

УДК: 519.8

Модификации алгоритма Frank–Wolfe в задаче поиска равновесного распределения транспортных потоков

И. Н. Игнашин^{1,a}, Д. В. Ярмошик^{1,2,b}

¹Национальный исследовательский университет «Московский физико-технический институт»,
Россия, 141701, г. Долгопрудный, Институтский пер., д. 9

²Институт проблем передачи информации РАН,
Россия, 127051, г. Москва, Большой Каретный пер., д. 19, стр. 1

E-mail: ^a ignashin.in@phystech.edu, ^b yarmoshik.dv@phystech.edu

Получено 23.12.2023.

Принято к публикации 23.12.2023.

В работе приведены различные модификации алгоритма Frank–Wolfe для задачи поиска равновесного распределения потоков. В качестве модели для экспериментов используется модель Бекмана. В этой статье в первую очередь уделяется внимание выбору направления базового шага алгоритма Frank–Wolfe (FW). Будут представлены алгоритмы: Conjugate Frank–Wolfe (CFW), Bi-conjugate Frank–Wolfe (BFW), Fukushima Frank–Wolfe (FFW). Каждой модификации соответствуют различные подходы к выбору этого направления. Некоторые из этих модификаций описаны в предыдущих работах авторов. В данной статье будут предложены алгоритмы N-conjugate Frank–Wolfe (NFW) и Weighted Fukushima Frank–Wolfe (WFFW). Эти алгоритмы являются некоторым идейным продолжением алгоритмов BFW и FFW. Таким образом, если первый алгоритм использовал на каждой итерации два последних направления предыдущих итераций для выбора следующего направления, сопряженного к ним, то предложенный алгоритм NFW использует N предыдущих направлений. В случае же Fukushima Frank–Wolfe в качестве следующего направления берется среднее от нескольких предыдущих направлений. Соответственно этому алгоритму предложена модификация WFFW, использующая экспоненциальное сглаживание по предыдущим направлениям. Для сравнительного анализа были проведены эксперименты с различными модификациями на нескольких наборах данных, представляющих городские структуры и взятых из общедоступных источников. За метрику качества была взята величина относительного зазора. Результаты экспериментов показали преимущество алгоритмов, использующих предыдущие направления для выбора шага, перед классическим алгоритмом Frank–Wolfe. Кроме того, было выявлено улучшение эффективности при использовании более двух сопряженных направлений. Например, на многих датасетах модификация 3-conjugate FW сходилась наилучшим образом. Кроме того, предложенная модификация WFFW зачастую обгоняла FFW и CFW, хотя и проигрывала модификациям NFW.

Ключевые слова: Conjugate Frank–Wolfe, Weighted Fukushima Frank–Wolfe, N-conjugate Frank–Wolfe

Работа И. Н. Игнашина была выполнена при поддержке ежегодного дохода ФЦК МФТИ (целевого капитала № 5 на развитие направлений искусственного интеллекта и машинного обучения в МФТИ, <https://fund.mipt.ru/capitals/ck5/>). Работа Д. В. Ярмошика была выполнена при поддержке Министерства науки и высшего образования Российской Федерации (госзадание), номер проекта FSMG-2024-0011.

UDC: 519.8

Modifications of the Frank – Wolfe algorithm in the problem of finding the equilibrium distribution of traffic flows

I. N. Ignashin^{1,a}, D. V. Yarmoshik^{1,2,b}

¹National Research University Moscow Institute of Physics and Technology,
9 Institute lane, Dolgoprudny, 141701, Russia

²Kharkevich Institute for Information Transmission Problems,
19/1 Bolshoy Karenty per., Moscow, 127051, Russia

E-mail: ^a ignashin.in@phystech.edu, ^b yarmoshik.dv@phystech.edu

Received 23.12.2023.

Accepted for publication 23.12.2023.

The paper presents various modifications of the Frank – Wolfe algorithm in the equilibrium traffic assignment problem. The Beckman model is used as a model for experiments. In this article, first of all, attention is paid to the choice of the direction of the basic step of the Frank – Wolfe algorithm. Algorithms will be presented: Conjugate Frank – Wolfe (CFW), Bi-conjugate Frank – Wolfe (BFW), Fukushima Frank – Wolfe (FFW). Each modification corresponds to different approaches to the choice of this direction. Some of these modifications are described in previous works of the authors. In this article, following algorithms will be proposed: N-conjugate Frank – Wolfe (NFW), Weighted Fukushima Frank – Wolfe (WFFW). These algorithms are some ideological continuation of the BFW and FFW algorithms. Thus, if the first algorithm used at each iteration the last two directions of the previous iterations to select the next direction conjugate to them, then the proposed algorithm NFW is using more than N previous directions. In the case of Fukushima Frank – Wolfe, the average of several previous directions is taken as the next direction. According to this algorithm, a modification WFFW is proposed, which uses a exponential smoothing from previous directions. For comparative analysis, experiments with various modifications were carried out on several data sets representing urban structures and taken from publicly available sources. The relative gap value was taken as the quality metric. The experimental results showed the advantage of algorithms using the previous directions for step selection over the classic Frank – Wolfe algorithm. In addition, an improvement in efficiency was revealed when using more than two conjugate directions. For example, on various datasets, the modification 3FW showed the best convergence. In addition, the proposed modification WFFW often overtook FFW and CFW, although performed worse than NFW.

Keywords: Conjugate Frank – Wolfe, Weighted Fukushima Frank – Wolfe, N-conjugate Frank – Wolfe

Citation: *Computer Research and Modeling*, 2024, vol. 16, no. 1, pp. 53–68 (Russian).

The work of I. Ignashin is supported by the annual income of the Endowment Fund of Moscow Institute of Physics and Technology (target capital no. 5 for the development of artificial intelligence and machine learning, <https://fund.mipt.ru/capitals/ck5/>). The work of D. Yarmoshik is supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye), project No. FSMG-2024-0011.

Введение

Задача поиска равновесного распределения потоков является одним из фундаментальных аспектов транспортного планирования. На данный момент методологии, основанные на машинном обучении, не достигли эффективности, сравнимой с подходами на основе теоретико-игровых моделей [Rahman, Hasan, 2022]. Поэтому исследования продолжают в рамках модели Бекмана. Проводятся исследования различных алгоритмов для решения задачи поиска равновесного распределения потоков. Так, в работе [Babazadeh et al., 2020] представлен алгоритм, использующий пути в явном виде, потому что требующий больше памяти, но достаточно эффективно работающий по времени. Также эффективными в данной задаче являются алгоритмы, основанные на использовании потоков по дорогам, к которым относится алгоритм Frank – Wolfe.

Алгоритм Frank – Wolfe нашел применение во многих областях, в том числе в транспортной логистике и телекоммуникациях. Этот метод оптимизации решает задачи математического программирования с ограничениями, но, несмотря на свою эффективность, его основной недостаток заключается в быстром замедлении сходимости после первых десятков итераций, что создает необходимость искать модификации этого алгоритма.

Алгоритм Frank – Wolfe (FW) был предложен в работе [Frank, Wolfe, 1956]. В статье [Pedregosa et al., 2020] проведен современный анализ сходимости алгоритма FW и получены оценки, которые могут быть использованы в контексте задачи, рассматриваемой в настоящей статье.

Одна из наиболее эффективных модификаций алгоритма FW — это Conjugate Frank – Wolfe (CFW). Данная модификация использует для выбора направления шага предыдущие направления, что позволяет бороться с известной проблемой «зигзага» в методе FW [Braun et al., 2022] аналогично методу сопряженных градиентов.

