

МЕТОДИКА ПРИМЕНЕНИЯ АРГМОР ДЛЯ ПРОГРАММНОГО РАЗДЕЛЕНИЯ СЕТЕВЫХ РЕСУРСОВ В РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

DOI: 10.36724/2072-8735-2026-20-9-47-58

Петров Алексей Юрьевич,
Санкт-Петербургский государственный университет
аэрокосмического приборостроения,
Санкт-Петербург, Россия, palexexq1998@yandex.ru

Manuscript received 30 May 2026;
Accepted 02 August 2026

Оленев Валентин Леонидович,
Санкт-Петербургский государственный университет
аэрокосмического приборостроения,
Санкт-Петербург, Россия, valentin.olenev@guap.ru

Статья подготовлена при финансовой поддержке Российского
научного фонда, проект № 22-19-00305-П <https://rscf.ru/project/22-19-00305/> "Пространственно-временные стохастические модели
беспроводных сетей с большим числом пользователей"

Тюрликов Андрей Михайлович,
Санкт-Петербургский государственный университет
аэрокосмического приборостроения,
Санкт-Петербург, Россия, turlikov@guap.ru

Ключевые слова: программное разделение сетевых
ресурсов; распределённая вычислительная сеть; АРГМОР;
многокритериальная оптимизация; Парето-архив; правило
распределения; управляющий узел; узел-исполнитель;
динамическое распределение ресурсов; обратная связь

Предметом исследования является методика применения Адаптивного Роево-Генетического Метода Оптимизации Распределения ресурсов (АРГМОР) для программного разделения сетевых ресурсов в распределённой вычислительной сети. Цель работы состоит в формализации перехода от результатов многокритериальной оптимизации к практическому управлению узлами-исполнителями с учётом изменяющегося состояния вычислительной среды. Область применения – централизованно управляемые распределённые вычислительные сети, в которых требуется динамическое программное распределение задач и ресурсов. Методология исследования основана на представлении текущего состояния сети $S(t)$, включающего сведения о задачах, доступности, занятости и загрузке узлов, временных характеристиках и действующих ограничениях и параметрах, определяющих допустимость текущих вариантов распределения ресурсов, а также на использовании Парето-архива A , формируемого АРГМОР по двум критериям – средней задержке и производительности алгоритма распределения ресурсов. Предложена архитектура, разделяющая управляющий и исполнительный уровни, и определена последовательность выбора рабочего решения x^* , его преобразования в правило распределения и далее в адресованные управляющие команды. Сформирован замкнутый управляющий цикл: сбор состояния сети, запуск АРГМОР, формирование Парето-архива, выбор решения, распределение подзадач, выполнение, получение результатов и актуализация состояния $S(t+1)$. Установлено, что повторный запуск оптимизации целесообразно выполнять при событиях, способных изменить применимость действующего правила: поступлении новой задачи, изменении доступности или загрузки исполнителей, перегрузке, превышении допустимого времени ожидания либо ошибке выполнения. Методика обеспечивает многокритериальность, адаптацию к динамике сети, поддержку неоднородных исполнителей и независимость от конкретной аппаратной платформы и протокола передачи данных. Практическое применение методики проиллюстрировано на четырёхузловой модели распределённой вычислительной сети при последовательном поступлении задач оценки числа ? методом Монте-Карло. В серии из 50 запусков результаты АРГМОР сопоставлены с PSO, GA и решением, выбранным из точного Парето-фронта, сформированного полным перебором.

Информация об авторах:

Петров Алексей Юрьевич, старший преподаватель, Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия, <https://orcid.org/0009-0005-0231-1544>

Оленев Валентин Леонидович, заведующий кафедрой аэрокосмических компьютерных и программных систем, кандидат технических наук, доцент, Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия, <https://orcid.org/0000-0002-1817-2754>

Тюрликов Андрей Михайлович, заведующий кафедрой инфокоммуникационных технологий и систем связи, доктор технических наук, профессор, Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия, <https://orcid.org/0000-0001-7132-094X>

Для цитирования:

Петров А. Ю., Оленев В. Л., Тюрликов А. М. Методика применения АРГМОР для программного разделения сетевых ресурсов в распределённой вычислительной сети // Т-Comm: Телекоммуникации и транспорт. 2026. Том 20. №9. С. 47-58.

For citation:

A. Yu. Petrov, V. L. Olenov, A. M. Tyurlikov, "Methodology for Applying ARG MOR to Software-Based Partitioning of Network Resources in a Distributed Computing Network," T-Comm, 2026, vol. 20, no. 9, pp. 47-58. (in Russian)

Введение

Развитие распределённых вычислительных систем сопровождается увеличением числа одновременно выполняемых задач, усложнением их структуры и повышением требований к эффективности использования доступных ресурсов [1]. В условиях распределённой сети назначение вычислительной работы отдельным узлам не может рассматриваться как статическая процедура, выполняемая один раз в начале работы. Состав доступных исполнителей, степень их занятости, текущая загрузка и характеристики взаимодействия изменяются во времени, а в систему продолжают поступать новые задачи. Вследствие этого эффективность программного разделения сетевых ресурсов определяется способностью системы учитывать текущее состояние вычислительной среды и своевременно корректировать принятое распределение.

Ограниченность вычислительных и сетевых ресурсов не позволяет во всех случаях резервировать для каждой задачи или потока объём ресурсов, соответствующий его максимальным требованиям. Кроме того, статическое закрепление ресурсов при изменяющейся интенсивности обмена способно приводить к их недоиспользованию. В связи с этим актуальным является динамическое перераспределение ресурсов с учётом текущего состояния сети и изменения характеристик обслуживаемых потоков [2, 3]. Современные исследования показывают, что аналогичная задача возникает в распределённых, периферийных и спутниковых вычислительных средах, где решение должно учитывать изменение нагрузки, неоднородность вычислительных узлов и ограниченность доступных ресурсов.

Для оценки качества распределения в работе используются два основных параметра – средняя задержка и производительность алгоритма распределения ресурсов. Задержка отдельной задачи характеризует временные издержки между её поступлением в систему и началом выполнения на выбранном узле-исполнителе. В условиях динамически изменяющегося состояния вычислительной сети величина задержки отдельной задачи зависит от текущей загрузки и доступности узлов, состава поступивших задач и принятого варианта распределения и поэтому рассматривается как случайная величина. При сравнении вариантов распределения используется оценка её среднего значения. Производительность алгоритма распределения ресурсов характеризует количество задач, успешно завершённых системой за единицу времени. Совместное рассмотрение средней задержки и производительности позволяет оценивать как временные издержки распределения, так и итоговую результативность функционирования распределённой вычислительной сети.

Задачи динамического распределения вычислительных и сетевых ресурсов с учётом временных характеристик и текущего состояния системы рассматриваются и в современных распределённых вычислительных средах [4-6]. При этом существенное значение имеют неоднородность вычислительных узлов, изменяющаяся загрузка и ограниченность доступных вычислительных ресурсов [7-9].

Для решения многокритериальной задачи распределения ресурсов ранее разработан Адаптивный Роево-Генетический Метод Оптимизации Распределения ресурсов – АРГМОР [10-12]. В рассматриваемой методике АРГМОР используется как компонент, формирующий множество недоминируемых

вариантов распределения. Использование популяционных методов и Парето-подхода является распространённым способом решения многокритериальных оптимизационных задач [13-15].

