

УДК 004.8

DOI: [10.26102/2310-6018/2026.57.6.007](https://doi.org/10.26102/2310-6018/2026.57.6.007)

Применение мультиагентных систем в задачах распределенной оптимизации

Т.В. Азарнова✉, Е.О. Калишкин

Воронежский государственный университет, Воронеж, Российская Федерация

Резюме. В данной статье проводится систематизация и детальный анализ современных методов распределенной оптимизации в мультиагентных системах (МАС). Мультиагентные подходы, определяемые как совокупность взаимодействующих автономных вычислительных сущностей, становятся критически востребованными в условиях, когда централизованная обработка данных невозможна из-за масштабов задач, жестких требований к скорости реакции или необходимости обеспечения приватности локальной информации. Цель работы заключается в комплексном исследовании ключевых подходов к децентрализованной оптимизации и выявлении фундаментальных факторов, определяющих их вычислительную эффективность, устойчивость к сбоям и практическую применимость. В рамках исследования подробно рассмотрены пять основных классов алгоритмов: консенсусный градиентный спуск (DGD), методы отслеживания градиента (Gradient Tracking), распределенный метод чередующихся направлений множителей (ADMM), а также современные стохастические и коммуникационно-эффективные подходы, включая Local SGD и FedAvg. В статье детально проанализированы системные ограничения, накладываемые топологией графа связности, алгоритмами компрессии данных и строгими требованиями дифференциальной приватности. Раскрыты ключевые теоретические и практические аспекты построения оптимальных процедур на основе сбалансированного сочетания архитектуры сети, характера локальных функций стоимости и доступной пропускной способности каналов связи. Сформулированы рекомендации по выбору конкретных алгоритмических решений в зависимости от специфики прикладной среды.

Ключевые слова: мультиагентные системы, распределенная оптимизация, консенсусный градиентный спуск, отслеживание градиента, коммуникационная эффективность, топология графа, дифференциальная приватность.

Для цитирования: Азарнова Т.В., Калишкин Е.О. Применение мультиагентных систем в задачах распределенной оптимизации. *Моделирование, оптимизация и информационные технологии*. 2026;14(6). URL: <https://moitvvt.ru/ru/journal/article?id=2283> DOI: 10.26102/2310-6018/2026.57.6.007

Application of multi-agent systems in distributed optimization problems

T.V. Azarnova✉, E.O. Kalishkin

Voronezh State University, Voronezh, the Russian Federation

Abstract. This paper systematizes and provides a detailed analysis of modern distributed optimization methods in multi-agent systems (MAS). Multi-agent approaches, defined as a set of interacting autonomous computing entities, are becoming critically in demand when centralized data processing is impossible due to the scale of problems, strict response time requirements, or the need to ensure the privacy of local information. The goal of the work is to comprehensively investigate key approaches to decentralized optimization and identify the fundamental factors that determine their computational efficiency, fault tolerance, and practical applicability. Within the framework of the study, five main classes of algorithms are considered in detail: consensus gradient descent (DGD), gradient tracking methods, the distributed alternating direction method of multipliers (ADMM), as well as modern

stochastic and communication-efficient approaches, including Local SGD and FedAvg. The paper thoroughly analyzes systemic limitations imposed by the connectivity graph topology, data compression algorithms, and strict differential privacy requirements. Key theoretical and practical aspects of constructing optimal procedures based on a balanced combination of network architecture, the nature of local cost functions, and available communication channel bandwidth are revealed. Recommendations for choosing specific algorithmic solutions depending on the specifics of the application environment are formulated.

Keywords: multi-agent systems, distributed optimization, consensus gradient descent, gradient tracking, communication efficiency, graph topology, differential privacy.

For citation: Azarnova T.V., Kalishkin E.O. Application of multi-agent systems in distributed optimization problems. *Modeling, Optimization and Information Technology*. 2026;14(6). (In Russ.). URL: <https://moitvvt.ru/ru/journal/article?id=2283> DOI: 10.26102/2310-6018/2026.57.6.007

Введение

Развитие распределенных вычислительных систем, сетей датчиков, робототехнических комплексов и автономного транспорта привело к резкому увеличению интереса к мультиагентным системам. Эти системы представляют собой совокупность взаимодействующих вычислительных сущностей, которые принимают решения на основе локальной информации и ограниченного обмена данными. Такой подход становится востребованным в ситуациях, когда централизованная обработка данных невозможна из-за масштабов задачи, требований к скорости реакции или ограничений на передачу данных.

