Методы оптимизации, лабораторная работа 2

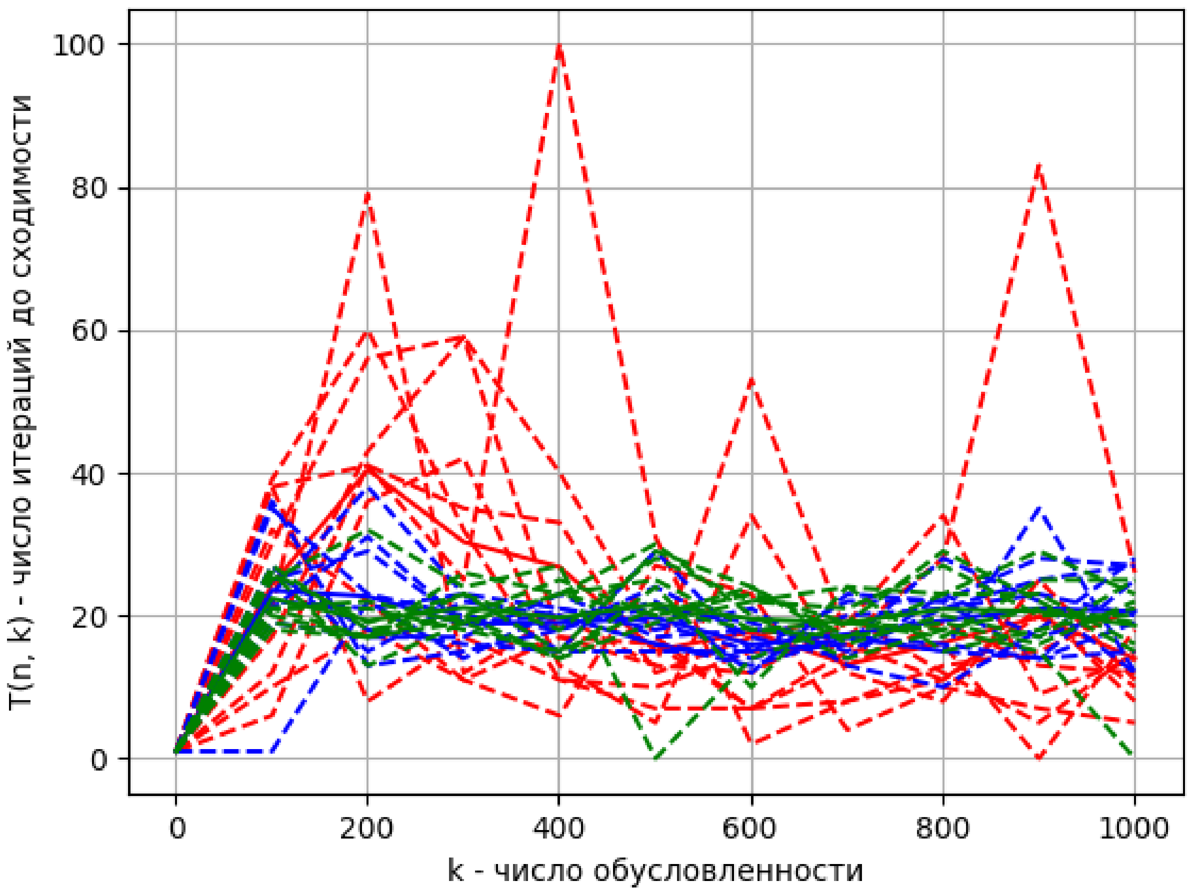
Миляуша Сабирова

1 Эксперимент: зависимость числа итерации метода сопряженных градиентов от числа обусловленности и размерности пространства