Алгоритм CFW хорошо описан в статье [Mitradjieva, Lindberg, 2012]. В статье доказывается глобальная сходимость алгоритма CFW. В статье также предлагается алгоритм Bi-conjugate Frank – Wolfe (BFW), в котором теперь берется выпуклая комбинация из двух предыдущих направлений и шага FW, а новое направление d^k должно быть сопряженным с этими предыдущими направлениями.

В текущей статье предлагается алгоритм N-conjugate Frank – Wolfe (NFW), обобщающий CFW до числа сопряженных направлений, равного $N > 2$. То есть используются 3 и более предыдущих направления для выбора следующего направления шага, сопряженного со всеми предыдущими N направлениями.

Другой известной модификацией FW является алгоритм Fukushima Frank – Wolfe (FFW), предложенный в статье [Fukushima, 1984]. Метод хорошо описан в статье [Arrache, Ouafi, 2008], в которой также предлагаются некоторые модификации алгоритма. Данный алгоритм из того же класса, что и CFW, BFW, NFW, ведь использует для следующего направления шага предыдущие направления FW, а именно, берет их среднее.

В текущей статье предлагается алгоритм-модификация Weighted Fukushima Frank – Wolfe (WFFW). Данный алгоритм использует вместо обычного среднего по направлениям экспоненциальное сглаживание.

Перечислим ключевые результаты данной статьи:

- 1) дан обзор существующих модификаций алгоритма FW для задачи распределения потоков;
- 2) предложены новые модификации алгоритма FW: NFW и WFFW;
- 3) проведены численные эксперименты с использованием открытого датасета [Stabler, Barga, Sall, 2023]. Установлено, что в большинстве случаев наиболее эффективным оказывается алгоритм NFW при $N = 3$.

Постановка задачи

В данной работе рассматривается модель Бекмана. Задача равновесного распределения потоков сводится к минимизации следующей потенциальной функции:

$$\Psi(f) = \sum_{e \in \mathcal{E}} \underbrace{\int_0^{f_e} \tau_e(z) dz}_{\sigma_e(f_e)} \longrightarrow \min_{f = \Theta x, x \in X}. \quad (\text{Primal})$$

В качестве функций затрат τ_e , оценивающих время проезда по дорогам при заданных потоках, используются классические BPR-функции:

$$\tau_e(f_e) = \bar{t}_e \left(1 + \rho \left(\frac{f_e}{\bar{f}_e} \right)^{1/\mu} \right), \quad (\text{Beckman})$$

$$\tau(f) \equiv \{\tau_e(f_e)\}_{e \in \mathcal{E}} \text{ — время проезда по дорогам с заданным потоком,} \quad (\text{times})$$

$\rho, \mu, \bar{t}_e, \bar{f}_e$ — константы в модели Бекмана.

Задан граф дорог, а задача — найти оптимальное распределение потоков:

$$\mathcal{E} \text{ — ребра дорожного графа,} \quad (\text{roads})$$

$$f \equiv \{f_e\}_{e \in \mathcal{E}} \text{ — потоки по дорогам.} \quad (\text{flows})$$

Предполагается, что заданы пары источников и стоков, а также корреспонденции по ним:

$$OD \equiv \{(i, j) \mid i \in O, j \in D\} \text{ — множество пар «источник–сток»,} \quad (\text{origin-destination})$$

$$\{d_{(i,j)}\}_{(i,j) \in OD} \text{ — корреспонденции или заданный поток машин из } i \text{ в } j, \quad (\text{correspondences})$$

$$\mathcal{P} = \{\mathcal{P}_{i,j}\}_{(i,j) \in OD} \text{ — множество всех путей из } O \text{ в } D.$$

Корреспонденции, соответствующие каждой паре «источник–сток», аддитивно разбиваются на потоки по путям, связывающим эти пары:

$$X = \left\{ x \mid d_{(i,j)} = \sum_{p \in \mathcal{P}_{i,j}} x_p, x_p \geq 0 \forall (i, j) \in OD \right\}.$$

Потоки по всем путям, проходящим по некоторой дороге, накладываются и порождают некоторую величину потока на этой дороге. Это можно записать компактно, введя матрицу принадлежности ребер путям:

$$\Theta \equiv \{\delta_{e,p}\}_{e \in \mathcal{E}, p \in \mathcal{P}} \text{ — матрица принадлежности ребер } e \text{ путям } p,$$

$$x \equiv \{x_p\}_{p \in \mathcal{P}} \text{ — потоки по заданным путям } p, \quad (\text{flow on path})$$

$$f = \Theta x \text{ — связь потоков по дорогам и по путям.}$$

Сама задача подробно описана в работе [Котлярова и др., 2021].

Алгоритмы

В данном параграфе все модификации алгоритма Frank–Wolfe используют общий подход: поиск направления шага d^k с использованием либо последних направлений FW y_k^{FW} , либо последних шагов $d^{k-m} \equiv s^{k-m} - f^{k-m}$.

Frank – Wolfe

Алгоритм Frank – Wolfe можно разбить на два этапа: линейная аппроксимация целевой функции с последующей минимизацией линейного функционала и поиск направления шага с применением одномерной минимизации.

В разложении в ряд Тейлора целевой функции оставляется только первый линейный член, который потом используется в минимизации функции:

$$\Psi_k(y) \equiv \Psi(f^k) + \langle \nabla \Psi(f^k), y - f^k \rangle. \quad (\text{linear approx})$$

Решается линейная задача:

$$\min_{y=\Theta x, x \in X} \Psi_k(y) \iff \min_{y=\Theta x, x \in X} \langle \nabla \Psi(f^k), y \rangle. \quad (\text{linear problem (LP)})$$

Далее делается шаг в направлении решения линейной задачи, а длина шага подбирается с помощью линейного поиска:

$$\begin{aligned} s_k^{FW} &\equiv \arg \min_{y=\Theta x, x \in X} \langle \nabla \Psi(f^k), y \rangle = \arg \min_{y=\Theta x, x \in X} \langle \tau(f^k), y \rangle, & (\text{solve LP}) \\ d_k^{FW} &= s_k^{FW} - f^k, & (\text{directional}) \\ f^{k+1} &= f^k + \gamma_k d_k^{FW}, \quad \gamma_k = \arg \min_{\gamma \in [0, 1]} (f^k + \gamma d_k^{FW}). & (\text{line search}) \end{aligned}$$

Стоит заметить, что наиболее вычислительно затратная операция в алгоритме FW для задачи распределения потоков — решение линейной подзадачи алгоритмами поиска кратчайших путей. Сама же целевая функция представляется в виде суммы простых одномерных функций, и вычисление ее и ее производных значительно проще поиска кратчайших путей в графе. Поэтому добавление в алгоритм FW процедур линейного поиска или описанного далее поиска сопряженных направлений не приводит к заметному увеличению сложности итерации.

В дальнейшем решение линейной задачи будет называться направлением FW и обозначаться как y_k^{FW} или s_k^{FW} . При этом данная линейная задача минимизации решается путем поиска кратчайших путей в графе дорог с текущими затратами $\tau(f^k)$, ведь $\langle \tau(f^k), y \rangle$ равно суммарным затратам при запуске потоков y по дорогам.