Мультиагентная система (МАС, англ. MAS – Multi-Agent System) – это распределенная вычислительная система, состоящая из нескольких взаимодействующих интеллектуальных агентов. Каждый агент в такой системе функционирует относительно автономно, обладая своими целями, знаниями и способностью к принятию решений. Взаимодействие и координация между агентами необходимы для коллективного достижения сложной общей цели, которую невозможно или неэффективно решить с помощью одного монолитного процесса или агента. МАС используются для интеллектуальной автоматизации и динамического решения проблем в реальном времени.

Актуальность темы определяется несколькими тенденциями. Во-первых, современные приложения работают в условиях высокой динамичности и неопределенности, где координация множества автономных устройств является критически важной. Во-вторых, рост числа источников данных и усложнение моделей приводит к невозможности передачи больших объемов информации на центральные сервера, что делает распределенные методы оптимизации естественным решением. В-третьих, во многих сферах требуется приватность данных, что ограничивает традиционные способы централизованного обучения и оптимизации.

В научной литературе представлено большое число работ, посвященных методам распределенной оптимизации, включая консенсусные алгоритмы, распределенные варианты градиентного спуска, алгоритмы на основе чередующихся направлений множителей (ADMM) и различные организационные структуры взаимодействия агентов. Развитие мультиагентных методов прослеживается также в задачах федеративного обучения и многоагентного обучения с подкреплением, в которых оптимизация параметров модели выполняется на распределенных данных и без централизованного хранилища. Несмотря на накопленный опыт, проблема сочетания ограниченных коммуникаций, частичной информации и устойчивости вычислений остается открытой.

Основное направление исследования связано с комплексным анализом зависимости эффективности процедур распределенной оптимизации от выбранного алгоритма, структуры сети агентов, характера локальных функций стоимости, пропускной способности каналов связи и устойчивости системы к сбоям. Эти факторы совместно влияют на скорость сходимости, точность решения и возможность достижения согласия между агентами. В работе: описаны особенности мультиагентных систем, влияющие на построение алгоритмов распределенной оптимизации; представлена классификация моделей взаимодействия агентов; рассмотрены и систематизированы основные методы распределенной оптимизации; проанализированы ограничения, связанные с коммуникациями, топологией сети, приватностью и свойствами оптимизационных задач; обозначены ключевые тенденции развития методов распределенной оптимизации в мультиагентных системах.

Материалы и методы

Рассматривается сеть из N агентов, пронумерованных $i = 1, \dots, N$. Каждый агент обладает собственной локальной функцией стоимости

$$f_i: R^d \rightarrow R.$$

Глобальная задача оптимизации формулируется как

$$\min_{x \in R^d} F(x) \triangleq \sum_{i=1}^N f_i(x). \quad (1)$$

При этом:

- агенту i известна только его локальная функция f_i ;
- обмен информацией ограничен графом коммуникаций $G = (V, E)$, где $V = 1, \dots, N$, а $(j, i) \in E$ означает, что агент i может получать сообщения от агента j .

В распределенной формулировке у каждого агента есть своя локальная копия переменной $x_i \in R^d$, и задача записывается как

$$\min_{x_1, \dots, x_N} \sum_{i=1}^N f_i(x_i) \text{ при условиях согласования } x_i = x_j, \forall (i, j) \in E. \quad (2)$$

Согласование локальных переменных x_i интерпретируется как консенсус в сети агентов. Методы распределенной оптимизации отличаются тем, как именно достигается это согласование и как обрабатывается информация о градиентах или субградиентах локальных функций [1].

Для описания взаимодействия, часто вводится матрица весов $W = [w_{ij}] \in R^{N \times N}$, связанная с графом G . Если агенты считают средние значения по своим соседям, то типичный шаг консенсуса имеет вид:

$$x_i^{k+1} = \sum_{j=1}^N w_{ij} x_j^k, \quad (3)$$

где $w_{ij} = 0$, если нет ребра (j, i) в графе. При таком подходе предполагается, что матрица W обладает свойствами, гарантирующими сходимость к среднему (например, двойная стохастичность для ненаправленных графов) [1].