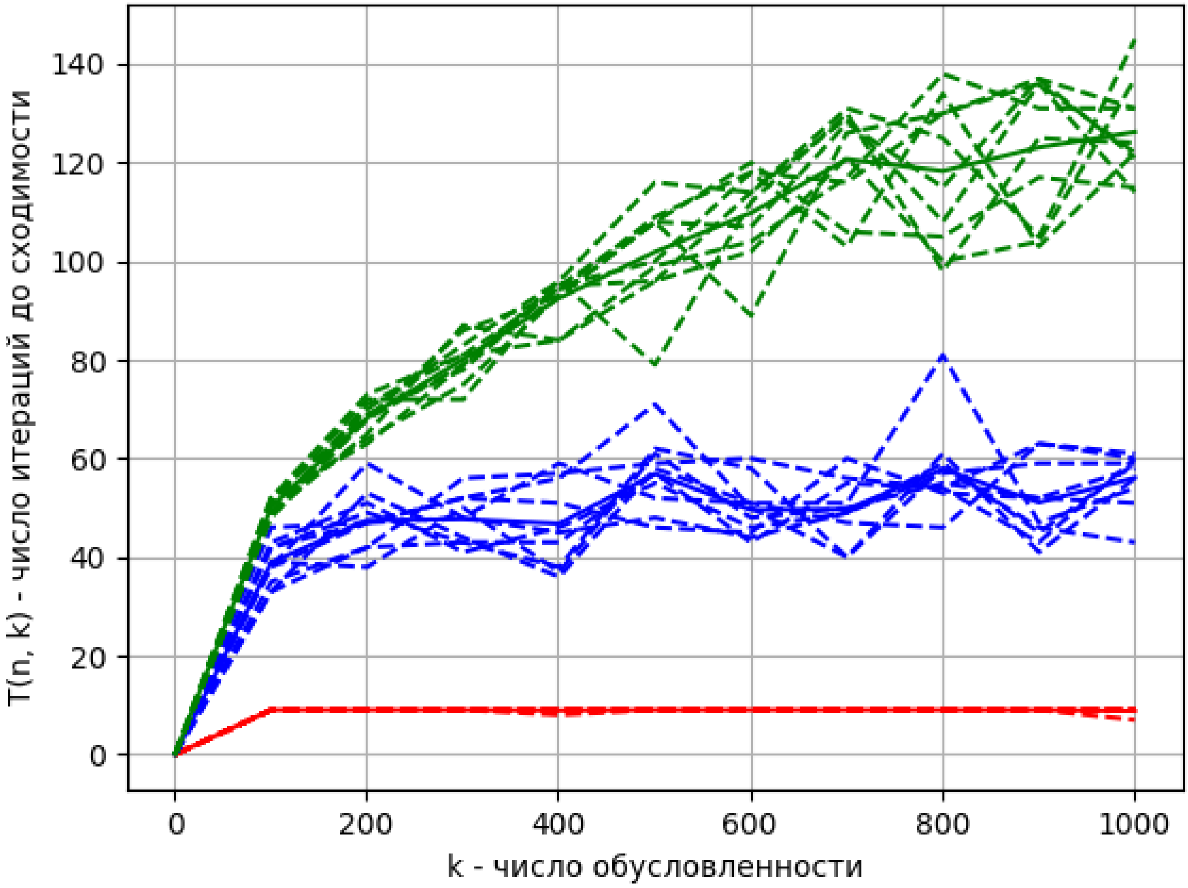
Порядок проведения эксперимента: в качестве n беру числа 10, 100 и 1000. Генерирую случайную диагональную матрицу A и вектор b так, как написано в условии эксперимента. В качестве стартовой точки всегда беру случайную. Для каждого n провожу по 10 итераций эксперимента. Для градиентного спуска я построила графики для всех стратегий поиска размера шага, в отчет вставлен график, где используются сильные условия Вульфа. Параметры для линейного поиска я использую стандартные из utils.py везде, кроме константного шага (там $c = 0.01$).

Первый график для градиентного спуска, второй - для метода сопряженных градиентов.

