САНКТ - ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет
Кафедра системного программирования

Модификация протокола локального голосования для оптимизации балансировки загрузки ресурсов распределенной системы

Отчет по производственной практике

Борисоглебская Елена Александровна
1 курс, группа 21.М04-мм

Научный руководитель:
д.ф.-м. н., профессор Граничин О. Н.

Санкт-Петербург
2022

Содержание

1. Введение

В последнее время все чаще при вычислениях используются распределенные системы параллельных вычислений, для которых актуальна задача разделения пакета заданий между несколькими вычислительными устройствами. Подобные задачи возникают не только в вычислительных сетях [1], но и в производственных [?], транспортных, логистических [2] и сетях обслуживания [?]. Обычно при распределении заданий по узлам их стараются распределять так, чтобы загрузка вычислительных узлов была равномерной.

На практике используются как централизованные стратегии динамической балансировки загрузки сети, так и полностью распределенные(децентрализованные). При централизованной стратегии создается отдельный ресурс, который собирает информацию о состоянии всей сети и распределяет задания между узлами [5], [6]. При полностью распределенной стратегии алгоритмы балансировки загрузки выполняются на каждом из узлов. Узлы в таком случае обмениваются информацией о состоянии, а при необходимости и задачами. Перемещение задач при этом происходит только между соседними (связанными) узлами [7–11]. В статье [12] рассмотрена децентрализованная стратегия для решения задачи балансировки — протокол локального голосования. Протокол локального голосования — это алгоритм управления, согласно которому задачи, поступающие на узлы сети, могут быть равномерно перераспределены между всеми узлами сети за счет обмена данными между соседями. Применение такого протокола позволяет уменьшить нагрузку на центр обработки данных, так как все вычисления будут производится локально на агенте.

Сложность балансировки загрузки ресурсов часто заключается в отсутствии априорной информации о задачах. Новые задачи появляются все время и при этом заранее неизвестно ни сколько их всего, ни когда они появятся, ни сколько времени потребуется на их выполнение.

Помимо отсутствия информации о задачах, количество узлов распределенной системы может быть очень велико, что также усложняет централизованное вычисление оптимального распределения задач между узлами. Здесь и появляются мультиагентные системы. Использование нескольких автономных агентов позволяет распараллелить, а значит и ускорить процесс принятия решения.

На сегодняшний день мультиагентные технологии нашли широкое применение в различных сферах: в производстве, в энергетике, в информационных технологиях и т.д. [13–15]. Одна из фундаментальных концепций мультиагентных технологий — это нахождение консенсуса. Под консенсусом в данном случае понимается соглашение между агентами относительно общего значения, чаще всего минимума некоторой функции потерь:

$$\bar{F}(x) = \sum_{i=0}^{n} F^i(x),$$
$$x \in \mathbb{R}, F^i(x) : \mathbb{R}^d \to \mathbb{R}.$$

Исследование алгоритмов консенсуса и распределенной оптимизации началось в 1970–1980-е гг. [16, 17]. На сегодняшний день существует ряд подходов для случая, когда функции $F^i(x)$ выпуклы: метод множителей с переменным направлением [19, 20], субградиентный метод [21, 22]. Для невыпуклых задач в работах [23, 24] разработан большой класс распределенных алгоритмов на основе различных «функционально-суррогатных единиц».

Существует ряд работ, в которых учтены возможные помехи. Помехи могут появляться как в связи с неточностями измерений, так и при передаче данных. Чаще всего предполагается, что помехи обладают некоторыми свойствами нормального распределения [25–27]. Кроме того, есть ряд статей, в которых единственные накладываемые на помехи условяе это их центрированность и ограниченность [28].

В статье [12] была рассмотрена задача балансировки загрузки сети при помощи протокола локального голосования. Алгоритм, используемый в статье, может быть ускорен при помощи метода Нестерова.

2. Постановка задачи

2.1. Сведения из теории графов

Граф — это пара $G = (N, E)$, где N — множество узлов или вершин и E — множество ребер. Предполагаем, что граф простой, т.е. $(i, i) \notin E, \forall i$ отсутствуют петли и между узлами может быть максимум одна дуга. Множеством соседей узла i называется множество $N^i = j : (j, i) \in E$, т.е. множество узлов с ребрами входящими в i.

