Теория графов.
Вторая презентация

Зайцев Дмитрий

Введение

- Рассматриваются реализации алгоритма **Борувки** на **GraphBLAS** и **PySpark**.
- В экспериментах **нет** прямого **сравнения** между библиотеками.
- Анализируется **влияние** конкретных **свойств** графа **на скорость** его обработки выбранной библиотекой.

Характеристики вычислительной машины

- **Процессор**: AMD Ryzen 7 3800X (8 ядер, 16 потоков)
  - Базовая частота: 3900 MHz
  - L1-кэш: 512 KB
  - L2-кэш: 4 МБ
  - L3-кэш: 32 МБ
- **RAM**: 32 GB
- **GPU**: AMD Radeon RX 5700 XT
- **Операционная система**: Ubuntu 24.04.2

Формулировка экспериментов

Эксперимент 1

Цель: Найти зависимость между плотностью графа и скоростью работы алгоритма на выбранной библиотеке.

Ход эксперимента:

1. Берём граф из датасета.
2. Начинаем менять количество рёбер в нём так, чтобы можно было построить график зависимости производительности библиотеки от плотности графа (1%, 3%, 5%, 7%, 10%, 13% и 15% от исходного числа рёбер).
3. Расширяем исходный датасет новыми графами.
4. Проверяем, будет ли наблюдаться зависимость для всех графов из датасета.

Как меняем графы:

- Не трогаем мосты.
- В приоритете менее значимые ребра (по весу и центральности).
- Для каждого получившегося графа выполняем дополнительную проверку на корректность.

Эксперимент 1

Гипотеза: Нет однозначной связи между плотностью графов, полученных описанным выше методом и скоростью их обработки на выбранной библиотеке и для предложенного датасета.

Получение результатов:

- Выполняем 100 запусков алгоритма на каждом из графов в расширенном датасете.
- Находим среднее, стандартное отклонение и доверительные интервалы.
- Эти данные используем для проверки гипотезы.

Проверка гипотезы:

1. Для каждой группы графов ищем коэффициент корреляции (*коэффициент корреляции Пирсона*).
2. Для анализа используем α (уровень значимости) = 0.05.
3. Делаем поправку на множественную проверку гипотез.
4. Пытаемся опровергнуть гипотезу для каждого графа из исходного датасета.
5. Получаем табличку с результатами для каждого графа.

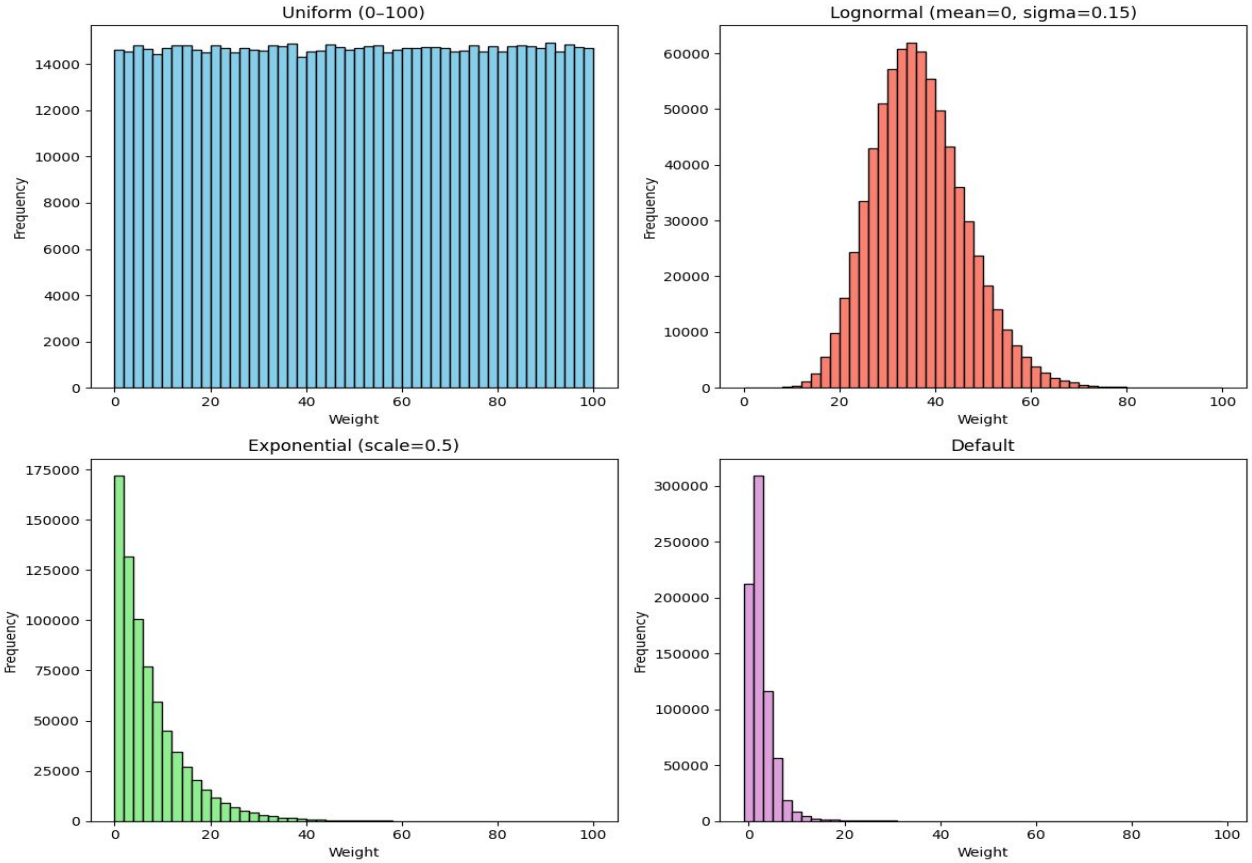
Эксперимент 2

Цель: Проанализировать производительность библиотеки на графах с одинаковой топологией, но разным распределением весов.