На данный момент известна сходимость алгоритма по прямой функции $\Psi(f)$ и по зазору FW из статьи [Pedregosa et al., 2020], где L — константа Липшица градиента, $\text{diam}(C)$ — диаметр множества $\{f \mid f = \Theta x, x \in X\}$,

$$g_k \equiv \max_{s=\Theta x, x \in X} \langle \nabla \Psi(f^k), f^k - s \rangle, \quad (\text{frank-wolfe gap})$$

$$g_k^* = \min_{m \in \overline{1, k}} g_m, \quad (\text{min FW gap})$$

$$g_k^* \leq \frac{\min \{2h_0, L \text{diam}(C)^2\}}{\sqrt{k+1}}, \quad (\text{FW gap convergence})$$

$$\Psi(f^k) - \Psi(f^*) \leq \frac{2L \text{diam}(C)^2}{k+1}. \quad (\text{primal convergence})$$

Эти результаты применимы и к FW в задаче поиска равновесного распределения потоков с моделью Бекмана. Ведь диаметр множества потоков по путям ограничен в силу того, что оно является прямым произведением многомерных симплексов. Липшицевость градиента обеспечивается на этом компакте.

Algorithm 1. Frank – Wolfe algorithm

-
- 1: $t^0 := \bar{t}, f^0 := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^0, s \rangle, k := 0$
 - 2: **repeat**
 - 3: $s^k := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^k, s \rangle, t_e^k := \frac{\partial \Psi(f^k)}{\partial f_e} = \tau_e(f^k)$
 - 4: $\gamma_k := \arg \min_{\gamma \in [0, 1]} ((1 - \gamma)f^k + \gamma s^k)$ or $\frac{2}{k+1}$
 - 5: $f^{k+1} := (1 - \gamma_k)f^k + \gamma_k s^k$
 - 6: $k := k + 1$
 - 7: **until** $k < \max_iter$
-

Conjugate Frank – Wolfe

Идея данного алгоритма заключается в выборе направления шага s^k как выпуклой комбинации предыдущего направления s^{k-1} и направления s_k^{FW} , соответствующего обычному шагу FW:

$$s^k = \alpha_0 s_k^{FW} + \alpha_1 s^{k-1}. \quad (\text{convex combination})$$

Выбор же коэффициентов в выпуклой комбинации находится из условия сопряженности нового шага d^k с предыдущим d^{k-1} по гессиану минимизируемого функционала $\nabla^2 \Psi(f^k)$:

$$(d^k)^T \nabla^2 \Psi(f^k) d^{k-1} = 0. \quad (\text{conjugate})$$

Гессиан $\nabla^2 \Psi(f^k)$ находится аналитически в случае модели Бекмана и вычисляется за такое же время, как и градиент.

В статье [Mitradjieva, Lindberg, 2012] также предлагается алгоритм Bi-conjugate Frank – Wolfe (BFW), в котором теперь берется выпуклая комбинация из двух предыдущих направлений и шага FW, а новое направление d^k должно быть сопряженным с этими предыдущими направлениями по гессиану в текущей точке f^k :

$$\begin{aligned} (d^k)^T \nabla^2 \Psi(f^k) d^{k-1} &= 0, \\ (d^k)^T \nabla^2 \Psi(f^k) d^{k-2} &= 0. \end{aligned} \quad (\text{bi-conjugate})$$

При этом делается предположение о сопряженности двух предыдущих направлений по новому гессиану, позволяющее получить коэффициенты выпуклой комбинации:

$$(d^{k-1})^T \nabla^2 \Psi(f^k) d^{k-2} = 0. \quad (\text{assumption})$$

Вывод этих коэффициентов написан в статье [Mitradjieva, Lindberg, 2012].

Fukushima Frank – Wolfe

Автор статьи [Fukushima, 1984] предлагает брать среднее от последних l направлений FW $\{y_{k-m}^{FW}\}_{m=0}^l$:

$$s^k = \frac{1}{l+1} \left(s_k^{FW} + \sum_{i=1}^l s_{k-i}^{FW} \right). \quad (\text{mean directions})$$

Также добавляется условие на оптимальность выбора этого среднего по сравнению с выбором только последнего направления FW.

Algorithm 2. Conjugate Frank – Wolfe algorithm

$t^0 := \bar{t}, f^0 := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^0, s \rangle, k := 0, x^* := f^0$
 2: FWM step for initialize f^1
 repeat
 4: $s^k := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^k, s \rangle, t_e^k := \frac{\partial \Psi(f^k)}{\partial f_e} = \tau_e(f^k)$
 $\alpha := \frac{(x^* - f^k)^T H^k (s^k - f^k)}{(x^* - f^k)^T H^k (s^k - f^{k-1})}, \text{ где } H^k = \nabla^2 \Psi(f^k)$
 6: Project α to $[0, \alpha_0]$
 $x^* := \alpha x^* + (1 - \alpha) s^k$
 8: $\gamma_k := \arg \min_{\gamma \in [0, 1]} ((1 - \gamma) f^k + \gamma x^*)$
 $f^{k+1} := (1 - \gamma_k) f^k + \gamma_k x^*$
 10: $k := k + 1$
 until $k < \max_iter$

Algorithm 3. Fukushima Frank – Wolfe algorithm

$l \in \mathbb{N}$
 $t^0 := \bar{t}, f^0 := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^0, s \rangle, k := 0$
 3: **repeat**
 $s^k := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^k, s \rangle, t_e^k := \frac{\partial \Psi(f^k)}{\partial f_e} = \tau_e(f^k)$
 $q = \min(k, l) - 1$
 6: $v_k = \sum_{i=k-q}^k \mu_i s^i - f^k, \sum_{i=k-q}^k \mu_i = 1, \mu_i \geq 0$
 $w_k = s^k - f^k$
 9: $d^k = \begin{cases} v_k, & \text{если } \frac{\nabla \Psi(f^k)^T v_k}{\|v_k\|} \leq \frac{\nabla \Psi(f^k)^T w_k}{\|w_k\|}, \\ w_k, & \text{иначе} \end{cases}$
 $\gamma_k := \arg \min_{\gamma \in [0, 1]} (f^k + \gamma d^k)$
 $f^{k+1} := f^k + \gamma_k d^k$
 12: $k := k + 1$
 until $k < \max_iter$

Weighted Fukushima Frank – Wolfe

Идея брать скользящее среднее по предыдущим направлениям в FFW привела к идее делать экспоненциальное сглаживание по этим направлениям. Поэтому предлагается Weighted FFW (WFFW), в котором производится экспоненциальное сглаживание по направлениям FW s_k^{FW} :

$$\begin{aligned}
 Q_k &= \beta Q_{k-1} + (1 - \beta) s_k^{FW}, & (\text{exponential smoothing}) \\
 d_k &= Q_k - f^k. & (\text{direction})
 \end{aligned}$$

Algorithm 4. Weighted Fukushima Frank – Wolfe algorithm

$W \in [0, 1]$
 $t^0 := \bar{t}, f^0 := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^0, s \rangle, k := 0, Q^0 = f^0$
repeat
 4: $s^k := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^k, s \rangle, t_e^k := \frac{\partial \Psi(f^k)}{\partial f_e} = \tau_e(f^k)$
 $Q^k = (1 - W)Q^{k-1} + Ws^k$
 $d^k = Q^k - f^k$
 $\gamma_k := \arg \min_{\gamma \in [0, 1]} (f^k + \gamma d^k)$
 8: $f^{k+1} := f^k + \gamma_k d^k$
 $k := k + 1$
until $k < \max_iter$

N-conjugate Frank – Wolfe

Как и в случае Conjugate Frank – Wolfe берется выпуклая комбинация s_k^{FW} и предыдущих направлений $\{s^{k-i}\}_{i=1}^{N_{curr}}$. Имеется степень свободы, связанная с выбором коэффициентов выпуклой комбинации. Эта степень свободы позволяет сделать следующее направление сопряженным с остальными, если выполняется предположение, что предыдущие направления сопряжены между собой по новому подсчитанному гессиану:

$$(d^{k-m})^T \nabla^2 \Psi(f^k) d^{k-n} = 0 \quad \forall m \neq n \in \overline{1, N_{curr}}. \quad (\text{assumption})$$

Оно техническое, ведь в дальнейших выкладках упрощает расчет коэффициентов, зануляя слагаемые в сумме. Причем оно не лишено смысла: если на предыдущих итерациях направления сопряжены по предыдущим гессианам, а гессиан медленно меняется, то с некоторой точностью предыдущие направления сопряжены и по новому гессиану.