Сопоставим каждому ребру $(j, i) \in E$ вес a^{ij}. Предположим, что все веса строго положительно. Граф может быть представлен матрицей смежности (или связности) $A = [a^{ij}]$ с весами $a^{ij} > 0$, если $(j, i) \in E$ и $a^{ij} = 0$ в противном случае. Отметим, что $a^{ii} = 0, \forall i$.

Определим взвешенную полустепень входа вершины i как количество входящих в i ребер или $d^i(A) = \sum_{j \in N} a^{ij}$, а полустепенью выхода вершины i — количество выходящих ребер или $d^i_0(A) = \sum_{j \in N} a^{ij}$. Если для всех вершин $i \in N$ взвешенная полустепень входа равна взвешенной полустепени выхода, то граф называется сбалансированным по весам.

Граф называется неориентированным, если $a^{ij} = a^{ji}, \forall i, j \in N$, т.е. если он является двунаправленным и веса ребер (i, j) и (j, i) совпадают. При этом матрица смежности симметрична. Неориентированный граф, в котором все узлы имеют одинаковые полустепени d называется d-регулярным.

Определим диагональную матрицу $D(A) = diag d^i(A)$ из полустепеней входа и лапласиан графа $\mathcal{L}(A) = D(A) - A$. Заметим, что сумма по строкам лапласиана равна нулю. Следовательно, любой вектор составленный из одинаковых констант является правым собственным вектором, соответствующим нулевому собственному значению.

Обозначим через $d_{max}(A)$ максимальную полустепень входа графа G. Применив круговой критерий Гершгорина [**?**], можно вывести еще одно важное свойство лапласиана: все собственные числа матрицы $\mathcal{L}(A)$ имеют неотрицательную вещественную часть и лежат в круге с центром на вещественной оси в точке $d_{max}(A)$ и радиусом $d_{max}(A)$.

Обозначим $\lambda_1, \lambda_2, \ldots, \lambda_n$ собственные числа матрицы $\mathcal{L}_t(A)$. Отсортируем их в порядке возрастания вещественной части: $0 \leq Re(\lambda_1) \leq Re(\lambda_2) \leq \cdots \leq Re(\lambda_n)$. Известно, что если у графа есть остовное дерево, то $\lambda_1 = 0$.

2.2. Задача балансировки загрузки ресурсов

Рассмотрим модель децентрализованной системы распределения однотипных заданий между разными узлами (агентами) для параллельной работы с обратной связью. Обозначим $N = 1, \ldots, n$ — набор интеллектуальных агентов (узлов), каждый из которых выполняет поступающие задания по принципу очереди. Задания поступают в систему на разные узлы, возможно, в различные моменты времени. Связь между узлами определяется, топологией динамической сети: в момент времен t графом $G_t = (N, E_t)$ с матрицей смежности A. Обозначим через $W_t = \mathcal{L}_t(A)$ лапласиан графа G_t.

В каждый момент времени t состояние агента $i \in N$ описывается двумя характеристиками:

1) q_t^i — загруженность очереди или длина очереди из атомарных элементарных заданий узла i в момент времени t,

2) r_t^i — производительность узла i в момент времени t.

Изменение состояния агента описывается следующим уравнением:

$$q_{t+1}^i = q_t^i - r_t^i + z_t^i + u_t^i; \quad i = 1, \dots, n, \quad t = 0, 1, 2, \dots,$$

где z_t^i — новое задание, поступившее на узел i в момент времени t, u_t^i — результат перераспределения задач между узлами (добавление или уменьшение), получающийся в итоге применения выбранного протокола перераспределения заданий. В уравнениях динамики предполагаем, что $\sum_i u_i^t = 0, \quad t = 0, 1, 2, \dots$.

В каждый момент времени t узел i имеет следующую информацию:

1) зашумленные данные о своей длине очереди:

$$y_t^{ii} = q_t^i + w_t^{ii},$$

2) зашумленные наблюдения о длинах очередей соседей, если $N_t^i \neq \emptyset$:

$$y_t^{ij} = q_{t-p_t^{ij}}^i + w_t^{ij}, \quad j \in N_t^i,$$

где w_t^{ij} — помехи, а $0 \leq p_t^{ij} \leq \bar{p}$ — целочисленная задержка, $\bar{p}$ — максимально возможная задержка,

3) данные о своей производительности r_t^i и о производительностях узлов соседей $r_t^j, j \in N_t^i$.

