Онлайн оптимизация на симлексе

Применение онлайн оптимизации к офлайн задачам

Никита Артюх

Научный руководитель: Александр Рогозин

Московский физико-технический институт

11 марта 2025 г.

Основная задача

Задача Max-Concurent-Flow

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

$$\text{s.t.} \sum_{p \in \mathcal{P}_e} x(p) \le b_e$$

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

$$\mathcal{P} = \bigcup_{s\text{-}t \in \mathcal{K}} \mathcal{P}_{s\text{-}t}$$ пути по парам s-t кореспонденций

$$\mathcal{P}_e = \{p \in \mathcal{P} \mid e \in p\}$$ пути проходящие по ребру e

Основная задача

Задача Max-Concurent-Flow

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

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

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

$$\mathcal{P} = \bigcup_{s\text{-}t \in \mathcal{K}} \mathcal{P}_{s\text{-}t}$$ пути по парам s-t кореспонденций

$$\mathcal{P}_e = \{p \in \mathcal{P} \mid e \in p\}$$ пути проходящие по ребру e

Основная задача

Задача Max-Concurent-Flow

$$\max \sum_{p \in \mathcal{P}} x(p)$$
$$\text{s.t.} \sum_{p \in \mathcal{P}_e} x(p) \leq b_e$$
$$x(p) \geq 0 \quad \forall p \in \mathcal{P}$$

$$\mathcal{P} = \bigcup_{s\text{-}t \in \mathcal{K}} \mathcal{P}_{s\text{-}t}$$ пути по парам s-t кореспонденций

$$\mathcal{P}_e = \{p \in \mathcal{P} \mid e \in p\}$$ пути проходящие по ребру e

Multiplicative[Wil19]/Exponential weights[Bub11]

Алгоритм 1 Multiplicative Weights

$w_1(i) = 1$ for $i = 1, \dots, N$
for $t = 1, \dots, T$ **do**
  Делаем действие i с вероятностью пропорциональной $w_t(i)$, получая значение $v_t(i)$
  $w_{t+1}(j) = w_t(j)(1 + \varepsilon v(j))$ for $j = 1, ..., N$
end for

Теорема

Пусть $\varepsilon \leq 1/2$. Тогда $\forall j \in \overline{1, \dots, N}$:

$$\sum_{t=1}^{T} \sum_{i=1}^{N} v_t(i) p_t(i) \geq (1 - \varepsilon) \sum_{t=1}^{T} v_t(j) - \frac{1}{\varepsilon} \log N$$

Multiplicative[Wil19]/Exponential weights[Bub11]

Алгоритм 1 Multiplicative Weights

```
1: w_1(i) = 1 for i = 1, ..., N
2: for t = 1, ..., T do
3:     Делаем действие i с вероятностью пропорциональной w_t(i), получая значение v_t(i)
4:     w_{t+1}(j) = w_t(j)(1 + εv(j)) for j = 1, ..., N
5: end for
```

Теорема

Пусть $\varepsilon \leq 1/2$. *Тогда* $\forall j \in \overline{1, \ldots, N}$:

$$\sum_{t=1}^{T} \sum_{i=1}^{N} v_t(i) p_t(i) \geq (1-\varepsilon) \sum_{t=1}^{T} v_t(j) - \frac{1}{\varepsilon} \log N$$

Алгоритм Гарга-Кёнемана [GK07] [Wil19]

Алгоритм 2 Алгоритм Гарга-Кёнемана для задачи о максимальном потоке.

$x(p) = 0 \quad \forall p \in \mathcal{P}$
$f(e) = 0,\ w(e) = 1 \quad \forall e \in E$
while $f(e)/u(e) < (\log m)/\varepsilon^2 \quad \forall e \in E$ **do**
    $P =$ наикратчайший путь из $\mathcal{P}$ по весам $w(e)/u(e)$
    $u = \min_{e \in P} u(e)$
    $x(P) = x(P) + u$
    $f(e) = f(e) + u \quad \forall e \in P$
    $w(e) = (1 + \varepsilon \frac{u}{u(e)}) w(e) \quad \forall e \in P$
end while