Благодаря предположению все направления, включая новое подсчитанное, $\{d^{k-j} \mid j \in \overline{0, N_{curr}}\}$, будут сопряжены между собой попарно.

Следующий шаг алгоритма — это выпуклая комбинация предыдущих направлений и нового подсчитанного направления FW:

$$s^k = \alpha_0 s_k^{FW} + \sum_{m=1}^{N_{curr}} \alpha_m s^{k-m}. \quad (\text{convex combination})$$

Это можно переписать в терминах d^k :

$$d^k = \alpha_0 s_k^{FW} + \sum_{m=1}^{N_{curr}} \alpha_m s^{k-m} - f^k. \quad (\text{direction})$$

Из определения $f^k = (1 - \gamma_{k-1})f^{k-1} + \gamma_{k-1}s^{k-1}$ и $d^{k-1} = s^{k-1} - f^{k-1}$ можно получить:

$$\bar{d}^{k-1} \equiv (1 - \gamma_{k-1})d^{k-1}, \quad (1)$$

$$s^{k-1} - f^k = \bar{d}^{k-1}. \quad (2)$$

Запишем телескопическую сумму и внедрим ее в сумму, задающую направление:

$$s^{k-m} - f^k = \sum_{i=0}^{m-2} (s^{k-m+i} - s^{k-m+i+1}) + s^{k-1} - f^k, \quad (\text{telescoping sum})$$

$$d^k = \alpha_0 d_k^{FW} + \sum_{m=1}^{N_{curr}} \alpha_m \left(\sum_{i=0}^{m-2} (s^{k-m+i} - s^{k-m+i+1}) + \bar{d}^{k-1} \right). \quad (3)$$

Перейдем от терминов s^k к терминам d^k и $\bar{d}^k$, воспользовавшись выражением (2), а после перепишем сумму в этих терминах и получим удобную запись направления шага NFW:

$$s^{k-m+i} - s^{k-m+i+1} = s^{k-m+i} - s^{k-m+i+1} - f^{k-m+i+1} + f^{k-m+i+1} = \bar{d}^{k-m+i} - d^{k-m+i+1}, \quad (4)$$

$$d^k = \alpha_0 d_k^{FW} + \sum_{m=1}^{N_{curr}} \alpha_m \left(\sum_{i=0}^{m-2} (\bar{d}^{k-m+i} - d^{k-m+i+1}) + \bar{d}^{k-1} \right), \quad (5)$$

$$d^k = \alpha_0 d_k^{FW} + \sum_{m=1}^{N_{curr}} \alpha_m \left(\sum_{i=0}^{m-1} \bar{d}^{k-m+i} - \sum_{i=0}^{m-2} d^{k-m+i+1} \right). \quad (\text{direction NFW})$$

Теперь внедрим результаты выкладок выше в условие сопряженности ($H^k \equiv \nabla^2 \Psi(f^k)$):

$$d^{k-m} H^k d^k = 0, \quad (\text{conjugate})$$

$$d^{k-m} H^k \left[\alpha_0 d_k^{FW} + \sum_{m=1}^{N_{curr}} \alpha_m \left(\sum_{i=0}^{m-1} \bar{d}^{k-m+i} - \sum_{i=0}^{m-2} d^{k-m+i+1} \right) \right] = 0. \quad (6)$$

Добавим некоторые переобозначения и выпишем еще раз предположение:

$$d^{k-m} H^k d^{k-n} = 0 \quad \forall m \neq n \in \overline{1, N_{curr}}, \quad (\text{assumption})$$

$$A_{k-m} \equiv d^{k-m} H^k d_k^{FW}, \quad (A)$$

$$B_{k-m} \equiv d^{k-m} H^k d^{k-m}. \quad (B)$$

Выражение (direction NFW) имеет много нулевых слагаемых, исходя из предположения. Поэтому после упрощения выражения можно последовательно выражать каждый коэффициент α_j , начиная с α_N :

$$d^{k-m} H^k \left[\alpha_0 d_k^{FW} + \bar{d}^{k-m} \sum_{i=m}^{N_{curr}} \alpha_i - d^{k-m} \sum_{i=m+1}^{N_{curr}} \alpha_i \right] = 0, \quad (\text{simplified})$$

$$\alpha_0 A_{k-m} + (1 - \gamma_{k-m}) B_{k-m} \alpha_m + B_{k-m} \left(\sum_{i=m+1}^{N_{curr}} \alpha_i (-\gamma_{k-m}) \right) = 0. \quad (\text{add A and B})$$

В итоге можно получить выражения для α_m . Делается переобозначение $\alpha_m \equiv \beta_m \alpha_0$, после можно последовательно вычислять β_m , начиная с $m = N_{curr}$, заканчивая $m = 1$, сохраняя в памяти

и последовательно изменяя сумму $\sum_{n=m+1}^{N_{curr}} \beta_n$:

$$\beta_{N_{curr}} = \frac{-A_{k-N_{curr}}}{B_{k-N_{curr}} (1 - \gamma_{k-N_{curr}})}, \quad (7)$$

$$\beta_m = \frac{-A_{k-m}}{B_{k-m} (1 - \gamma_{k-m})} + \frac{\gamma_{k-m}}{1 - \gamma_{k-m}} \sum_{n=m+1}^{N_{curr}} \beta_n \quad \forall m = N_{curr} - 1, \dots, 1, \quad (8)$$

$$\alpha_m \equiv \beta_m \alpha_0 \quad \forall m \in \overline{1, N_{curr}}, \quad (9)$$

$$\alpha_0 = \frac{1}{1 + \sum_{i=1}^{N_{curr}} \beta_i}. \quad (\text{convex combination})$$

Если $\gamma_k = 1$, то текущие потоки f^k не учитываются в получении новых f^{k+1} . В случае FW f^{k+1} будет находиться на вершине симплекса, т. е. не в относительной внутренности. Истинное решение f^* зачастую лежит в относительной внутренности. Для избежания такой ситуации используется следующая эвристика: когда $\gamma_k \geq \gamma_{\max}$, то «обнуляются» накопленные N_{curr} направлений и получение N сопряженных направлений начинается с начала.

