Федеральное государственное автономное образовательное учреждение
высшего образования «Московский физико-технический институт
(государственный университет)»

Физтех-школа прикладной математики и информатики
Основная образовательная программа
Прикладная математика и информатика

ПРОЕКТ ПО КУРСУ
МЕТОДЫ ОПТИМИЗАЦИИ

"Алгоритм Франко-Вульфа"

Выполнил студент группы М05-313а, 1 курса магистратуры,
Печёнкин Александр Алексеевич

Консультант:
Федор Стонякин

Москва 2023

Содержание

1 Введение

Метод Франка-Вульфа решения оптимизационных задач был разработан уже более 65 лет назад в статье [1] Американскими математиками Маргаритой Франк и Филипом Вульфом. Однако именно за последнее десятилетие к нему появился большой интерес в связи с появлением потребности в быстрых и надежных методах оптимизации первого порядка.

Основная идея метода достаточно проста: построить последовательность вычислительных итераций, двигаясь на каждом шаге в направлении минизации линеаризированной цели.

После появления данного метода вышла серия статей о приложениях данного метода в теории оптимального управления. Также, появилось обощение данного метода для сглаживания оптимизации надо замкнутыми подмножествами банаховых пространств, где линейный оракул минимизации допустим.

В дальнейшем было доказано, что асимптотика сходимости $O\left(\frac{1}{k}\right)$ является оптимальной, когда решение лежит на границе допустимого множества. В следствие этого, появились попытки улучшить показатели при более строгих условиях. В статьях [2] и [3] была доказана линейная скорость сходимости в случае сильно выпуклых областей, предполагающих нижнюю границу градиентной нормы. В дальнейшем, результат был расширен при более общих градиентых неравенствах, а в итоге результат получилось перенести для сильно выпуклых задач с минимумом, полученным во внутренней части допустимого множества.

В последние 5 лет метод Франка-Вульфа вновь обрел огромную популяронсть благодаря своей способности достаточно эффективно спрвляться с ограничениями, возникающими в приложениях машинного обучения и Data Science.

Важной особенностью данного алгоритма является тот факт, что линейная минизация, требуемая в алгоритме, обходится дешевле, чем поиск проекций, требуемый во многих методах. В текущих реалиях важно заметить, что если даже две операции имеют одинаковую сложность, константы, определяющие соответствующие границы, могут существенно отличаться. Следовательно, при решении крупномасштабных задач метод Франка-Вульфа имеет гораздо меньшие затраты на итерацию по сравнению с методами прогнозируемого градиента. Это также верно и для затрат по памяти.

2 Основные определения, обозначения и вспомогательные утверждения

Определение 1. *Пусть* $f, g : \mathbb{N} \to \mathbb{N}$. $f = O(g)$ *если* $\exists C > 0$ *такая что* $\forall n \in \mathbb{N}$ *выполняется неравество* $f(n) \leq C \cdot g(n)$.

Определение 2. *Градиентом дифференцируемой функции* $f : \mathbb{R}^n \to \mathbb{R}$ *называется вектор частных производных функции* f. *Обозначение* ∇f.

Определение 3. *Дифференцируемая функция* f *удовлетворяет условию Липшица с константой* $L > 0$ *на множестве* $S \subset \mathbb{R}^n$ *если* $\forall \mathbf{x}, \mathbf{y} \in S$ *выполняется неравенство:*

$$||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}||$$

Утверждение 1. *(Лемма о спуске). Пусть* f *– дифференцируемая функция, удовлетворяющая условию Липшица с константой* $L > 0$ *на множестве* S. *Тогда* $\forall\, x, y \in S$ *выполняется следующее неравенство:*

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2}||\mathbf{x} - \mathbf{y}||_2^2$$

3 Описание схемы работы алгоритма

На протяжении всей работы рассматривается следующая задача:

Задача. *Пусть $S \subset \mathbb{R}^n$ – компакт и выпуклое подмножество, f – дифференцируемая функция, удовлетворяющая условию Липшица с некоторой константой L. Необходимо вычислить следующую величину:*

$$\min_{\mathbf{x} \in S} f(\mathbf{x})$$

Решение этой задачи будем обозначать через $\mathbf{x}^*$, а также через f^* будем обозначать $f(\mathbf{x}^*)$.