Ход эксперимента:

1. Берём граф из датасета.
2. Несколько раз меняем в нём распределение весов, но не трогаем топологию.
3. Расширяем исходный датасет новыми графами.
4. Проверяем, будет ли наблюдаться отличие в скорости обработки графов на обновлённом датасете.

Эксперимент 2. Распределения



Эксперимент 2

Гипотеза: Нет никакой статистически значимой разницы в производительности выбранной библиотеки на графах с различным распределением весов, но полностью одинаковой топологией.

Получение результатов:

- Выполняем 100 запусков алгоритма на каждом из графов в обновлённом датасете.
- Находим среднее, стандартное отклонение и доверительные интервалы.
- Эти данные используем для проверки гипотезы.

Проверка гипотезы:

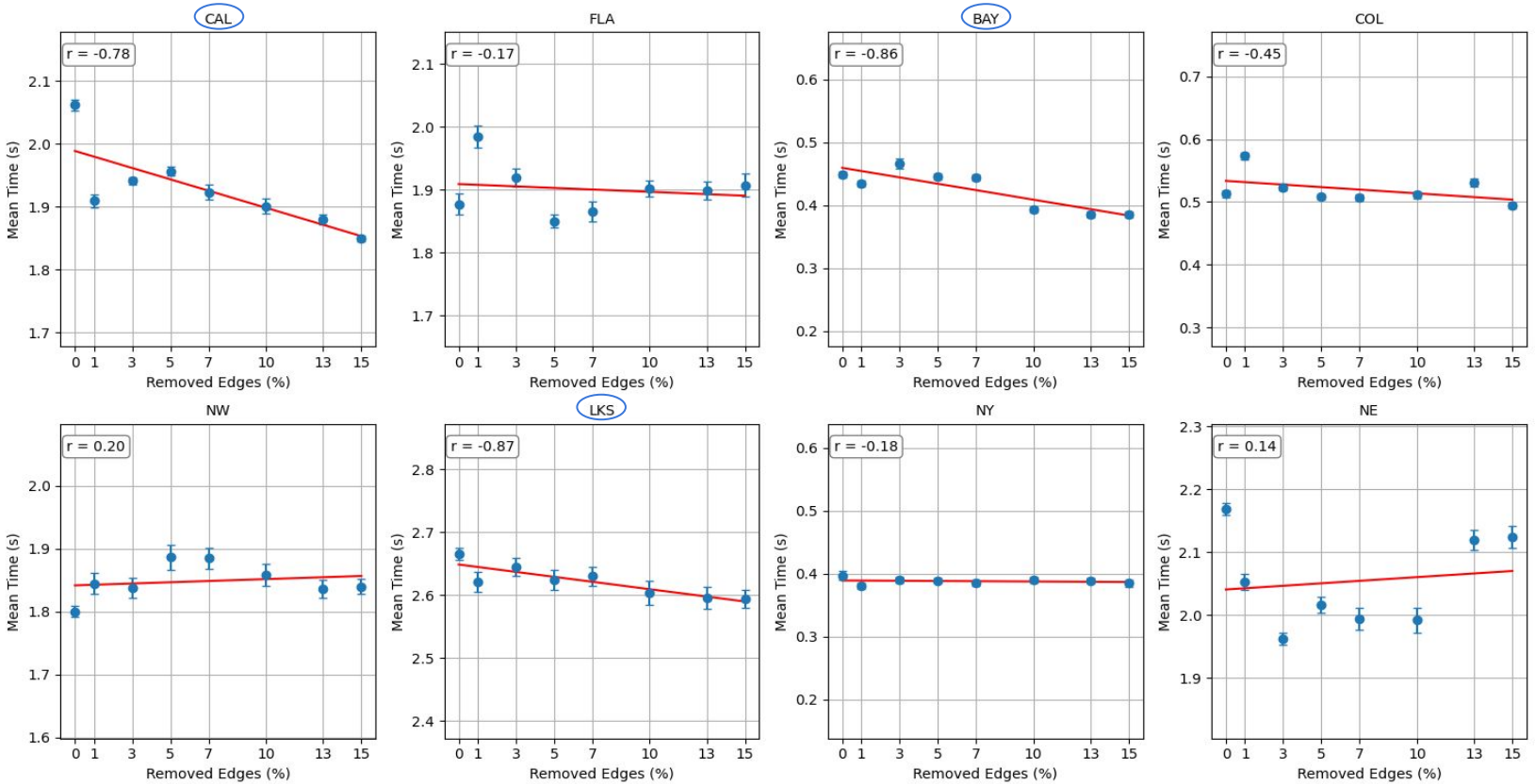
1. Для анализа используем α (уровень значимости) = 0.05.
2. Применяем однофакторный дисперсионный анализ для всего датасета.
3. Пытаемся опровергнуть гипотезу.