Algorithm 5. N-conjugate Frank – Wolfe algorithm

$N \in \mathbb{N}$, $\gamma_{\max} \in [0, 1]$

$t^0 := \bar{t}$, $f^0 := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^0, s \rangle$, $k := 0$, $N_{curr} = 1$

repeat

$s_k^{FW} := \arg \min_{s \in \{\Theta x : x \in X\}} \langle t^k, s \rangle$, $t_e^k := \frac{\partial \Psi(f^k)}{\partial f_e} = \tau_e(f^k)$

5: Вычислить $\{\alpha_i\}_{i=0}^{N_{curr}}$

$s^k = \alpha_0 s_k^{FW} + \sum_{i=1}^{N_{curr}} \alpha_i s^{k-i}$

$d^k = s^k - f^k$

$\gamma_k := \arg \min_{\gamma \in [0, 1]} (f^k + \gamma d^k)$

$f^{k+1} := f^k + \gamma_k d^k$

10: $N_{curr} = \begin{cases} N_{curr} + 1, & \text{если } N_{curr} < N, \\ N_{curr}, & \text{если } N_{curr} = N, \\ 1, & \text{если } \gamma_k > \gamma_{\max} \end{cases}$

$k := k + 1$

until $k < \max_iter$

Эксперименты

В качестве оценки эффективности алгоритмов взята величина относительного зазора, которая также используется в статьях [Mitradjieva, Lindberg, 2012; Arrache, Ouafi, 2008]:

$$RG_k \equiv \frac{UBD_k - BLB_k}{BLB_k}. \quad (\text{relative gap})$$

Относительный зазор представляет собой долю разницы между верхней оценкой минимума прямой функции и ее нижней оценкой относительно последней. Эта величина служит индикатором близости текущего значения прямой функции к ее минимуму в данное время.

В качестве верхней оценки берется значение прямой функции в следующей точке:

$$UBD_k \equiv \Psi(f^{k+1}). \quad (\text{upper bound})$$

Чтобы получить нижнюю оценку в относительном зазоре, вводится следующая нижняя оценка:

$$LB_k \equiv \min_{y \in \Theta_X, x \in X} \{ \Psi(f^k) + \langle \nabla \Psi(f^k), y - f^k \rangle \}. \quad (\text{def lower bound})$$

Ее можно представить в виде разности прямой функции и зазора Frank – Wolfe:

$$\begin{aligned} g^k &\equiv \langle \nabla \Psi(f^k), f^k - s_k^{FW} \rangle, & (\text{FW gap}) \\ LB_k &= \Psi(f^k) + \langle \nabla \Psi(f^k), f^k - s_k^{FW} \rangle = \Psi(f^k) - g^k. & (\text{LB with FW gap}) \end{aligned}$$

В качестве нижней оценки для относительного зазора берется наибольшая нижняя оценка среди ранее посчитанных:

$$BLB_k \equiv \max_{m \in \overline{1, k}} LB_m. \quad (\text{best lower bound})$$

С использованием относительного зазора возможно провести верхнюю оценку отношения между разностью значения прямой функции в следующей точке и ее минимальным значением к самому минимуму функции. Это можно показать, воспользовавшись выпуклостью прямой функции:

$$\begin{aligned} \Psi(f^*) &\geq \Psi(f^k) + \langle \nabla \Psi(f^k), f^* - f^k \rangle \geq \min_{y \in \Theta_X, x \in X} \{ \Psi(f^k) + \langle \nabla \Psi(f^k), y - f^k \rangle \} \equiv LB_k, & (\text{convex}) \\ \Psi(f^*) &\geq BLB_k, \\ \frac{\Psi(f^{k+1}) - \Psi(f^*)}{\Psi(f^*)} &\leq \frac{\Psi(f^{k+1}) - BLB_k}{BLB_k} \equiv RG_k. \end{aligned}$$

Также можно заметить, что данная оценка более точная, чем относительный зазор Frank – Wolfe:

$$RG_k \leq \frac{\Psi(f^k) - LB_k}{BLB} = \frac{g^k}{BLB}.$$

Сравнение 3-conjugate FW с Bi-conjugate FW

В данном разделе уделим внимание сравнению Bi-conjugate Frank – Wolfe с его обобщением N-conjugate Frank – Wolfe, так как в статье [Mitradjieva, Lindberg, 2012] показано, что BFW эффективней, чем CFW и FW. Но для того, чтобы показать превосходство алгоритмов NFW и BFW, в эксперименты также добавлены алгоритмы FW, WFFW, CFW.

Эксперименты были проведены на датасетах Anaheim, SiouxFalls, Berlin-Tiergarten, Terrassa-Asymmetric, Chicago-Sketch, Berlin-Mitte-Center, Berlin-Friedrichshain, Barcelona, Berlin-Mitte-Prenzlauerberg-Friedrichshain-Center. Результаты изображены на рис. 1, 2, 3.

В итоге в 6 из 9 случаев NFW с $N = 3$ побеждает BFW, а в остальных случаях — проигрывает. Также можно заметить, что методы CFW, FW значительно уступают по эффективности остальным рассмотренным методам.

Сравнение FFW с Weighted FFW

В данном разделе не только проведен сравнительный анализ между алгоритмом FFW и его модификацией, но также включен в рассмотрение CFW, поскольку CFW выступает в качестве базовой модификации алгоритма. Результаты приведены на рис. 4, 5.

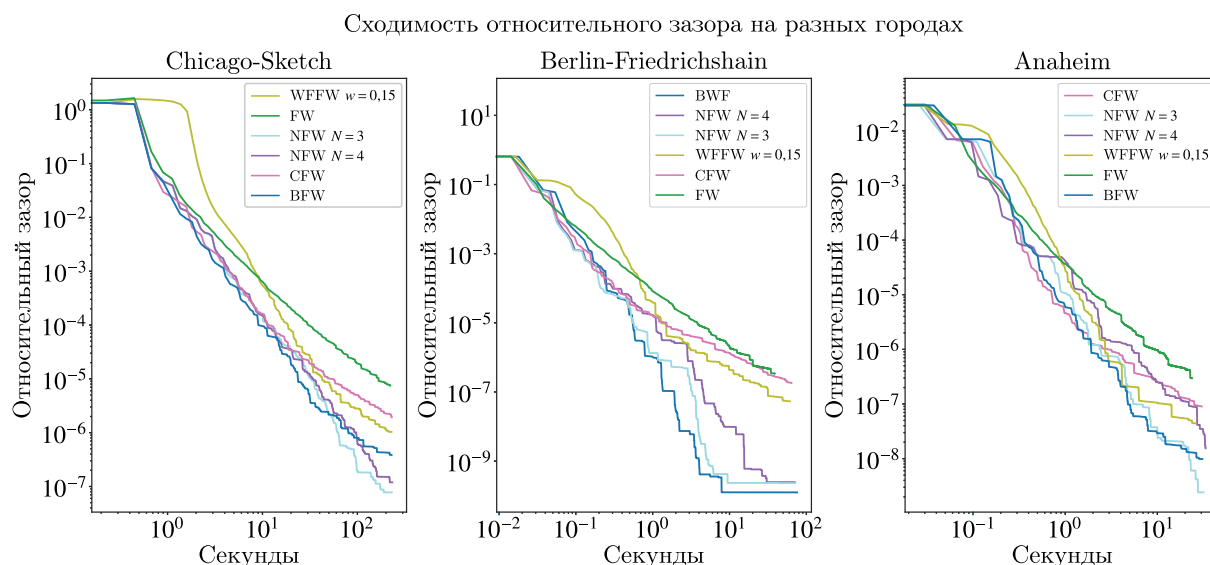


Рис. 1. Сходимость относительного зазора для алгоритмов NFW, CFW, BFW, WffW, FW по времени (в секундах) на датасетах: Chicago-Sketch, Berlin-Friedrichshain, Anaheim