Базовый класс методов распределенной оптимизации – консенсусные градиентные методы, в которых шаг градиентного спуска совмещается с шагом согласования состояний с соседями. Общая форма распределенного градиентного спуска (Distributed Gradient Descent, DGD) для агента i имеет вид [1]:

$$x_i^{k+1} = \sum_{j=1}^N w_{ij} x_j^k - \alpha_k \nabla f_i(x_i^k), \quad (4)$$

где $\alpha_k > 0$ – шаг в направлении антиградиента. Сущность данного метода заключается в том:

- агент усредняет оценки соседей первая часть $\sum_{j=1}^N w_{ij}x_j^k$ – шаг консенсуса;
- агент осуществляет локальный шаг градиентного спуска по собственной функции стоимости $-\alpha_k \nabla f_i(x_i^k)$.

Для выпуклых задач при подходящем выборе матрицы W и шага α_k гарантируется сходимость к окрестности оптимума или к самому оптимуму, в зависимости от режима выбора шага и предположений о гладкости и сильной выпуклости.

Наиболее часто используются следующие базовые варианты метода DGD:

1. Спуск с убывающим шагом $\alpha_k \rightarrow 0$ обеспечивает сходимость к оптимуму, но снижает скорость сходимости.
2. Спуск с постоянным шагом $\alpha_k = \alpha$ обеспечивает более быструю сходимость, но только в окрестность оптимума.
3. Если задача имеет ограничения $x \in X$, то применяют проекционный DGD, в котором добавляется проекция:

$$x_i^{k+1} = P_X(\sum_{j=1}^N w_{ij}x_j^k - \alpha_k \nabla f_i(x_i^k)). \quad (5)$$

На основе базовых методов DGD разработан ряд модификаций: введения отслеживания градиента, адаптивных шагов, стохастичности и т. д.

Ограничения классического DGD состоит в том, что локальные градиенты $\nabla f_i(x_i^k)$ не дают агентам точной информации о глобальном градиенте $\nabla F(x)$. Это приводит к смещению и ухудшению скорости сходимости, особенно в случае постоянного шага. Для устранения смещения разработан класс методов с отслеживанием градиента (gradient tracking) [2].

Основная идея методов с отслеживанием градиента состоит в том, что каждый агент хранит дополнительную переменную y_i^k , аппроксимирующую глобальный градиент. Типичный алгоритм GT имеет вид:

1. Шаг консенсуса по переменным оптимизациям

$$x_i^{k+1} = \sum_{j=1}^N w_{ij}x_j^k - \alpha y_i^k. \quad (6)$$

2. Шаг отслеживания градиента

$$y_i^{k+1} = \sum_{j=1}^N w_{ij}y_j^k + \nabla f_i(x_i^{k+1}) - \nabla f_i(x_i^k). \quad (7)$$

Благодаря введению второго шага, сумма $\sum_i y_i^k$ приближается к сумме $\sum_i \nabla f_i(x_i^k) = \nabla F(x^k)$, что эффективно устраняет смещение и позволяет достигать сходимость к оптимуму при постоянном шаге для выпуклых задач.

Аналогичный подход используется в методах exact diffusion, в которых шаги диффузии и коррекции комбинируются таким образом, чтобы обеспечить сходимость даже на направленных или несбалансированных графах [3].

Вторым крупным классом методов является класс методов чередующихся направлений множителей (ADMM). Для задачи:

$$\min_x \sum_{i=1}^N f_i(x) \quad (8)$$

вводятся локальные копии x_i и вспомогательная переменная z , отражающая общее согласованное решение:

$$\min_{x_i} \sum_{i=1}^N f_i(x_i) \text{ при условиях } x_i = z, i = 1, \dots, N. \quad (9)$$

Классический ADMM для данной задачи содержит следующие шаги [4]:

1. Обновление локальных переменных:

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}} \left(f_i(x_i) + \frac{\rho}{2} \|x_i - z^k + u_i^k\|^2 \right). \quad (10)$$

где $\rho > 0$ – параметр штрафа, u_i^k – масштабированный двойственный множитель.