Выход: $x / \max_{e \in E}(f(e)/u(e))$

Цели исследования

1. Ускорить алгоритм Гарга-Кёнемана
   1. Применить параллельные методы поиска почти точных путей [Li20] (их нам достаточно [Fle99])
   2. Изменить структуру онлайн задачи и предоставить другую среду
   3. Найти потенциал и использовать адаптивное взвешивание аналогично [VAM21]
2. Применить алгоритм Гарга-Кёнемана для решения других потоковых задач (min-cost-flow, ...)
3. Использовать какой-нибудь другой алгоритм решения онлайн задачи [Ora23]

Цели исследования

1. Ускорить алгоритм Гарга-Кёнемана
   1. Применить параллельные методы поиска почти точных путей [Li20] (их нам достаточно [Fle99])
   2. Изменить структуру онлайн задачи и предоставить другую среду
   3. Найти потенциал и использовать адаптивное взвешивание аналогично [VAM21]
2. Применить алгоритм Гарга-Кёнемана для решения других потоковых задач (min-cost-flow, ...)
3. Использовать какой-нибудь другой алгоритм решения онлайн задачи [Ora23]

Цели исследования

1. Ускорить алгоритм Гарга-Кёнемана
   1. Применить параллельные методы поиска почти точных путей [Li20] (их нам достаточно [Fle99])
   2. Изменить структуру онлайн задачи и предоставить другую среду
   3. Найти потенциал и использовать адаптивное взвешивание аналогично [VAM21]
2. Применить алгоритм Гарга-Кёнемана для решения других потоковых задач (min-cost-flow, ...)
3. Использовать какой-нибудь другой алгоритм решения онлайн задачи [Ora23]

Цели исследования

1. Ускорить алгоритм Гарга-Кёнемана
   1. Применить параллельные методы поиска почти точных путей [Li20] (их нам достаточно [Fle99])
   2. Изменить структуру онлайн задачи и предоставить другую среду
   3. Найти потенциал и использовать адаптивное взвешивание аналогично [VAM21]
2. Применить алгоритм Гарга-Кёнемана для решения других потоковых задач (min-cost-flow, ...)
3. Использовать какой-нибудь другой алгоритм решения онлайн задачи [Ora23]

Цели исследования

1. Ускорить алгоритм Гарга-Кёнемана
    1. Применить параллельные методы поиска почти точных путей [Li20] (их нам достаточно [Fle99])
    2. Изменить структуру онлайн задачи и предоставить другую среду
    3. Найти потенциал и использовать адаптивное взвешивание аналогично [VAM21]
2. Применить алгоритм Гарга-Кёнемана для решения других потоковых задач (min-cost-flow, ...)
3. Использовать какой-нибудь другой алгоритм решения онлайн задачи [Ora23]

Цели исследования

1. Ускорить алгоритм Гарга-Кёнемана
   1. Применить параллельные методы поиска почти точных путей [Li20] (их нам достаточно [Fle99])
   2. Изменить структуру онлайн задачи и предоставить другую среду
   3. Найти потенциал и использовать адаптивное взвешивание аналогично [VAM21]
2. Применить алгоритм Гарга-Кёнемана для решения других потоковых задач (min-cost-flow, ...)
3. Использовать какой-нибудь другой алгоритм решения онлайн задачи [Ora23]

Список литературы I

- Sébastien Bubeck, Introduction to online optimization, **Link**, 2011.
- L.K. Fleischer, Approximating fractional multicommodity flow independent of the number of commodities, 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039), 1999, doi:10.1109/SFFCS.1999.814573, pp. 24–31.
- 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.

Список литературы II

- Jason Li, Faster parallel algorithm for approximate shortest path, Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (New York, NY, USA), STOC 2020, Association for Computing Machinery, 2020, doi:10.1145/3357713.3384268, p. 308–321.
- Francesco Orabona, A modern introduction to online learning, 2023, doi:10.48550/arXiv.1912.13213.
- Dong Quan Vu, Kimon Antonakopoulos, and Panayotis Mertikopoulos, Fast routing under uncertainty: Adaptive learning in congestion games via exponential weights, Advances in Neural Information Processing Systems (M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, eds.), vol. 34, Curran Associates, Inc., 2021, **Link**, pp. 14708–14720.

Список литературы III

David P. Williamson, Network flow algorithms, Cambridge University Press, 2019, doi:10.1017/9781316888568.