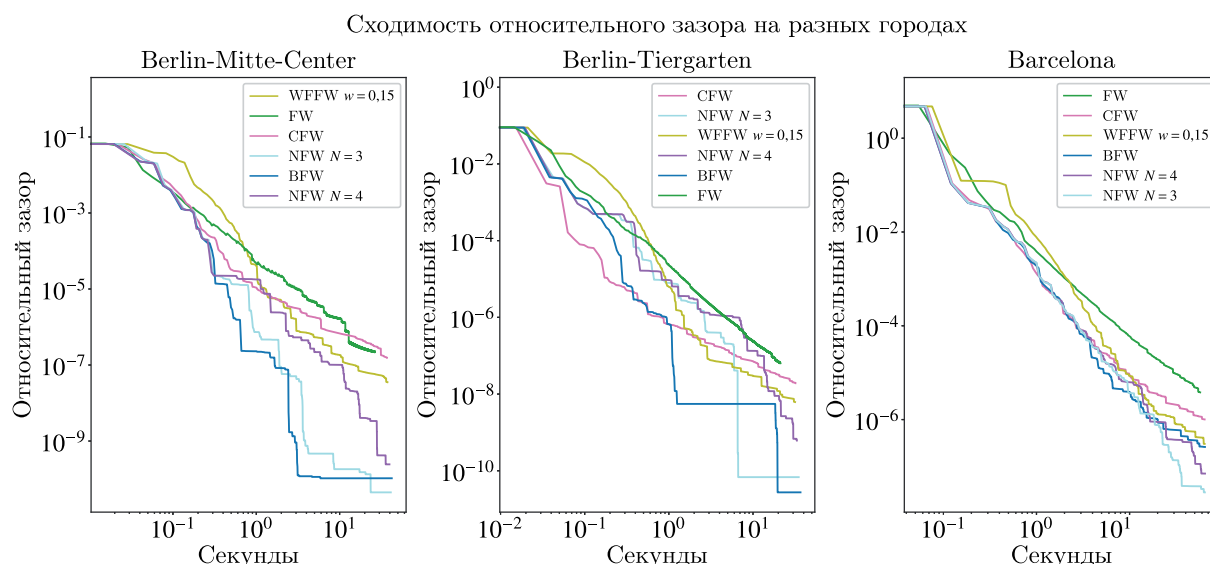


Рис. 2. Сходимость относительного зазора для алгоритмов NFW, CFW, BFW, WFFW, FW по времени (в секундах) на датасетах Berlin-Mitte-Center, Berlin-Tiergarten, Barcelona

Отмечается, что CFW проявляет сходство в процессе сходимости с FFW, в то время как WFFW демонстрирует значительное превосходство над обоими. Также можно заметить характерное поведение WFFW: в начальный момент времени он отстает от конкурирующих методов, набирая скорость, после чего обгоняет их в дальнейшем.

Сравнение NFW для разных N

Также становится интересно проверить: как на эффективность алгоритмов NFW влияет гиперпараметр N . Результаты экспериментов приведены на рис. 6, 7.

На датасете SiouxFalls лучшей моделью оказалась модель с $N = 3$. На датасете Berlin-Friedrichshain, в свою очередь, лидировал Bi-conjugate Frank–Wolfe. На датасете Terrassa-Asymmetric лучшей моделью оказалась модель с $N = 6$. Интересным является неожиданное

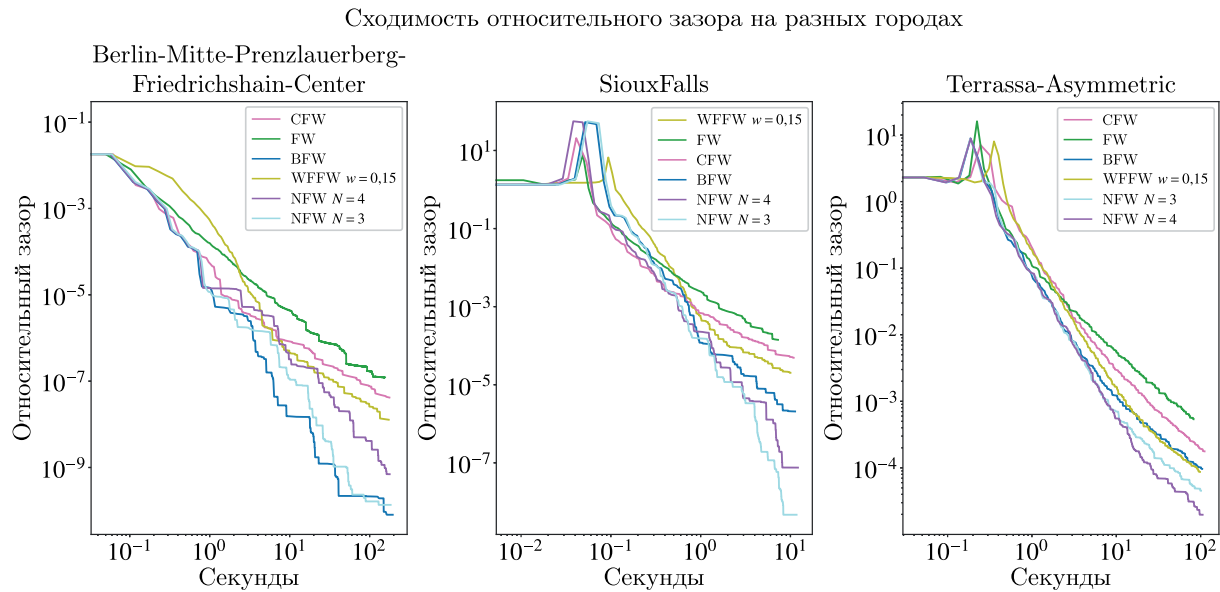


Рис. 3. Сходимость относительного зазора для алгоритмов NFW, CFW, BFW, WFFW, FW по времени (в секундах) на датасетах Berlin-Mitte-Prenzlauerberg-Friedrichshain-Center, Sioux Falls, Terrassa-Asymmetric

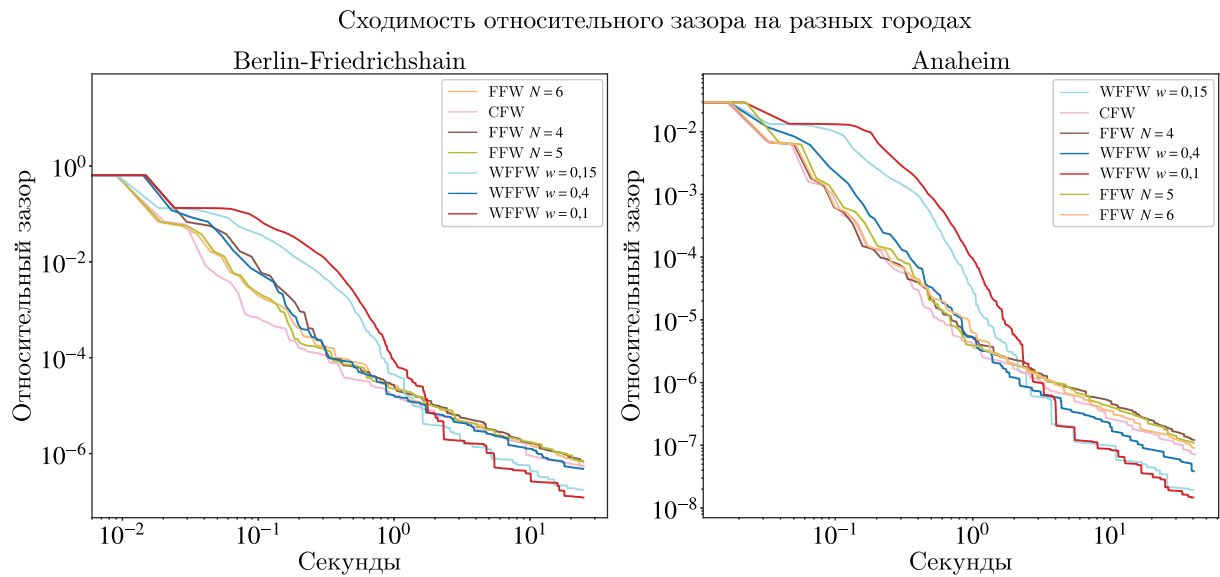


Рис. 4. Сходимость относительного зазора для алгоритмов FFW, WFFW, CFW по времени (в секундах) на датасетах Anaheim, Berlin-Friedrichshain

обстоятельство, заключающееся в том, что на данном датасете практически все модификации при различных значениях N продемонстрировали превосходство над моделью с $N = 3$. Более того, Bi-conjugate FW показал себя менее эффективным, отстающим от всех рассмотренных алгоритмов.

