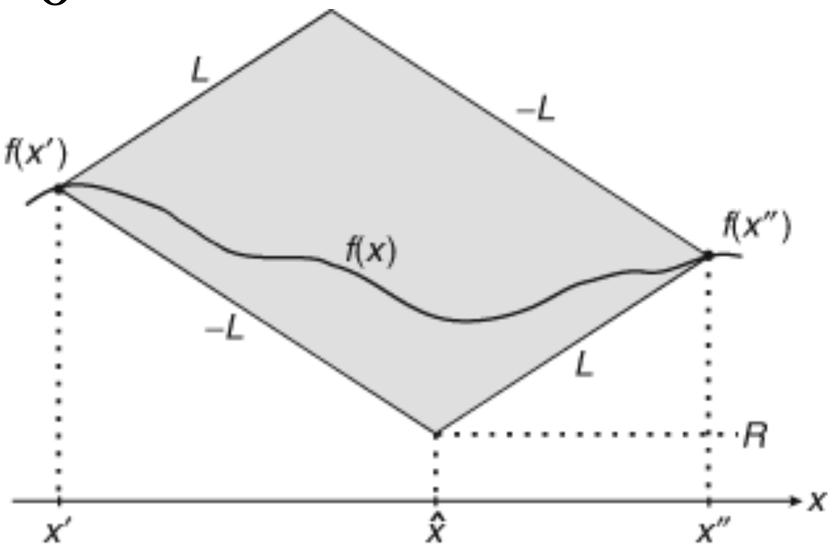
Разработка высокопроизводительных методов глобальной оптимизации

Выполнил: Иванов Семен, 182 группа
Руководитель: Посыпкин Михаил Анатольевич

Рассматриваемая задача

Липшицева глобальная оптимизация

- Задача глобальной оптимизации функции $f : \mathbb{R}^N \to \mathbb{R}$:
  $f^* = f(x^*) = \min\limits_{x \in D} f(x)$, где $D = \left\{ x \in \mathbb{R}^N : \; a_j \le x_j \le b_j, 1 \le j \le N \right\}$
- Условие Липшица:
  $\left| f(x') - f(x'') \right| \le L \; \left\| x' - x'' \right\|, \forall x', x'' \in D, L > 0$
- black-box функция
- Вычислительно трудозатратная функция



Практическая применимость

Диагональный безызбыточный подход [1]

- Особое разбиения, позволяющее переиспользовать значения несоклько раз
- Сравнение с другими подходами по числу вызовов функции (последние два – диагональный подход):

n	ε	Class	DIRECT	BISECTION	SMOOTHD	MULTK
2	10^{-4}	Simple	198.89	432.75	151.11	74.75
2	10^{-4}	Hard	1,063.78	707.03	404.79	162.11
3	10^{-6}	Simple	1,117.70	3,369.76	1,011.00	783.49
3	10^{-6}	Hard	$\gg$6,322.65	4,934.85	1,756.18	618.32
4	10^{-6}	Simple	$\gg$11,282.89	4,061.15	4,598.97	3,512.92
4	10^{-6}	Hard	$\gg$29,540.12	>59,581.96	7,276.23	6,127.09
5	10^{-7}	Simple	>6,956.97	>40,772.45	4,281.42	3,583.20
5	10^{-7}	Hard	$\gg$72,221.24	>50,223.86	33,246.18	19,688.68

План работы

1. Освоить 1-d случай
2. Изучить диагональные схемы и безызбыточную стратегию разбиения
3. Эффективная последовательная реализация
4. Распараллеливание
5. Исследование эффективности полученной реализации

Основная задача: проверить, насколько успешно диагональный безызбыточный подход может быть распараллелен

Диагональный подход

Общее описание

- best first search
- подзадача описывается двумя точками на концах главной диагонали
- выбирается по наибольшему критерию для $R(\hat{L}, \text{iter}, a_i, b_i, f(a_i), f(b_i))$