Требуется поддерживать равномерную загрузку всех узлов сети. Будем рассматривать две постановки задачи: стационарную и нестационарную. В стационарном случае все задания поступают в систему на разные узлы в начальный момент времени $t = 0$. Если все задания выполняются только тем агентом, которому они поступили, то время реализации всех заданий определяется как

$$T_{max} = \max_{i \in N} q_0^i / r_0^i.$$

В нестационарном — новые задания могут поступать в систему на любой из n узлов в различные моменты времени t.

Для момента времени t определим T_t — время до окончания выполнения всех заданий на всех узлах.

$$T_t = \max_{i \in N} q_t^i / r_t^i. \tag{1}$$

Будем называть отношение q_t^i / r_t^i загруженностью узла i в момент времени t. Поставим цель управления

$$T_t \to \min_{u_t}. \tag{2}$$

В статье [33] показано, что в стационарном случае из всех возможных вариантов распределения общего количества заданий, не обработанных к моменту времени t, наименьшее время работы системы соответствует тому, при котором:

$$q_t^i/r_t^i = q_t^j/r_t^j, \quad \forall i, j \in N.$$

В соответствии с этим утверждением можно переформулировать (1) и (2) следующим образом:

$$F_t(q_t) = \sum_{i,j \in \{1,\dots,m\}} a_t^{j,i} \left(\frac{q_t^j}{r_t^j} - \frac{q_t^i}{r_t^i} \right)^2 \to \min_{q_t}. \tag{3}$$

2.3. Протокол локального голосования

Рассмотрим способ решения описанной выше задачи с применением методов мультиагентных технологий. Будем рассматривать каждый узел системы в качестве агента. Необходимо решить задачу достижения агентами консенсуса при наличии неизвестных, но ограниченных помех. Каждый агент i в момент времени t вычисляет значение некоторой функции потерь:

$$z_t^i = F_t^i(y_t^{i,i}) + \upsilon_t^i, \tag{4}$$

где q_t^i — количество задач в очереди агента i в момент времени t,

$F_t^i = \sum_{j \in N_t^i} a_t^{j,i} (y_t^{i,j}/r^j - y_t^{i,i}/r^i)^2$ — функция потерь, то насколько его количество задач отличается от среднего количества задач соседей,

υ_t^i — неизвестные ограниченные помехи, связанные с передачей данных между устройствами.

Агент i в каждый момент времени t обменивается данными с соседями и пытается оценить оптимальное значение q_t^i. Для этого решается задача нестационарной оптимизации среднего риска, иначе говоря ищется минимум $\hat{q}_t$ функционала $F_t^i(q)$:

$$\bar{F}_t(q) = \sum_{i \in \mathbb{N}} F_t^i(q) \to \min_q. \tag{5}$$

При меняющихся условиях минимум функционала (5) может также меняться со временем. Поэтому очень важно быстро и эффективно найти данный минимум.

Рассмотрим алгоритм минимизации функционала (5) при помощи алгоритма локального голосования:

$$\begin{gathered} T_t^i(\hat{q}_{t-1}^i) = \sum_{j \in \mathcal{N}_t^i} b_t^{i,j} \left(\frac{y_{t-1}^{i,j}}{r^j} - \frac{\hat{q}_{t-1}^i}{r^i} \right), \\ \hat{q}_t^i = \hat{q}_{t-1}^i - \gamma T_t^i(\hat{q}_{t-1}^i), \end{gathered} \tag{6}$$

где $\gamma > 0$ — размер шага протокола локального голосования, $b_t^{i,j} > 0 \quad \forall j \in N_t^i$. Обратим внимание, что функция T_t^i в (6) является производной функции F_t из уравнения (3).

3. Быстрый градиентный метод Нестерова

Метод Нестерова — это итеративный алгоритм, основанный на градиентном методе. На каждом шаге k оптимальное значение q_k обновляется в соответствии с антиградиентом и направлением на предыдущем шаге v_{k-1}. Алгоритм быстрого градиента Нестерова представляет собой последовательность следующих шагов:

1) Выбрать $\hat{q}_0 \in \mathbb{R}^d$ и $\gamma_0 > 0$, $v_0 = \hat{q}_0$

2) На k-ой итерации:

   а) Найти $\alpha_k \in (0, 1)$:
   $$L\alpha_k^2 = (1 - \alpha_k) * \gamma_k + \alpha_k * \mu.$$

   б) Вычислить $\gamma_{k+1} = (1 - \alpha_k) * \gamma_k + \alpha_k * \mu$.