Из результатов экспериментов можно сделать вывод, что при использовании гиперпараметра модели $N = 3$ не всегда достигаются лучшие результаты по сравнению с другими значениями N . Однако в большинстве датасетов алгоритм все же превосходит другие модификации. Интересно отметить, что иногда модели с $N > 3$ демонстрируют лучшие результаты, в то время как в некоторых случаях побеждает Bi-conjugate Frank – Wolfe.

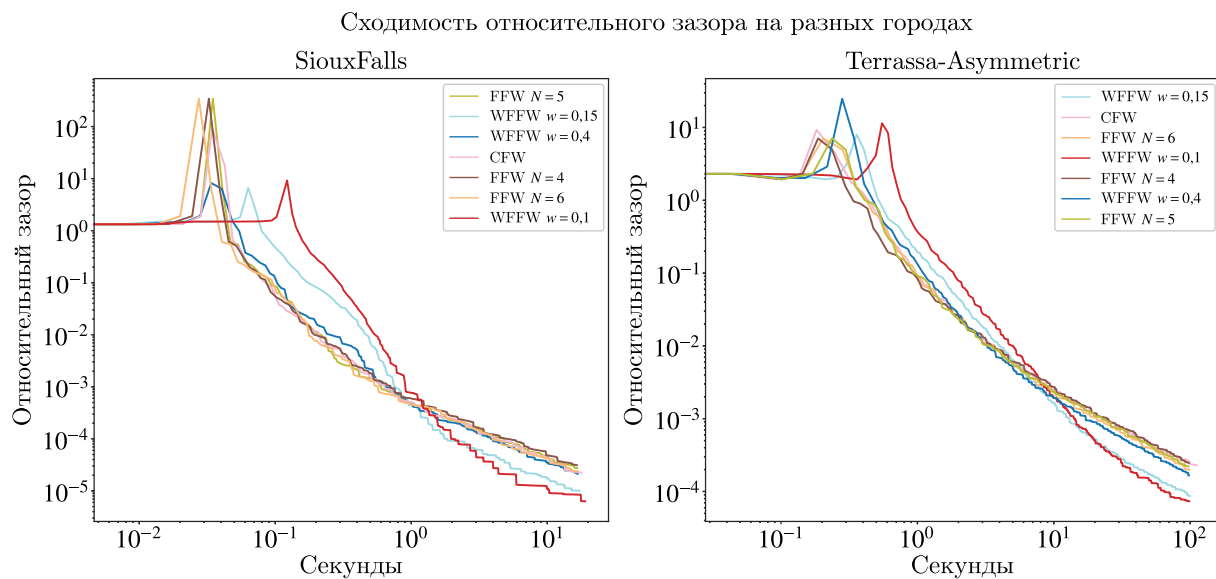


Рис. 5. Сходимость относительного зазора для алгоритмов FFW, WFFW, CFW по времени (в секундах) на датасетах SiouxFalls, Terrassa-Asymmetric

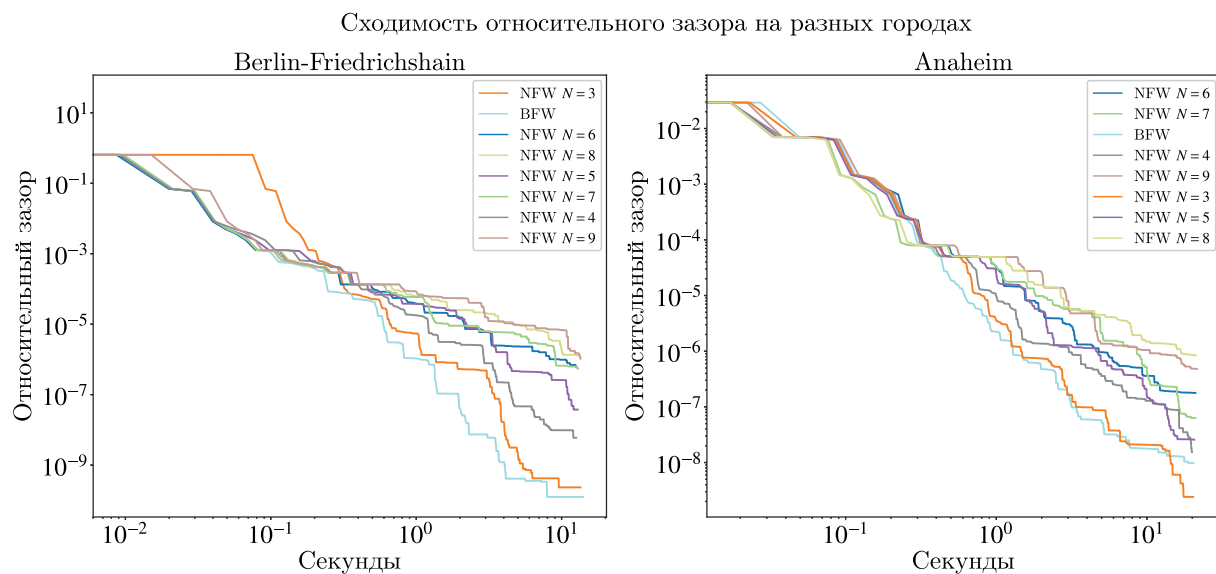


Рис. 6. Сходимость относительного зазора для алгоритмов NFW, BFW (в секундах) на датасетах Anaheim, Berlin-Friedrichshain

Сравнение WFFW для разных параметров

На рис. 8 изображены графики сходимостей WFFW при различных гиперпараметрах.

