

Использование тензорных сетей для изучения квантового отжига

Олег Лаврин

10 июня 2022 г.

1 Введение

Комбинаторная оптимизация - одно из наиболее важных направлений математической оптимизации, встречающейся как в промышленности, так и в физике. Задача заключается в отыскании экстремума объектной функции на некотором комбинаторно-заданном множестве. Например на путях в графе, или среди покрытий какого-то множества. Как правило, эти задачи сводятся принятию большого количества решений "ДА/НЕТ", при выполнении некоторых ограничений на эти комбинации.

Примером такой задачи, востребованной в промышленности, в том числе военной[1], является покрытие множеств. Пусть у нас имеется некоторое множество \mathcal{N} , которое мы хотим покрыть множествами из набора $\mathcal{S} = \{S_i \subset \mathcal{N}\}$, минимизируя стоимость этого покрытия.

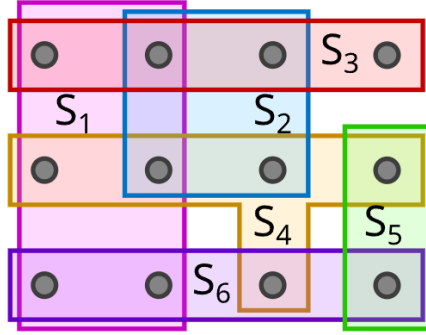


Рис. 1: Пример задачи покрытия. Необходимо покрыть серые точки минимальным кол-вом цветных фигур. Изображение взято с <https://www.gcardone.net/2019-08-31-visa-free-travel/>

Строгая формулировка задачи выглядит следующим образом:

$$\sum_{s \in \mathcal{S}} w_s x_s \rightarrow \min_{x_s \in \mathcal{B}} \quad (1)$$

$$s.t. \forall j \in \mathcal{N} \hookrightarrow \sum_{s \in \mathcal{S}} (j \in S) x_s \geq 1 \quad (2)$$

В случае, если кол-во элементов $|\mathcal{N}| = N$ и $|\mathcal{S}| = M$ конечно, то задачу можно переформулировать в матричном виде:

$$w^T x \rightarrow \min_{x \in \mathcal{B}^M} \quad (3)$$

$$s.t. Ax \succeq 1 \quad (4)$$

где A - матрица, столбцами которой мы хотим покрыть вектор единиц.

Следующим примером комбинаторной оптимизации можно рассмотреть задачу странствующего торговца. Имеется набор городов и сеть дорог, которыми они соединены. Странствующий торговец пытается посетить их все, минимизируя пройденный путь.



Рис. 2: Пример из реального мира. Оптимальный маршрут коммивояжёра через 15 крупнейших городов Германии. Изображение взято с https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden#/media/Datei:TSP_Deutschland_3.png

Эту задачу тоже можно сформулировать в матричном виде:

$$\sum_{n,m,t}^N d_{n,m} x_{n,t} x_{m,t+1} \rightarrow \min_{x_{n,t} \in \mathcal{B}} \quad (5)$$

$$s.t \forall n \hookrightarrow \sum_t x_{n,t} = 1 \quad (6)$$

$$\forall t \hookrightarrow \sum_n x_{n,t} = 1 \quad (7)$$

Эти задачи, как и многие другие практически значимые задачи, чрезвычайно трудно решаются. Традиционный подход их решения заключается в формировании эмпирик, основанных на математической структуре конкретных задач. Несмотря на то, что этот подход дает приемлимые результаты, он обладает качественным недостатком - для каждой конкретной задачи требуется разработать свой алгоритм, практически неприменимый для других задач.

Основным барьером на пути создания универсального алгоритма для их решений встают ограничения на множество, по которому ведется поиск. Одним из стандартных способов обойти этот барьер - воспользоваться релаксацией ограничений, предложенной в следующем гайдбуке[2].

