УДК 004.75

М.В. Степанова, О.Ю. Ерёмин

НАЗНАЧЕНИЕ ЗАДАНИЙ УЗЛАМ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ПЛАТФОРМЫ ИНТЕРНЕТА ВЕЩЕЙ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Степанова Мария Владимировна, окончила факультет «Информатика и системы управления» Московского государственного технического университета им. Н.Э. Баумана, аспирант кафедры «Компьютерные системы и сети» МГТУ им. Н.Э. Баумана. Имеет статьи по использованию адаптивных технологий в распределенных системах и в образовании. [e-mail: stepanova@bmstu.ru].

Ерёмин Олег Юрьевич, кандидат технических наук, окончил Аэрокосмический факультет МГТУ им. Н.Э. Баумана, доцент кафедры «Компьютерные системы и сети» факультета «Информатика и системы управления» МГТУ им. Н.Э. Баумана. Имеет статьи по использованию адаптивных технологий для обработки неструктурированных и слабоструктурированных данных. [e-mail: ereminou@bmstu.ru].

Аннотация

В работе рассматриваются вопросы применения адаптивного подхода, основанного на машинном обучении с подкреплением, для распределения вычислительных заданий в распределенной инфраструктуре Интернета вещей (ИВ, IoT – Internet of Things). Платформа ИВ состоит из гетерогенных элементов, являющихся вычислительными узлами. Особенностью системы является постоянное изменение характеристик вычислительных узлов, что не позволяет использовать классические подходы к реализации распределенных вычислительных систем. Метод машинного обучения с подкреплением позволяет решить задачу построения распределенной системы за счет адаптивного формирования последовательности вычислительных узлов и соответствующих вычислительных заданий. Результатом автоматизации процесса является возможность использования узлов платформы ИВ для выполнения программного кода, разработанного для классических распределенных и параллельных систем, и решения вычислительных задач.

Ключевые слова: Интернет вещей, распределенные системы, распределенные вычисления, вычислительные узлы, программный код, параллельные вычисления, машинное обучение, обучение с подкреплением, адаптивный метод.

doi: 10.35752/1991-2927-2021-1-63-27-33

THE ASSIGNMENT OF TASKS TO THE NODES OF THE IOT DISTRIBUTED SYSTEM BASED ON REINFORCEMENT MACHINE LEARNING

Maria Vladimirovna Stepanova, graduated from the Faculty of Informatics and Control Systems of Bauman Moscow State Technical University (BMSTU) with Master's degree in Computer Science; Postgraduate Student at the Department of Computer Systems and Networks of BMSTU, an author of scientific papers in the field of adaptive technologies in distributed systems and in education. e-mail: stepanova@bmstu.ru.

Oleg Iurevich Eremin, Candidate of Sciences in Engineering; graduated from the Aerospace Faculty of Bauman Moscow State Technical University; Associate Professor of the Department of Computer Systems and Networks of the Faculty of Informatics and Control Systems at BMSTU; an author of articles in the field of adaptive technologies for processing unstructured and semi-structured data. e-mail: ereminou@bmstu.ru.

Abstract

The article describes issues of applying an adaptive approach based on reinforcement learning for assignment of the computing tasks to nodes of distributed Internet of Things (IoT) platform. The IoT platform consists of heterogeneous elements that are computing nodes. Classical approaches, methods, and algorithms for distributed and parallel systems are not suitable for task assignment in IoT systems due to its characteristics. The reinforcement learning method allows you to solve the problem of building a distributed system due to the adaptive formation of a sequence of computational nodes and the corresponding computational tasks. Thus, the article represents a method that makes IoT nodes capable of execution computing tasks, especially, which were previously designed for classical distributed and parallel systems.

Keywords: Internet of Things, distributed systems and computations, computations nodes, programme code, parallel computing, machine learning, reinforcement learning, adaptive method.

ВВЕДЕНИЕ

Решение сложных вычислительных задач требует использования специализированных распределенных или параллельных вычислительных систем, таких как многопроцессорные, многоядерные, многопоточные системы и аналогичные им. Платформа Интернета вещей (ИВ, IoT – Internet of Things) представляет множество гетерогенных вычислительных узлов, на основе которых возможно построение распределенной вычислительной системы.