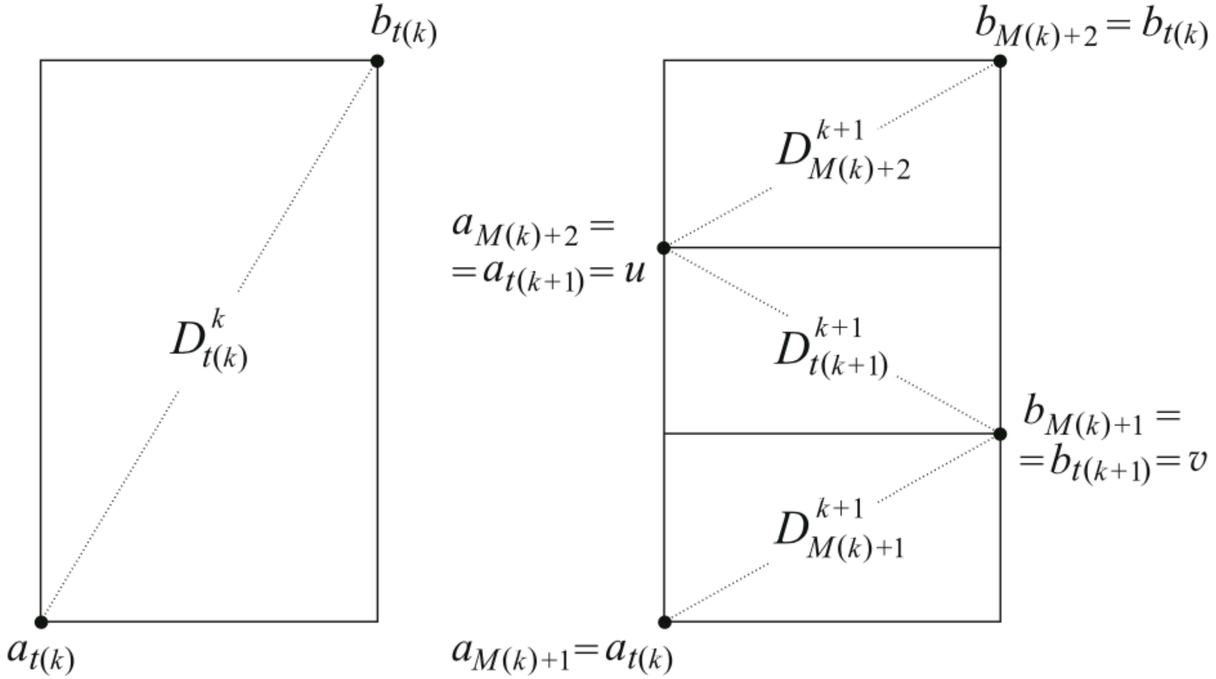
```
while not Tasks.StoppingCriterion():
    task = Tasks.PopBestTask()
    taskDivided = DivideTask(task)
    for task in taskDivided:
        Tasks.PushTask(task)
    Tasks.Recalc()
```

- Константу Липшица можно оценивать снизу как $\max_i \left(\frac{|f(a_i) - f(b_i)|}{\| a_i - b_i \|} \right) \leq \hat{L}$