Квадратичную оптимизацию с линейными ограничениями вида **равно** и **больше-или-равно** можно переформулировать в оптимизацию со **штрафами**. Таким образом многие задачи комбинаторной оптимизации могут быть сведены к виду:

$$x^T Q_{obj} x + f_{penalty}(x) \rightarrow \min_{x_i \in \mathcal{B}} \quad (8)$$

где $f_{penalty}$ - штрафы которые формируют из ограничений оригинальной задачи. Зачастую, эти ограничения тоже получаются квадратичными.

Classical Constraint	Equivalent Penalty
$x + y \leq 1$	$P(xy)$
$x + y \geq 1$	$P(1 - x - y + xy)$
$x + y = 1$	$P(1 - x - y + 2xy)$
$x \leq y$	$P(x - xy)$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1 x_2 + x_1 x_3 + x_2 x_3)$
$x = y$	$P(x + y - 2xy)$

Рис. 3: Список из некоторых ограничений, сводимых к квадратичным штрафам. Таблица взята из того же гайдбука[2].

В случае, если нам таки удалось представить штраф в виде квадратичной функции, то наша задача сводится к так называемой Quadratic Unconstrained Binary Optimization (QUBO):

$$x^T Q_{eff} x \rightarrow \min_{x_i \in \mathcal{B}} \quad (9)$$

2 Изинговские цепочки

Используя замену $x_i = \frac{1+s_i}{2}$, можно показать, что эта задача эквивалентна поиску основного состояния в модели Изинга со следующим гамильтонианом:

$$\hat{H} = - \sum_{i=1}^n \sum_{j=1}^i J_{i,j} \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} - \sum_{i=1}^n h_i \hat{\sigma}_z^{(i)} \quad (10)$$

Особенно интересен случай с внешним полем $h = 0$. В таком случае эта задача имеет следующую эмпирику, называемую local search(LS)[3]:

Для каждой вершины случайно решаем, в какую сторону направлен спин, а затем переворачиваем один спин за один шаг, улучшая решение на каждом шаге до тех пор, пока улучшение возможно.

Неожиданно, но подобный алгоритм хорошо работает в случаях простых физических задач. К сожалению, на практике встречаются и задачи, в которых этот подход быстро сходится к локальному минимуму и глохнет. Одним из качественных примеров являются **фрустрированные** Изинговские цепочки[4].

Фрустрацией цепочки называют ситуацию, когда геометрия связей такова, что невозможно направить спины, одновременно обеспечивая локальный минимум энергии на всех связях.

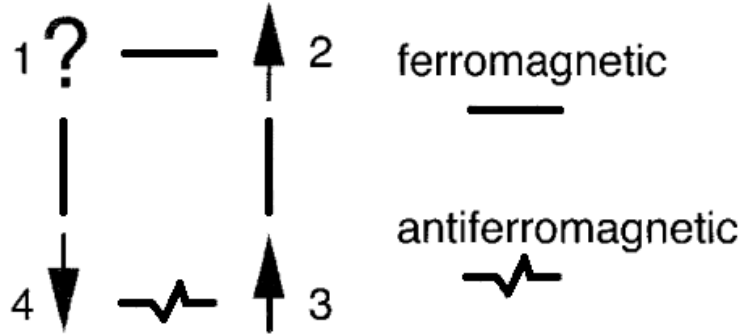


Рис. 4: Пример фрустрированной цепочки. Независимо от того, куда направлен первый спин, на одной из его связей не будет минимума энергии, а все остальные связи будут иметь минимальную энергию.

Задача о поиске основного состояния в подобных системах крайне трудна для классических алгоритмов. Наиболее распространенным является метод ветвей и границ, который можно назвать умным полным перебором.[4]

Теоретически, квантовые компьютеры могут дать экспоненциальное ускорение для решения подобных задач.[5] К сожалению, современные универсальные квантовые компьютеры слишком маленькие и шумные для применения в этих задачах.

Компромиссным решением является **квантовый отжиг**(d-wave) и **квантово-вдохновленные алгоритмы**(тензорные сети) для классических компьютеров. В отличие от квантовых компьютеров, эти решения уже существуют и доступны. D-wave предоставляет облачный доступ к своему компьютеру, а для тензорных сетей написана удобная библиотека на python.