GraphBlas

Датасет DIMACS 9th

Description	Nodes	Edges
Great Lakes	2,758,119	3,397,404
California and Nevada	1,890,815	2,315,222
Northeast USA	1,524,453	1,934,010
Northwest USA	1,207,945	1,410,387
Florida	1,070,376	1,343,951
Colorado	435,666	521,200
San Francisco Bay Area	321,270	397,415
New York City	264,346	365,050

Эксперимент 1. Результаты для GraphBlas

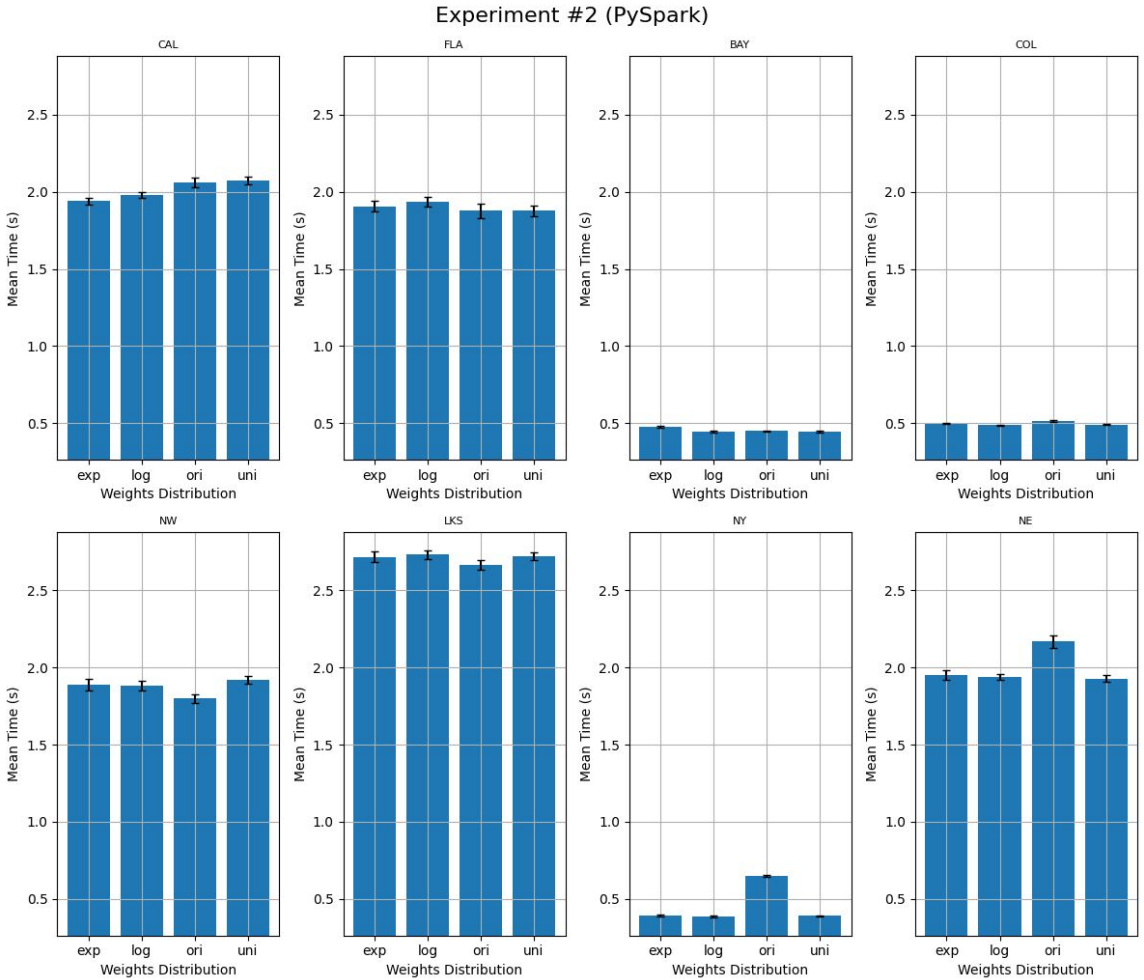


Эксперимент 1. Результаты для GraphBlas

#	Graph Name	R-value	P-value	Reject H0
1	CAL	-0.782567	0.021690	🟡False
2	FLA	-0.166988	0.692670	❌False
3	BAY	-0.861912	0.005920	✅ True
4	COL	-0.454869	0.257462	❌False
5	NW	0.195723	0.642283	❌False
6	LKS	-0.868481	0.005141	✅ True
7	NY	-0.181951	0.666296	❌False
8	NE	0.144117	0.733499	❌False

Вывод: На некоторых графах из датасета есть сильная отрицательная корреляция между плотностью графа и скоростью его обработки. С ростом плотности графов скорость работы алгоритма иногда может снижаться.

Эксперимент 2. Результаты для GraphBlas



Эксперимент 2. Результаты для GraphBlas

Distribution	N	Mean	Values
exp	8	1.471	[1.94, 1.906, 0.474, 0.497, 1.888, 2.718, 0.391, 1.952]
lognorm	8	1.474	[1.98, 1.935, 0.445, 0.486, 1.885, 2.733, 0.385, 1.94]
original	8	1.523	[2.062, 1.877, 0.449, 0.513, 1.8, 2.666, 0.646, 2.168]
uni	8	1.481	[2.073, 1.878, 0.446, 0.491, 1.918, 2.722, 0.39, 1.928]

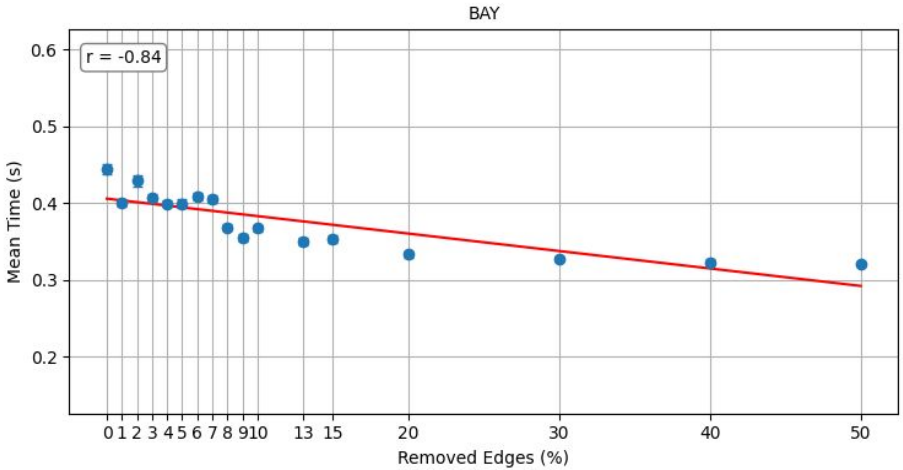
❌ P-value: 9.9935e-01 → гипотеза не опровергнута