Диагональный подход

Безызбыточная стратегия

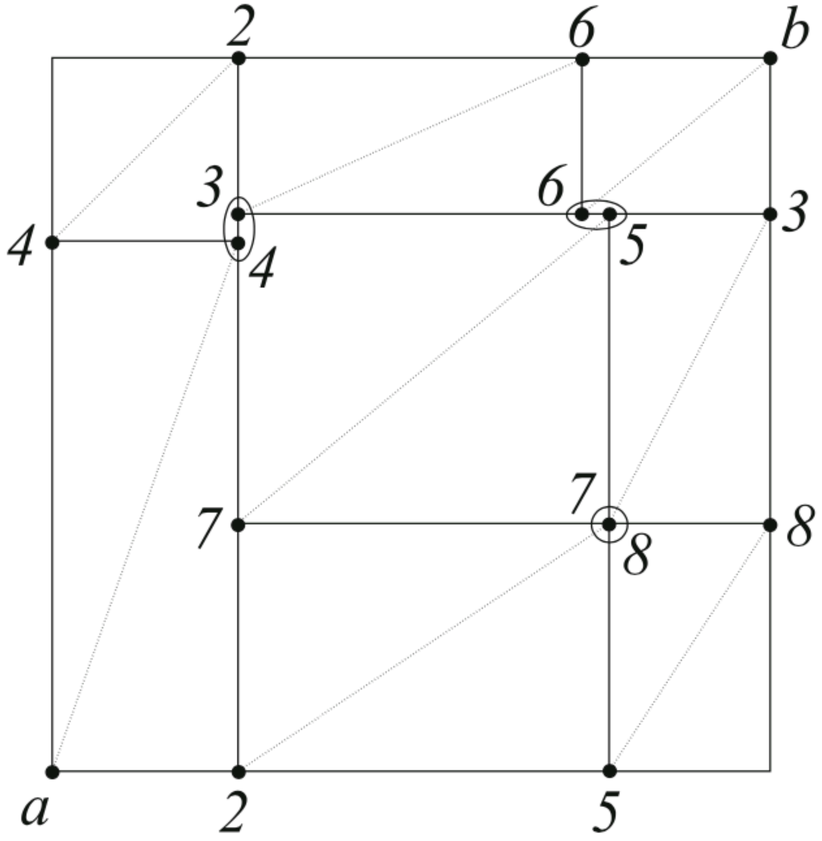
- Разбиваем гиперкуб на 3 равные части плоскостями, перпендикулярными самой длинной стороне