Красным цветом обозначены графики для $n = 10$, синим - для $n = 100$, зеленым - для $n = 1000$



T(n, k) - число итераций до сходимости
k - число обусловленности
0
20
40
60
80
100
0
200
400
600
800
1000



Выводы:

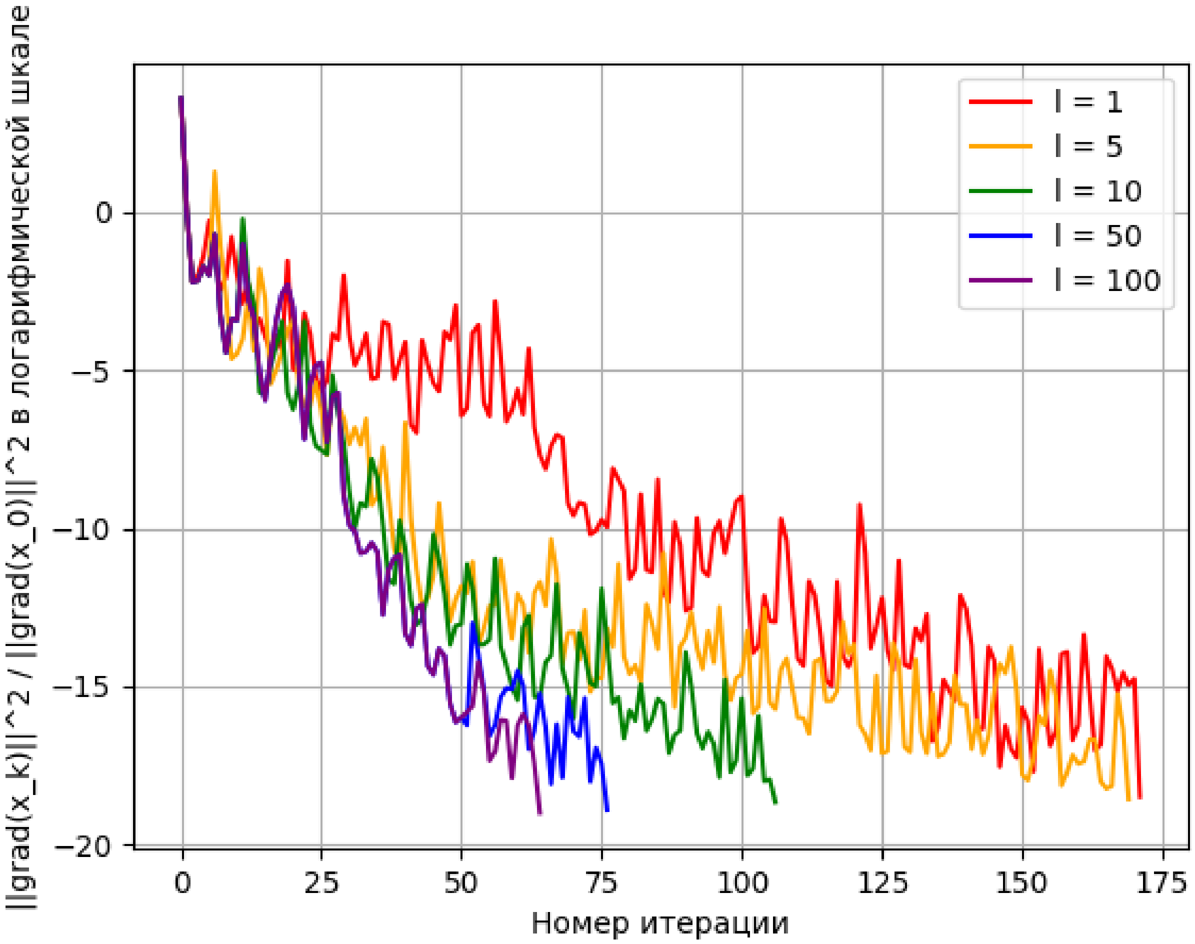
1) На графиках видно, что в методе сопряженных градиентов число итераций всегда не превосходит размерность пространства. Особенно ярко это преимущество заметно для $n = 10$: на моем графике максимальное число итераций достигает 100, но количество итераций может быть еще больше и быть порядка нескольки тысяч, тк оно зависит от разных факторов: самой оптимизируемой функции, стратегии подбора длины шага спуска (тут график для стратегии с использованием сильных условий Вульфа и бэктрекинга), критерия остановки и т.д. В теории, для $n = 10$ мы могли получить чуть больше итераций из-за неточности вычислений дробных чисел, но это маловероятно и гарантированный порядок сходимости все равно в разы лучше, чем у метода градиентного спуска. Для больших же n все вообще хорошо с этой точки зрения.

