зрения его длины и уровня феромона.

В качестве входных данных задаются: множество ФМ ( $F$ ), множество ВМ ( $V$ ), множество ИМ ( $A_{ИМ}$ ), список ФМ ( $L_F$ ), список ВМ ( $L_V$ ), список ИМ ( $L_{ИМ}$ ) показатель

уровня феромона для каждой пары ВМ-ФМ  $(v, F) - \tau_{vF,a}$ , параметры алгоритма  $\delta, \beta, \omega, q_0$ .

**Шаг 1 Инициализация параметров.** Алгоритм начинается с инициализации необходимых параметров, включая количество ИМ, количество циклов завершения (nCycleTerm) и параметры алгоритма  $(\beta, \omega, \delta, q_0)$ .

Он также устанавливает начальное значение феромона  $(\tau_{vF,a})$  равным  $\tau_0$  и инициализирует матрицу миграции ВМ  $(MM_G)$ , с наиболее оптимальным размещением ВМ по ФМ, пустым решением.

**Шаг 2 Итерация ИМ.** В начале каждой итерации алгоритма каждый  $a$ -й ИМ запускается с пустой  $MM_a$ , набором пустых  $L_F$ , имеющих общую емкость ресурсов, и набор  $L_V$ , имеющих общие потребности в ресурсах, и рандомизирует ВМ в  $L_V$  [блок 7] для внесения рандомизации в последующий процесс поиска.

Необходимо отметить, что при назначении ВМ для  $F$  ИМ учитывают, где в данный момент размещены ВМ, и соответствующие издержки миграции, определяемые как накладные расходы на процесс живой миграции ВМ, учитываются при принятии решений о миграции.

**Шаг 3 Генерация матрицы миграции ВМ.** На данном этапе все ИМ генерируют свои  $MM_a$ , используя псевдослучайное правило пропорционального выбора  $s$ :

$$s = \begin{cases} \operatorname{argmax}_{v \in LV(F_a)} \{ \tau_{v,F} \times [\eta_{v,F}]^\beta \} & \text{при } q \leq q_0, \\ S & \text{в противном случае} \end{cases} \quad (6)$$

где  $q$  – случайное число, равномерно распределенное в диапазоне  $[0, \dots, 1]$ ;  $q_0$  – параметр, удовлетворяющий неравенству  $0 \leq q_0 \leq 1$  и определяющий важность функции выбора пары значений параметров  $v$  и  $F$  в сравнении с функцией продолжения поиска подходящей пары;  $\eta_{v,F,a}$  – величина вектора дисбаланса ресурсов;  $\beta$  – параметр, имеющий неотрицательное значение и определяющий относительную важность параметра  $\tau_{v,F,a}$  относительно значения  $\eta_{v,F,a}$ ;  $S$  – случайная величина, выбранная в соответствии со следующим распределением вероятности  $p_a(vF, a)$  включения ВМ в  $MM_a$ :

$$p_a(vF, a) = \begin{cases} \frac{\tau_{v,F} \times [\eta_{v,F}]^\beta}{\sum_{F_k \in F} \tau_{v,F_k} \times [\eta_{v,F_k}]^\beta} & \text{при } F_k \in F \\ 0 & \text{в противном случае} \end{cases} \quad (7)$$

ИМ выбирается случайно, и, если в его списке ВМ –  $L_V$  есть хотя бы одна ВМ, то он выбирает вариант ЖМ ВМ из возможных вариантов на основе значений феромонов (уравнение 7), с учетом накладных расходов. Затем ИМ добавляет в свою  $MM_a$  пару  $(v, F)$ , а ВМ  $v$  удаляется из списка  $L_V$ .

Если же список  $L_V$  выбранного ИМ пуст, то ИМ принимает решение о завершении переноса ВМ из списка  $L_V$ , вычисляет значение целевой функции  $f$  в соответствии с уравнением 8 (блок 17):

$$\max f(MM_G) = \frac{N_{F_{HA}}}{HP_{MM_G}}, \quad (8)$$

где  $N_{F_{HA}}$  – количество неактивных ФМ ( $F_{HA} \subset F$ );  $HP_{MM_{i,j}}$  – накладные расходы на процесс ЖМ ВМ матрицы  $MM_G$ .

Физический смысл целевой функции можно интерпретировать как увеличение количества ФМ, на которые не распределена ни одна ВМ и которые можно исключить из схемы энергопотребления ЦОД и/или использовать их для нового цикла выделения ВМ, с учетом влияния накладных расходов на миграцию ВМ на ФМ.