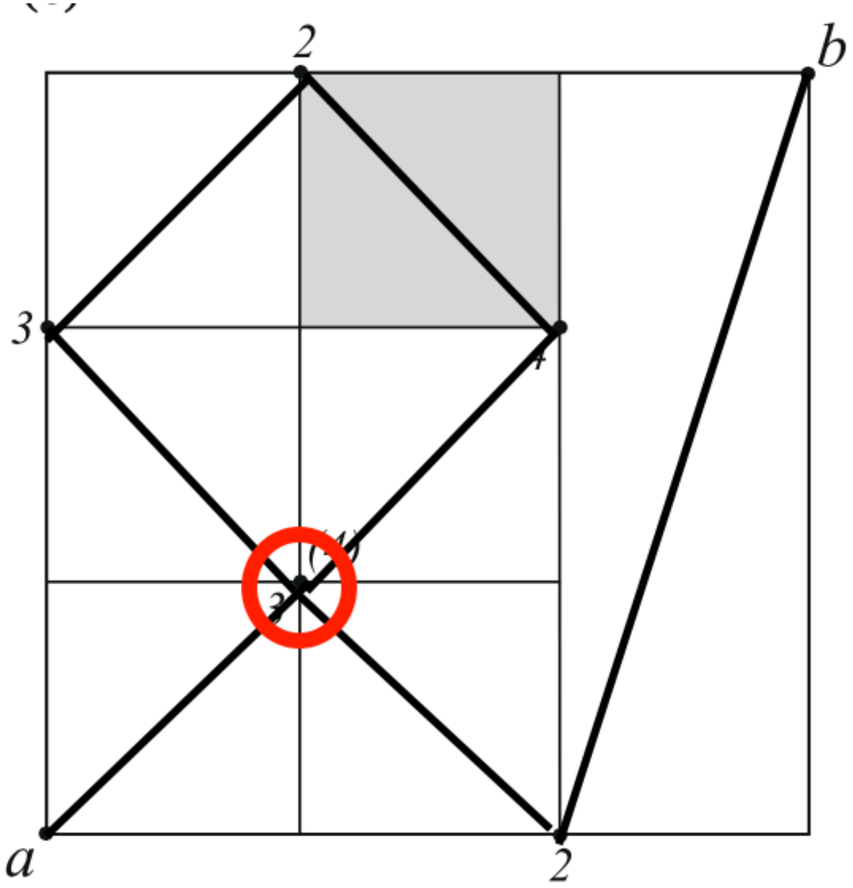


Диагональный подход

Безызбыточная стратегия

- Стратегия разбиения получается безызбыточной и позволяет переиспользовать значения (до 2^N раз)