3.1 Схема методов первого порядка

Общая схема методов первого порядка, которые мы рассматриваем для решения *задачи 1*, основана на наборе $F(\mathbf{x}, g)$. Он вычисляется как некоторый набор направлений из точки $\mathbf{x}$ по локальной информации первого порядка g вокруг $\mathbf{x}$. В случае когда f является дифференцируемой, $g = \nabla f$. Из данного набора мы выбираем некоторый $\mathbf{d}$ с некоторым коэффициентом не больше a_{max}. При этом, этот a_{max} может зависеть от информации, которая является доступной для этого метода. В каждом методе есть информации о номере итерации, поэтому можно писать a^k_{max}.

Общая схема по итогу выглядит следующим образом:

Algorithm 1 Общая схема методов первого порядка

1: **Initialization** Выбираем $\mathbf{x}_0 \in S$
2: **for** $k = 0, \ldots$ **do**
3: Если [$\mathbf{x}_k$ удовлетворяет некоторому условию, остановить цикл.
4: Выбрать $\mathbf{d}_k \in F(\mathbf{x}_k, \nabla f(\mathbf{x}_k))$
5: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, где $\alpha_k \in (0, \alpha^k_{max}$
6: **end for**

3.2 Описание метода Франка-Вульфа

Теперь перейдем непосредственно к методу Франка-Вульфа. Напишем общую схему алгоритма, идею которой опишем после:

Algorithm 2 Метод Франка-Вульфа

1: **Initialization** Выбираем $\mathbf{x}_0 \in S$
2: **for** $k = 0, \ldots$ **do**
3: Если $\mathbf{x}_k$ удовлетворяет некоторому условию, остановить цикл.
4: Посчитать $\mathbf{s}_k \in LMO_S(\nabla f(\mathbf{x}_k))$
5: $\mathbf{d}_k := \mathbf{s}_k - \mathbf{x}_k$
6: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \cdot \mathbf{d}_k$, $\alpha_k \in (0, 1]$.
7: **end for**

Как видим, алгоритм генерирует последовательность точек $\mathbf{x}_k$ на предположении о том, что f является дифференцируемой. На k-ой итерации мы двигаемся в направление, минимизирующее скалярное произведение с текущим градиентом $\nabla f(\mathbf{x}_k)$. Следовательно, возникает необходимость в использовании оракула $LMO_S(\nabla f(\mathbf{x}_k))$, минимизирующего линейную функцию для множества S в следующем виде:

$$LMO_S(\nabla f(\mathbf{x}_k)) \in \arg\min_{\mathbf{y} \in S} \langle \nabla f(\mathbf{x}_k), y \rangle$$

Далее, определяем направление спуска как

$$\mathbf{d}_k := \mathbf{s}_k - \mathbf{x}_k, \;\; \mathbf{s_k} \in LMO_S(\nabla f(\mathbf{x}_k)) \tag{1}$$

Затем идет вычисление новой точки последовательности стандартным образом (строка 6). Заметим, что данный шаг согласно (1)можно переписать как

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k(\mathbf{s}_k - \mathbf{x}_k) = (1 - \alpha_k)\mathbf{x}_k + \alpha_k \mathbf{s}_k$$

4 Выбор размера шага

В данной секции приведем алгоритмы для определения размера шага на k-ой итерации.

4.1 Уменьшающийся шаг

Задается следующим образом:

$$\alpha_k = \frac{2}{k+2}$$

Данный шаг используется в классической реализации алгоритма Франка-Вульфа.

4.2 Точный линейный поиск

Задается следующим образом:

$$\alpha_k = \min \arg\min_{\alpha \in (0, \alpha^k_{max}} \varphi(\alpha), \quad \varphi(\alpha) := f(\mathbf{x}_k + \alpha \mathbf{d}_k)$$

Отметим, что мы выбираем наименьшую точку минимума для функции $\varphi(\alpha)$ для того, чтобы алгоритм остался детерминированным даже в случае нескольких минимумов.

