ЛАБОРАТОРНАЯ РАБОТА №4

Задача максимального разреза (Max-Cut Problem)

Дисциплина: Квантовые подходы к обработке информации и искусственному интеллекту

1. ПОСТАНОВКА ЗАДАЧИ

Дано:

- Неориентированный граф $G = (V, E)$
- Множество вершин: $V = \{0, 1, ..., n-1\}$
- Множество рёбер: $E \subseteq V \times V$
- Веса рёбер: $w_{ij} \geq 0$ для каждого $(i, j) \in E$ (по умолчанию $w_{ij} = 1$)

Требуется найти:

Разбиение вершин графа на два непересекающихся множества S и $\bar{S} = V \setminus S$ такое, что вес разреза максимален:

$$\mathrm{Cut}(S, \bar{S}) = \sum_{\substack{i \in S, j \in \bar{S} \\ (i,j) \in E}} w_{ij}$$

Целевая функция:

$$\max_{S \subset V} \mathrm{Cut}(S, \bar{S})$$

Практический смысл:

Задача возникает в:

- **Кластеризация данных:** разделение на две группы с максимальным числом связей между группами
- **VLSI-проектирование:** размещение элементов схемы для минимизации внутренних связей
- **Анализ социальных сетей:** выявление противостоящих сообществ
- **Балансировка нагрузки:** распределение задач между двумя серверами

2. QUBO-ФОРМУЛИРОВКА

2.1. Бинарные переменные

$$x_i \in \{0, 1\}, \quad i \in V$$

Интерпретация:

- $x_i = 1$ означает, что вершина i принадлежит множеству S
- $x_i = 0$ означает, что вершина i принадлежит множеству $\bar{S}$

Число переменных: n

2.2. Целевая функция

Ребро (i, j) попадает в разрез, если вершины i и j находятся в разных множествах, то есть когда $x_i \neq x_j$.

Индикатор разреза для ребра:

$$1_{\text{cut}}(i, j) = x_i(1 - x_j) + x_j(1 - x_i) = x_i + x_j - 2x_i x_j$$

Вес разреза:

$$\text{Cut}(x) = \sum_{(i,j) \in E} w_{ij}(x_i + x_j - 2x_i x_j)$$

Максимизация разреза эквивалентна **минимизации**:

$$E_{obj}(x) = -\sum_{(i,j) \in E} w_{ij}(x_i + x_j - 2x_i x_j)$$

Раскрывая и группируя:

$$E_{obj}(x) = 2 \sum_{(i,j) \in E} w_{ij} x_i x_j - \sum_{i \in V} d_i x_i$$

где $d_i = \sum_{j:(i,j) \in E} w_{ij}$ — взвешенная степень вершины i.

2.3. Итоговая энергия

$$E_{total}(x) = E_{obj}(x)$$

Задача: $\min_{x \in \{0,1\}^n} E_{total}(x)$

Примечание: В задаче Max-Cut нет жёстких ограничений — каждая вершина однозначно либо в S (если $x_i = 1$), либо в $\bar{S}$ (если $x_i = 0$). Поэтому штрафы за нарушение ограничений не нужны.

3. МЕТОДЫ РЕШЕНИЯ

3.1. Имитация отжига (Simulated Annealing)

Идея: Метод глобальной стохастической оптимизации, основанный на аналогии с физическим процессом отжига металлов. Алгоритм начинает с высокой "температуры", при которой допускаются переходы в состояния с большей энергией (для исследования пространства решений и избежания локальных минимумов). Затем температура постепенно снижается, и алгоритм становится более "жадным".

Основные компоненты:

1. **Температурный график:** Управляет вероятностью принятия ухудшающих решений

$$T(t) = T_0 \cdot \alpha^t$$

   где T_0 — начальная температура, $\alpha \in (0, 1)$ — коэффициент охлаждения, t — номер итерации
2. **Критерий принятия решения:** Для перехода из состояния с энергией E в состояние с энергией E':
   - Если $\Delta E = E' - E < 0$ (улучшение) — принять всегда
   - Если $\Delta E \geq 0$ (ухудшение) — принять с вероятностью:

$$P(\text{accept}) = e^{-\Delta E / T}$$

   При высокой температуре T эта вероятность близка к 1 (принимаем почти любые ухудшения). При низкой температуре $T \to 0$ эта вероятность стремится к 0 (становимся жадными).

Параметры для подбора:

- **Начальная температура** T_0: Обычно подбирается так, чтобы в начале принималось около 80% ухудшающих переходов. Типичный диапазон: $T_0 \in [1.0, 3.0]$

- **Коэффициент охлаждения** $\alpha \in [0.93, 0.99]$: Чем ближе к 1, тем медленнее охлаждение. Типичные значения: 0.95-0.97.
- **Критерий остановки:** Обычно $T < T_{min}$ (например, $T_{min} = 0.001$) или фиксированное число итераций.