Диагональный подход

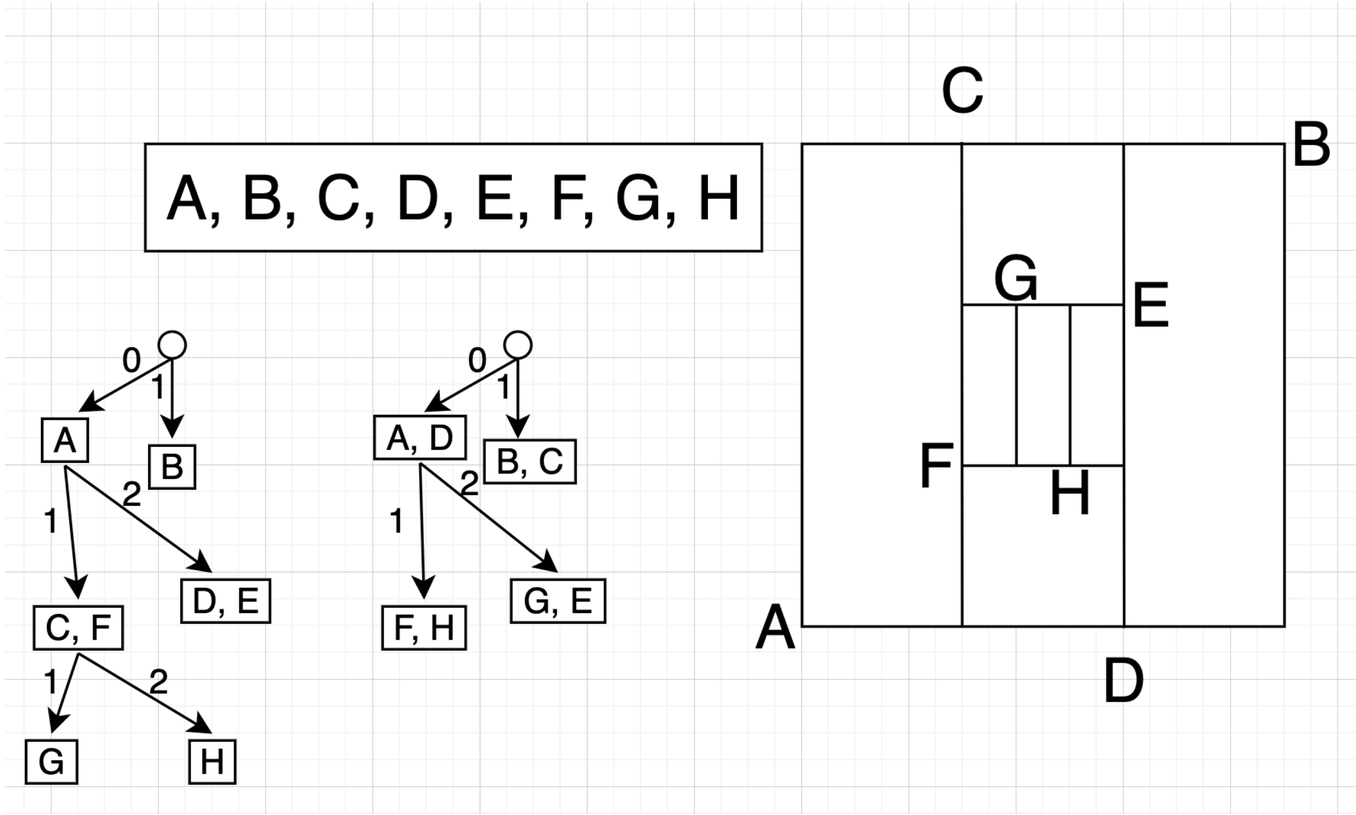
Переиспользование точек

- Предлагается кодировать точки и подзадачи N строчками-числами из алфавита $\{0,1,2\}$ и построить соответствие из множества подзадач в множество точек
- Строится база точек, к которой происходит обращение при разбиении
- Предложенный алгоритм был немного уточнен с точки зрения реализации

Эффективная последовательная реализация

- Предлагается организовать точки в структуру из N деревьев, где путь от корня до вершины с точкой определял бы код по соответствующей размерности

- Также предлагается отказаться от кодирования подзадач



Многопоточная реализация

Основные проблемы

- Так как это best first search, возникает неясность, как адаптивно строить расписание обработки подзадач
- Необходимо обеспечить многопоточно-эффективную схему для доступа к базе точек
- Так же необходимо сохранить безызбыточность

Параллельная реализация

База точек

- Вершины деревьев, в которых хранятся точки, предлагается хранить в виде листов
- Брать lock только на конец листа в вершине при добавлении
- Поиск по базе получается lock-free
- Существует проблема с одновременным поиском и добавлением той же точки, она незначительна и никак не сказывается на эффективности