В основе ИВ содержится идея внедрения в объекты окружающего мира специализированных модулей, которые позволяют реализовывать возможности идентификации, передачи данных, измерения (сенсоры и датчики) и обработки данных. Первоначальные подходы по взаимодействию элементов ИВ состояли в передаче данных со встроенных вычислительно-датчиковых модулей на центральный вычислительный узел [1]. Несмотря на наличие вычислительного модуля у узлов ИВ, хранение и обработка данных, а также формирование управляющих сигналов для модулей осуществляются только по централизованному типу [2]. Данные передаются на центральный управляющий узел, который проводит необходимые преобразования данных. Центральный управляющий узел в ИВ является либо сервером, осуществляющим прием сообщений от объектов ИВ, хранение, обработку и передачу поступившей информации пользователям, либо шлюзом, выполняющим согласование протоколов ИВ и сети Интернет, а также ограниченно выполняющим все роли сервера. Подобная архитектура имеет ряд значительных недостатков, таких как:

- большая временная задержка из-за используемых объектами ИВ протоколов связи;
- сложность поддержки многообразия технологического стека и сложность организации их согласованного взаимодействия с ростом числа разнотипных модулей;
- рост нагрузки на пропускную способность вычислительной сети из-за передачи существенного количества необработанных данных.

Для устранения данных недостатков были предложены архитектуры по выносу вычислительной нагрузки с централизованного узла согласно принципу: часть вычислительных задач, которая может быть решена без их непосредственной отправки на централизованный узел, должна быть выполнена соответствующим узлом. Подобный подход реализован в парадигме граничных вычислений, где периферийные узлы могут быть организованы в распределенную вычислительную систему с целью повышения скорости обработки данных [3]. Вычислительные возможности встраиваемых систем, в том числе объектов ИВ, существенно растут, поэтому вынесение обработки сенсорных и локальных данных на узлы

платформы ИВ может дать преимущества, аналогичные полученным в системах граничных вычислений [1, 2].

Существующие алгоритмы распределения заданий по вычислительным узлам, используемые в классических распределенных системах [4], не применимы к платформе ИВ. Подобное ограничение вызвано природой её инфраструктуры: непостоянным числом вычислительных узлов и их гетерогенностью, непостоянным качеством сети передачи данных и т. д. Таким образом, является актуальной задача разработки метода, учитывающего изменчивость инфраструктуры и обеспечивающего распределение заданий по вычислительным узлам платформы ИВ. В работе в качестве такого метода предлагается использовать машинное обучение с подкреплением.

1 РАСПРЕДЕЛЕННАЯ СИСТЕМА НА БАЗЕ ПЛАТФОРМЫ ИВ

Распределенная вычислительная система, реализованная на платформе ИВ, состоит из множества гетерогенных узлов, имеющих различные технические ограничения. Узлы осуществляют обмен сообщениями по проводным или беспроводным сетям. В распределенных вычислительных системах для реализации вычислительной задачи необходимо осуществить размещение вычислительных заданий (которые являются элементарными блоками исходной вычислительной задачи) с учетом следующего:

- минимизации времени простоя узлов;
- отсутствия перезагрузки узлов;
- обеспечения качества обработки вычислительных заданий на узлах.

Поскольку к вычислительным системам на платформе ИВ неприменимы подходы для классических систем, то к ним также невозможно применять используемые в них параметры эффективности [5].

Задача распределения заданий в вычислительной системе на платформе ИВ является NP-полной задачей. Решение таких задач может выполняться методами на основе эвристик, которые позволяют получить близкое к оптимальному решение в условиях ограничений [6]. Эвристические подходы предполагают полное описание накладываемых на систему ограничений и состояний вычислительных узлов.

Процедура распределения заданий по узлам в общем случае может иметь следующий вид [7]:

- задание алгоритма вычислительной задачи для работы на распределенной системе;
- составление вычислительного графа алгоритма;
- определение взаимной зависимости заданий;
- составление последовательности заданий;
- сопоставление заданий и вычислительных узлов;
- отправка заданий на вычислительные узлы;
- получение ответов (выполненных заданий) с каждого вычислительного узла.