4.3 Метод Армихо

Данный метод итеративно уменьшает размер шага, чтобы гарантировать достаточное уменьшение целевой функции. Это хороший способ заменить точный поиск по строке в случаях, когда он становится слишком дорогостоящим.

На практике мы фиксируем параметры $\delta \in (0, 1)$ и $\gamma \in (0, 1)$ и

пробуем шаги $\alpha := \delta^m \alpha^k_{max}$, где m пробегает множество натуральных чисел по возрастанию, пока выполняется неравенство

$$f(\mathbf{x}_k + \alpha \mathbf{d}_k) \leq f(\mathbf{x}_k) + \gamma \alpha \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

4.4 Шаг с константой Липшица

Задается следующим образом:

$$\alpha_k = \alpha_k(L) := \min \Big\{ -\frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{L \cdot ||\mathbf{d}_k||^2,}, \alpha^k_{max} \Big\}$$

Отметим, что размер шага с Липшицевой константой может рассматриваться как минимум для квадратичной модели следующего вида:

$$m_k(\alpha, L) = f(x_k) + \alpha \nabla f(\mathbf{x}_k)^T \mathbf{d}_k + \frac{L\alpha^2}{2} ||\mathbf{d}_k||^2$$

5 Сходимость метода

В данной секции мы посмотрим на скорость сходимости метода Франка-Вульфа при различных свойствах функции f и различных подходах выбора шага.

Основным параметром, который используется при измерении сходимости метода Франка-Вульфа является

$$G(\mathbf{x}) = \max_{s \in S} -\nabla f(\mathbf{x})^T (\mathbf{s} - \mathbf{x})$$

Данная величина всегда является положительной и равна нулю тогда и только тогда, когда $\mathbf{x}$ является *стационарной* точкой. Заметим, что данная величина уже по определению доступна в алгоритме.

Также отметим, что если f является дифференцируемой функцией,

то

$$G(\mathbf{x}) \geq -\nabla f(\mathbf{x})(\mathbf{x}^* - \mathbf{x}) \geq f(\mathbf{x}) - f^*$$

Таким образом, получается что $G(\mathbf{x})$ является оценкой сверху на зазор между искомым минимумом функции и значением в точке $\mathbf{x}$.

Теорема 1. *Если f является невыпуклой, то асимптотика сходимости $G(\mathbf{x})$ составляет $O\left(\frac{1}{\sqrt{k}}\right)$.*

Доказательство можно посмотреть в [4].

В случае когда f является выпуклой функцией, то достигается скорость сходимости $O\left(\frac{1}{k}\right)$ при всех вышеописанных размерах шага. Мы же рассмотрим доказательство для случая, когда шаг зависит от константы Липшица.

Перед тем, как доказывать теорему, докажем вспомогательную лемму:

Лемма 1. *Пусть $f : \mathbb{R}^n \to \mathbb{R}$ является выпуклой. Последовательности $\mathbf{d}_k$ и $\mathbf{x}_k$ генерируется по алгоритму Франка-Вульфа с шагом, описанным в методе 4. Также, $\alpha_k = 1 \ \forall k$. Тогда, верно следующее неравенство:*

$$f(\mathbf{x}_{k+1}) - f^* \leq \frac{1}{2} \min\left\{f(\mathbf{x}_k) - f^*, L||\mathbf{d}_k||^2\right\}$$

Доказательство. Заметим, что

$$G(\mathbf{x}_k) = -\nabla f(\mathbf{x}_k)^T \mathbf{d}_k \geq L||\mathbf{d}_k||^2 \tag{2}$$

Последнее неравенство в строке выше следует из того, что $\alpha_k = 1$ и определения липшицевого шага. Далее, используя *лемму о спуске*,

имеем

$$f(\mathbf{x}_{k+1}) - f^* = f(\mathbf{x}_k + \mathbf{d}_k) - f^* \leq f(\mathbf{x}_k) - f^* + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k + \frac{L}{2}||\mathbf{d}_k||^2 \quad (3)$$

Теперь, используя определение $\mathbf{d}_k$ и выпуклость f, получаем

$$f(\mathbf{x}_k) - f^* + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k \leq f(\mathbf{x}_k) - f^* + \nabla f(\mathbf{x}_k)^T (\mathbf{x}^* - \mathbf{x}_k) \leq 0 \quad (4)$$

Из (3)и (4)следует, что $f(\mathbf{x}_{k+1}) - f^* \leq \frac{L}{2}||\mathbf{d}_k||^2$ и для левой части минимума доказательство завершено.