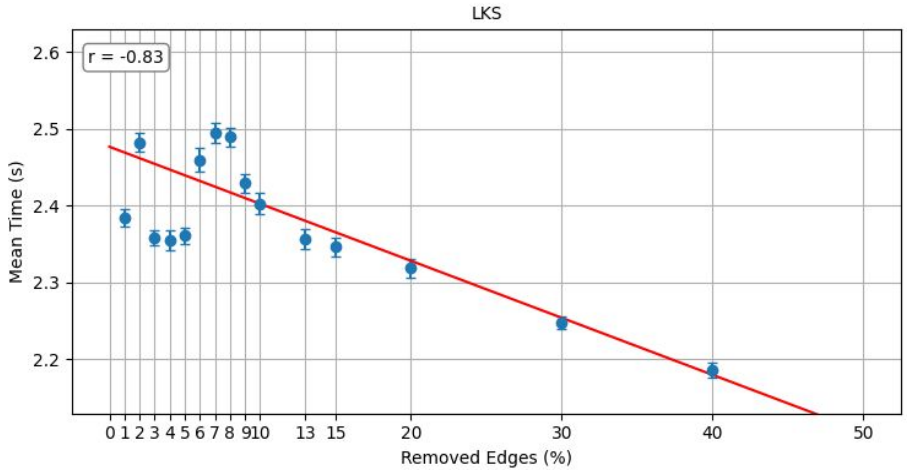
Вывод: Не получилось опровергнуть гипотезу, можно сказать, что для GraphBlas на этом датасете нет большой разницы, какое распределение будет у графа. *Возможно следует провести другой эксперимент для уточнения.*

GraphBlas
Проверка результатов

(для первого эксперимента)

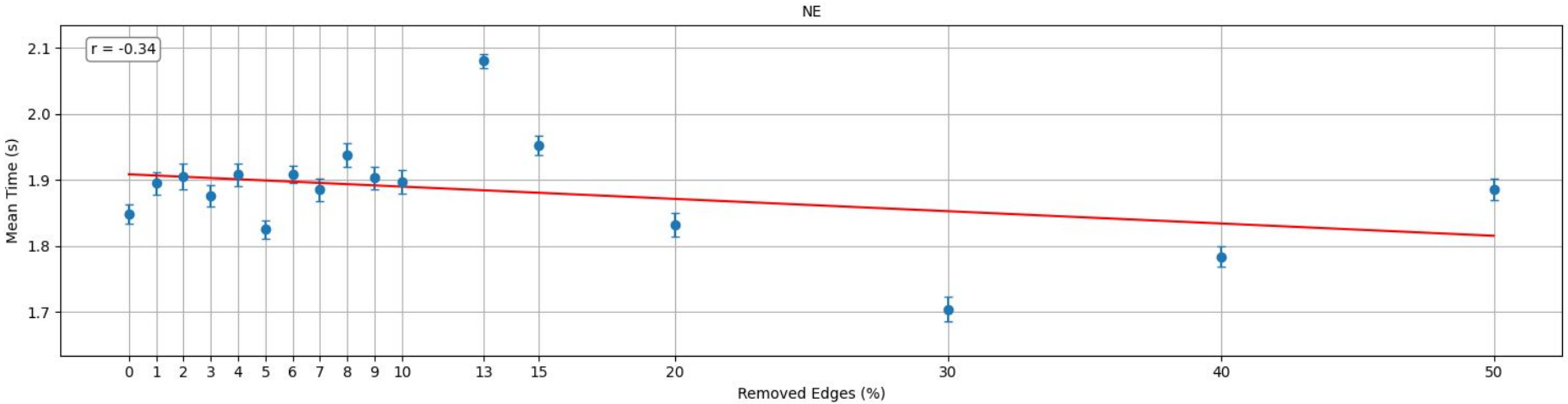
Эксперимент 1. Проверка результатов





- Корреляция действительно есть на графах BAY и LKS
- А что по остальным графам?

Эксперимент 1. Проверка результатов



- Корреляция на графе NE всё ещё недостаточна для опровержения гипотезы
- Почему на одних графах корреляция видна сразу, а на других нет?

GraphBlas
Анализ результатов

(для первого эксперимента)

Эксперимент 1. Анализ результатов

Вопросы:

1. Почему алгоритм на GraphBlas начинает работать быстрее?
2. Какие свойства графов влияют на корреляцию?