Алгоритм:

1. **Инициализация:**
   - Случайное начальное состояние: $x_i \in \{0, 1\}$ для всех i
   - Начальная температура: $T_0 = 2.0$
   - Минимальная температура: $T_{min} = 0.001$
   - Коэффициент охлаждения: $\alpha = 0.96$
   - Вычислить начальную энергию E_0
2. **Основной цикл** (пока $T > T_{min}$):

   а) **Выбор переменной:** Выбрать случайную вершину $i \in V$

   б) **Вычисление изменения энергии:** При перевороте бита $x_i \to 1 - x_i$ изменение энергии:

$$\Delta E = (1 - 2x_i) \left(2 \sum_{j:(i,j) \in E} w_{ij} x_j - d_i \right)$$

   Это $O(\deg(i))$ вычисление — нужно учесть только соседей вершины i.

   в) **Критерий принятия:**
   - Если $\Delta E < 0$ → принять всегда
   - Если $\Delta E \geq 0$ → принять с вероятностью $e^{-\Delta E / T}$

   г) **Обновление:**
   - Если принят: $x_i \leftarrow 1 - x_i$, $E \leftarrow E + \Delta E$
   - Обновить лучшее решение

   д) **Охлаждение:** $T \leftarrow \alpha \cdot T$
3. **Возврат:** Лучшее найденное разбиение

3.2. Жадный алгоритм + локальные улучшения (Baseline)

Идея: Простой жадный алгоритм с последующими локальными улучшениями.

Алгоритм:

1. **Жадное построение:**
   - Начать с произвольного разбиения (например, случайное)
   - Для каждой вершины вычислить выгоду переноса в другое множество:

$$\text{gain}_i = \sum_{j \in \bar{S}_i, (i,j) \in E} w_{ij} - \sum_{j \in S_i, (i,j) \in E} w_{ij}$$

где S_i — текущее множество вершины i, $\bar{S}_i$ — противоположное
- Переносить вершины с положительной выгодой

2. **Локальные улучшения (2-opt):**
   - Повторять:
     - Для каждой вершины i: вычислить изменение разреза при переносе
     - Если есть улучшение → перенести вершину
   - Пока есть улучшения
3. **Возврат:** Итоговое разбиение

4. МЕТРИКИ СРАВНЕНИЯ

4.1. Основные метрики

Вес разреза (целевая метрика):

$$\text{Cut} = \sum_{\substack{i \in S, j \in \bar{S} \\ (i,j) \in E}} w_{ij}$$

Доля от суммарного веса рёбер:

$$\text{Ratio}(\%) = \frac{\text{Cut}}{\sum_{(i,j) \in E} w_{ij}} \times 100\%$$

Баланс разбиения:

$$\text{Balance} = \frac{\min(|S|, |\bar{S}|)}{n}$$

Идеально сбалансированное разбиение: $\text{Balance} = 0.5$

Gap от baseline:

$$\text{Gap}(\%) = \frac{\text{Cut}_{SA} - \text{Cut}_{baseline}}{\text{Cut}_{baseline}} \times 100\%$$

4.2. Сравнение времени выполнения

Для каждого метода измерять:

- Время до лучшего найденного решения
- Общее время работы

5. ПЛАН ЭКСПЕРИМЕНТОВ

5.1. Требования

1. **Генерация инстанса:** Сгенерировать граф согласно вашему варианту (n вершин, вероятность ребра p, seed)
2. **Реализация методов:**
   - Имитация отжига (обязательно)
   - Жадный алгоритм с локальными улучшениями (baseline для сравнения)
3. **Множественные запуски:** Для имитации отжига выполнить **не менее 10 запусков** с разными random seed
4. **Сбор данных:** Для каждого запуска записать:
   - Вес разреза
   - Баланс разбиения
   - Время работы
   - Траекторию лучшего решения (для графика)

5.2. Анализ результатов

Статистический анализ:

- Вычислить среднее, стандартное отклонение, медиану для имитации отжига
- Построить boxplot распределения весов разреза
- Сравнить SA с baseline

Визуализация:

- График сходимости SA: зависимость лучшего разреза от времени
- Визуализация графа и лучшего разреза (для небольших n)
- Гистограмма размеров множеств $|S|$ и $|\bar{S}|$

Сравнительная таблица:

Метод	Средний разрез	Лучший	Худший	Время (сек)	Balance
SA	...	...	...	...	...
Greedy	...	...	...	...	...

5.3. Подбор параметров

Провести эксперименты с варьированием:

- $T_0 \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$
- $\alpha \in \{0.90, 0.93, 0.96, 0.99\}$

Построить графики качества решения - параметры.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Почему задача Max-Cut является NP-трудной?
2. Объясните, как бинарные переменные $x_i \in \{0, 1\}$ кодируют разбиение вершин на два множества.
3. Выведите формулу для быстрого вычисления ΔE при перевороте бита x_i. Какова сложность?
4. Объясните физический смысл температуры T в имитации отжига. Что происходит при $T \to 0$ и при $T \to \infty$?
5. Какие значения параметров T_0 и α вы выбрали? Обоснуйте выбор на основе экспериментов.
6. Достигается ли в вашем лучшем решении хороший баланс между множествами? Важен ли баланс?
7. Приведите примеры реальных задач, которые сводятся к Max-Cut.
8. Как зависит сложность задачи от плотности графа (вероятности ребра p)?
9. В чём преимущество имитации отжига перед жадным алгоритмом для Max-Cut?
10. Предложите модификацию: Max-Cut с ограничением на минимальный размер меньшей части разбиения. Как изменится QUBO?

УДАЧИ В РАБОТЕ!