Наличие оптимизационного метода само по себе не определяет порядок фактического распределения задач. Из множества недоминируемых решений необходимо выбрать рабочий недоминируемый вариант распределения, соответствующий текущим требованиям; затем выбранный вариант должен быть преобразован в правило распределения, пригодное для исполнения; на его основе должны быть сформированы управляющие команды, адресованные конкретным узлам-исполнителям; после выполнения задач должны быть получены результаты и актуализированы сведения о состоянии сети. Эти операции связывают результаты многокритериальной оптимизации с практическим управлением сетью.

В исследованиях посвящённых распределению сетевых ресурсов также рассматриваются механизмы динамического регулирования выделяемых ресурсов, справедливого распределения нескольких типов ресурсов и программного разделения функций управления и непосредственного исполнения [16-18].

Целью статьи является разработка и описание методики применения АРГМОР для программного разделения сетевых ресурсов в распределённой вычислительной сети. Для достижения поставленной цели необходимо определить входные данные, архитектуру взаимодействия компонентов, порядок выбора рабочего решения, формирования правила распределения и организации обратной связи.

Постановка задачи

Рассмотрим распределённую вычислительную сеть, в которой множество вычислительных узлов используется для выполнения поступающих задач или их отдельных подзадач. В каждый момент времени часть узлов может быть свободна, часть – выполнять ранее назначенную работу, а часть – быть временно недоступной для использования. Одновременно в систему могут поступать новые задачи, требующие назначения доступным ресурсам. Следовательно, процедура распределения должна учитывать не только состав вычислительной работы, но и текущие характеристики среды исполнения.

Вариант распределения, формируемый АРГМОР, далее обозначается через x . Рассматриваются только варианты распределения, удовлетворяющие действующим ограничениям и соответствующие текущему состоянию вычислительной сети.

Каждый допустимый вариант распределения x оценивается по двум показателям качества – средней задержке D и производительности P [6, 8, 19]:

$$D \rightarrow \min, P \rightarrow \max \quad (1)$$

Для отдельной задачи i задержка D_i определяется интервалом времени между поступлением задачи в систему и началом её выполнения на назначенном вычислительном узле:

$$D_i = t_i^{\text{start}} - t_i^{\text{in}}, \quad (2)$$

где t_i^{in} – момент поступления задачи в систему, t_i^{start} – момент начала выполнения задачи на назначенном узле.

Величина D_i зависит от текущего состояния вычислительной сети, занятости и загрузки узлов, состава поступающих

задач и сформированного варианта распределения. Для оценки качества конкретного варианта распределения используется оценка её среднего значения. Для N наблюдаемых задач средняя задержка определяется выражением:

$$\bar{D} = \frac{\sum_{i=1}^N D_i}{N} \quad (3)$$

Значение D_i характеризует задержку отдельной задачи, тогда как $\bar{D}$ используется при сравнении вариантов распределения. Единицей измерения задержки и средней задержки являются секунды или миллисекунды.

Производительность алгоритма распределения ресурсов определяется количеством задач, успешно завершённых системой за единицу времени:

$$P = \frac{N_{done}}{\Delta t}, \quad (4)$$

где N_{done} – количество задач, успешно завершённых за интервал наблюдения Δt . Единицей измерения производительности алгоритма является задач/с.

На основании сформированных входных данных и текущего состояния сети выполняется запуск АРГМОП, результатом которого является Парето-архив A – множество недоминируемых решений [14, 15]. Формирование архива A завершает оптимизационный этап, однако для практического применения полученного решения требуется определить порядок его преобразования в правило распределения и управляющие команды узлам-исполнителям. Из архива необходимо выбрать рабочее решение x^* , после чего преобразовать его в правило распределения и управляющие команды.

Дальнейшая задача состоит в переходе от полученного оптимизационного результата к практическому управлению сетевыми ресурсами.

Входные данные и состояние сети

Для применения АРГМОП управляющий контур должен располагать актуальными сведениями о задачах и доступных ресурсах. Эти сведения относятся к определённому моменту времени и изменяются по мере выполнения назначений, освобождения узлов, изменения их состояния и поступления новых задач. Текущее состояние распределённой вычислительной сети обозначим $S(t)$.

Под состоянием сети $S(t)$ далее понимается совокупность актуальных сведений о задачах, доступных узлах, временных характеристиках и действующих ограничениях, что схематично представлено на рисунке 1 [3, 7, 20].

К блоку «Задачи» относятся исходные вычислительные задачи и, при необходимости, подзадачи, ожидающие распределения между узлами сети.

Блок «Узлы сети» описывает множество вычислительных узлов, которые могут участвовать в текущем цикле распределения. Для каждого узла учитываются его доступность, занятость и загрузка. Под доступностью понимается способность узла принять назначенную подзадачу и корректно вернуть результат. Занятость отражает факт выполнения узлом ранее назначенной работы в рассматриваемый момент времени. Загрузка характеризует степень текущего использования вычислительного ресурса и может определяться по выбранным в реализации показателям [4, 8].

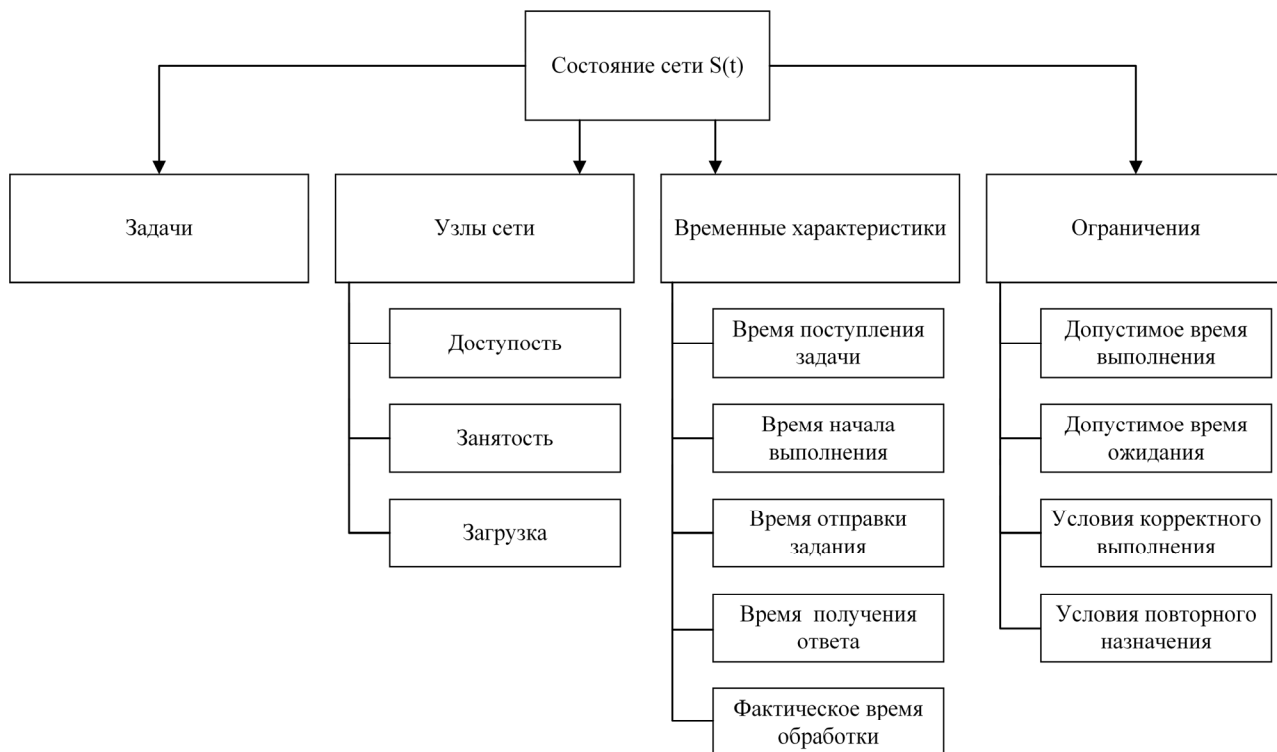


Рис. 1. Состав состояния сети $S(t)$

Блок «Временные характеристики» включает параметры, используемые для оценки текущего режима функционирования сети: время отправки задания, время получения ответа и фактическое время обработки подзадачи. Эти сведения позволяют оценивать временные затраты распределения и уточнять текущее состояние сети [1, 3].