2. Обновление глобальной переменной:

$$z^{k+1} = \frac{1}{N} \sum_{i=1}^N (x_i^{k+1} + u_i^k). \quad (11)$$

3. Обновление двойственных переменных:

$$u_i^{k+1} = u_i^k + x_i^{k+1} - z^{k+1}. \quad (12)$$

В мультиагентной сети, централизованное вычисление z может быть заменено на распределенный консенсусный шаг (или использование вспомогательных переменных на ребрах графа). Возникают децентрализованные варианты ADMM, где согласование выполняется по структуре графа.

Преимущества ADMM-подхода:

- удобная работа с разделимыми структурами задач;
- естественная поддержка ограничений и регуляризаторов, которые легко обрабатываются через проксимальные операторы;
- хорошая практическая сходимость для большого класса выпуклых задач, включая задачи статистического обучения и распределенной регрессии.

Во многих приложениях (федеративное обучение, многороботные системы, распределенные сенсорные сети) вычислительная мощность агентов невелика, а коммуникационные ресурсы ограничены. Здесь важны методы, уменьшающие число обменов сообщениями и объем передаваемых данных [5].

Третий обширный класс методов распределенной оптимизации образуют стохастические методы SGD. В стохастических методах градиент заменяется оценкой по батчу (batch) данных. В локальном SGD каждый агент выполняет несколько локальных шагов перед очередной синхронизацией с соседями или центральным узлом:

1. Осуществляется совокупность шагов локальной оптимизации на каждом устройстве i :

$$x_i^{k,\tau+1} = x_i^{k,\tau} - \eta \nabla f_i(x_i^{k,\tau}; \xi_i^{k,\tau}), \tau = 0, \dots, E - 1, \quad (13)$$

где $\xi_i^{k,\tau}$ – стохастический образец, η – локальный шаг, E – число локальных шагов.

2. Для достижения консенсуса проводится синхронизация:

$$x_i^{k+1} = \sum_{j=1}^N w_{ij} x_j^{k,E}. \quad (14)$$

Если число локальных шагов E уменьшается, то, с одной стороны, уменьшается частота коммуникаций, но, с другой стороны, усиливается расхождение локальных копий. Поэтому важно достигнуть определенного компромисса между частотой коммуникаций и расхождением локальных копий.

Правильный выбор E и шага η – ключевой элемент компромисса между скоростью сходимости и стоимостью коммуникаций [6].

В задачах федеративного обучения достаточно распространен метод FedAvg. В рамках данного метода множество агентов обучает общую модель, периодически передавая обновления параметров. Каждый агент k в процессе работы алгоритма выполняет E локальных шагов SGD, которые формируют глобальные параметры $w_t^{(k,E)}$, на основании данных параметров сервер или координатор вычисляет взвешенное усреднение [7]:

$$w_{t+1} = \sum_{k=1}^N \frac{n_k}{n_{\text{tot}}} w_t^{(k,E)}, \quad (15)$$

где n_k – число локальных данных у клиента k , $n_{\text{tot}} = \sum_k n_k$. Метод федеративного обучения FedAvg можно рассматривать как частный случай распределенной оптимизации с редкой синхронизацией и параметрическими сообщениями.

$$x_i^{k+1} = \sum_{j=1}^N w_{ij} (x_j^k + Q(\Delta_j^k)) - \alpha_k \nabla f_i(x_i^k), \quad (16)$$

где Δ_j^k – это приращение, которое передает агент j (например, изменение модели), а Q – стохастический оператор с конечной дисперсией [8]. Чтобы компенсировать искажения, вносимые сжатием, и не потерять сходимость, в таких подходах приходится использовать дополнительные механизмы – например, накопление ошибок или обратную связь по ошибке (error feedback) [9].