Параллельная реализация

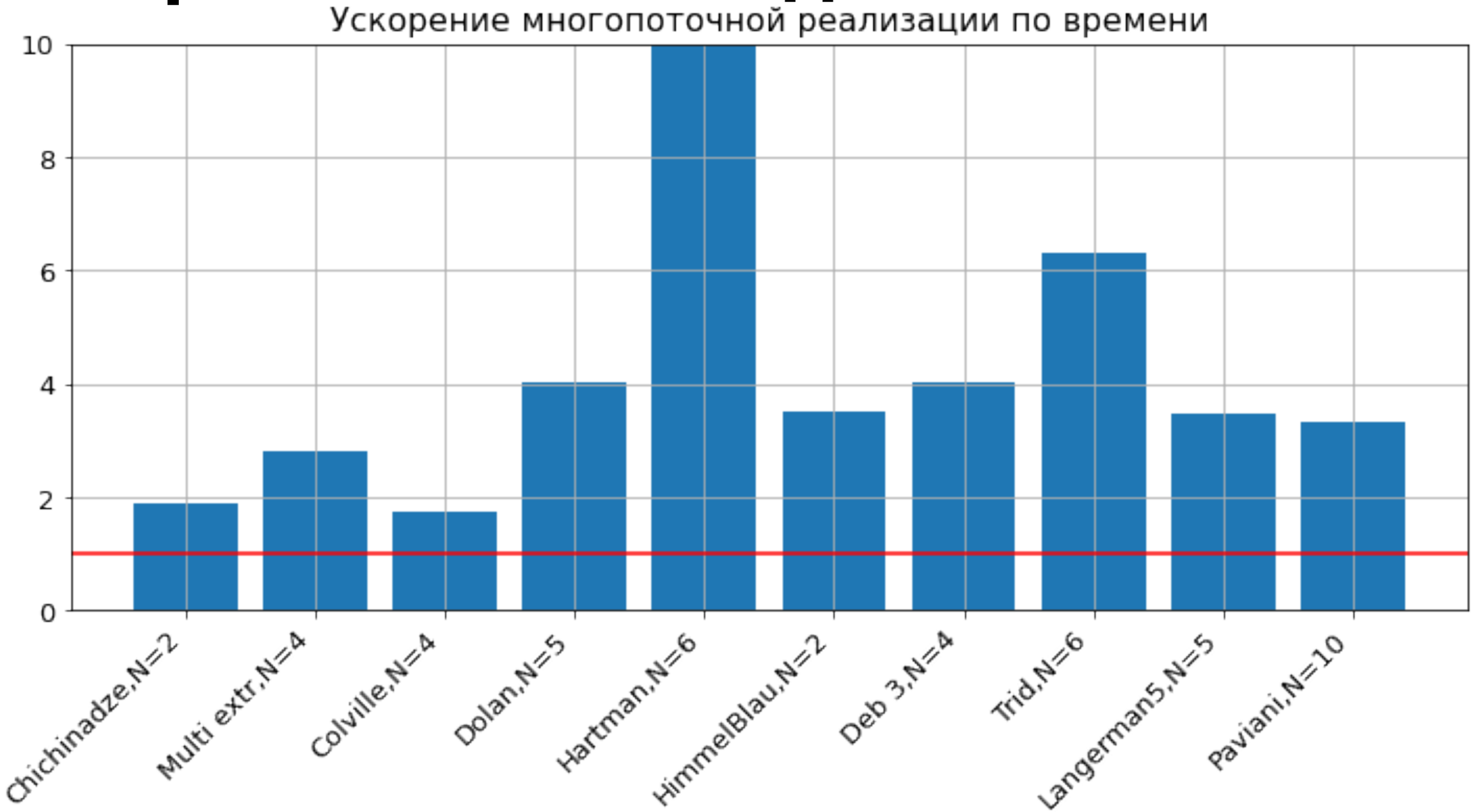
Построение расписания

- Приоритетная очередь из Intel Threading Building Blocks [5]
- Узкое место при большом числе потоков
- Поддерживать несколько очередей и обращаться к случайной
- Необходимо регулярно пересчитывать приоритеты
- Предлагается пересчитывать при сильном изменении оценки константы Липшица с прошлого пересчета