В результате анализа выявлен оптимальный выбор коэффициента затухания для заданного временного интервала. Выбор небольшого коэффициента подразумевает, что модель в значительной степени ориентируется на предыдущие направления и производит небольшие коррекции с их учетом, используя направление FW. Установлено, что уменьшение значения коэффициента приводит к увеличению времени, в течение которого относительный зазор незначительно изменяется со временем. После накопления скорости алгоритм в первую очередь движется в направлении, противоположном оптимальному направлению. Позже направление начинает настраиваться на направления FW, становясь оптимальным. Тем не менее, несмотря на данное простояние,

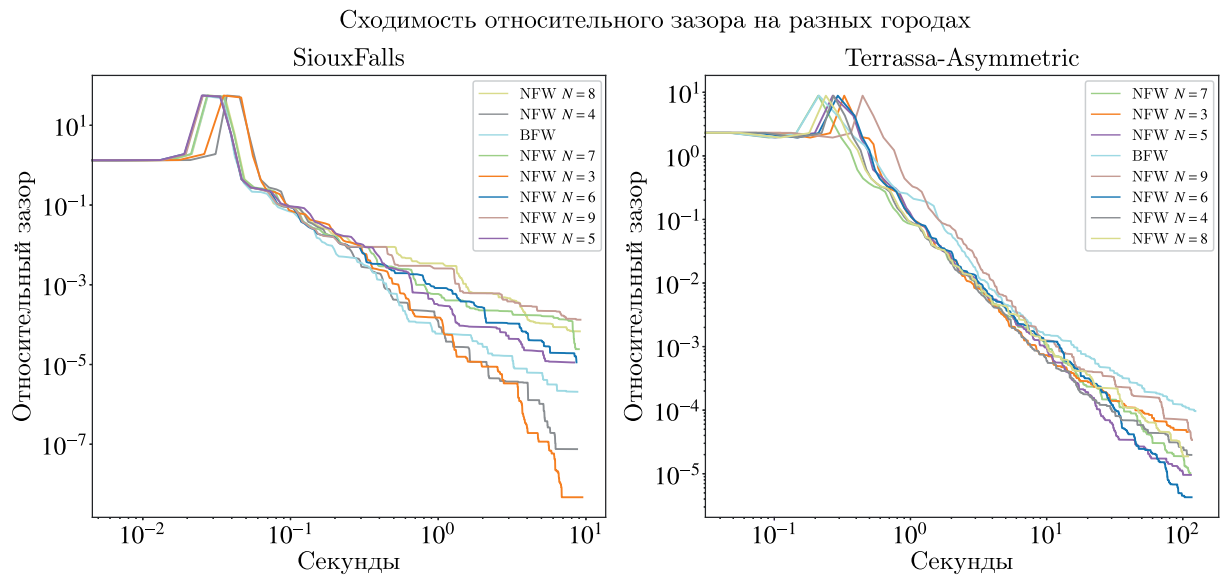


Рис. 7. Сходимость относительного зазора для алгоритмов NFW, BFW (в секундах) на датасетах SiouxFalls, Terrassa-Asymmetric

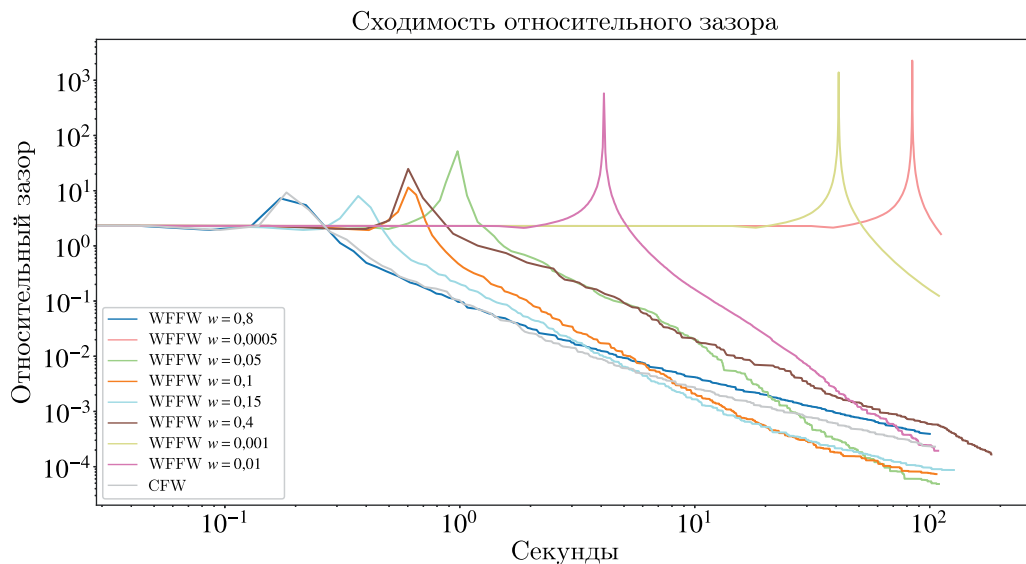


Рис. 8. Сходимость относительного зазора для алгоритмов WFFW с различными весами затухания (в секундах) на датасете Terrassa-Asymmetric

модель начинает догонять модели с более высокими значениями коэффициента. Однако при использовании очень малых значений коэффициента наблюдается значительное увеличение времени простоя.

Вывод

В настоящей статье были представлены разнообразные модификации алгоритма FW, подвергнуты сравнительному анализу на различных датасетах, представляющих реальные городские условия. В свете полученных результатов можно сделать вывод о том, что среди всех представленных модификаций наилучшие характеристики проявляют модели, относящиеся к классу N-conjugate Frank – Wolfe. Замечено, что часто оптимальным выбором является модель с $N = 3$,

что соответствует использованию трех последних направлений в процессе оптимизации. Также было показано абсолютное превосходство алгоритма Weighted FFW перед алгоритмом FFW.

Список литературы (References)

- Котлярова Е. В., Гасников А. В., Гасникова Е. В., Ярмошик Д. В.* Поиск равновесий в двухстадийных моделях распределения транспортных потоков по сети // Компьютерные исследования и моделирование. — 2021. — Т. 13, № 2. — С. 365–379.
- Kotlyarova E. V., Gasnikov A. V., Gasnikova E. V., Yarmoshik D. V.* Poisk ravnovesiy v dvukhstadiynykh modelyakh raspredeleniya transportnykh potokov po seti // Computer Research and Modeling. — 2021. — Vol. 13, No. 2. — P. 365–379 (in Russian).
- Arrache S., Ouafi R.* Accelerating convergence of the Frank–Wolfe algorithm for solving the traffic assignment problem // IJCSNS International Journal of Computer Science and Network Security. — 2008. — Vol. 8, No. 5. — P. 181.
- Babazadeh A., Javani B., Gentile G., Florian M.* Reduced gradient algorithm for user equilibrium traffic assignment problem // Transportmetrica A: Transport Science. — DOI: 10.1080/23249935.2020.1722279
- Braun G., Carderera A., Combettes C., Hassani H., Karbasi A., Mokhtari A., Pokutta S.* Conditional gradient methods // arXiv preprint. — 2022. — arXiv:2211.14103
- Frank M., Wolfe P.* An algorithm for quadratic programming // Naval Research Logistics Quarterly. — 1956. — Vol. 3, No. 1–2. — P. 95–110.
- Fukushima M.* A modified Frank–Wolfe algorithm for solving the traffic assignment problem // Transportation Research Part B: Methodological. — 1984. — Vol. 18, No. 2. — P. 169–177. — <https://www.sciencedirect.com/science/article/pii/0191261584900298>
- Mitradjieva M., Lindberg P. O.* The stiff is moving-conjugate direction Frank–Wolfe methods with applications to traffic assignment // Transportation Science. — 2012. — Vol. 47, No. 2. — P. 280–293.
- Pedregosa F., Negiar G., Askari A., Jaggi M.* Linearly convergent Frank–Wolfe with backtracking line-search // Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. — 2020. — <https://arxiv.org/pdf/1806.05123.pdf>
- Rahman R., Hasan S.* Data-driven traffic assignment: A novel approach for learning traffic flow patterns using a graph convolutional neural network // arXiv preprint. — 2022. — arXiv:2202.10508
- Stabler B., Bar-Gera H., Sall E.* Transportation Networks for Research // <https://github.com/bstabler/TransportationNetworks> (accessed: 20.12.2023).