После чего ИМ удаляется из списка ИМ –  $L_{ИМ}$  (блок 18).

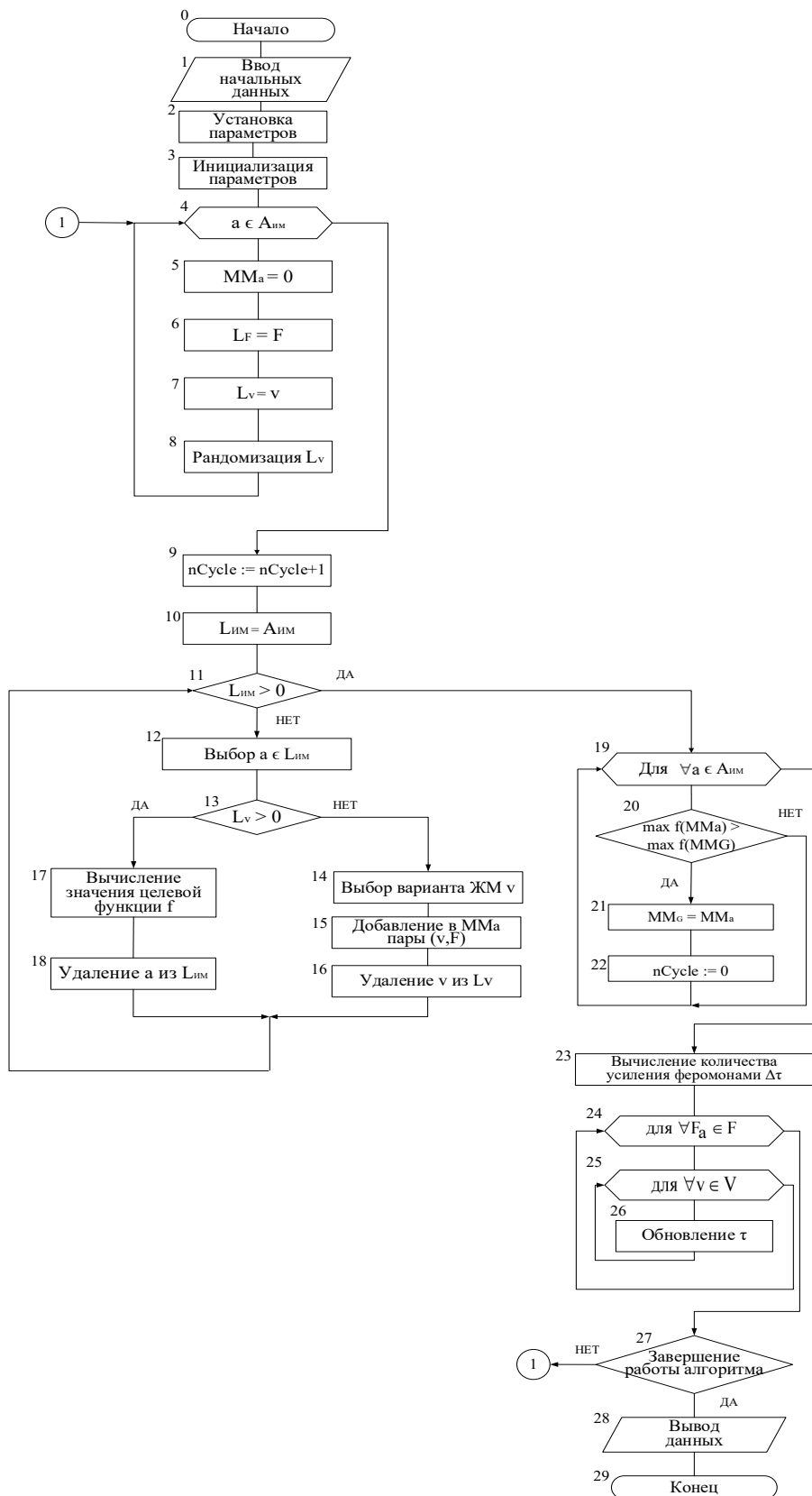


Рисунок 2 – Схема алгоритма ACS  
Figure 2 – The scheme of the ACS algorithm



**ШАГ 4 Обновление матрицы миграции.** Когда все ИМ завершили построение своих  $MM_a$ , происходит процесс определения лучшей/оптимальной  $MM_G$ . Алгоритм проверяет, имеет ли какая-либо  $MM_a$  лучшее значение целевой функции  $f$ , чем текущая  $MM_G$  (блок 20). Если это так, то  $MM_G$  обновляется (блок 21), счетчик  $nCycle$  сбрасывается (блок 22).