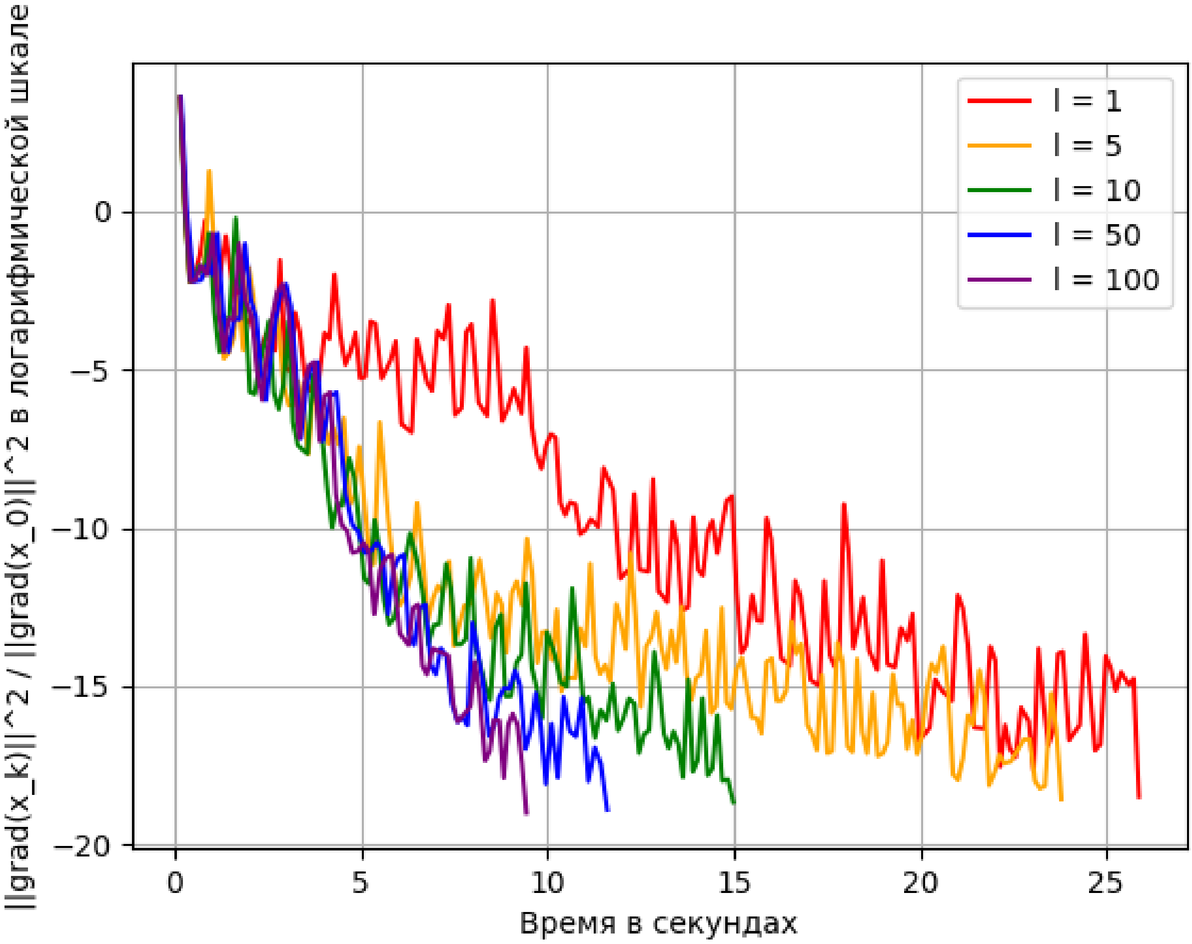
2) Количество итераций зависит от числа обусловленности примерно как $C \cdot \sqrt{n}$, что подтверждает теоретическую оценку. При достаточно больших n эта размерность пространства для роста $T(n, k)$ начинает играть существенно меньшую роль, чем число обусловленности.

2 Эксперимент: Выбор размера истории в методе L- BFGS

Размер требуемой памяти: для метода с размером истории l для каждого "момента истории"мы храним по 2 вектора размерности n, поэтому нам требуется $\mathcal{O}(ln)$ памяти.

