Онлайн оптимизация и задача максимального многопродуктового потока

Улучшение и адаптация алгоритма Гарга-Кёнемана

Никита Артюх

Научный руководитель: Александр Рогозин

Московский физико-технический институт

17 мая 2025 г.

Введение и Мотивация

- **Задача максимального многопродуктового потока (Max-Multicommodity Flow, MCFP):**
  - Одновременная маршрутизация нескольких потоков (товаров, данных) между различными парами источник-сток в сети.
  - Цель: максимизировать суммарный поток с учетом пропускных способностей ребер.
- **Актуальность:**
  - Проектирование телекоммуникационных сетей.
  - Логистика и транспортные задачи.
  - Распределение ресурсов в вычислительных системах.
- **Связь с онлайн-оптимизацией:**
  - MCFP может быть решена итеративно, где на каждом шаге выбирается путь и объем потока.
  - Алгоритмы онлайн-оптимизации (например, Multiplicative Weights) предоставляют теоретическую основу для таких итеративных методов.

Постановка Задачи: Max-Multicommodity Flow

Задача Max-Multicommodity Flow

$$\max \sum_{p \in \mathcal{P}} x(p)$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_e} x(p) \leq b_e \quad \forall e \in E$$

$$x(p) \geq 0 \quad \forall p \in \mathcal{P}$$

- $\mathcal{P} = \bigcup_{k \in \mathcal{K}} \mathcal{P}_{s_k, t_k}$: множество всех допустимых путей для всех k корреспонденций (пар источник s_k - сток t_k).
- $\mathcal{P}_e = \{p \in \mathcal{P} \mid e \in p\}$: множество путей, проходящих по ребру e.
- $x(p)$: объем потока по пути p.
- b_e: пропускная способность ребра e.

Обзор Подходов: Онлайн Обучение и Multiplicative Weights

Multiplicative Weights (MW) [Bub11, Wil19]

Идея: итеративно обновлять веса “экспертов” (в нашем случае, ребер графа) на основе их “производительности”.

Алгоритм 1 Multiplicative Weights (общая идея)

1: Инициализировать веса $w_1(i) = 1$ для всех экспертов $i = 1, \ldots, N$.
2: **for** $t = 1, \ldots, T$ **do**
3: Выбрать эксперта i с вероятностью, пропорциональной $w_t(i)$.
4: Получить “потери” (или “выигрыши”) $v_t(i)$ для каждого эксперта.
5: Обновить веса: $w_{t+1}(j) = w_t(j) \cdot \mathrm{factor}(v_t(j))$ (напр., $w_t(j)(1 - \varepsilon v_t(j))$ для потерь).
6: **end for**

MW гарантирует, что средние потери алгоритма не сильно хуже потерь лучшего эксперта “в ретроспективе”.

Обзор Подходов: Алгоритм Гарга-Кёнемана (ГК)

Алгоритм Гарга-Кёнемана [GK07, Wil19]

Применяет идеи, схожие с MW, для решения задачи MCFP.

Алгоритм 2 Алгоритм Гарга-Кёнемана для MCFP

1: $x(p) = 0 \quad \forall p \in \mathcal{P}$; $f(e) = 0$ (текущий поток по ребру), $w(e) = 1$ (вес ребра) $\quad \forall e \in E$.
2: **while** условие не выполнено (напр., пока есть недогруженные ребра) **do**
3: Найти кратчайший путь P по всем корреспонденциям по весам $w(e)/b(e)$.
4: Найти bottleneck $u = \min_{e \in P} b(e)$.
5: $x(P) \leftarrow x(P) + u$.
6: $f(e) \leftarrow f(e) + u \quad \forall e \in P$.
7: $w(e) \leftarrow w(e) \cdot (1 + \varepsilon \frac{u}{b(e)}) \quad \forall e \in P$
8: **end while**

Выход: $x / \max_e(f(e)/b(e))$.

Цели Исследования и Подход

1. **Улучшить алгоритм Гарга-Кёнемана (ГК):**
   - ✓ Реализация базового алгоритма ГК.
   - ✓ Параллелизация вычислений по корреспонденциям.
   - ✓ Адаптивный подбор шага с использованием AdaGrad.
   - ✓ Теоретическое исследование MCMC для выбора путей.

Технологический стек

- Язык программирования: **Rust**(1.87.0) для производительности и безопасности памяти и **Python**(3.12.4) для анализа и визуализации.
- Среда: Intel(R) Core(TM) i9-7980XE CPU @ 2.60GHz, 64GB RAM.

Улучшение ГК: Параллелизация по Корреспонденциям

Идея

На каждой итерации алгоритма Гарга-Кёнемана поиск кратчайшего пути для *каждой* из K корреспонденций (пар источник-сток). При этом корреспонденции могут обрабатываться параллельно.

- **Реализация:**
  - Параллельно считаем кратчайший путь для каждой корреспонденции, используя текущие веса $w(e)/b(e)$.
  - Находим минимум по всем путям для каждой корреспонденции.
  - Продолжаем как в стандартном ГК.

