Алгоритмы и структуры данных

Кратчайшие пути: продолжение

CS Center, Новосибирск

P2P кратчайший путь

- Ищем кратчайший путь s → t
  - Алг. Дейкстры с ранней остановкой
    - останавливаемся, когда t стала чёрной
    - T ~ r^d
  - Алг. Дейкстры с двух сторон
    - останавливаемся, когда есть вершина, чёрная с обеих точек зрения
    - кратчайший путь:
      - s→v + (v, u) + u→t
        - v – чёрная для s, u – чёрная для t
    - T ~ 2*(r/2)^d
      - В ½^(d-1) раз быстрее, чем однонаправленный алгоритм

Алгоритм Дейкстры с потенциалами

- Идея: выбрать новые длины l': $E \to \mathbb{R}$, $l'(e) \geq 0$
  - Путь p – кратчайший для l ⇔ p – кратчайший для l'
- Можно выбрать потенциалы ϕ: $V \to \mathbb{R}$
  - $l'(u, v) = l(u, v) + \phi(u) - \phi(v)$
  - Тогда для пути $p: x \to y$
    - $d'(p) = d(p) + \phi(x) - \phi(y)$
- Нет отрицательных циклов ⇔ существуют потенциалы ϕ: $l'(e) \geq 0$
  - (<=) длина цикла не зависит от потенциалов
  - (=>) можно взять $dist(s, v)$ в качестве $\phi(v)$
    - длина $l'(u, v) = l(u, v) + dist(s, u) - dist(s, v) \geq 0$
      - это верно, т.к. $dist(s, u) + l(u, v) \geq dist(s, v)$

Алгоритм Джонсона

- Вычислим потенциалы алгоритмом Форда-Беллмана
- Теперь можно запускать алг. Дейкстры из любой вершины

- Алг. Джонсона:
  - Задача APSP
  - Форд-Беллман
  - |V| раз Дейкстра
  - Сложность $O(|V| * |E| * \log|V|)$
    - Быстрее, чем алг. Флойда, если граф разреженный

P2P: алгоритм A*

- Умеем оценивать насколько каждая вершина близка к t
- Предположение: знаем dist(v, t)
  - $\phi(v) := -dist(v, t)$
  - $l'(u, v) = l(u, v) – dist(u, t) + dist(v, t)$
  - $l'(u, v) = 0$, если (u, v) на кратчайшем пути
    - Алг. Дейкстры пройдёт только по этому пути
- Можно выбрать $\phi(v) = -d_{est}(v, t)$, $d_{est}(v, t) \leq dist(v, t)$
  - граф в пространстве, $d_{est}(v) = eucl\_dist(v, t)$ – допустимы для алг. Дейкстры
    - $l'(u, v) = l(u, v) - eucl\_dist(u, t) + eucl\_dist(v, t) \geq 0$
      - Верно, т.к. $l(u, v) + eucl\_dist(v, t) \geq eucl\_dist(u, t)$

- A* - это алг. Дейкстры, где выбираем $argmin\{ d(v) + d_{est}(v) \}$

P2P: алгоритм ALT

- Как получить оценку d_{est} для A*?
  - Выбираем множество вершин L
  - Для l из L и для каждой вершины v вычислим d(v, l)
    - $O(|L||E|\log|V|)$ на предподсчёт
  - Решаем задачу d(s, t) - ?
    - $d(v, t) + d(t, l) \geq d(v, l)$
      - d(t, l) и d(v, l) предподсчитаны
      - $d(v, t) \geq d(v, l) - d(t, l)$
    - $d_{est}(v) = \max\{ d(v, l) - d(t, l) \}$ по всем l из L
  - $\phi_l(v) = d(t, l) - d(v, l)$
    - $\phi(v) = \min\{ \phi_l(v) \}$ по всем l из L

P2P: highway hierarchy

- Алгоритм для графов дорог
- Предобработка:
  - Разбиваем граф на части из не более чем k вершин
    - При этом минимизируем число вершин, инцидентных разрезанным рёбрам
    - Для этого используется эвристика
  - Для каждой пары граничных вершин каждой части вычисляем кратчайшее расстояние
  - Каждую часть заменяем на полный граф с соответствующими весами рёбер
- Вычисление d(s, t):
  - в части с s запускаем алг. Дейкстры и находим $d(s, exit_s)$
  - в части с t находим $d(exit_d, t)$
  - находим $d(exit_s, exit_d)$
- Подбираем параметр k, чтобы получить примерно равные по размеру графы на общем и частном уровнях

P2P: hub system

- $H: V \to 2^V$
- Для любых (s, t) существует $x \in H(s) \cap H(t)$:
  - x на кратчайшем пути $s \to t$
- Предподсчёт:
  - расстояния между (v, x), $x \in Hx$
- Нахождение расстония:
  - $T = |H(s)| + |H(t)|$