Во многих практических сценариях – таких как работа с персональными устройствами, медицинскими данными или промышленными сетями – агенты не имеют права раскрывать исходные данные или точные значения градиентов. Это стимулирует развитие методов распределенной оптимизации, в которых используются механизмы дифференциальной приватности и другие способы защиты информации [10]. Один из распространенных приемов – добавление случайного шума к локальным градиентам или к обновлениям модели. Пусть агент i вычислил стохастический градиент $\nabla f_i(x_i^k; \xi_i^k)$. Тогда перед отправкой он формирует «обезличенное» сообщение:

$$g_i^k = \nabla f_i(x_i^k; \xi_i^k) + \zeta_i^k, \quad (17)$$

где ζ_i^k – случайный шум (например, гауссовский $N(0, \sigma^2 I)$). Далее агрегирование или консенсус выполняются уже над зашумленными величинами:

$$x_i^{k+1} = \sum_{j=1}^N w_{ij} x_j^k - \alpha_k g_i^k. \quad (18)$$

При правильном выборе масштаба шума σ достигается заданный уровень дифференциальной приватности, но при этом растет дисперсия градиентной оценки и ухудшается точность решения. Задача состоит в том, чтобы сбалансировать требования приватности и качество оптимизации.

Во многих мультиагентных системах граф коммуникаций может быть направленным, несбалансированным и изменяющимся во времени (например, при движении роботов или динамической маршрутизации пакетов). В таких условиях стандартные консенсусные методы с симметричными, двустохастическими матрицами W неприменимы.

Для направленных графов разработаны:

- push-sum/push-pull методы, где дополнительно отслеживается нормирующая масса, позволяющая корректно усреднять значения при односторонних связях;
- модификации gradient tracking и exact diffusion, адаптированные к левостохастическим и правостохастическим матрицам;
- ускоренные методы первого порядка, учитывающие структуру направленного графа и обеспечивающие точную сходимость [11, 12].

Типичный шаг push-sum подобного метода имеет следующий вид:

$$s_i^{k+1} = \sum_{j \in N_i^i} a_{ij} s_j^k - \alpha_k \nabla f_i(x_i^k), r_i^{k+1} = \sum_{j \in N_i^i} a_{ij} r_j^k, \quad (19)$$

$$x_i^{k+1} = \frac{s_i^{k+1}}{r_i^{k+1}}. \quad (20)$$

Таким образом, корректируется смещение, вызванное асимметрией графа.

В задачах интеллектуального анализа данных, особенно при обучении нейронных сетей и многоагентном обучении с подкреплением, функции стоимости f_i обычно невыпуклые [13]. В этом случае классические гарантии сходимости к глобальному минимуму не выполняются. Вместо этого анализ строится на сходимости к стационарным точкам (точкам, где $\|\nabla F(x)\| \rightarrow 0$) или на эмпирических характеристиках [14].

Для невыпуклых задач применяются:

- стохастические варианты DGD и gradient tracking;
- локальный SGD и FedAvg;
- распределенные версии методов оптимизации в MARL, где локальные функции стоимости связаны с оценкой ценности стратегий (value functions, policy gradients).

Классический стохастический GT-метод для невыпуклых задач имеет вид:

$$x_i^{k+1} = \sum_{j=1}^N w_{ij} x_j^k - \alpha y_i^k, \quad (21)$$

$$y_i^{k+1} = \sum_{j=1}^N w_{ij} y_j^k + \nabla f_i(x_i^{k+1}; \xi_i^{k+1}) - \nabla f_i(x_i^k; \xi_i^k), \quad (22)$$

где ξ_i^{k+1} – стохастические образцы, а анализ сходимости строится в терминах ожидаемого значения $E\|\nabla F(x^k)\|$ [14].

Результаты

Анализ алгоритмов распределенной оптимизации показывает четкую зависимость сходимости от типа используемого метода и условий среды. Для классического консенсусного спуска (DGD) сходимость к оптимуму x^* гарантируется только при убывающем шаге, тогда как постоянный шаг α ведет лишь в окрестность решения, размер которой зависит от спектрального разрыва графа. Более совершенные алгоритмы – такие как Gradient Tracking и Exact Diffusion – позволяют избавиться от этого смещения и добиться точной сходимости $x_i^k \rightarrow x^*$ с линейной скоростью даже при постоянном шаге, если задача является гладкой и сильно выпуклой. Метод ADMM также показывает линейную сходимость и высокую точность, но его сложность связана с необходимостью подбирать штрафной параметр ρ .