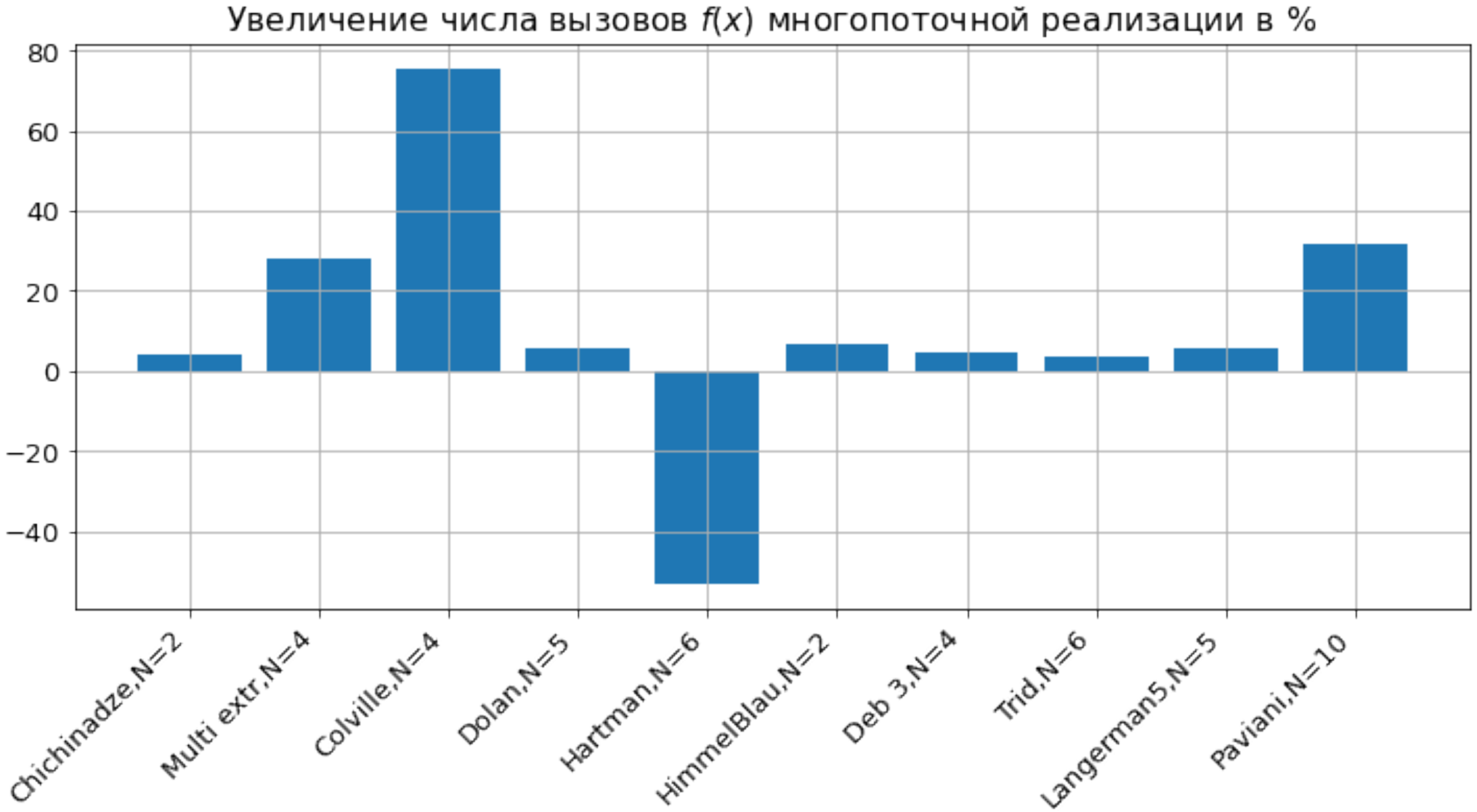
Эксперименты

- Вычислительно трудозатратные функции обычно имеют прикладную и физическую природу
- Были проведены эксперименты на достаточно популярных сложных Липшицевых функциях [4]
- Исследовалось ускорение времени работы, чтобы оценить эффективность распараллеливания
- И число увеличение числа вызовов функции, чтобы оценить сохранение безызбыточности
- Эксперименты были проведены на 5 потоках
- Для каждого теста была выбрана необходимая точность, до которой следует продолжать исполнение

Эксперименты и выводы



Эксперименты и выводы



Итог

- Был рассмотрен диагональный безызбыточный подход Липшицевой оптимизации
- Предложена эффективная последовательная схема
- Разработан подход для многопоточного исполнения
- Проведено сравнение многопоточного и последовательного исполнения

Список литературы

[1] Д.Е.Квасов Я.Д.Сергеев. Диагональные метода глобальной оптимизации. Москва, Нижний Новгород: ФИЗМАТЛИТ, 2008.
isbn : 9785922110327
[2] Dmitri E. Kvasov Yaroslav D. Sergeyev. Deterministic global optimization: an introduction to the diagonal approach. New York, NY: Springer, 2017.
isbn : 9781493971978
[3] М.А. Посыпкин А.Ю. Горчаков. СРАВНЕНИЕ ВАРИАНТОВ МНОГОПОТОЧНОЙ РЕАЛИЗАЦИИ МЕТОДА ВЕТВЕЙ И ГРАНИЦ ДЛЯ МНОГОЯДЕРНЫХ СИСТЕМ. г. Москва, Россия: Федеральный исследовательский центр “Информатика и управление” Российской академии наук, 2018
[4] Xin-She Yang Momin Jamil. A literature survey of benchmark functions for global optimization problems. Int. Journal of Mathematical Modelling и Numerical Optimisation, 2013. url
[5] Intel Threading Building Blocks. url
[6] cppreference. url
[7] Реализация описанных подходов. url