Как на них можно ответить:

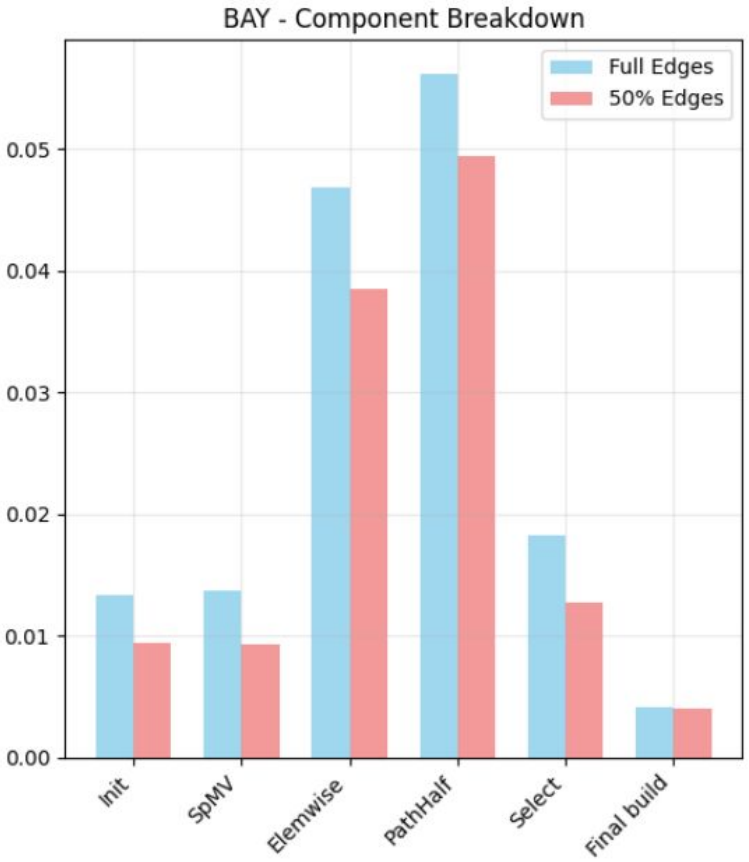
1. Попробуем выделить в алгоритме самые долгие по времени обработки графа сегменты и посмотреть, что изменится на 50% удаленных рёбер
2. Попробуем предположить на основе деталей работы алгоритма, какие свойства могут быть ответственны за результат и измерим их у каждого графа в датасете. Возможно сможем понять, в чём именно отличие.

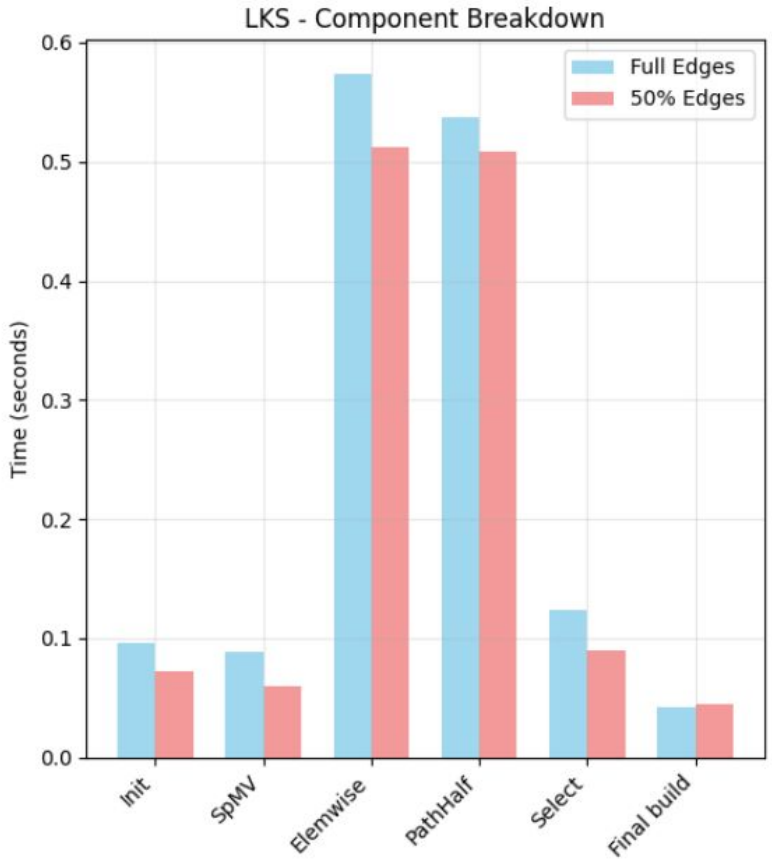
Эксперимент 1. Анализ результатов

Сегменты для профилирования:

1. Инициализация (init).
2. Нахождение вектора, содержащего минимальные рёбра инцидентные каждой вершине (перемножение вектора и матрицы смежности, SpMV).
3. Нахождение вектора, содержащего минимальные рёбра для каждой компоненты связности (elemwise).
4. Обновление вектора с компонентами связности и матрицы смежности (PathHalf и Select).
5. Завершение работы.