В ситуациях с ограниченными коммуникационными возможностями применяют методы Local SGD и FedAvg. Они уменьшают частоту обменов, однако это ведет к расхождению локальных моделей – величина расхождения растет пропорционально числу локальных шагов E . Для таких методов характерна сублинейная скорость сходимости ($O(1/k)$ или $O(1/t)$). Если дополнительно используются сжатие данных или механизмы приватности, то искажения возрастают. Сходимость при этом ограничивается некоторой окрестностью оптимума, радиус которой определяется степенью сжатия или уровнем шума. Для невыпуклых задач все рассмотренные подходы гарантируют сходимость только к стационарным точкам, где $E\|\nabla F(x^k)\|^2 \rightarrow 0$.

На основе проведенного анализа можно дать ряд рекомендаций по выбору алгоритмов для прикладных задач.

В задачах экологического мониторинга, где вычислительными узлами выступают маломощные датчики (температуры, влажности, загрязнений), наиболее подходят методы типа DGD и Diffusion. Например, для построения тепловой карты региона путем усреднения показаний датчиков не требуется сложных вычислений, и простота реализации градиентного шага здесь важнее, чем невысокая скорость сходимости.

Задача экономического диспетчирования в энергосистемах (Economic Dispatch Problem) – распределение генерации между источниками при соблюдении баланса

мощностей – эффективно решается с помощью ADMM. Эти методы хорошо работают с жесткими ограничениями-равенствами, позволяют учитывать физические законы сетей и при этом не раскрывают информацию о локальном потреблении отдельных домохозяйств.

В робототехнике, при управлении движением группы роботов, где требуется высокая точность согласования траекторий и скоростей, критически важна точность достижения консенсуса. Здесь наилучшие результаты дают методы Gradient Tracking. Они устраняют стационарную ошибку, свойственную обычному DGD, и позволяют группе двигаться как единое целое даже при постоянно меняющихся скоростях.

В задачах федеративного обучения на мобильных устройствах или в медицине (например, обучение модели для анализа МРТ-снимков или предиктивной модели клавиатуры на смартфоне) стандартом стал алгоритм FedAvg. Несмотря на сублинейную сходимость, возможность выполнять много локальных эпох обучения сильно снижает нагрузку на сеть – это подтверждается как экспериментами Google, так и опытом медицинских консорциумов.

В задачах федеративного обучения в медицине и на мобильных устройствах, таких как обучение предиктивных моделей клавиатур на смартфонах или анализа МРТ-снимков, стандартом стал алгоритм FedAvg. Несмотря на сублинейную сходимость, возможность проводить множество локальных эпох обучения радикально снижает нагрузку на сеть, что подтверждается экспериментами Google и медицинских консорциумов.

В Таблице 1 собранная по результатам исследования информация о характеристиках различных методов распределенной оптимизации.

Таблица 1 – Сводная таблица характеристик методов распределенной оптимизации
Table 1 – Summary of characteristics of distributed optimization methods

Метод	Тип задачи	Сходимость	Скорость	Требования к графу	Коммуникации
DGD	выпуклая	в окрестность (const шаг), к точке оптимума (убыв. шаг)	сублин.	ненаправленный, связный	высокая частота
Gradient Tracking	выпуклая, сильно выпуклая	к точке оптимума	линейная	ненаправленный/направленный	средняя
ADMM	выпуклая	к точке оптимума	линейная	общий или консенсусный	высокая нагрузка вычислений
Local SGD	невыпуклая	к точке оптимума	сублин.	любая	низкая частота
FedAvg	невыпуклая	к точке оптимума	сублин.	звезда/дерево	редкие коммуникации
Компрессированные методы	выпуклая, невыпуклая	в окрестность	зависит от компрессии	любая	низкая нагрузка

Обсуждение

Анализ полученных результатов подтверждает, что эффективность распределенной оптимизации в мультиагентных системах определяется сложным взаимодействием трех факторов: свойствами локальных функций стоимости, топологией сети и коммуникационными ограничениями. Проведенный анализ указывает на фундаментальные недостатки базового метода DGD, который из-за неустранимого смещения уступает алгоритмам с отслеживанием градиента (Gradient Tracking) и ADMM. Последние демонстрируют лучшую приспособленность к распределенным

вычислениям, обеспечивая сходимость и устойчивость даже в невыпуклых задачах, однако требуют больших вычислительных ресурсов и более сложной структуры обмена данными.