Преобразование алгоритма вычислительной проблемы к вычислительному графу [8] представляет решение в виде набора заданий, имеющих связь с другими заданиями. Также необходимо определить, какому заданию для начала своего выполнения требуется окончание выполнения других заданий. Учет этой связи позволит вычислительной системе не оказаться в состоянии клинча (deadlock). Также задание может требовать различные ограничения на характеристики вычислительного узла, на котором оно может быть выполнено.

Сформированная с учётом ограничений последовательность заданий распределяется по узлам распределенной системы на платформе ИВ. Поскольку узлы на платформе ИВ имеют разные характеристики, то метод на основе обучения с подкреплением учитывает специфику каждого задания и каждого вычислительного узла.

2 СОПОСТАВЛЕНИЕ ЗАДАНИЙ ПО ВЫЧИСЛИТЕЛЬНЫМ УЗЛАМ

Задача распределения заданий по вычислительным узлам распределенной системы на базе платформы ИВ может быть решена использованием алгоритма многорукого бандита, относящегося к методам машинного обучения с подкреплением.

Согласно данным методам [9], инфраструктура ИВ может быть представлена в виде окружающей среды, которая определяется числом узлов, их характеристиками и т. д. Со средой взаимодействует агент, являющийся узлом распределения вычислительных заданий, реализованный на центральном управляющем узле (на сервере или на шлюзе). При данном подходе агент не нуждается в формировании модели окружающей среды, а принимает решения на основании полученных от окружающей среды реакций (см. рис. 1) и предпринятых ранее действий.

Обучение с подкреплением — одно из направлений машинного обучения, однако, в отличие от обучения с учителем, не требует наличия размеченных данных, и в отличие от обучения без учителя, не предназначен для нахождения скрытых структур без учета многообразия влияющих факторов.

Ключевой особенностью данного типа машинного обучения является самостоятельное составление набора благоприятных для принятия решений действий в за-

висимости от текущей ситуации окружающей среды и постоянная адаптация к изменяющимся неопределённым условиям окружающей среды.

Предлагается применение алгоритма многорукого бандита, согласно которому агент осуществляет воздействие на окружающую среду. Все возможные воздействия принадлежат совокупности известных действий A , а R — совокупность сигналов вознаграждения [10]. Каждому определенному действию $a \in A$ ставится в соответствие сигнал вознаграждения $r \in R$.

Следовательно, для момента времени t получаемое вознаграждение R на любое действие из A :

$$q_*(a) = E[R_t | A_t = a], \quad (1)$$

где $q_*(a)$ — значение ценности выполненного действия при ожидаемом вознаграждении, которое получает агент, выбравший произвольное действие в какой-то момент времени.

$$R^a(t) = P(r|a), \quad (2)$$

где $R^a(t)$ — описывает неизвестное распределение вероятности вознаграждения при условии выбора действия a .

В поведении агента можно выделить этапы исследования среды (режим исследования) и этапы воздействия на среду (режим эксплуатации). Для получения оптимального (субоптимального) поведения агента необходимо обеспечить равновесие в работе фаз алгоритма: процесса исследования и процесса эксплуатации [11, 12]. В алгоритме многорукого бандита для этого могут использоваться различные подходы, в нашем случае рассматривается ε -жадная (ε -greedy) стратегия. При данной стратегии используется изменяемый коэффициент $\varepsilon \in [0; 1]$, который определяет вероятность выбора каждого действия:

$$a_t = \begin{cases} a_t^*, & \text{с вероятностью } 1 - \varepsilon; \\ \text{случайное действие,} & \text{с вероятностью } \varepsilon. \end{cases} \quad (3)$$

Варьируя параметр ε в алгоритме, можно настраивать поведение агента: делать его полностью жадным, когда алгоритм находится в режиме эксплуатации, или полностью вариативным, когда алгоритм находится в режиме исследования.

При $\varepsilon = 0$ агент не пытается выполнить новое действие, стратегия становится полностью жадной. То есть выполняются только те действия, которые приводят к наибольшему вознаграждению.

При $\varepsilon = 1$ алгоритм случайным образом выбирает действия на каждом этапе.