Эксперимент 1. Анализ результатов

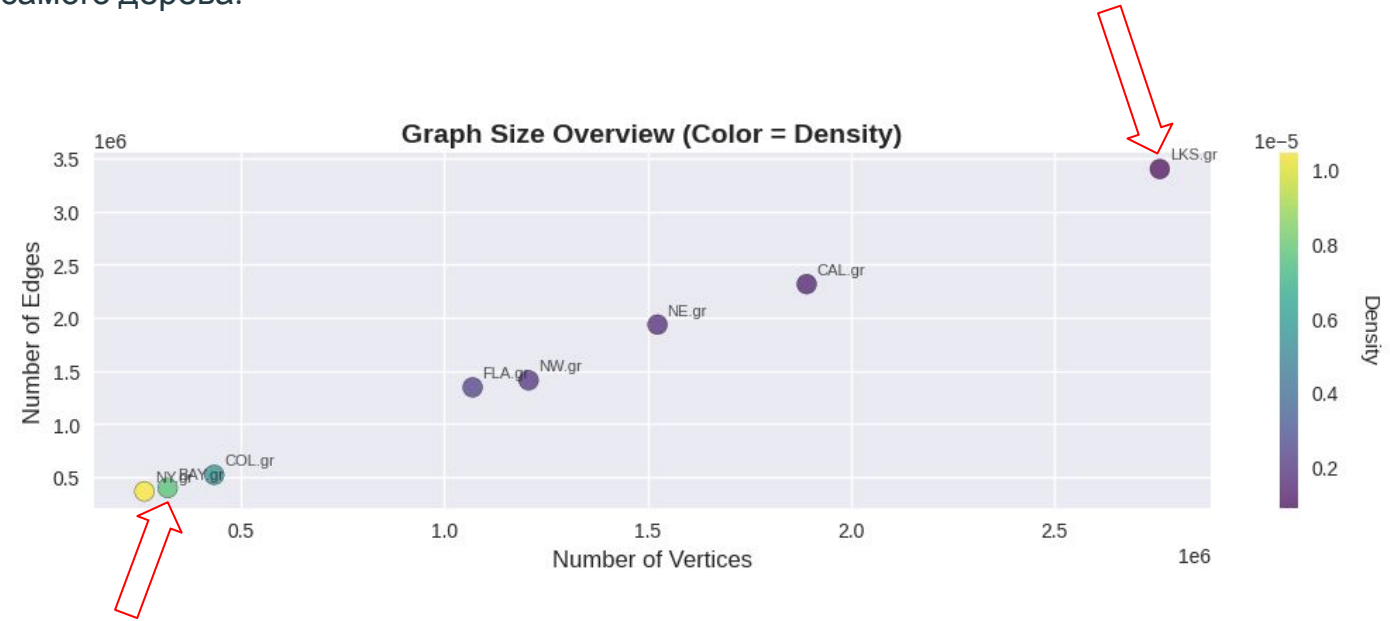




Эксперимент 1. Анализ результатов

Какие свойства графов могут влиять на результат:

- Изначальная плотность графа и размер. Возможно в некоторых графах слишком много “избыточных” рёбер, которые алгоритм обрабатывает, но не использует для построения самого дерева.



PySpark

Проблемы с датасетом

Проблема: PySpark слишком долго работает на графах из датасета DIMACS.

Попытки решения:

1. Оставить прежний датасет, но уменьшать графы ещё сильнее.
2. Сгенерировать похожие графы.
3. Найти новый датасет с такой же спецификой, но подходящим размером графов.
4. ???

Проблемы с датасетом

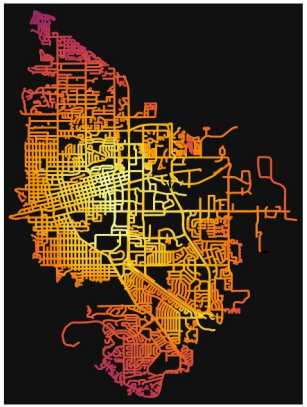
Проблема: PySpark либо падает с ошибкой, либо слишком долго работает на графах из датасета DIMACS.

Попытки решения:

1. Оставить прежний датасет, но уменьшать графы ещё сильнее.
2. Сгенерировать похожие графы.
3. Найти новый датасет с такой же спецификой, но подходящим размером графов.
4. Использовать OSMnx.