**ШАГ 5 Обновление уровня феромона.** На данном этапе вычисляется количество усиления феромонами  $\Delta\tau_{vF,a}$  на основе качества  $MM_G$  в соответствии с уравнением 9 (блок 23):

$$\Delta\tau_{vF,a} = \begin{cases} \max(MM_G) \text{ при } (v,F) \in MM_G \\ 0 \text{ в противном случае} \end{cases} \quad (9)$$

Значения феромонов обновляется для каждой пары  $(v, F)$ , имитируя испарение и осаждение феромонов, в соответствии с уравнением 10 (блоки 24–26):

$$\tau_{vF,a} = (1-\delta)*\tau_{vF,a} + \delta*\Delta\tau_{vF,a}. \quad (10)$$

Алгоритм усиливает значение феромона только в парах  $(v, F)$ , которые принадлежат  $MM_G$ .

**ШАГ 6 Проверка завершения алгоритма.** Основной процесс продолжает итерации до тех пор, пока не будет достигнуто максимальное количество циклов алгоритма  $nCycleTerm$  (блок 27).

Результатом работы алгоритма является  $MM_G$  с оптимальным размещением ВМ по ФМ, что позволяет значительно увеличить число размещенных ВМ и тем самым повысить показатель утилизации ВКР ФМ. Это также способствует уменьшению количества процессов ЖМ ВМ и снижению связанных с этих накладных расходов, что в итоге помогает сократить количество задействованных ФМ.

### Заключение

Статья посвящена решению проблемы перераспределения виртуализированных ВКР в ЦОД с целью повышения их эффективности, учитывая накладные расходы на ЖМ ВМ. Анализ существующих исследований в данной области позволил установить, что проблема оптимального распределения ресурсов в ЦОД является NP-трудной задачей, решаемой в рамках многомерной векторной упаковки. Для ее решения был осознанно выбран метод на основе метаэвристики муравьиной колонии.

На базе этого метода был разработан усовершенствованный вариант алгоритма муравьиной колонии – алгоритм ACS, который формирует оптимальное размещение ВМ на ФМ. Это позволяет не только увеличить число размещенных ВМ, но и сократить количество задействованных ФМ.

Дальнейшие исследования направлены на разработку новой распределенной структуры оптимизации.

### СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Бумажкина Н.Ю., Захарова И.Н., Кочкуров А.Е. К вопросу об использовании технологий живой миграции виртуальных машин в задаче оптимизации ресурсов центра обработки данных. В сборнике: *Современные проблемы информатизации в области анализа и синтеза технологических и телекоммуникационных систем: Сборник статей по материалам 29-й международной открытой научной конференции, ноябрь 2023 года – январь 2024 года, Воронеж, Россия*. Воронеж: Научная книга; 2024. С. 133–137.