Алгоритм распределения заданий:

Шаг 0. Определяется среда, то есть задается количество узлов и их характеристики в инфраструктуре ИВ. Распределяющий узел составляет последовательность заданий, учитывая их взаимосвязи.

Рис. 1. Схема взаимодействия распределительно-го узла (агента) и вычислительных узлов (окружающей среды)

Шаг 1 (этап исследования). Распределяющий узел осуществляет опрос вычислительных узлов, получая значения R . На основе текущего состояния каналов связи и доступности вычислительных узлов строится множество значений сигнала вознаграждения R — интегральной характеристики. Характеристика учитывает параметры вычислительного узла (время обработки пакетов, расход энергии и т. д.).

Шаг 2. Интегральная характеристика каждого вычислительного узла преобразуется в значения вероятностей. Вероятности узлов характеризуют готовность того или иного узла принятия задания.

Шаг 3 (этап эксплуатации). Распределяющий узел учитывает вероятности предыдущего шага и сопоставляет задания с вычислительными узлами, которые дадут максимальный сигнал вознаграждения.

Шаг 4. Распределяющий узел пересылает задания соответствующим вычислительным узлам.

Шаг 5. Вычислительные узлы осуществляют расчёты и отправляют результаты (включая данные о состоянии каналов связи и вычислительных узлов) на распределяющий узел, который пересчитывает значения сигналов вознаграждения R .

Шаг 6. Если последовательность содержит невыполненные задания, то осуществляется переход на Шаг 3, иначе — переход на Шаг 7.

Шаг 7. Конец алгоритма.

Реализация алгоритма многорукого бандита может происходить в различных условиях поведения окружающей среды [13]:

- статический режим, когда параметры среды со временем не меняются;
- динамический режим, когда параметры среды изменяются во времени.

Вариант работы в динамическом режиме, например, возможен в случае, когда объекты ИВ не постоянны [14] и перемещаются в пространстве, и связь с ними может прерываться, либо меняться пропускная способность канала связи.

3 РЕЗУЛЬТАТЫ РАБОТЫ АЛГОРИТМА В СТАТИЧЕСКОМ И ДИНАМИЧЕСКОМ РЕЖИМАХ

Возможность использования метода распределения заданий по вычислительным узлам платформы ИВ может быть продемонстрирована экспериментом *in silico*, в котором рассматриваемый метод и модели вычислительных узлов ИВ были реализованы на языке программирования Python. Вычислительные задания и сигналы вознаграждения на этапе симуляции представляют собой сообщения, которыми обменивается распределяющий узел и вычислительные узлы ИВ.

При проведении симуляции были сделаны следующие предположения: если от вычислительного узла поступил сигнал вознаграждения, то узел выполнил полученное задание (то есть увеличение нагрузки на вычислительные узлы никак не сказывается на их функционировании); у вычислительных узлов могут меняться

параметры, но сами узлы при этом не могут исчезать и появляться; вычислительные узлы имеют одинаковую структуру и отличаются только сигналом вознаграждения (который при натурном эксперименте является интегральной характеристикой каждого узла, учитывающей текущее состояние узла и состояние вычислительной сети).

Для анализа возможности применимости предложенного метода алгоритм рассматривается в двух режимах: в статическом и в динамическом.

Рассматривается выполнение алгоритма многорукого бандита в статическом режиме с целью оценки поведения алгоритма, когда окружающая среда стационарна, то есть значения вероятностей выбора каждого действия неизменны, количество активных элементов постоянно, а параметр ε принимает следующие значения:

$$\varepsilon \in [0; 0,001; 0,05; 0,25; 0,1; 0,2; 0,3; 0,4; 0,5; 1].$$

Изменяя параметр ε , необходимо оценить поведение системы распределения заданий на некотором интервале времени.

Демонстрация результатов на рисунке 2 показывает, что поведение системы со временем становится стабильным, и система будет предоставлять одинаковое качество сопоставления заданий и вычислительных узлов, а также для большего числа вычислительных узлов требуется большее время на стабилизацию. Полученное поведение системы совпадает с результатами ранее проведенных исследований [15, 16]: при увеличении параметра ε алгоритм все больше осуществляет работу в режиме исследования окружающей среды и все меньше в режиме эксплуатации.