Для правой части минимума заметим, что

$$f(\mathbf{x}_k) - f^* + + \nabla f(\mathbf{x}_k)^T \mathbf{d}_k + \frac{L}{2}||\mathbf{d}_k||^2 \leq f(\mathbf{x}_k) - f^* - \frac{1}{2}G(\mathbf{x}_k) \leq \frac{f(\mathbf{x}_k) - f^*}{2}$$

В первом переходе было использовано неравенство (2), во втором – неравенство $G(\mathbf{x}_k) \geq f(\mathbf{x}_k) - f^*$. □

Лемма 2. *Пусть $f : \mathbb{R}^n \to \mathbb{R}$ является выпуклой. Последовательности $\mathbf{d}_k$ и $\mathbf{x}_k$ генерируется по алгоритму Франка-Вульфа с шагом, описанным в методе 4. Также, $\alpha_k < 1 := a^k_{max}$ $\forall k$. Тогда, для любого $k \in \mathbb{N}$ верно следующее неравенство:*

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) - \frac{1}{2L}(\nabla f(\mathbf{x}_k)^T \hat{\mathbf{d}}_k)^2$$

Доказательство. Согласно *лемме о спуске*, имеем

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k + \frac{L\alpha_k^2}{2}||\mathbf{d}_k||^2$$

Так как $\alpha_k < 1 = \alpha^k_{max}$, то $\alpha_k = -\frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{L||\mathbf{d}_k||^2}$. Подставляя данное значе-

ние, получаем

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) - \frac{(\nabla f(\mathbf{x}_k)^T \mathbf{d}_k)^2}{2L||\mathbf{d}_k||^2} = f(\mathbf{x}_k) - \frac{1}{2L}(\nabla f(\mathbf{x}_k)^T \hat{\mathbf{d}}_k)^2$$

□

Теперь перейдем к доказательству теоремы:

Теорема 2. *Пусть $f : \mathbb{R}^n \to R$ является выпуклой. Последовательность x_k генерируется по алгоритму Франка-Вульфа с шагом, описанным в методе 4. Также, пусть D – диаметр множества S. Тогда, для любого $k \in \mathbb{N} \setminus \{0\}$ выполняется следующее неравенство:*

$$f(\mathbf{x}_k) - f^* \leq \frac{2LD^2}{k+2}$$

Доказательство. Если $\alpha_0 = 1$, то по *лемме 1* имеем

$$f(x_1) - f^* \leq \frac{L||\mathbf{d}_0||^2}{2} \leq \frac{LD^2}{2}$$

Если же $\alpha_0 < 1$, то

$$f(\mathbf{x}_0) - f^* \leq G(\mathbf{x}_0) < L||\mathbf{d}_0||^2 \leq LD^2$$

Как видим, неравенство из условия теоремы в обоих случаях выполняется, это и можно считать за базу индукции.

Перейдем к шагу индукции. Если $\alpha_k = 1$, то из неравенства $f(\mathbf{x}_{k+1}) - f^* \leq \frac{1}{2}(f(\mathbf{x}_k) - f^*)$ очевидно, что условие теоремы будет выполняться.