Berkeley, California, USA

Cambridge, Massachusetts, USA



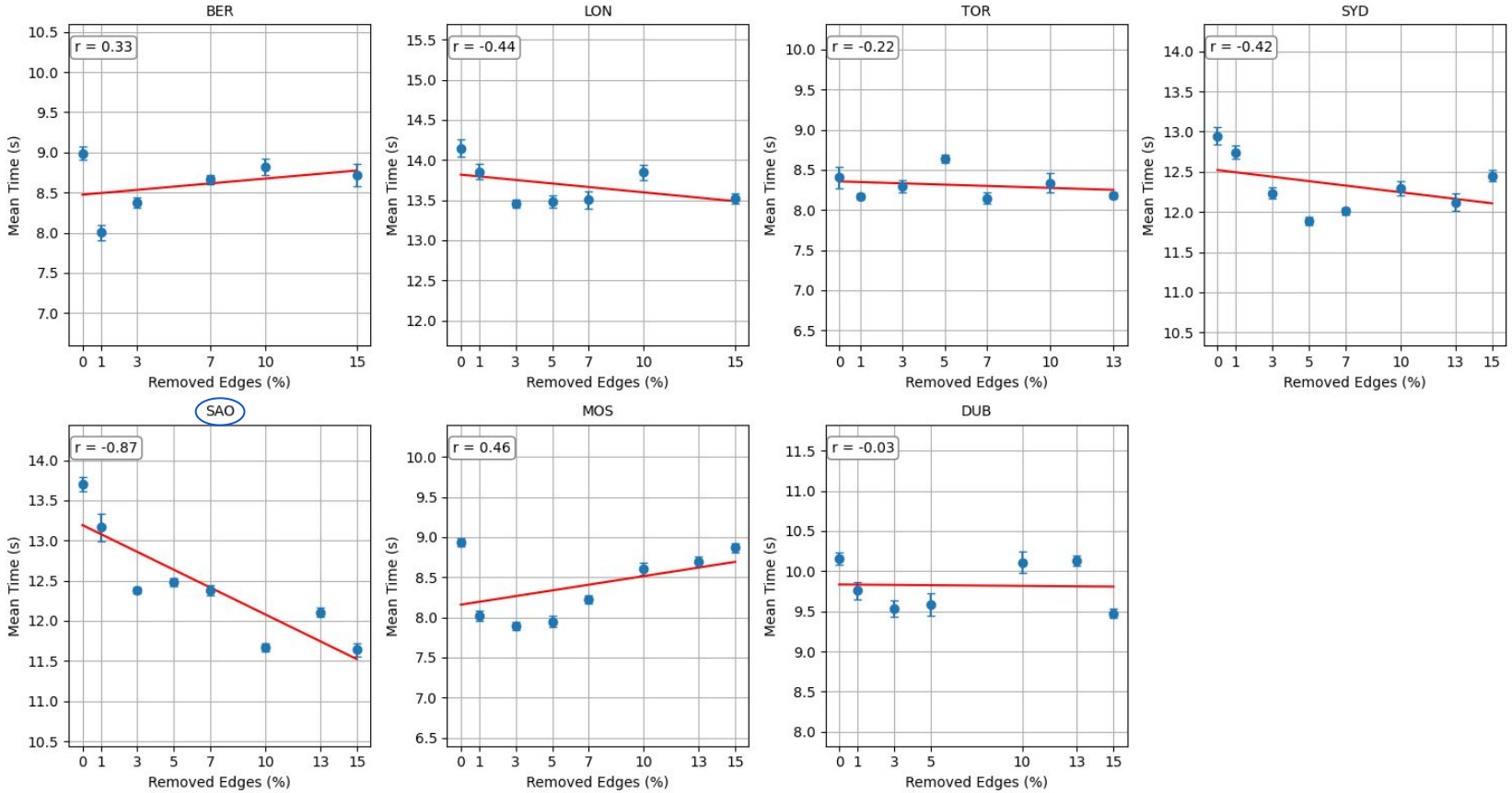
Boulder, Colorado, USA

Датасет для PySpark

Description	Nodes	Edges
London, England, UK	129,453	327,198
São Paulo, Brazil	121,896	355,542
Sydney, Australia	117,404	315,076
Dubai	60,603	174,918
Berlin, Germany	28,277	84,620
Toronto, Canada	27,390	80,396
Moscow, Russia	27,067	77,686

- Веса брались как количество времени (секунд), необходимого для преодоления дороги.

Эксперимент 1. Результаты для PySpark

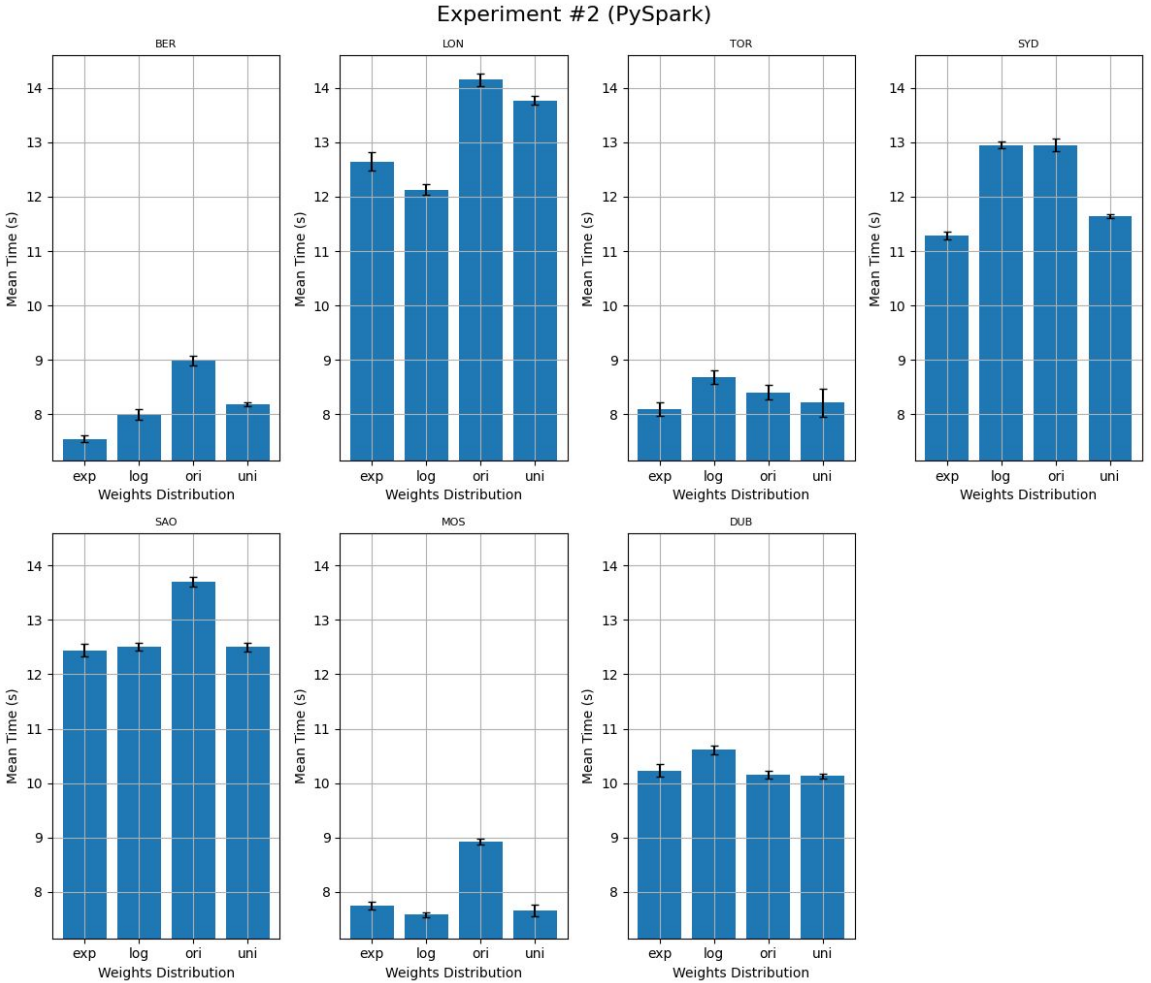


Эксперимент 1. Результаты для PySpark

#	Graph Name	R-value	P-value	Reject H0
1	BER	0.328922	0.524411	❌False
2	LON	-0.441955	0.320789	❌False
3	TOR	-0.223925	0.629312	❌False
4	SYD	-0.420699	0.299321	❌False
5	SAO	-0.872076	0.004744	✅ True
6	MOS	0.462876	0.248105	❌False
7	DUB	-0.034333	0.941749	❌False

Вывод: только на одном графе из датасета получилось обнаружить достаточно сильную корреляцию.