   в) Вычислить
   $$z_k = \frac{\alpha_k \gamma_k v_k + \gamma_{k+1} \hat{q}_k}{\gamma_k + \alpha_k \mu}.$$

   г) Найти $\hat{q}_{k+1}$ такое, что
   $$\hat{q}_{k+1} = z_k - \frac{1}{L} \nabla F(z_k)$$

   д) Вычислить $v_{k+1} = \frac{1}{\gamma_{k+1}} \Big[(1 - \alpha_k)\gamma_k v_k + \alpha_k \mu z_k - \alpha_k \nabla F(z_k) \Big]$.

4. Ускоренный протокол локального голосования

4.1. Постановка задачи и модель

Рассматривается модель для конкретного узла i. Единственной доступной информацией служат измерения протокола локального голосования $Y_k(q)$, искаженные аддитивным шумом ξ_k:

$$Y_k(q) = \nabla F_k(q) + \xi_k, \quad k = 0, 1, 2, \dots. \tag{7}$$

Задача найти последовательность оценок $\{\hat{q}_k\}_{k=0}^{\infty}$ таких, что

$$\exists N, C < \infty : \forall k > N \quad \mathbb{E}\|\hat{q}_k - q_k\|^2 \le C. \tag{8}$$

Предполагаем, что функции F_n и шум ξ_n обладают следующими свойствами:

1) Функции F_n имеют общую константу Липшица $L > 0$ и общую константу строгой выпуклости $\mu > 0$:
$$\forall x \in \mathbb{R}^d : \|\nabla F_n(x)\| \le L\|x - q_n\|,$$
$$\langle \nabla F_n(x), x - q_n \rangle \ge \mu \|x - q_n\|^2.$$

2) Для $\forall n > 0 > 0$ изменение функции F_n ограничено, т.е. существуют такие константы a, b, c, что для $\forall n > 0$ выполняется:
$$\|F_n(x) - F_{n+1}(x)\| \le a\|\nabla F_n(x)\| + b,$$
$$\|\nabla F_{n+1}(x) - \nabla F_n(x)\| \le c.$$

3) Шум ξ_n имеет нулевое математическое ожидание и ковариация шума в среднем ограничены:
$$\mathbb{E}\xi_n = 0, \mathbb{E}Q \leq \sigma^2_{max} I,$$
где Q - матрица ковариации случайного вектора ξ_n, I - единичная матрица, σ^2_{max} - максимальное собственное число матрицы Q.

4.2. Алгоритм

В статье [34] представлен алгоритм ускорения по Нестерову для градиентного спуска в задаче трекинга. Для решения задачи, определенной в (8) с моделью наблюдения, определенной в (7), удовлетворяющей предположениям 1–3 рассмотрим следующий алгоритм трекинга:

1) Выбрать $\hat{q}_0 \in \mathbb{R}^d$ и $\gamma_0 > 0$, $v_0 = \hat{q}_0$. Выбрать $h > 0, \eta \in (0, \mu), \alpha_x \in (0, 1)$ такое, что неравенство (9) всегда выполнялось. Вычисляется значение $H_1 = h - \frac{h^2 L}{2}$.

2) На k-ой итерации:

   а) Найти $\alpha_k \in [\alpha_x, 1)$:
   $$H_1 - \frac{\alpha_k^2}{2\gamma_{k+1}} > 0. \tag{9}$$

   б) Вычислить $\gamma_{k+1} = (1 - \alpha_k) * \gamma_k + \alpha_k * (\mu - \eta)$.

   в) Вычислить
   $$z_k = \frac{\alpha_k \gamma_k v_k + \gamma_{k+1} \hat{q}_k}{\gamma_k + \alpha_k(\mu - \eta)}$$
   и посчитать значение $Y_k(z_k)$.

   г) Вычислить $\hat{q}_{k+1}$:
   $$\hat{q}_k = z_k - hY_k(z_k).$$

   д) Вычислить $v_{k+1} = \frac{1}{\gamma_k}\Big[(1 - \alpha_k)\gamma_k v_k + \alpha_k(\mu - \eta) z_k - \alpha_k Y_k(z_k)\Big]$.