Если же $\alpha_k < 1$, то согласно *лемме 2* имеем

$$f(\mathbf{x}_{k+1}) - f^* \leq f(\mathbf{x}_k) - f^* - \frac{1}{2L}(\nabla f(\mathbf{x}_k)^T \hat{\mathbf{d}}_k)^2 \leq$$
$$\leq f(\mathbf{x}_k) - f^* - \frac{(\nabla f(\mathbf{x}_k)^T \mathbf{d}_k)^2}{2LD^2} \leq f(\mathbf{x}_k) - f^* - \frac{(f(\mathbf{x}_k) - f^*)^2}{2LD^2} =$$
$$= (f(\mathbf{x}_k) - f^*)\big(1 - \frac{f(\mathbf{x}_k) - f^*}{2LD^2}\big) \leq \frac{2LD^2}{k+3} \quad (1)$$

Во третьем неравенстве был использован факт, что $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k = G(\mathbf{x}_k) \leq f(\mathbf{x}_k) - f^*$. В последнем – предположение индукции. □

Далее можно рассмотреть скорость сходимости метода и в более строгих ограничениях, однако это сделать уже заметно труднее. Поэтому, в завершение просто предоставим таблицу с известными результатами:

Класс Функций	Множество S	Доп. Предположения	Сходимость
невыпуклые	произвольное	-	$O\big(\frac{1}{\sqrt{k}}\big)$
выпуклые	произвольное	-	$O\big(\frac{1}{k}\big)$
сильно выпуклые	строго выпуклое	-	$O\big(\frac{1}{k^2}\big)$
сильно выпуклые	произвольное	$\mathbf{x}^* \in int(S)$	линейная
сильно выпуклые	строго выпуклое	$\min \|\|\nabla f(\mathbf{x})\|\| > 0$	линейная

6 Примеры задач

Приведем два примера задач, в которых использование метода Франка-Вульфа достаточно эффективно.

6.1 LASSO Problem

Изначальная постановка задачи выглядит так: нужно было построить хороший инструмент для разреженной линейной регрессии. То есть,

дан тестовый набор

$$T = \{(a_i, b_i) \in \mathbb{R}^n \times \mathbb{R}\}$$

Основная цель – построить разреженную линейную модель, которая будет хорошо описывать данные. Под разреженной моделью понимается модель с небольшим количеством ненулевых параметров.

Данная задача напрямую связана с задачей BPD в анализе сигналов. Постановка задачи в BPD немного другая, однако обе они сводятся к следующей оптимизационной задаче:

$$\min_{\mathbf{x} \in \mathbb{R}^n} ||\mathbf{Ax} - \mathbf{b}||_2^2 \quad s.t. \ ||\mathbf{x}||_1 \leq \tau$$

Заметим, что область определения в данном случае можно описать как

$$C = \{\mathbf{x} \in \mathbb{R}^n : \ ||\mathbf{x}||_1 \leq \tau\} = conv\{\pm\tau\mathbf{e}_i, \ i \in \{1, ..., n\}\}$$

Также отметим, что для данной задачи оракул $LMO_S(\nabla f(\mathbf{x_k}))$ выглядит как $sign(-\nabla_{i_k} f(\mathbf{x}_k)) \cdot \tau\mathbf{e}_i$, где $i_k = \arg\max_i |\nabla_i f(\mathbf{x}_k)|$. Таким образом, получается время работы одной итерации составляет $O(n)$.

Остается заметить, что функция f в данном случае является сильно выпуклой, а также S является сильно выпуклым. Это наводит нас на тот факт, что скорость сходимости может быть линейной. И действительно, согласно, например, [5], метод Франка-Вульфа для данной задачи имеет линейную скорость сходимости при некоторых видах шага.

6.2 Поиск максимальной клики в графе

Пусть $G = (V, E)$ – неориентированный граф, V – множество вершин, $E \subset V \times V$ – набор ребер. Кликой в графе G называется $C \subset V$, такое что $\forall\, i, j \in C$, $i \neq j$ верно что $(i, j) \in E$.

Основная задача – найти такую клику C для заданного G, чтобы $|C|$ было максимальным. Данная задача имеет широкое применение в различных отраслях, таких как телекоммуникационные связи, биоинформатика и другие.

Поставленная задача может быть сведена к оптимизационной задаче следующего вида:

$$\max \left\{ \mathbf{x}^T A_G \mathbf{x} + \frac{1}{2} |\mathbf{x}|_2^2, \, \mathbf{x} \in \delta_{n-1} \right\}$$

Здесь A_G – это матрица смежности графа G. Данная постановка задачи позволяет эффективно использовать алгоритм Франка-Вульфа, см. [6].