Эксперимент 2. Результаты для PySpark



Эксперимент 2. Результаты для PySpark

Distribution	N	Mean	Values
exp	7	10.000	[7.548, 12.643, 8.096, 11.289, 12.444, 7.75, 10.232]
lognorm	7	10.352	[8.003, 12.132, 8.684, 12.949, 12.501, 7.581, 10.614]
original	7	11.039	[8.987, 14.147, 8.404, 12.947, 13.703, 8.929, 10.159]
uni	7	10.300	[8.192, 13.77, 8.217, 11.635, 12.501, 7.656, 10.131]

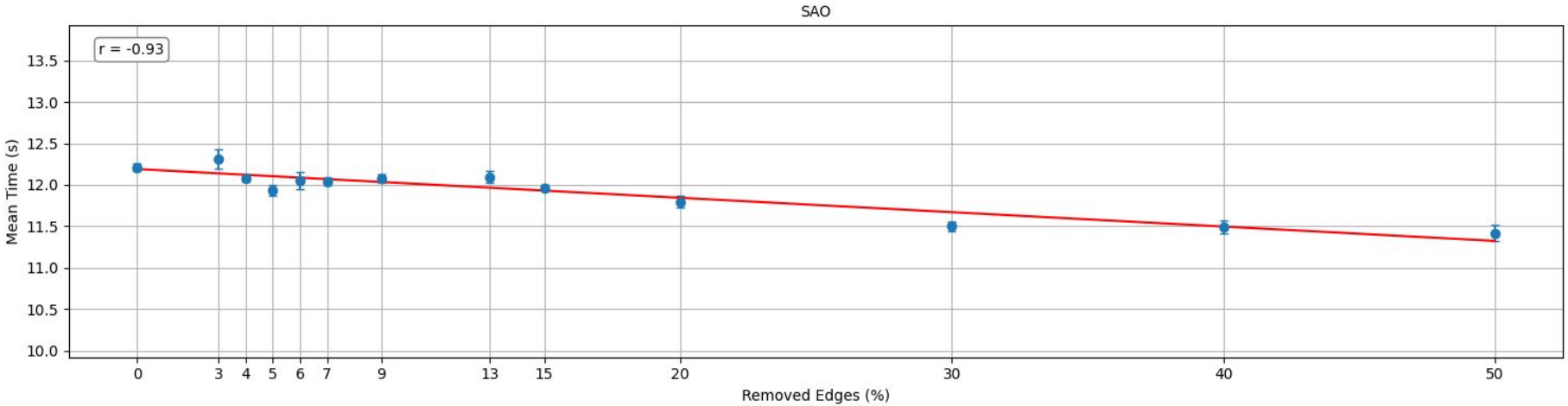
❌ P-value: 8.6256e-01 → гипотеза не опровергнута

Вывод: Не получилось опровергнуть гипотезу, можно сказать, что для PySpark на этом датасете нет большой разницы, какое распределение будет у графа.

PySpark
Проверка результатов

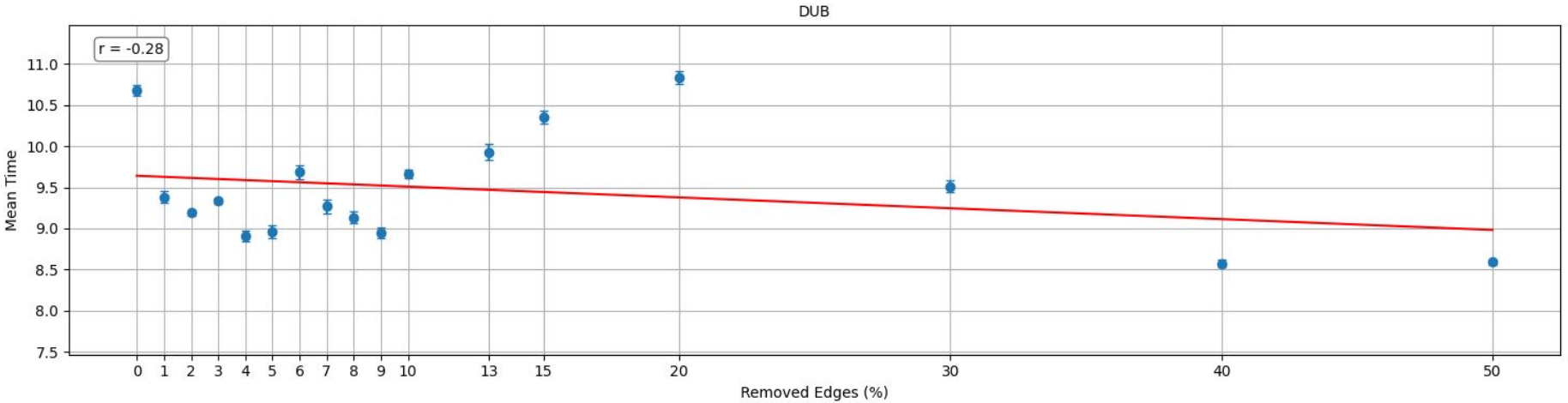
(для первого эксперимента)

Эксперимент 1. Проверка результатов



- Корреляция действительно есть на графе SAO

Эксперимент 1. Проверка результатов



- Корреляция на графе DUB всё ещё недостаточна для опровержения гипотезы
- Почему на одних графах корреляция видна сразу, а на других нет?

PySpark
Анализ результатов

(для первого эксперимента)

Эксперимент 1. Анализ результатов