Блок «Ограничения» задаёт условия допустимого распределения. К ним относятся допустимое время выполнения, допустимое время ожидания, условия корректного завершения и условия повторного назначения подзадачи. Совокупность указанных ограничений определяет, какие варианты распределения могут рассматриваться как допустимые в текущих условиях функционирования сети [5, 21].

Помимо параметров текущего состояния сети, при формировании входных данных используются заданные параметры оптимизационного цикла АРГМОП.

Архитектура применения АРГМОП

Архитектура применения АРГМОП основывается на разделении процессов принятия решения и непосредственного исполнения вычислительной работы. На управляющем уровне выполняется подготовка задачи, формируется текущее состояние сети $S(t)$, запускается модуль АРГМОП и формируется Парето-архив A . Из архива выбирается рабочее решение x^* , на основе которого формируется правило распределения. Далее средствами сетевой диспетчеризации правило преобразуется в управляющие команды, адресуемые соответствующим узлам-исполнителям. На исполнительном уровне узлы принимают назначения, выполняют соответствующие подзадачи и возвращают результаты и служебные сведения, используемые для последующего обновления состояния сети. Указанная иерархия представлена на рисунке 2 [9, 18].

В архитектуре можно выделить несколько функциональных компонентов. Компонент подготовки задачи определяет параметры исходной вычислительной задачи и, если её структура допускает декомпозицию, формирует набор подзадач, предназначенных для последующего распределения. Компонент формирования текущего состояния сети объединяет сведения о задачах, узлах-исполнителях, их состоянии, временных характеристиках и действующих ограничениях в составе $S(t)$. Оптимизационный модуль АРГМОП формирует Парето-архив A , содержащий множество недоминируемых вариантов распределения [7]. Из него выбирается рабочее решение x^* , соответствующее текущим требованиям. Выбранное решение преобразуется в правило распределения, после чего сетевой диспетчер формирует адресованные управляющие команды для узлов-исполнителей.

Представленная архитектура не привязана к конкретной аппаратной платформе или физическому интерфейсу передачи данных. Существенными для методики являются функциональные этапы подготовки задачи, формирования состояния сети, многокритериальной оптимизации, выбора рабочего решения, формирования правила распределения и передачи управляющих воздействий узлам-исполнителям [22]. После выполнения назначенных подзадач результаты и сведения о фактическом состоянии исполнителей могут использоваться для актуализации $S(t)$ и последующего цикла распределения [19, 20].



Рис. 2. Архитектура применения АРГМОП в распределённой вычислительной сети

В конкретной программно-аппаратной реализации функции управляющего уровня могут выполняться одним компонентом либо распределяться между несколькими компонентами. Например, подготовка и декомпозиция исходной задачи могут выполняться контролирующим устройством, тогда как функции сетевой диспетчеризации, передачи команд и получения результатов – master-узлом. Такое разделение носит функциональный характер и не изменяет общей последовательности применения АРГМОП.

Формирование правила распределения

Центральным этапом методики является преобразование результата многокритериальной оптимизации в прикладное правило распределения. Парето-архив A содержит несколько недоминируемых вариантов, поэтому первым шагом после завершения работы АРГМОП становится выбор рабочего решения x^* [14, 15]. Выбор осуществляется в соответствии с текущими требованиями цикла распределения: приоритет может отдаваться минимизации средней задержки, максимизации производительности алгоритма распределения ресурсов или компромиссному варианту.

Рабочее решение x^* не является непосредственной командой узлу-исполнителю, оно описывает выбранный вариант распределения в обозначениях АРГМОР. Для практического применения его необходимо преобразовать в правило распределения. Правило распределения определяет, какие подзадачи каким исполнителям передаются, в каком объёме, в каком порядке и при каких условиях контроля.

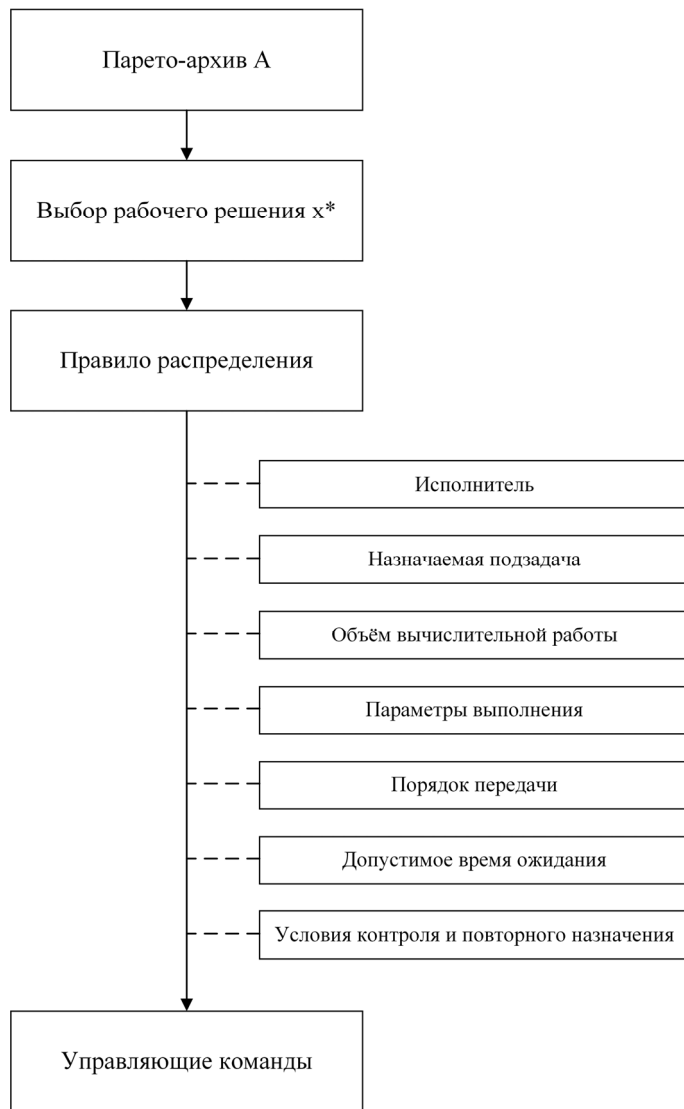


Рис. 3. Переход от результата оптимизации к управлению исполнителями

Таким образом, необходимо чётко различать три уровня представления результата. Первый уровень — оптимизационное решение x^* , выбранное из Парето-архива. Второй уровень — правило распределения, которое интерпретирует выбранное решение с учётом текущего состояния сети. Третий уровень — управляющие команды, представляющие собой адресованные узлам-исполнителям сообщения, посредством которых правило реализуется на практике. Такое разграничение позволяет избежать смешения математического результата оптимизации и сетевого механизма его исполнения.