Рис. 2. Средняя производительность при выборе из 20 действий в зависимости от параметра ε

Результаты, показанные на рисунке 3, демонстрируют, что с увеличением количества повторений растет общее вознаграждение распределяющего узла. Для оценки оптимального поведения алгоритма многорукого бандита может использоваться функция сожаления (см. рис. 4). Логарифмический вид функции соответствует

Рис. 3. Среднее значение вознаграждения (20 действий) при 1000 симуляций

Рис. 4. Среднее значение функции сожаления (20 действий) при 1000 симуляций

ранее проведенным исследованиям [17, 18]. При 20 действиях оптимальным значением ϵ является 0,1 и 0,2.

В динамическом режиме принимается условие, что окружающая среда стационарна некоторый промежуток времени до момента непосредственного назначения очередного вычислительного задания узлу. Затем среда может изменять состояние.

В предыдущих работах приводилась реализация адаптивного изменения параметра ϵ посредством его постоянного уменьшения с течением времени или переход из режима исследования в режим эксплуатации осуществлялся после четко определенного промежутка времени [19, 20].

В работе вариативный коэффициент адаптивно изменяет значение непрерывно из диапазона $[1;0]$ в зависимости от числа правильно выбранного узла, суммарно полученного вознаграждения от среды, статистики вознаграждения от каждого узла в зависимости от задания. В отличие от стационарного режима, где алгоритм функционирует постоянно только в одном из двух состояний (исследование или эксплуатация), динамическое изменение вариативного показателя позволяет

непрерывно адаптивно менять поведение агента к среде. Тем самым, обеспечивается оптимальный (субоптимальный) уровень опроса и выбора узлов.

Результаты на рисунке 5 показывают, что вследствие изменения среды получаемое значение функции сожаления со временем не стабилизируется. Это означает, что агент выбирает оптимальную стратегию назначения задания узлу зависимости от среды – состояний узлов.

Рис. 5. Среднее значение функции сожаления при адаптивном параметре ϵ (20 действий) при 1000 симуляций

Среднее значение вознаграждения при адаптивном ϵ достигается быстрее, и по значению является больше, чем при статическом режиме.

В процессе функционирования системы качество ее работы с адаптивным подходом улучшается, что далее приводит к назначению заданий на оптимальные узлы. В противном случае, без адаптивного подхода, система выбирает узлы, однако они не являются оптимальными под задания.

Результаты на рисунке 6 показывают, что вследствие постоянного изменения параметров вычислительной системы график имеет более «рваный» вид.

Рис. 6. Среднее значение вознаграждения при адаптивном параметре ϵ (20 действий) при 1000 симуляций

Значение функции вознаграждения не убывает в динамической системе, что свидетельствует о возможности использования предложенного метода для реализации в вычислительной системе, параметры которой изменяются со временем.

ЗАКЛЮЧЕНИЕ

В работе представлен метод, позволяющий распределять задания по вычислительным узлам, для построения распределенной системы на платформе ИВ.

Использование методов машинного обучения с подкреплением позволяет учитывать многообразие и изменчивость окружающей среды.

Предложенный подход позволяет задействовать сами вычислительные узлы для выполнения вычислений вместо передачи всего потока данных в централизованное хранилище на обработку. Также особенностью предложенного метода является возможность использования классических принципов разработки алгоритмов для распределенных параллельных систем и имеющегося программного обеспечения для распределенных вычислений для работы на платформах с изменяющимися во времени характеристиками, такими, как платформы ИВ.