В условиях жестких ограничений на пропускную способность каналов стохастические методы (Local SGD, FedAvg) и алгоритмы с компрессией позволяют существенно снизить трафик, но ценой снижения точности и возникновения дрейфа локальных моделей, что согласуется с аналогичными эффектами в федеративном обучении. Дополнительные внешние требования, такие как обеспечение дифференциальной приватности, или работа на слабосвязных графах с малым спектральным разрывом неизбежно увеличивают остаточную ошибку и замедляют консенсус.

На основе проведенного анализа можно сформулировать ключевые векторы дальнейших исследований в области распределенной оптимизации:

1. Асинхронность и устойчивость к задержкам. Большинство рассмотренных методов, такие как DGD, Gradient Tracking, предполагают синхронные обновления, где вся сеть ждет самого медленного агента. Разработка асинхронных версий алгоритмов, которые математически гарантируют сходимость даже при использовании устаревшей информации от соседей, является критически важной для гетерогенных сетей [2, 14].

2. Оптимизация на изменяющихся во времени направленных графах. Существующие методы для направленных графов, такие как Push-Pull, Directed-GT, часто требуют, чтобы граф был сильно связным в каждый момент времени. Ослабление этого условия до «совместной связности» во времени для ускоренных алгоритмов первого порядка остается актуальной проблемой в области многоагентной оптимизации [13].

3. Совместное сжатие и приватность. Методы компрессии [10] и методы дифференциальной приватности [12] развивались параллельно. Однако их объединение часто приводит к кумулятивному росту дисперсии и падению точности. Создание адаптивных схем, где уровень сжатия динамически подстраивается под требуемый бюджет приватности, представляет собой также современную актуальную задачу.

4. Робастность к византийским атакам. В открытых мультиагентных системах часть агентов может быть скомпрометирована и посылать ложные градиенты. Интеграция механизмов статистической оптимизации непосредственно в итерации методов типа ADMM или Gradient Tracking без потери скорости линейной сходимости требует разработки новых теоретических подходов.

Таким образом, универсального решения не существует: выбор между точностью (ADMM, Gradient Tracking) и эффективностью (Local SGD, сжатие) должен базироваться на конкретных требованиях к архитектуре сети и безопасности, а не только на внутренних характеристиках алгоритма.

Заключение

В рамках исследования проведен обзор методов распределенной оптимизации в мультиагентных системах, направленный на выделение ключевых закономерностей, определяющих эффективность данных методов и область применимости. Эффективность анализируемых методов в конечном счете определяется свойствами локальных функций, структурой графа коммуникаций и ограничениями на обмен данными. Все эти настройки вместе задают пределы производительности распределенных алгоритмов. Зависимость параметров сходимости от характеристик графа и от доступности связи подчеркивает, что важно проектировать согласованно и сами алгоритмы, и архитектуру системы.

Методы, основанные на консенсусном градиентном спуске, отличаются стабильностью и простотой. Их главный недостаток – наличие смещения при использовании постоянного шага. Подходы с отслеживанием градиента дают более высокую точность и лучше адаптируются к ограничениям каналов связи. ADMM-методы надёжны и предсказуемы при решении задач с ограничениями, но требуют значительных вычислительных ресурсов. Стохастические и коммуникационно-эффективные алгоритмы хорошо работают в гетерогенных или слабосвязанных системах, однако точность решения при этом снижается. Методы с приватностью и компрессией позволяют соблюдать внешние ограничения, но усиливают расхождение между локальными и глобальными оценками.

Таким образом, выбор алгоритма распределенной оптимизации должен подстраиваться под конкретную задачу и среду: топологию сети, качество каналов связи, уровень шума, требования к приватности. Универсального решения нет, эффективность достигается за счет комбинирования методов и их адаптации к имеющейся инфраструктуре.

Перспективные направления дальнейших исследований – это создание алгоритмов, устойчивых к динамическим и ориентированным структурам графа; снижение чувствительности к невыпуклости локальных функций; повышение точности при минимальных коммуникационных затратах; интеграция механизмов приватности без существенного ухудшения сходимости. Также интерес представляет изучение способов согласования оптимизационных методов с интеллектуальными модулями агентов, которые могли бы прогнозировать состояние сети и адаптировать стратегию обмена данными.