Правило распределения включает указание исполнителя, назначаемую подзадачу, объём вычислительной работы, параметры выполнения, порядок передачи, допустимое время

ожидания результата, а также условия контроля и повторного назначения. Конкретный состав и способ представления этих элементов могут уточняться в зависимости от архитектуры и программно-аппаратной реализации вычислительной сети.

На основе сформированного правила подготавливаются управляющие команды для узлов-исполнителей. Команда должна содержать, по крайней мере, идентификатор или адрес исполнителя, идентификатор подзадачи, объём работы, необходимые параметры вычисления и служебные признаки контроля. Конкретный формат управляющей команды определяется используемым протоколом и программно-аппаратной реализацией сети [18, 20].

Последовательность применения методики

Последовательность применения АРГМОР представляет собой совокупность взаимосвязанных этапов, охватывающих путь от постановки вычислительной задачи до получения результатов и актуализации состояния сети. Она описывает взаимодействие управляющего и исполнительного уровней сети, тогда как внутренние процедуры многокритериального оптимизационного поиска выполняются внутри модуля АРГМОР.



Рис. 4. Последовательность применения АРГМОР

Полный цикл начинается с постановки вычислительной задачи управляющим узлом. Затем выполняются инициализация сети и определение множества потенциальных исполнителей. На следующем этапе собираются актуальные параметры вычислительной среды и формируется текущее состояние сети $S(t)$, включающее сведения о задачах, доступности, занятости и загрузке узлов, временных характеристиках и действующих ограничениях. Если структура исходной задачи допускает декомпозицию, она разделяется на подзадачи, после чего формируются входные данные для АРГМОР [3, 4, 7].

В результате оптимизационного цикла АРГМОР формируется Парето-архив A , содержащий множество недоминируемых вариантов распределения. Из него выбирается рабочее решение x^* , соответствующее текущим требованиям по средней задержке и производительности алгоритма распределения ресурсов. Выбранное решение преобразуется в правило распределения, на основании которого формируются управляющие команды, адресованные узлам-исполнителям.

После передачи команд исполнители выполняют назначенные подзадачи и возвращают результаты и служебные сведения. На их основе рассчитываются фактические параметры работы и актуализируется состояние сети $S(t+1)$. Обновлённые сведения используются при последующем цикле распределения, что обеспечивает обратную связь между результатами исполнения и очередным запуском АРГМОР.

Управляющий цикл и обратная связь

Программное разделение сетевых ресурсов в динамической вычислительной среде не может ограничиваться однократным формированием распределения. В процессе выполнения задач меняются доступность, занятость и загрузка узлов, могут поступать новые задания, часть исполнителей может выходить из числа доступных ресурсов, а отдельные назначения могут завершаться ошибкой. Поэтому применение АРГМОР должно рассматриваться как замкнутый управляющий цикл, включающий обратную связь [1, 3, 4].

Общий цикл можно описать последовательностью: состояние сети $S(t)$ – запуск АРГМОР – Парето-архив A – выбор рабочего решения x^* – правило распределения – управляющие команды – результаты выполнения – обновлённое состояние сети $S(t+1)$.

Обратная связь выполняет две основные функции. Во-первых, она позволяет фиксировать успешность или неуспешность текущих назначений и получать собственно вычислительные результаты. Во-вторых, она служит источником сведений для актуализации состояния сети. После завершения или срыва исполнения изменяются характеристики доступности, занятости и загрузки ресурсов; кроме того, могут уточняться временные характеристики взаимодействия. Всё это используется при подготовке следующего цикла применения метода.

Новый цикл следует запускать только при изменениях, способных повлиять на действующее правило распределения. К таким событиям относятся поступление новой задачи, изменение доступности исполнителя, существенное изменение загрузки, перегрузка, превышение допустимого времени ожидания, ошибка выполнения либо отсутствие корректного результата. При возникновении одного из указанных событий обновлённое состояние $S(t+1)$ используется как вход для нового запуска АРГМОР. Если же значимых изменений не произошло, система может продолжать исполнение уже сформированного правила или перейти в режим ожидания.

Свойства и ограничения методики

Предлагаемая методика обладает рядом свойств, непосредственно вытекающих из её структуры.

Первым свойством является многокритериальность: формирование распределения опирается на Парето-архив, в котором одновременно учитываются средняя задержка и производительность алгоритма распределения ресурсов. Это

позволяет не сводить задачу к единственной интегральной оценке, а сохранять несколько недоминируемых вариантов и выбирать из них рабочее решение в соответствии с текущими приоритетами [14, 15–19].

Вторым свойством является адаптивность методики к изменению состояния вычислительной сети. Состояние сети актуализируется по мере поступления новых данных, а правило распределения может формироваться повторно при изменении условий функционирования. Результаты выполнения используются для обновления состояния сети и подготовки следующего цикла АРГМОР [1, 3, 4].

Третьим важным свойством является разделение оптимизационного и исполнительного уровней. АРГМОР отвечает за формирование множества недоминируемых решений, после чего выбранный вариант преобразуется в правило распределения и далее – в управляющие команды. Поэтому методика не зависит от конкретного физического интерфейса передачи данных и может применяться в сетях с различной аппаратной организацией. Ещё одним свойством является возможность работы с неоднородными исполнителями: различия между узлами учитываются в составе текущего состояния сети и не требуют изменения общей последовательности применения метода [7–9].

Наряду со свойствами необходимо выделить и ограничения методики. Во-первых, её применение требует достаточно актуального и достоверного состояния сети. Если сведения о доступности, занятости или загрузке существенно расходятся с фактическими условиями, выбранное решение может потерять актуальность ещё до начала исполнения. Во-вторых, повторная оптимизация требует вычислительных затрат, поэтому слишком частый запуск АРГМОР способен сам по себе увеличивать служебные временные издержки. В-третьих, методика ориентирована на наличие централизованного управляющего контура, что упрощает принятие согласованных решений, но предъявляет повышенные требования к производительности и устойчивости управляющего уровня. Наконец, качество практического применения существенно зависит от корректности подготовки подзадач и от полноты входных данных.

Демонстрационный пример применения методики

Для количественной иллюстрации её применения далее рассматривается четырёхузловой экспериментальный сценарий, позволяющий проследить переход от текущего состояния сети к конкретному распределению вычислительной нагрузки, а также сопоставить АРГМОР с PSO, GA и решением, выбранным из точного Парето-фронта, сформированного полным перебором.

Для пояснения практического применения предложенной методики рассмотрим задачу оценки числа π методом Монте-Карло [10]. Геометрической основой данной задачи является определение площади четверти единичного круга. Площадь этой области может быть представлена выражением:

$$\pi = 4 \int_0^1 \sqrt{1-x^2} dx. \quad (5)$$

При оценке числа π методом Монте-Карло генерируются независимые случайные точки (x, y) , равномерно распределённые внутри единичного квадрата:

$$0 \leq x \leq 1, \quad 0 \leq y \leq 1. \quad (6)$$

Для каждой сгенерированной точки проверяется условие принадлежности четверти единичной окружности:

$$x^2 + y^2 \leq 1. \quad (7)$$

Следовательно отдельное испытание представляет собой генерацию одной случайной точки и проверку условия её принадлежности четверти единичной окружности. Поскольку результаты отдельных испытаний не зависят друг от друга, общий объём вычислений может быть разделён на независимые части и распределён между вычислительными узлами.