Сложность итерации: на каждой итерации делаем не более l шагов для вычисления H_k. На каждом шаге проводим константное количество операций с векторами размерности $n \implies \mathcal{O}(ln)$.





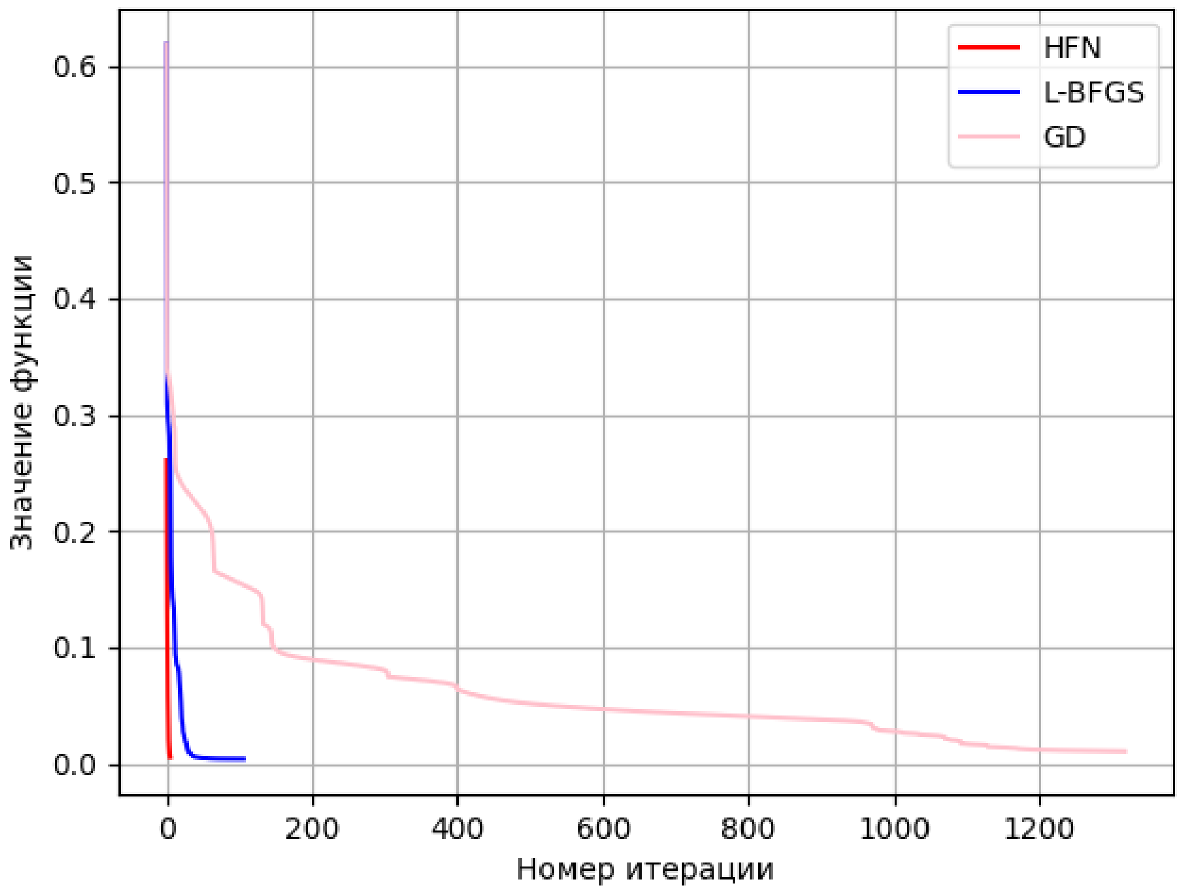
Выводы: 1) Чем больше размер истории в L-BFGS, тем меньше итераций нужно для сходимости алгоритма.