Эти направления закладывают основу для разработки устойчивых, масштабируемых и адаптивных мультиагентных систем, способных функционировать в условиях ограниченных ресурсов и высокой неопределенности.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Yang T., Yi X., Wu J., et al. A survey of distributed optimization. *Annual Reviews in Control*. 2019;47:278–305. <https://doi.org/10.1016/j.arcontrol.2019.05.006>
2. Li J., Su H. *Gradient Tracking: A Unified Approach to Smooth Distributed Optimization*. arXiv. URL: <https://arxiv.org/abs/2202.09804> [Accessed 25th December 2025].
3. Yuan K., Ying B., Zhao X., et al. *Exact Diffusion for Distributed Optimization and Learning – Part I: Algorithm Development*. arXiv. URL: <https://arxiv.org/abs/1702.05122> [Accessed 25th December 2025].
4. Boyd S., Parikh N., Chu E., et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends[®] in Machine Learning*. 2011;3(1):1–122. <https://doi.org/10.1561/22000000016>
5. Halsted T., Shorinwa O., Yu J., et al. *A survey of distributed optimization methods for multi-robot systems*. arXiv. URL: <https://arxiv.org/abs/2103.12840> [Accessed 25th December 2025].
6. McMahan B., Moore E., Ramage D., et al. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017), 20–22 April 2017, Fort Lauderdale, FL, USA*. PMLR; 2017. P. 1273–1282.
7. Wang Y., Lin H., Lam J., et al. Differentially private consensus and distributed optimization in multi-agent systems: A review. *Neurocomputing*. 2024;597:127986. <https://doi.org/10.1016/j.neucom.2024.127986>

8. Alistarh D., Grubic D., Li J., et al. QSGD: Communication-efficient SGD via gradient quantization and encoding. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 04–09 December 2017, Long Beach, CA, USA*. 2017. P. 1709–1720.
9. Koloskova A., Lin T., Stich S.U., et al. *Decentralized deep learning with arbitrary communication compression*. arXiv. URL: <https://arxiv.org/abs/1907.09356> [Accessed 25th December 2025].
10. Yurdem B., Kuzlu M., Gullu M.K., et al. Federated learning: Overview, strategies, applications, tools and future directions. *Heliyon*. 2024;10(19):e38137. <https://doi.org/10.1016/j.heliyon.2024.e38137>
11. Wang Zh., Wang Ch., Wang J., et al. An accelerated exact distributed first-order algorithm for optimization over directed networks. *Journal of the Franklin Institute*. 2023;360(14):10706–10727. <https://doi.org/10.1016/j.jfranklin.2023.08.015>
12. Xin R., Pu Sh., Nedić A. A general framework for decentralized optimization with first-order methods. *Proceedings of the IEEE*. 2020;108(11):1869–1889. <https://doi.org/10.1109/JPROC.2020.3024266>
13. Lian X., Zhang C., Zhang H., et al. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 04–09 December 2017, Long Beach, CA, USA*. 2017. P. 5330–5340.
14. Nedic A., Ozdaglar A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*. 2009;54(1):48–61. <https://doi.org/10.1109/TAC.2008.2009515>

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Азарнова Татьяна Васильевна, доктор технических наук, профессор, профессор Воронежского государственного университета, Воронеж, Российская Федерация.
e-mail: ivdas92@mail.ru

Tatyana V. Azarnova, Doctor of Engineering Sciences, Professor, Professor of Voronezh State University, Voronezh, the Russian Federation.

Калишкин Евгений Олегович, аспирант, Воронежский государственный университет, Воронеж, Российская Федерация
e-mail: e2697517@gmail.com

Evgeny O. Kalishkin, Postgraduate, Voronezh State University, Voronezh, the Russian Federation.

Статья поступила в редакцию 11.03.2026; одобрена после рецензирования 03.06.2026; принята к публикации 14.06.2026.

The article was submitted 11.03.2026; approved after reviewing 03.06.2026; accepted for publication 14.03.2026.