- Bumazhkina N.Yu., Zakharova I.N., Kochkurov A.E. K voprosu ob ispol'zovanii tekhnologii zhivoi migratsii virtual'nykh mashin v zadache optimizatsii resursov tsentra obrabotki dannykh. In: *Sovremennye problemy informatizatsii v oblasti analiza i sinteza tekhnologicheskikh i telekommunikatsionnykh sistem: Sbornik statei po materialam 29-i mezhdunarodnoi otkrytoi nauchnoi konferentsii, November 2023 – January 2024, Voronezh, Russia*. Voronezh: Nauchnaya kniga; 2024. pp. 133–137. (In Russ.).
2. Choudhary A., Govil M.Ch., Singh G., Awasthi L.K., Pilli E.S., Kapil D. A critical survey of live virtual machine migration techniques. *Journal of Cloud Computing*. 2017;6(1). <https://doi.org/10.1186/s13677-017-0092-1>
  3. Mishra M., Sahoo A. On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach. In: *2011 IEEE 4th International Conference on Cloud Computing, 04-09 July 2011, Washington, DC, USA*. IEEE; 2011. pp. 275–282. <https://doi.org/10.1109/CLOUD.2011.38>
  4. Li X., Qian Z., Lu S., Wu J. Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Mathematical and Computer Modelling*. 2013;58(5-6):1222–1235. <https://doi.org/10.1016/j.mcm.2013.02.003>
  5. Murtazaev A., Oh S. Sercon: Server Consolidation Algorithm using Live Migration of Virtual Machines for Green Computing. *IETE Technical Review*. 2011;28(3):212–231.
  6. Shen H., Chen L. A Resource Usage Intensity Aware Load Balancing Method for Virtual Machine Migration in Cloud Datacenters. *IEEE Transactions on Cloud Computing*. 2020;8(1):17–31. <https://doi.org/10.1109/TCC.2017.2737628>
  7. Marzolla M., Babaoglu O., Panzieri F. Server consolidation in Clouds through gossiping. In: *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 20-24 June 2011, Lucca, Italy*. IEEE; 2021. pp. 1–6. <https://doi.org/10.1109/WoWMoM.2011.5986483>
  8. Feller E., Morin C., Esnault A. A case for fully decentralized dynamic VM consolidation in clouds. In: *4th IEEE International Conference on Cloud Computing Technology and Science: Proceedings, 03-06 December 2012, Taipei, Taiwan*. IEEE; 2012. pp. 26–33. <https://doi.org/10.1109/CloudCom.2012.6427585>
  9. Caprara A., Toth P. Lower bounds and algorithms for the 2-dimensional vector packing problem. *Discrete Applied Mathematics*. 2001;111(3):231–262. [https://doi.org/10.1016/S0166-218X\(00\)00267-5](https://doi.org/10.1016/S0166-218X(00)00267-5)
  10. Пантелеев А.В. *Метаэвристические алгоритмы поиска глобального экстремума*. Москва: Издательство МАИ-Принт; 2009. 160 с.  
Panteleev A.V. *Metaheuristic algorithms for finding the global extremum*. Moscow: Izdatel'stvo MAI-Print; 2009. 160 p. (In Russ.).
  11. Blum Ch., Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*. 2003;35(3):268–308. <https://doi.org/10.1145/937503.937505>
  12. Reeves C.R. *Modern Heuristic Techniques for Combinatorial Problems*. New York: John Wiley & Sons, Inc.; 1993. 320 p.
  13. Glover F.W., Kochenberger G.A. *Handbook of Metaheuristics*. Dordrecht: Kluwer Academic Publishers; 2003. 557 p. <https://doi.org/10.1007/b101874>
  14. Nesmachnow S. An overview of metaheuristics: accurate and efficient methods for optimisation. *International Journal of Metaheuristics*. 2014;3(4):320–347. <https://doi.org/10.1504/ijmheur.2014.068914>
  15. Алексеев В.Е., Захарова Д.В. *Теория графов*. Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского; 2012. 57 с.  
Alekseev V.E., Zakharova D.V. *Teoriya grafov*. Nizhny Novgorod: National Research Lobachevsky State University of Nizhny Novgorod; 2012. 57 p. (In Russ.).

16. Reeves C. Hybrid genetic algorithms for bin-packing and related problems. *Annals of Operations Research*. 1996;63(3):371–396. <https://doi.org/10.1007/BF02125404>
17. Кааоуаче М.А., Вуамамма С. Solving bin Packing Problem with a Hybrid Genetic Algorithm for VM Placement in Cloud. *Procedia Computer Science*. 2015;60:1061–1069. <https://doi.org/10.1016/j.procs.2015.08.151>
18. Junjie P., Dingwei W. An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem. In: *First International Conference on Innovative Computing, Information and Control (ICICIC'06): Volume I, 30 August 2006 - 01 September 2006, Beijing, China*. IEEE; 2006. pp. 210–213. <https://doi.org/10.1109/ICICIC.2006.40>
19. Dorigo M., Stützle T. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: *Handbook of Metaheuristics*. Dordrecht: Kluwer Academic Publishers; 2003. pp. 250–285. [https://doi.org/10.1007/0-306-48056-5\\_9](https://doi.org/10.1007/0-306-48056-5_9)
20. Dorigo M., Birattari M., Stutzle T. Ant colony optimization. *IEEE Computational Intelligence Magazine*. 2006;1(4):28–39. <https://doi.org/10.1109/mci.2006.329691>

### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Бумажкина Наталья Юрьевна**, сотрудник **Natalya Y. Bumazhkina**, employee of the Академии Федеральной службы охраны Russian Federation Security Guard Service Российской Федерации, Орёл, Российская Федерация, Oryol, the Russian Federation. Федерация.  
*e-mail:* [kislaya90@mail.ru](mailto:kislaya90@mail.ru)

*Статья поступила в редакцию 01.07.2024; одобрена после рецензирования 09.07.2024; принята к публикации 17.07.2024.*

*The article was submitted 01.07.2024; approved after reviewing 09.07.2024; accepted for publication 17.07.2024.*