СПИСОК ЛИТЕРАТУРЫ

1. Ai Y., Peng M., Zhang K. Edge computing technologies for Internet of Things: a primer // *Digital Communications and Networks*. 2018. Vol. 4, no. 2. pp. 77–86.
2. Adi E., Anwar A., Baig Z. Machine learning and data analytics for the IoT // *Neural Comput & Applic.* Oct. 2020. Vol. 32, iss. 20. pp. 16205–16233.
3. Chiang M., Zhang T. Fog and IoT: an overview of research opportunities // *IEEE Internet Things J.* 2016. Vol. 3. pp. 854–864.
4. M. van Steen, Tanenbaum A.S. *Distributed Systems*, 3rd ed., 2017. 596 p.
5. Черненький В.М., Семкин П.С. Метод описания процессов выполнения заданий в мультипрограммных и мультипроцессорных системах // *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*. 2014. № 1 (94). С. 121–132.
6. Yeckle J., Rivera W. Mapping and characterization of applications in heterogeneous distributed systems // *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI2003)*. 2003. pp. 1–6.
7. Ерёмин О.Ю., Степанова М.В. Распределение заданий по узлам вычислительной системы на платформе Интернета вещей на основе машинного обучения // *Динамика сложных систем*. 2020. Т. 14, № 2. С. 84–92.
8. Воеводин В.В., Воеводин Вл.В. *Параллельные вычисления*. СПб. : БХВ-Петербург. 2002. 608 с.
9. Sutton Richard S., Barto Andrew G. *Reinforcement learning: an introduction*. The MIT Press, 2008.
10. Robbins H. Some aspects of the sequential design of experiments // *Bull. Am. Math. Soc.* 1952. No. 58. pp. 527–535.
11. Rejeb L., Guessoum Z., M'Hallah R. An Adaptive Approach for the Exploration-Exploitation Dilemma for Learning Agents // Pěchouček M., Petta P., Varga L.Z. (eds) *Multi-Agent Systems and Applications IV. CEEMAS 2005. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2005. Vol. 3690. pp. 316–325.
12. Yogeswaran M., Ponnambalam S.G. Reinforcement learning: exploration–exploitation dilemma in multi-agent foraging task // *OPSEARCH*. 2012. No. 49. pp. 223–236.
13. A Multi-armed Bandit Algorithm Available in Stationary or Non-stationary Environments Using Self-organizing Maps / N. Manome, S. Shinohara, K. Suzuki, K. Tomonaga, S. Mitsuyoshi // *Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation. ICANN 2019. Lecture Notes in Computer Science* / I. Tetko, V. Kurkova, P. Karpov, F. Theis (eds). Springer, Cham, 2019. Vol. 11727. pp. 529–540.
14. Hakim A., Amirat A., Oussalah M.C. Non-intrusive contextual dynamic reconfiguration of ambient intelligent IoT systems // *J. Ambient Intell Human Comput.* 2020. No. 11. pp. 1365–1376.
15. Akanmu S., Garg R., Gilal A. Towards an Improved Strategy for Solving Multi-Armed Bandit Problem // *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*. 2019. Vol. 8, no. 12. pp. 5060–5064.
16. Mignona A., Rocha R. An Adaptive Implementation of ϵ -Greedy in Reinforcement Learning // *Procedia Computer Science* 109C. 2017. pp. 1146–1151.
17. Peter A., Fischer N., Fischer P. Finite-time Analysis of the Multiarmed Bandit Problem // *Machine Learning*. 2002. No. 47. pp. 235–256.
18. Agrawal R. Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem // *Advances in Applied Probability*. 1995. No. 27. pp. 1054–1078.
19. Tokic M. Adaptive ϵ -Greedy Exploration in Reinforcement Learning Based on Value Differences / Dillmann R., Beyerer J., Hanebeck U.D., Schultz T. (eds) // *KI 2010: Advances in Artificial Intelligence. KI 2010. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg. 2010. Vol. 6359. pp. 203–210.
20. Caelen O., Bontempi G. Improving the exploration strategy in bandit algorithms // *Learning and Intelligent Optimization*. Number 5313 in LNCS. Springer, 2008. pp. 56–68.

REFERENCES

1. Ai Y., Peng M., Zhang K. Edge Computing Technologies for Internet of Things: A Primer. *Digital Communications and Networks*, 2018, vol. 4, no. 2, pp. 77–86.
2. Adi E., Anwar A., Baig Z. Machine Learning and Data Analytics for the IoT. *Neural Comput & Applic.*, Oct. 2020, vol. 32, iss. 20, pp. 16205–16233.
3. Chiang M., Zhang T. Fog and IoT: An Overview of Research Opportunities. *IEEE Internet Things J.*, 2016, vol. 3, pp. 854–864.