4.3. Сходимость

Для доказательства сходимости метода введем следующие обозначения: Пусть $\{\alpha_k\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}, \{A_k\}_{k=0}^{\infty}, \{Z_k\}_{k=0}^{\infty}, \{D_k\}_{k=0}^{\infty}$ последовательности определенные следующим образом:

$$\begin{gathered}
\alpha_k \in [\alpha_x, 1), \lambda_0 = 1, \lambda_{k+1} = (1 - \alpha_k)\lambda_k, \\
A_0 = 0, \\
A_{k+1} = (1 - \alpha_k)((1 - \alpha_k)a + A_n), \\
Z_n = (1 - \lambda_k)(b + ac) + A_k c, \\
D_0 = 0, \\
D_{k+1} = (1 - \alpha_k)D_k + \frac{a(1 + \alpha_k) + hc}{4\epsilon} + (1 + \alpha_k)b + \\
+ (1 - \alpha_k)Z_k + h^2 \frac{L}{2}\sigma^2 + \frac{\alpha_k c^2}{2\eta}.
\end{gathered}$$

Тогда имеем:

$$D_\infty = \alpha_x^{-1}\left[\frac{2a+hc}{4\epsilon} + 2b + (1-\alpha_x)(b + A_\infty c) + h^2\frac{L}{2}\sigma^2 + \frac{c^2}{2\eta}\right],$$

где

$$\Gamma = \max_{n\geq 0}\gamma_k,$$

$$\epsilon \in \left(0, \frac{1}{a(1+\alpha_x)+hc}\left(H_1 - \frac{\alpha_x^2}{2\Gamma}\right)\right]$$

Теорема Если предположения 1–3 выполнены, то алгоритм, описанный выше, решает проблему (8) со следующими параметрами:

$$C = \frac{2}{\mu}D_\infty$$

Ошибка оценки после конечного числа итераций ограничена:

$$\mathbb{E}_k F_k(\hat{q}_k) - F_k(q_k) \leq \prod_{i=1}^{k}(1-\alpha_k)(\phi_0(q_0) - F_k(q_k) + \Phi) + D_k,$$

где $\phi_0(x) = F_0(\hat{q}_0) + \frac{\gamma_0}{2}\|x - v_0\|^2, \Phi = \dfrac{\gamma_0 c^2}{2\eta^2}$.

5. Результат

В результате проделанной работы был найден метод оптимизации протокола локального голосования для балансировки загрузки сети. А также разобраны теоремы, доказывающие сходимость данного алгоритма.

Список литературы

[1] N. Amelina, A. Fradkov, Y. Jiang, and D. J. Vergados, “Approximate consensus in stochastic networks with application to load balancing,” IEEE Trans. Inf. Theory, vol. 61, no. 4, pp. 1739–1752, Apr. 2015.

[2] Towards Autonomous AI Systems for Resource Management: Applications in Industry and Lessons Learned

[3] D. Armbruster, A.S. Mikhailov, K. Kaneko (eds.), "Networks of Interacting Machines: Production Organization in Complex Industrial Systems and Biological Cells,"World Scientific Singapore, p. 267, 2005.

[4] A. Glashenko, S. Inozemtzev, I. Grachev, P. Skobelev, "Magenta Technology: case studies of Magenta i-scheduler for road transportation,"Proc. of Int. Conf. on Autonomous Agents and Multi Agent Systems (AAMAS-6), p. 1385-1392, 2007.

[5] О.Н.Граничин, “Стохастическая оптимизация и системное программирование,” Стохастическая оптимизация в информатике, No. 6, pp.3-44, 2010.

[6] А.Т. Вахитов, О.Н.Граничин, М.А. Паньшенсков. "Методы оценвания скорости передачи данных в грид,"Нейрокомпьютеры: разработка, применение, vol.11, pp.45-52, 2009

[7] T.A. Friedrich, T.B. Sauerwald, D.C. Vilenchik, "Smoothed analysis of balancing networks,"Random Structures and Algorithms, Vol. 39. No. 1, pp. 115-138, Aug. 2011.

[8] H. Li, "Load balancing algorithm for heterogeneous P2P systems based on Mobile Agent,"Proc. of ICEICE 2011, pp. 1446-1449, 2011.