Для демонстрации работы методики принимаются следующие допущения.

1. Исходная вычислительная задача содержит 50 000 независимых испытаний Монте-Карло.

2. Минимальной единицей вычислительной работы, передаваемой отдельному узлу, является вычислительный квант. Под вычислительным квантом далее понимается группа из 250 независимых испытаний Монте-Карло.

3. В рассматриваемой вычислительной сети используются четыре узла-исполнителя.

При указанных допущениях общее число вычислительных квантов M составляет:

$$M = \frac{50\,000}{250} = 200. \quad (8)$$

Следовательно, задача распределения состоит в назначении 200 вычислительных квантов четырём узлам-исполнителям. Объём вычислений, назначаемый каждому узлу, определяется непосредственно в процессе формирования решения.

Вариант распределения вычислительной нагрузки между K узлами представим вектором:

$$z = (z_1, z_2, \dots, z_K), \quad (9)$$

где z_i – число вычислительных квантов, назначаемых i -му узлу.

Для допустимого варианта распределения выполняются условия

$$\sum_{i=1}^K z_i = M. \quad (10)$$

$$z_i \in \mathbb{Z}, \quad z_i \geq 1, \quad i = 1, \dots, K. \quad (11)$$

В рассматриваемом четырёхузловом примере $K = 4$, $M = 200$, поэтому:

$$z = (z_1, z_2, z_3, z_4), \quad (12)$$

а допустимые варианты распределения удовлетворяют условиям:

$$z_1 + z_2 + z_3 + z_4 = 200 \quad (13)$$

Поскольку один вычислительный квант содержит 250 испытаний, фактический объём вычислительной работы N_i , назначаемой i -му узлу, составляет:

$$N_i = 250 * z_i. \quad (14)$$

При любом допустимом варианте распределения выполняется

$$\sum_{i=1}^4 N_i = 50\,000 \quad (15)$$

то есть общий объём исходной вычислительной задачи остаётся неизменным.

Распределение вычислительных квантов между узлами представляет собой известную комбинаторную задачу [23] определения числа положительных целочисленных решений уравнения:

$$z_1 + z_2 + \dots + z_K = M. \quad (16)$$

При условии $z_i \geq 1$ число различных вариантов распределения N_{var} определяется биномиальным коэффициентом:

$$N_{var}(M, K) = \binom{M-1}{K-1} = \frac{(M-1)!}{(K-1)!(M-K)!} \quad (17)$$

Для $M = 200$ квантов и $K = 4$ узлов число допустимых вариантов составляет

$$N_{var}(200, 4) = \binom{199}{3} = \frac{199!}{3!196!} = 1\,293\,699 \quad (18)$$

Таким образом, даже при использовании всего четырёх вычислительных узлов и однородных вычислительных квантов полный перебор должен проверить более 1,29 млн вариантов распределения.

Поскольку при увеличении объёма вычислительной задачи число распределяемых квантов существенно превышает число вычислительных узлов $M \gg K$, для оценки роста числа вариантов распределения целесообразно использовать приближение биномиального коэффициента при $n \gg k$. При $k = o(n)$, то есть $k/n \rightarrow 0$, биномиальный коэффициент имеет вид [24]:

$$\binom{n}{k} \sim \left(\frac{ne}{k}\right)^k \frac{1}{\sqrt{2\pi k}} \exp\left(-\frac{k^2}{2n}(1 + o(1))\right). \quad (19)$$

Для рассматриваемой задачи $n = M - 1$, $k = K - 1$. Поэтому число допустимых вариантов распределения может быть оценено как:

$$N_{var}(M, K) = \binom{M-1}{K-1} \sim \left(\frac{e(M-1)}{K-1}\right)^{K-1} \frac{1}{\sqrt{2\pi(K-1)}} \exp\left(-\frac{(K-1)^2}{2(M-1)}(1 + o(1))\right). \quad (20)$$

Полученная оценка показывает, что при увеличении числа распределяемых квантов и вычислительных узлов количество допустимых вариантов распределения быстро возрастает, вследствие чего применение полного перебора становится вычислительно значительно более затратным.

Рассмотрим отдельный цикл функционирования распределённой вычислительной сети. Помимо приведённых выше общих допущений, для данного цикла задаётся конкретное начальное состояние узлов.

В момент поступления новой вычислительной задачи все четыре узла уже выполняют ранее назначенную работу. Остаточное время до освобождения узлов составляет соответственно 2,9846; 1,9214; 1,6115; 0,6010 с. Поступившая задача не вытесняет ранее выполняемые вычисления и не выполняется параллельно с ними на одном вычислительном узле. Каждый узел сначала завершает ранее назначенную работу, после чего начинает выполнение назначенных ему квантов новой задачи.

Вследствие этого алгоритм распределения должен определить такой вектор

$$z = (z_1, z_2, z_3, z_4) \quad (21)$$

который учитывает различное остаточное время занятости вычислительных узлов.

Для заданного начального состояния в первом рассматриваемом цикле АРГМОР формирует вектор распределения:

$$z^{(1)} = (2, 44, 57, 97) \quad (22)$$

Верхний индекс (1) обозначает номер рассматриваемого цикла распределения, а элементы вектора определяют число квантов, назначенных соответственно первому, второму, третьему и четвёртому вычислительным узлам.

При размере вычислительного кванта 250 испытаний указанному вектору соответствует распределение - (500; 11 000; 14 250; 24 250).

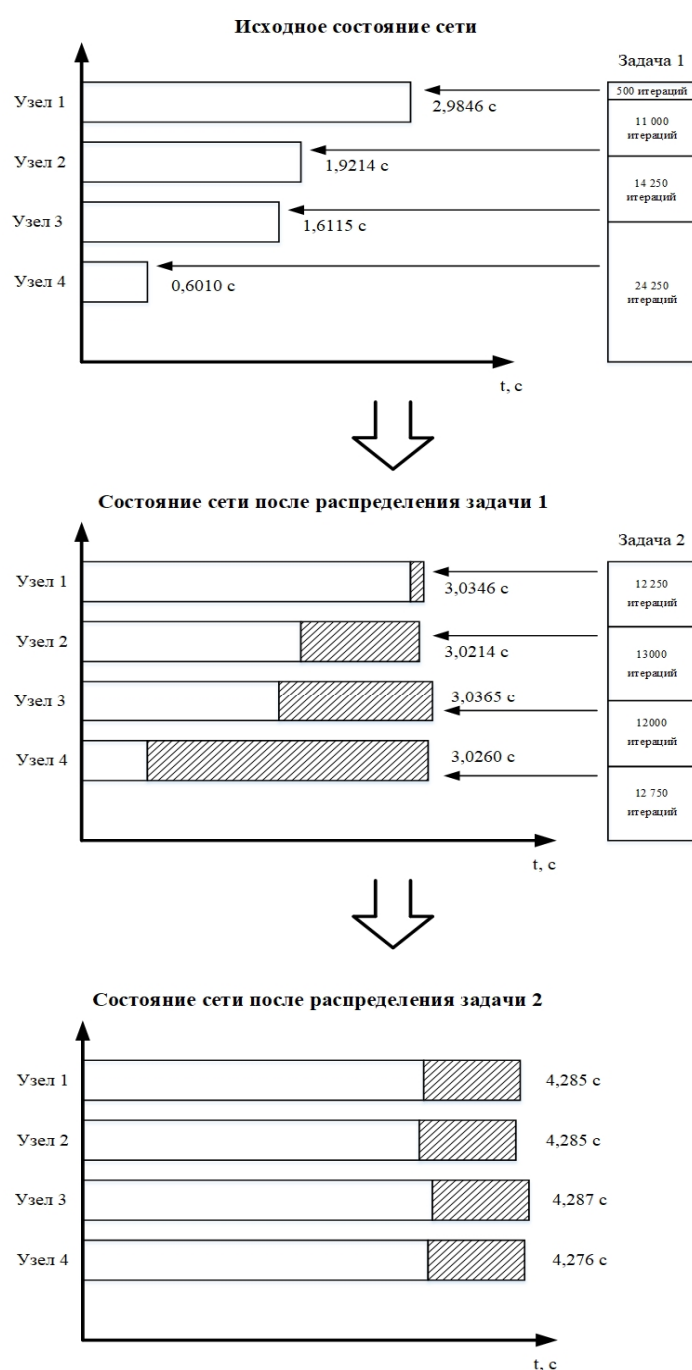


Рис. 5. Пример последовательного распределения двух вычислительных задач между четырьмя узлами