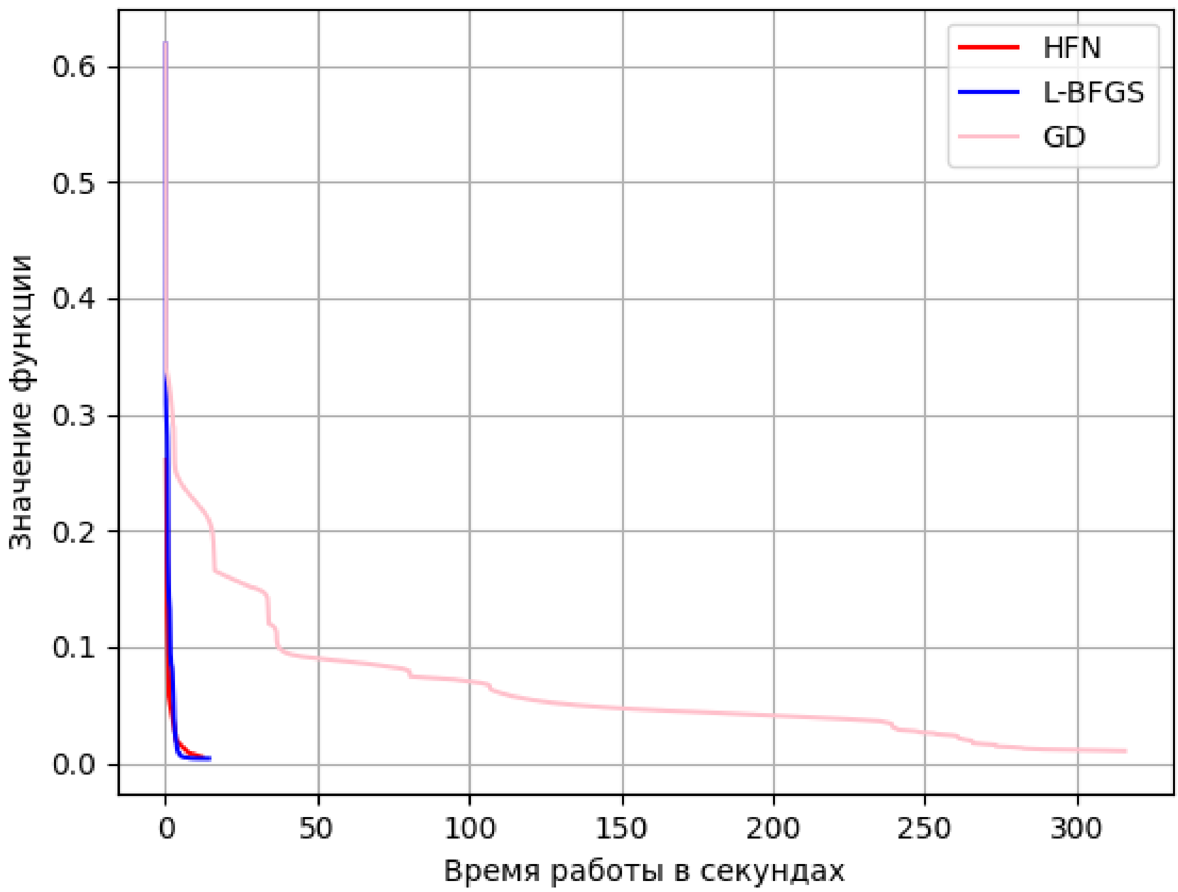
2) Все графики имеют одинаковый характер, нет каких - либо аномалий (например резких скачков), что может говорить о стабильности работы алгоритма.

3) Графики очень похожи, оси X относительно номера итераций и времени пропорциональны друг другу, что говорит о том, что на практике больший размер истории почти не замедляет время одной итерации алгоритма, поэтому имеет смысл брать l порядка 50 или 100, чтобы уменьшить количество итераций и затраченного времени, необходимых для сходимости алгоритма, то есть повысить его производительность.