[9] M.-T. Kechadi, I.K. Savvas, "Dynamic task scheduling for irregular network topologies,"Parallel Computing. Vol. 31. No. 7. pp.757-776, 2005.

[10] Я.В.Кутаева. "Балансировка загрузки несимметричного вычислительного комплекса при решении задачи статистического оценивания,"Информатика и системы управления, No.2(12), pp.88-93, 2006.

[11] ly K. Gil, C. Juiz, R. Puigjaner, "An up-to-date survey in web load balancing,"World Wide Web, Vol. 14. No. 2. pp. 105-131, 2011.

[12] N. Amelina, A. Fradkov, Y. Jiang, and D. J. Vergados, “Approximate consensus in stochastic networks with application to load balancing,” IEEE Trans. Inf. Theory, vol. 61, no. 4, pp. 1739–1752, Apr. 2015.

[13] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” IEEE Trans. Autom. Control, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[14] W. Ren and R. W. Beard, Distributed Consensus in Multi-Vehicle Cooperative Control. New York, NY, USA: Springer, 2008.

[15] F. L. Lewis,H. Zhang, K. Hengster-Movric, andA.Das, Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches. New York, NY, USA: Springer, 2013.

[16] M. DeGroot, "Reaching a consensus," J. Amer. Stat. Assoc., vol. 69, pp. 118–121, 1974.

[17] V. Borkar and P. Varaiya, "Asymptotic agreement in distributed estimation," IEEE Trans. Autom. Control, vol. AC-27, no. 3, pp. 650–655, Jun. 1982.

[18] J.N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," in Proc. Amer. Control Conf., pp. 484–489, 1984.

[19] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, 2011.

[20] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," Automatica,vol. 117, Art. no. 108962, 2020.

[21] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks,"in Proc. 3rd Int. Symp. Inf. Process. Sensor Netw., 2004, pp. 20–27, 2004.

[22] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," IEEE Trans.Autom.Control, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[23] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," Automatica, vol. 46, no. 2, pp. 322–329, 2010.

[24] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," IEEE Trans.Signal Inf. Process. Netw., vol. 2, no. 2, pp. 120–136, Jun. 2016.

[25] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," IEEE Aerospace and Electronic Systems Magazine, vol. 19, no. 1, pp. 5–18, 2004.

[26] M. R. Leonard and A. M. Zoubir, "Multi-target tracking in distributed sensor networks using particle phd filters," Signal Processing, vol. 159, pp. 130–146, 2019.

[27] X. R. Li and Y. Bar-Shalom, "Design of an interacting multiple model algorithm for air traffic control tracking," IEEE Transactions on Control Systems Technology, vol. 1, no. 3, pp. 186–194, 1993.

[28] O. Granichin, V. Erofeeva, Y.Ivanskiy, Y.Jiang, "Simultaneous Perturbation Stochastic Approximation-Based Consensus for Tracking Under Unknown-But-Bounded Disturbances" IEEE Trans. Autom. Control, vol. 66, no. 8, pp. 3710–3717, Aug. 2021.

[29] О.Н.Граничин, "Процедура стохастической аппроксимации с возмущением на входе," Автоматика и телемеханика, vol. 2, pp. 97–104, 1992.

[30] О.Н.Граничин, В.Н. Фомин, “Адаптивное управление с использованием пробных сигналов в канале обратной связи,” Автоматика и телемеханика, vol. 2, pp. 100–112, 1986.

[31] О.Н.Граничин, В.Н. Фомин, “Оптимальные порядки точности поисковых алгоритмов стохастической оптимизации,” Проблемы передачи информации, vol. 26, no. 2, pp. 126–133, 1990.

[32] J. C. Spall, “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation,” IEEE Transactions on Automatic Control, vol. 37, no. 3, pp. 332–341, 1992.

[33] Н.О.Амелина, О.Н.Граничин. "Управление балансировкой загрузки в вычислительных сетях,"Глава 13 в монографии «Проблемы сетевого управления» СПб.: Наука, pp.297-318, 2015

[34] D. Kosaty, A. Vakhitov, O. Granichin, and M. Yuchi, “Stochastic fast gradient for tracking,” in American Control Conference (ACC). IEEE, pp. 1476–1481, 2019.