Наиболее занятый первый вычислительный узел получает только два кванта, то есть 500 новых испытаний, тогда как четвёртый узел, имеющий минимальное остаточное время занятости, получает 97 квантов, соответствующих 24 250 испытаниям. Тем самым сформированное распределение учитывает текущее состояние сети и компенсирует исходную неравномерность занятости исполнителей.

На рисунке 5 показан последовательный характер применения методики. Для первой поступившей задачи распределение формируется исходя из остаточной занятости вычислительных узлов. После назначения вычислительных квантов состояние сети изменяется. Поэтому при поступлении следующей задачи используется уже не исходное, а обновлённое состояние вычислительной сети.

В рассматриваемом примере после первого назначения прогнозируемая занятость узлов составляет:

$$S(t_1) = (3,035; 3,021; 3,037; 3,026) \quad (23)$$

При поступлении второй вычислительной задачи АРГМОР запускается повторно уже для этого обновлённого состояния. Поскольку после первого распределения занятость узлов практически выровнена, следующая однотипная задача может распределяться близко к равномерному варианту. Таким образом, правило распределения не является заранее зафиксированным: оно формируется заново при поступлении каждой новой задачи с учётом состояния сети в соответствующий момент времени.

Для количественной оценки методики эксперимент проводился серией из 50 воспроизводимых запусков. В каждом запуске АРГМОР, PSO, GA и полный перебор получали одинаковое исходное состояние четырёх узлов, одинаковый объём поступающей вычислительной задачи и одинаковые случайные отклонения фактического времени исполнения.

На рисунке 6 представлены значения задержки, полученные в каждом из 50 экспериментальных запусков. Каждая точка соответствует одному состоянию вычислительной сети и результату распределения поступившей задачи при использовании соответствующего метода. Пунктирная линия соответствует рабочему решению, выбранному из точного Парето-фронта, сформированного полным перебором, и используется в качестве эталона для рассматриваемой четырёхузловой модели.

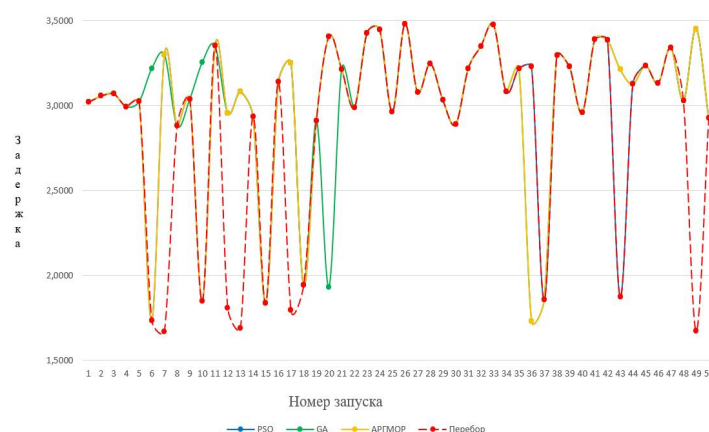


Рис. 6. Задержка алгоритмов распределения ресурсов для серии из 50 запусков

На рисунке 7 представлена производительность распределённой системы при использовании четырёх рассматриваемых способов формирования распределения.

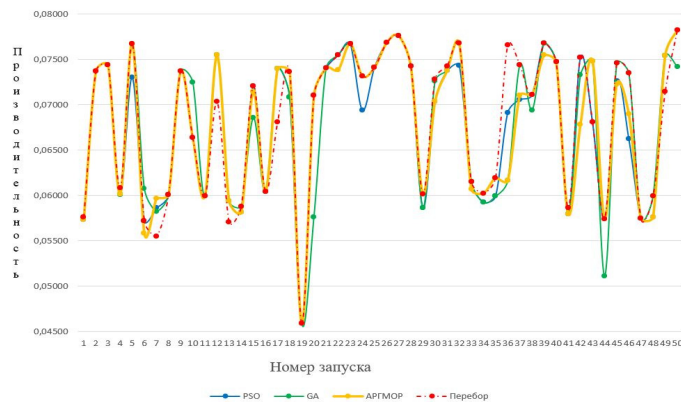


Рис. 7. Производительность алгоритмов распределения ресурсов для серии из 50 запусков

По результатам серии из 50 запусков были рассчитаны средние значения задержки и производительности для АРГМОП, PSO, GA и полного перебора. Полученные значения представлены в таблице 1.

Таблица 1

Средние значения задержки и производительности по результатам 50 запусков

Метод	Средняя задержка D, с	Средняя производительность P, задач/с
АРГМОП	3,00506	0,067877
PSO	3,00825	0,067732
GA	3,03338	0,067850
Полный перебор	2,86006	0,068230

В рассматриваемой четырёхузловой модели пространство решений сравнительно невелико, а задача распределения в четырёхузловой модели может быть достаточно эффективно решена всеми сравниваемыми эвристическими методами. Поэтому различия между АРГМОП, PSO и GA в данном эксперименте имеют ограниченную величину. С увеличением числа вычислительных узлов и размерности пространства распределений возрастает количество допустимых вариантов назначения вычислительной нагрузки и усложняется поиск сбалансированного решения. В этих условиях преимущество гибридного механизма АРГМОП может проявляться более заметно за счёт сочетания роевого поиска, внешнего Парето-архива и генетической коррекции при стагнации.

Дополнительным направлением исследования является переход от распределения однородных вычислительных квантов к задаче назначения неоднородных вычислительных блоков. В такой постановке отдельные подзадачи могут иметь различную вычислительную трудоёмкость, а узлы — различную производительность, остаточную занятость и временные характеристики взаимодействия. Решение должно определять не только объём нагрузки каждого исполнителя, но и состав конкретных подзадач, назначаемых каждому узлу. Такое усложнение существенно увеличивает размер и неоднородность пространства решений и позволяет более полно оценить эффективность адаптивной гибридизации АРГМОП по сравнению с отдельным применением PSO и GA.

Заключение

В статье предложена методика применения ранее разработанного Адаптивного Роево-Генетического Метода Оптимизации Распределения ресурсов АРГМОП в распределённой вычислительной сети. Методика определяет порядок включения АРГМОП в контур управления распределённой вычислительной сетью и переход от текущего состояния сети и результатов оптимизации к правилу распределения и управляющим командам.