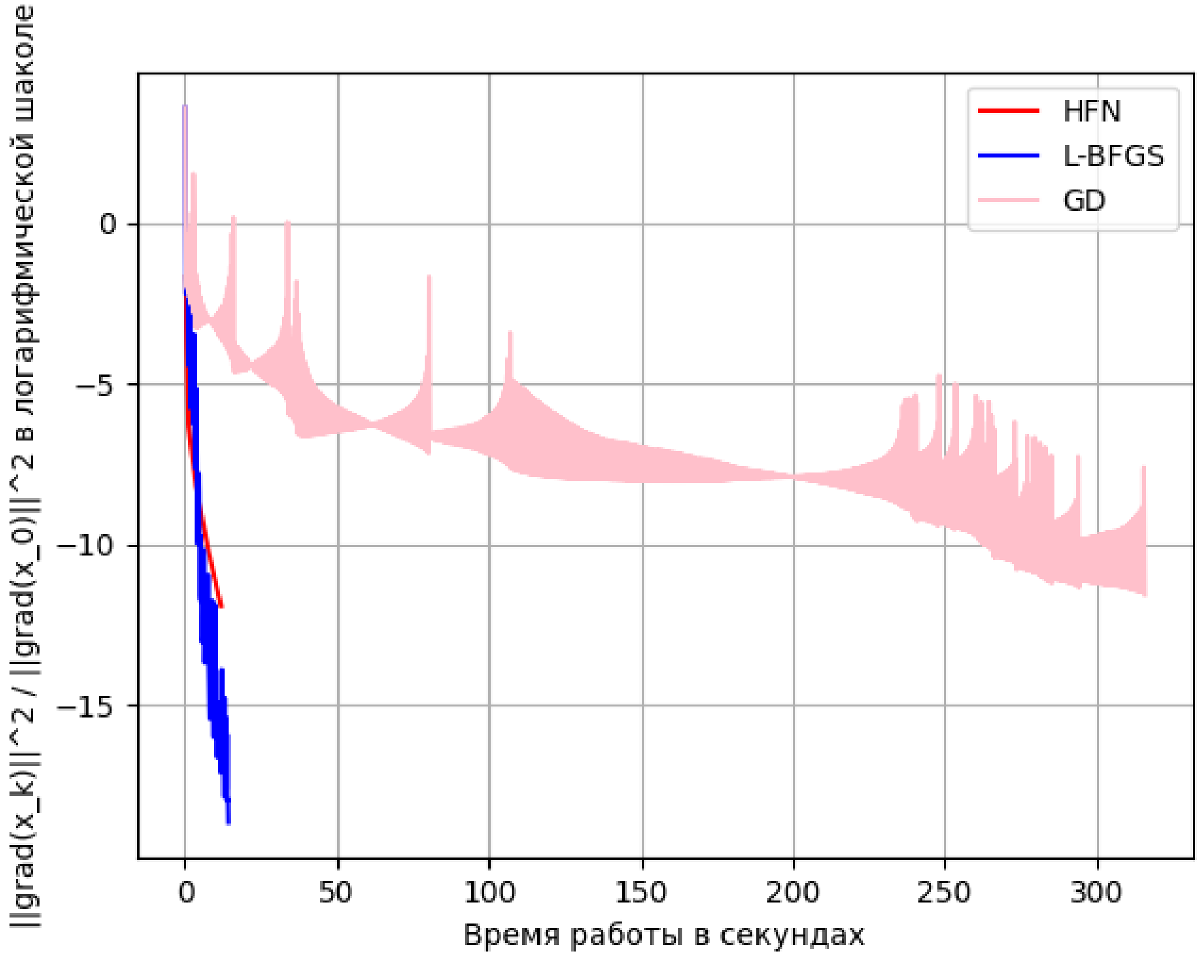
3 Эксперимент: Сравнение методов на реальнои задаче логистическои регрессии

В отчете графики для датасета gisette.


HFN
L-BFGS
GD
Значение функции
Номер итерации
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
200
400
600
800
1000
1200



HFN
L-BFGS
GD
Значение функции
Время работы в секундах
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
50
100
150
200
250
300



Выводы: 1. Обычный градиентный спуск на реальной задаче логистической регрессии работает значительно медленнее, чем квазиньютоновские методы, поэтому не стоит использовать его в подобных задачах.

2. В среднем, усеченный метод Ньютона и L-BFGS дают примерно одинаковые результаты по времени работы в задачах с реальными данными, поэтому в подобных задачах они взаимозаменяемы и работают одинаково хорошо.