  По факту делаем $\mathrm{Map} - \mathrm{Reduce}$ по корреспонденциям.
- **Ожидаемый эффект:**
  - Ускорение за счет параллельной обработки.
  - Малое количество синхронизаций не должно сильно влиять на производительность даже в маленьких графах.

Улучшение ГК: Параллелизация по Корреспонденциям

Идея

На каждой итерации алгоритма Гарга-Кёнемана поиск кратчайшего пути для *каждой* из K корреспонденций (пар источник-сток). При этом корреспонденции могут обрабатываться параллельно.

- **Реализация:**
  - Параллельно считаем кратчайший путь для каждой корреспонденции, используя текущие веса $w(e)/b(e)$.
  - Находим минимум по всем путям для каждой корреспонденции.
  - Продолжаем как в стандартном ГК.

  По факту делаем $\mathrm{Map} - \mathrm{Reduce}$ по корреспонденциям.
- **Ожидаемый эффект:**
  - Ускорение за счет параллельной обработки.
  - Малое количество синхронизаций не должно сильно влиять на производительность даже в маленьких графах.

Адаптация ГК: Использование AdaGrad [Ora23]

Мотивация

В стандартном алгоритме ГК параметр ε (шаг или скорость обучения) обычно фиксирован. Адаптивный подбор шага может улучшить сходимость.

- **Идея AdaGrad (Adaptive Gradient):**
  - Масштабировать скорость обучения для каждого параметра (в нашем случае, для каждого ребра при обновлении его веса) обратно пропорционально сумме квадратов предыдущих градиентов.
  - Для ребер, которые часто обновляются (сильно загружены), шаг будет уменьшаться быстрее.
- **Применение в ГК (концептуально):**
  - При обновлении веса ребра $w(e) \leftarrow w(e) \cdot (1 + \eta_e \frac{\Delta f(e)}{b(e)})$:
  - Адаптивный шаг η_e для ребра e можно взять как $\eta_e = \frac{\eta}{\sqrt{G_e + \epsilon}}$, где G_e - сумма квадратов градиентов для ребра e.

Адаптация ГК: Использование AdaGrad [Ora23]

Мотивация

В стандартном алгоритме ГК параметр ε (шаг или скорость обучения) обычно фиксирован. Адаптивный подбор шага может улучшить сходимость.

- **Идея AdaGrad (Adaptive Gradient):**
  - Масштабировать скорость обучения для каждого параметра (в нашем случае, для каждого ребра при обновлении его веса) обратно пропорционально сумме квадратов предыдущих градиентов.
  - Для ребер, которые часто обновляются (сильно загружены), шаг будет уменьшаться быстрее.
- **Применение в ГК (концептуально):**
  - При обновлении веса ребра $w(e) \leftarrow w(e) \cdot (1 + \eta_e \frac{\Delta f(e)}{b(e)})$:
  - Адаптивный шаг η_e для ребра e можно взять как $\eta_e = \frac{\eta}{\sqrt{G_e + \epsilon}}$, где G_e - сумма квадратов градиентов для ребра e.