Для описания методики используются представление допустимого решения x , два целевых параметра – средняя задержка $\bar{D}$ и производительность P , обозначение текущего состояния сети $S(t)$, Парето-архива A и выбранного рабочего решения x^* . На их основе определена последовательность перехода от текущего состояния сети к правилу распределения и управляющим воздействиям.

Определены состав текущего состояния сети, архитектура применения АРГМОП, порядок выбора рабочего решения из Парето-архива, формирование правила распределения, построение управляющих команд и общий управляющий цикл с обратной связью. Показано, что в рассматриваемой централизованной архитектуре правило распределения может быть функционально связано как с подготовкой вычислительной задачи, так и с сетевой диспетчеризацией исполнителей. Тем самым устанавливается связь между многокритериальным оптимизационным алгоритмом и практическим контуром управления распределённой вычислительной сетью.

Для иллюстрации применения методики рассмотрена работа распределённой вычислительной сети из четырёх узлов в условиях различной остаточной занятости исполнителей и последовательного поступления новых вычислительных задач. Показано, что решение АРГМОП непосредственно определяет объём вычислительной работы, назначаемый каждому узлу, а после формирования распределения состояние сети обновляется и используется при поступлении следующей задачи. На малой четырёхузловой модели результаты метода сопоставлены с PSO, GA и решением, выбранным из точного Парето-фронта, сформированного полным перебором. Проведение серии из 50 запусков позволило непосредственно оценить два принятых в работе показателя – среднюю задержку и производительность алгоритма распределения ресурсов. Полученные результаты подтверждают эффективность практическую применимость предложенной методики для динамического формирования правил распределения в условиях изменяющегося состояния вычислительной сети.

Ограниченная размерность рассматриваемой модели приводит к относительно небольшим различиям между АРГМОП и отдельным применением PSO и GA. Дальнейшее исследование целесообразно направить на оценку масштабируемости метода при увеличении числа вычислительных узлов, поскольку рост размерности существенно расширяет пространство допустимых распределений.

Финансовая поддержка

Статья подготовлена при финансовой поддержке Российского научного фонда, проект № 22-19-00305-П <https://rscf.ru/project/22-19-00305/> "Пространственно-временные стохастические модели беспроводных сетей с большим числом пользователей".

Литература

1. Оленев В.Л. Эволюция коммуникационных технологий через автоматизацию процессов разработки протоколов // Известия Самарского научного центра Российской академии наук, т. 28, № 2, 2026, С. 185-195
2. Суворова Е. А. Динамическое распределение сетевых ресурсов в сети SpaceFibre на основе мультиагентных самообучающихся систем // Наукоемкие технологии в космических исследованиях Земли. 2025. Т. 17, № 4. С. 15–27.
3. Суворова Е. А. Оценка состояния сети SpaceFibre для динамического распределения ресурсов буферного пространства сетевых устройств между потоками данных // Системный анализ и логистика. 2026. № 1 (49). С. 3-13.
4. Zhang Q., Zhang C., Zhao J., Wang D., Xu W. Dynamic Resource Allocation for Multi-Access Edge Computing in Urban Rail Transit // IEEE Transactions on Vehicular Technology. 2025. Vol. 74, No. 2. P. 3296-3310.
5. Huang Z., Wu X., Dong S. Multi-objective task offloading for highly dynamic heterogeneous Vehicular Edge Computing: An efficient reinforcement learning approach // Computer Communications. 2024. Vol. 225. P. 27-43.
6. Liu Y., Yan B., Yu J. Multi objective task resource allocation method based on hierarchical Bayesian adaptive sparsity for edge computing in low voltage stations // IET Cyber-Physical Systems: Theory & Applications. 2024. Vol. 9, No. 1. P. 63-71.
7. Aslanpour M. S., Toosi A. N., Cheema M. A., Chhetri M. B., Salehi M. A. Load balancing for heterogeneous serverless edge computing: A performance-driven and empirical approach // Future Generation Computer Systems. 2024. Vol. 154. P. 266-280.
8. Aminu J., Latip R., Hanafi Z. M., Kamarudin S., Gabi D. Efficient Task Allocation for Energy and Execution Time Trade-Off in Edge Computing Using Multi-Objective IPSO // Computers, Materials & Continua. 2025. Vol. 84, No. 2. P. 2989-3011.
9. Li H., Yu J., Cao L., Zhang Q., Song Z., Hou S. Multi-agent reinforcement learning based computation offloading and resource allocation for LEO Satellite edge computing networks // Computer Communications. 2024. Vol. 222. P. 268-276.
10. Петров А. Ю. Программно-аппаратная архитектура демонстратора алгоритма АРГМОП // Инновационное приборостроение. 2026. Т. 5, № 1. С. 54-58.
11. Петров А. Ю. Гибридный Алгоритм PSO-GA в Вычислительных системах // Аэрокосмическое приборостроение и эксплуатационные технологии: Пятая Международная научная конференция. Сборник докладов. В 2-х частях, СПб. Том Часть 2. Санкт-Петербург: Санкт-Петербургский государственный университет аэрокосмического приборостроения, 2025. С. 220-223.
12. Петров А.Ю., Оленев В.Л. Формализация гибридного метода распределения задач с использованием Сетей Петри // Научно-технический вестник Поволжья. 2025. № 6. С. 228-232.
13. Kennedy J., Eberhart R. Particle swarm optimization // Proceedings of the IEEE International Conference on Neural Networks. Perth: IEEE, 1995. P. 1942-1948.
14. Deb K., Pratap A., Agarwal S., Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II // IEEE Transactions on Evolutionary Computation. 2002. Vol. 6, No. 2. P. 182-197.
15. Coello Coello C. A., Pulido G. T., Lechuga M. S. Handling multiple objectives with particle swarm optimization // IEEE Transactions on Evolutionary Computation. 2004. Vol. 8, No. 3. P. 256-279.
16. Kelly F. P. Charging and rate control for elastic traffic // European Transactions on Telecommunications. 1997. Vol. 8, No. 1. P. 33-37.
17. Ghodsi A., Zaharia M., Hindman B., Konwinski A., Shenker S., Stoica I. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types // Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation. Boston: USENIX Association, 2011. P. 323-336.
18. McKeown N., Anderson T., Balakrishnan H. et al. OpenFlow: Enabling Innovation in Campus Networks // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38, No. 2. P. 69-74.
19. Nagalakshmi B., Subramanian S. An efficient multi-objective task scheduling in edge computing using adaptive honey badger optimization // International Journal of Web Engineering and Technology. 2024. Vol. 19, No. 2. P. 110-126.
20. Pei X., Hu Y., Dong Y., Li D. Task Scheduling in Edge Network System Using Resource Interleaving and Reinforcement Learning // IEEE Transactions on Information and Systems. 2025. Vol. E108.D, No. 9. P. 1142-1145.
21. Liu X., Lv J., Kim B.-G. et al. Cooperative Digital Healthcare Task Scheduling and Resource Management in Edge Intelligence Systems // Tsinghua Science and Technology. 2025. Vol. 30, No. 2. P. 926-945.
22. Olenov V., Lavrovskaya I., Korobkov I., Sheynin Yu. Design and Simulation of Onboard SpaceWire Networks // Conference of Open Innovations Association, FRUCT. 2019. №. 24. P. 291-299.
23. Виленкин Н. Я., Виленкин А. Н., Виленкин П. А. Комбинаторика. 6-е изд., стер. М.: МЦНМО: ФИМА, 2017. 400 с.
24. Spencer J. Asymptopia. Providence, RI: American Mathematical Society, 2014. 189 p.