4. M. van Steen, Tanenbaum A.S. *Distributed Systems*. 3rd ed., 2017. 596 p.
5. Chernenkiy V.M., Semkin P.S. Metod opisaniia protsessov vypolneniia zadani v multiprogrammnykh i multiprotsessornykh sistemakh [Method for Description of Task in Multiprogramming and Multiprocessor Systems]. *Vestnik MGTU im. N.E. Baumana. Ser. Priborostroenie* [Herald of the Bauman Moscow State Technical University. Instrument Engineering], 2014, no. 1 (94), pp. 121–132.
6. Yeckle J., Rivera W. Mapping and Characterization of Applications in Heterogeneous Distributed Systems. *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI2003)*. 2003, pp. 1–6.
7. Eremin O.Iu., Stepanova M.V. Raspredelenie zadani po uzlam vychislitelnoi sistemy na platforme Interneta veshchei na osnove mashinnogo obucheniia [Applying Reinforcement Learning in Distribution Computational Systems – Internet of Things]. *Dinamika slozhnykh sistem* [Dynamics of Complex Systems], 2020, vol. 14, no. 2, pp. 84–92.
8. Voevodin V.V., Voevodin V.V. *Parallelnye vychisleniia* [Parallel Computing]. St. Petersburg, BHV-Peterburg Publ., 2002. 608 p.
9. Sutton Richard S., Barto Andrew G. *Reinforcement Learning: An Introduction*. The MIT Press, 2008.
10. Robbins H. Some Aspects of the Sequential Design of Experiments. *Bull. Am. Math. Soc.*, 1952, no. 58, pp. 527–535.
11. Rejeb L., Guessoum Z., M'Hallah R. An Adaptive Approach for the Exploration-Exploitation Dilemma for Learning Agents. Pěchouček M., Petta P., Varga L.Z. (eds). *Multi-Agent Systems and Applications IV. CEEMAS 2005. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2005, vol. 3690, pp. 316–325.
12. Yogeswaran M., Ponnambalam S.G. Reinforcement Learning: Exploration-Exploitation Dilemma in Multi-Agent Foraging Task. *OPSEARCH*, 2012, no. 49, pp. 223–236.
13. Manome N., Shinohara S., Suzuki K., Tomonaga K., Mitsuyoshi S. A Multi-armed Bandit Algorithm Available in Stationary or Non-stationary Environments Using Self-organizing Maps. *Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation. ICANN 2019. Lecture Notes in Computer Science*. I. Tetko, V. Kurkova, P. Karpov, F. Theis (eds.). Springer, Cham, 2019, vol. 11727, pp. 529–540.
14. Hakim A., Amirat A., Oussalah M.C. Non-Intrusive Contextual Dynamic Reconfiguration of Ambient Intelligent IoT Systems. *J. Ambient Intell. Human Comput.*, 2020, no. 11, pp. 1365–1376.
15. Akanmu S., Garg R., Gilal A. Towards an Improved Strategy for Solving Multi-Armed Bandit Problem. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2019, vol. 8, no. 12, pp. 5060–5064.
16. Mignona A., Rocha R. An Adaptive Implementation of ϵ -Greedy in Reinforcement Learning. *Procedia Computer Science 109C*, 2017, pp. 1146–1151.
17. Peter A., Fischer N., Fischer P. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 2002, no. 47, pp. 235–256.
18. Agrawal R. Sample Mean Based Index Policies with $O(\log n)$ Regret for the Multi-Armed Bandit Problem. *Advances in Applied Probability*, 1995, no. 27, pp. 1054–1078.
19. Tokic M. Adaptive ϵ -Greedy Exploration in Reinforcement Learning Based on Value Differences. *Advances in Artificial Intelligence. KI 2010. Lecture Notes in Computer Science*. Dillmann R., Beyerer J., Hanebeck U.D., Schultz T. (eds.). Springer, Berlin, Heidelberg, 2010, vol. 6359, pp. 203–210.
20. Caelen O., Bontempi G. Improving the Exploration Strategy in Bandit Algorithms. *Learning and Intelligent Optimization*. Number 5313 in LNCS. Springer, 2008, pp. 56–68.