Экспериментальные Результаты

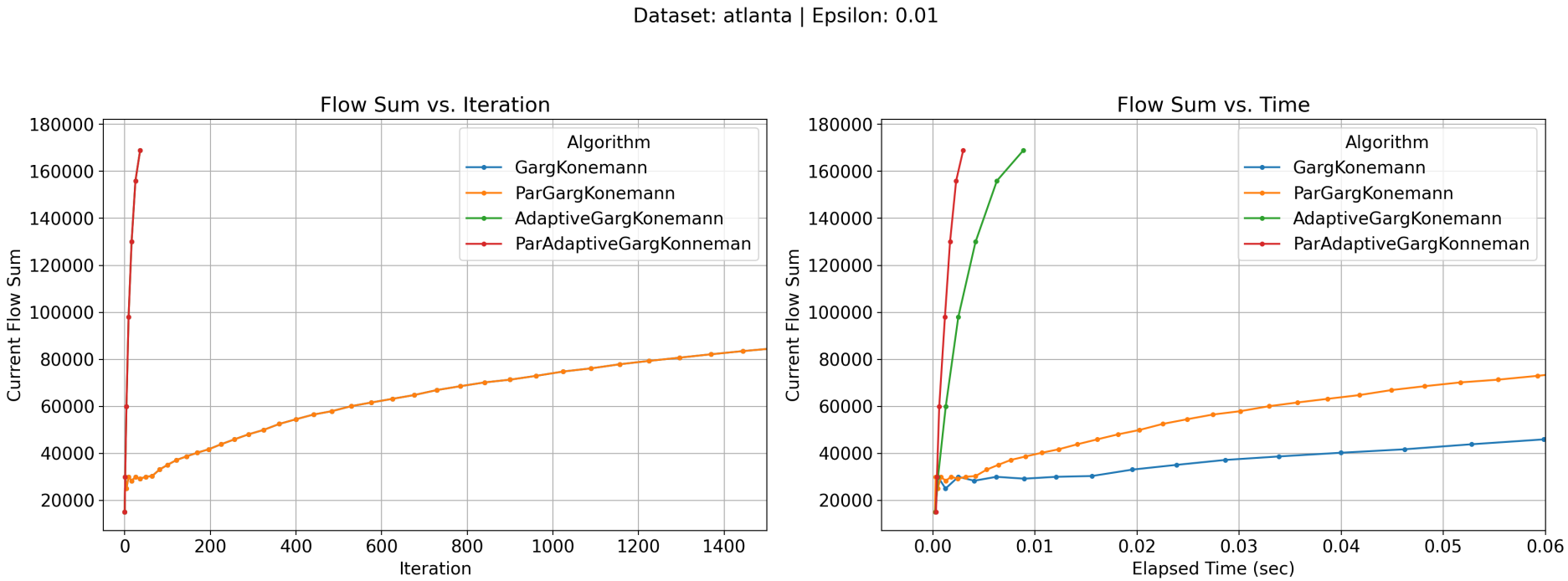

Рис.: Работа алгоритмов на графе Atlanta из sndlib [OPTW07, OPTW10].

Примечание: Результаты для остальных графов можно найти в репозитории на GitHub.

Теоретическое Направление: MCMC для Сэмплирования Путей

Идея

Вместо детерминированного поиска кратчайшего пути, можно сэмплировать пути из распределения, где вероятность пути зависит от его стоимости (длины по текущим весам).

- **Связь с Exponential Weights / Распределением Больцмана:**
  - В онлайн обучении, выбор действия (пути) часто производится согласно распределению $\mathrm{softmax}$/Больцмана: $P(\text{путь } i) \propto \exp(-\eta w(i))$.

- **MCMC (Markov Chain Monte Carlo) для сэмплирования:**
  - Если пространство путей слишком велико, можно использовать MCMC (например, на основе динамики Ланжевена или модификаций на графах [MP15]) для генерации выборки путей из нужного распределения.
  - На каждой итерации ГК:
    1. Обновить веса ребер $w(e)$.
    2. Используя MCMC, сэмплировать путь P из распределения, зависящего от $w(e)/b(e)$.
    3. Продолжить как в стандартном ГК.
- Потенциально полезно для очень больших графов или при сложных ограничениях на пути.

Основные Результаты и Заключение

Достигнутые результаты в течение семестра:

- **Реализован базовый алгоритм Гарга-Кёнемана** для MCFP на языке Rust.
- **Разработан и реализован вариант ГК с параллелизацией** по корреспонденциям.
- **Разработан и реализован адаптивный вариант ГК** с использованием идей, аналогичных AdaGrad, для динамического подбора шага обновления весов ребер.
- Проведено теоретическое исследование возможности применения MCMC для сэмплирования путей.
- Подготовлена основа для проведения сравнительных бенчмарков различных версий алгоритма.

Значение работы:

- Полученные реализации и модификации алгоритма ГК могут привести к более эффективным решениям задачи MCFP.
- Адаптивные и параллельные подходы важны для масштабируемости на больших и сложных сетях.
- Работа закладывает основу для дальнейших исследований в области применения онлайн-методов к комбинаторным задачам оптимизации.

Дальнейшая Работа

- **Углубленное исследование MCMC для сэмплирования путей:**
  - Реализация и тестирование MCMC-подхода.
  - Сравнение с детерминированными методами выбора путей.
- **Применение к другим потоковым задачам:**
  - Адаптация алгоритма Гарга-Кёнемана для задачи min-cost multicommodity flow.
- **Исследование других онлайн-алгоритмов:**
  - Рассмотрение более современных подходов из области онлайн-обучения (например, [Ora23]) и их применимости.
- **Оптимизация реализации на Rust:**
  - Профилирование и улучшение производительности кода.

Список литературы I

- Sébastien Bubeck, Introduction to online optimization, **Link**, 2011.
- Naveen Garg and Jochen Könemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, SIAM Journal on Computing **37** (2007), no. 2, 630–652, doi:10.1137/S0097539704446232.
- Sandro Montanari and Paolo Penna, On sampling simple paths in planar graphs according to their lengths, Mathematical Foundations of Computer Science 2015 (Berlin, Heidelberg) (Giuseppe F. Italiano, Giovanni Pighizzini, and Donald T. Sannella, eds.), Springer Berlin Heidelberg, 2015, doi:10.1007/978-3-662-48054-0_41, pp. 493–504.

Список литературы II

- S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, SNDlib 1.0–Survivable Network Design Library, Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, April 2007, http://sndlib.zib.de, extended version accepted in Networks, 2009. (English).
- _____, SNDlib 1.0–Survivable Network Design Library, Networks **55** (2010), no. 3, 276–286 (English).
- Francesco Orabona, A modern introduction to online learning, 2023, doi:10.48550/arXiv.1912.13213.
- David P. Williamson, Network flow algorithms, Cambridge University Press, 2019, doi:10.1017/9781316888568.