METHODOLOGY FOR APPLYING ARGMOR TO SOFTWARE-BASED PARTITIONING OF NETWORK RESOURCES IN A DISTRIBUTED COMPUTING NETWORK

Alexey Yu. Petrov, Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia, palexeq1998@yandex.ru
Valentin L. Olenev, Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia, valentin.olenov@guap.ru
Andrey M. Tyurlikov, Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia, turlikov@guap.ru

Abstract

This study addresses a methodology for applying the Adaptive Swarm-Genetic Method for Resource Allocation Optimization (ARGMOR) to software-based partitioning of network resources in a distributed computing network. The aim is to formalize the transition from multi-objective optimization results to practical control of worker nodes, taking into account changes in the computing environment. The methodology is intended for centrally managed distributed computing networks requiring dynamic software-based allocation of tasks and resources. The research approach is based on a representation of the current network state $S(t)$, comprising information on tasks, node availability, occupancy and load, timing characteristics, and the applicable constraints and parameters that determine the feasibility of resource allocation options. It also uses a Pareto archive A generated by ARGMOR according to two criteria: average delay and throughput achieved using the resource allocation algorithm. An architecture separating the control and execution layers is proposed, and a sequence is defined for selecting a working solution x^* , converting it into an allocation rule, and subsequently generating control commands addressed to individual nodes. A closed-loop control process is established: collecting network state information, running ARGMOR, generating the Pareto archive, selecting a solution, distributing subtasks, executing them, receiving results, and updating the network state to $S(t+1)$. Optimization is found to be appropriate when triggered by events that may affect the applicability of the current allocation rule, including the arrival of a new task, changes in worker availability or load, overload, excessive waiting time, or an execution error. The methodology supports multiple optimization criteria, adaptation to network dynamics, heterogeneous worker nodes, and independence from any specific hardware platform or data transmission protocol. Its practical application is illustrated using a four-node model of a distributed computing network with successive arrivals of tasks for estimating τ by the Monte Carlo method. Across 50 runs, ARGMOR results are compared with those obtained using PSO, GA, and a solution selected from the exact Pareto front generated by exhaustive search.

Keywords: software-based partitioning of network resources; distributed computing network; ARGMOR; multi-objective optimization; Pareto archive; allocation rule; control node; worker node; dynamic resource allocation; feedback.

References

- [1] V. L. Olenov, "Evolution of communication technologies through automation of protocol development processes," *Izvestia of the Samara Scientific Center of the Russian Academy of Sciences*, vol. 28, no. 2, pp. 185-195, 2026.
- [2] E. A. Suvorova, "Dynamic allocation of network resources in the SpaceFibre network based on multiagent self-learning systems," *H&ES Research*, vol. 17, no. 4, pp. 15-27, 2025.
- [3] E. A. Suvorova, "The SpaceFibre network state evaluation for dynamic allocation of buffer space resources for network devices between data flows," *System Analysis and Logistics*, no. 1 (49), pp. 3-13, 2026.
- [4] Q. Zhang, C. Zhang, J. Zhao, D. Wang, and W. Xu, "Dynamic resource allocation for multi-access edge computing in urban rail transit," *IEEE Trans. Veh. Technol.*, vol. 74, no. 2, pp. 3296-3310, 2025.
- [5] Z. Huang, X. Wu, and S. Dong, "Multi-objective task offloading for highly dynamic heterogeneous vehicular edge computing: An efficient reinforcement learning approach," *Comput. Commun.*, vol. 225, pp. 27-43, 2024.
- [6] Y. Liu, B. Yan, and J. Yu, "Multi objective task resource allocation method based on hierarchical Bayesian adaptive sparsity for edge computing in low voltage stations," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 9, no. 1, pp. 63-71, 2024.
- [7] M. S. Aslanpour, A. N. Toosi, M. A. Cheema, M. B. Chhetri, and M. A. Salehi, "Load balancing for heterogeneous serverless edge computing: A performance-driven and empirical approach," *Future Gener. Comput. Syst.*, vol. 154, pp. 266-280, 2024.
- [8] J. Aminu, R. Latip, Z. M. Hanafi, S. Kamarudin, and D. Gabi, "Efficient task allocation for energy and execution time trade-off in edge computing using multi-objective IPSO," *Comput., Mater. Continua*, vol. 84, no. 2, pp. 2989-3011, 2025.
- [9] H. Li, J. Yu, L. Cao, Q. Zhang, Z. Song, and S. Hou, "Multi-agent reinforcement learning based computation offloading and resource allocation for LEO satellite edge computing networks," *Comput. Commun.*, vol. 222, pp. 268-276, 2024.
- [10] A. Yu. Petrov, "Hardware and software architecture of an ARGMOR algorithm demonstrator," *Innovative Instrumentation*, vol. 5, no. 1, pp. 54-58, 2026.
- [11] A. Yu. Petrov, "A hybrid PSO-GA algorithm in computing systems," in *Proc. 5th Int. Sci. Conf. Aerospace Instrumentation and Operational Technologies*, pt. 2. Saint Petersburg, Russia: Saint Petersburg State University of Aerospace Instrumentation, 2025, pp. 220-223.
- [12] A. Yu. Petrov and V. L. Olenov, "Formalization of a hybrid task allocation method using Petri nets," *Scientific and Technical Volga Region Bulletin*, no. 6, pp. 228-232, 2025.

- [13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.
- [15] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256-279, 2004.
- [16] F. P. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33-37, 1997.
- [17] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in *Proc. 8th USENIX Symp. Netw. Syst. Des. Implement. (NSDI)*, Boston, MA, USA, 2011, pp. 323-336.
- [18] N. McKeown et al., "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69-74, 2008.
- [19] B. Nagalakshmi and S. Subramanian, "An efficient multi-objective task scheduling in edge computing using adaptive honey badger optimisation," *Int. J. Web Eng. Technol.*, vol. 19, no. 2, pp. 110-126, 2024.
- [20] X. Pei, Y. Hu, Y. Dong, and D. Li, "Task scheduling in edge network system using resource interleaving and reinforcement learning," *IEICE Trans. Inf. Syst.*, vol. E108-D, no. 9, pp. 1142-1145, 2025.
- [21] X. Liu et al., "Cooperative digital healthcare task scheduling and resource management in edge intelligence systems," *Tsinghua Sci. Technol.*, vol. 30, no. 2, pp. 926-945, 2025.
- [22] V. Olenov, I. Lavrovskaya, I. Korobkov, and Yu. Sheynin, "Design and simulation of onboard SpaceWire networks," in *Proc. 24th Conf. Open Innovations Assoc. (FRUCT)*, 2019, pp. 291-299.
- [23] N. Ya. Vilenkin, A. N. Vilenkin, and P. A. Vilenkin, *Combinatorics*, 6th ed. Moscow, Russia: MCCME and FIMA, 2017.
- [24] J. Spencer, *Asymptopia*. Providence, RI, USA: American Mathematical Society, 2014.

Information about authors:

Alexey Yu. Petrov, Senior Lecturer, Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia, <https://orcid.org/0009-0005-0231-1544>

Valentin L. Olenov, Head of department of Aerospace computer and software systems, Candidate of Technical Sciences, Associate Professor, Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia, <https://orcid.org/0000-0002-1817-2754>

Andrey M. Tyurlikov, Head of Department of Infocommunication technologies and communication systems, Doctor of Technical Sciences, Professor, Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, Russia, <https://orcid.org/0000-0001-7132-094X>