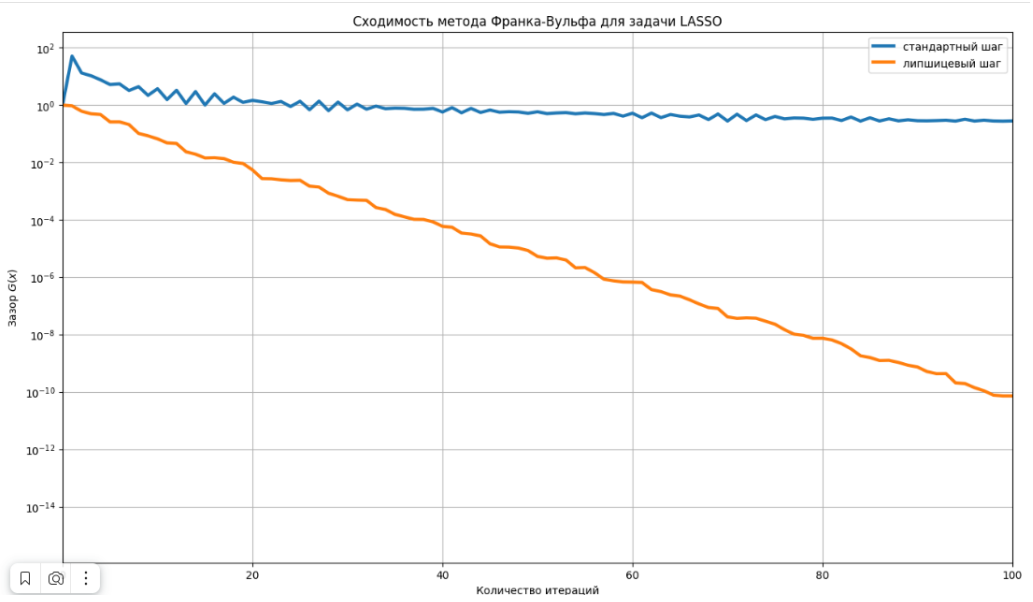
7 Практическая часть

В данном разделе мы реализуем метод Франка-Вульфа для задачи LASSO. Рассмотрим два метода выбора шага – стандартный и липшицевый. Эксперимент проводить будем следующим образом:

Из библиотеки sklearn мы берем пакет datasets и генерируем синтетическую матрицу $\mathbf{A}$ размером $10^4 \times 10^4$, а также вектор $\mathbf{b}$.

Далее мы производим расчеты по методу Франка-Вульфа с заданным шагом, параллельно записывая промежуточные значения величины $G(x) = -\nabla f(\mathbf{x_k})(\mathbf{s} - \mathbf{x}_k)$, чтобы оценивать скорость сходимости метода. Затем мы все выведем на график с помощью библиотеки matplotlib. График показывает отношение количества пройденных итераций к зазору $G(x)$ в логарифмической шкале.

Весь код можно посмотреть по ссылке https://github.com/Piachonkin-Alex/Stats-ML/tree/main/opts-masters. По итогу, у нас получается такой график:



Как можно заметить, сходимость при липшицевом шаге – линейная, а при стандартном шаге – сублинейная. Следовательно, в данной задаче липшицевый шаг значительно лучше.

7 Источники

1. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly 3(1-2), 95–110 (1956)

2. Levitin, E.S., Polyak, B.T.: Constrained minimization methods. USSR Computational Mathematics and Mathematical Physics 6(5), 1–50 (1966)

3. Demyanov, V.F., Rubinov, A.M.: Approximate methods in optimization problems. American Elsevier (1970)

4. Lacoste-Julien, S.: Convergence rate of Frank-Wolfe for non-convex objectives. arXiv preprint arXiv:1607.00345 (2016)

5. Jaggi, M.: Sparse convex optimization methods for machine learning. Ph.D. thesis, ETH Zurich (2011)

6. Hungerford, J.T., Rinaldi, F.: A general regularized continuous formulation for the maximum clique problem. Mathematics of Operations Research 44(4), 1161–1173 (2019)

7. https://github.com/paulmelki/Frank-Wolfe-Algorithm-Python