Для двух этих подходов было интересно провести их сравнительный анализ. В качестве тестовой задачи было взято отыскания основного состояния для следующей фрустрированной цепочки Изинга:

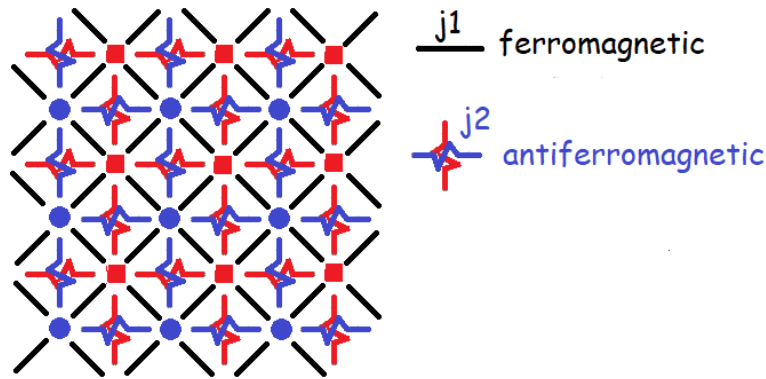


Рис. 5: Решетка для тестовой задачи. Она состоит из двух антиферромагнитных подрешеток, сдвинутых на пол-периода, взаимодействующих друг с другом ферромагнитным образом.

В зависимости от отношения $\frac{j_2}{j_1}$ меняется и поведение системы. При $\frac{j_2}{j_1} \rightarrow 0$ решетка обладает ферромагнитными свойствами, а при $\frac{j_2}{j_1} \rightarrow \infty$ антиферромагнитными. Именно наличие двух фаз и делает конкретно эту задачу сложной и интересной.

3 Принцип работы d-wave

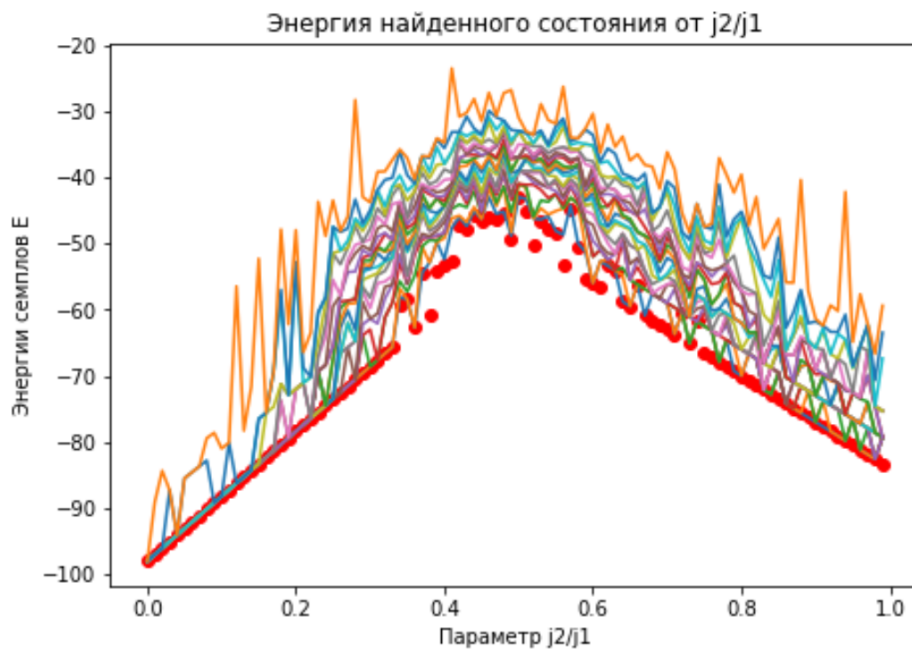
Принцип работы квантового отжига, устройство d-wave и источники ошибок (вырождение уровней).

4 Принцип работы тензорных сетей

Что такое тензорная сеть. Крестовая аппроксимация. Оптимизация и приближение MPS. Поиск основного состояния через DMRG(ALS).

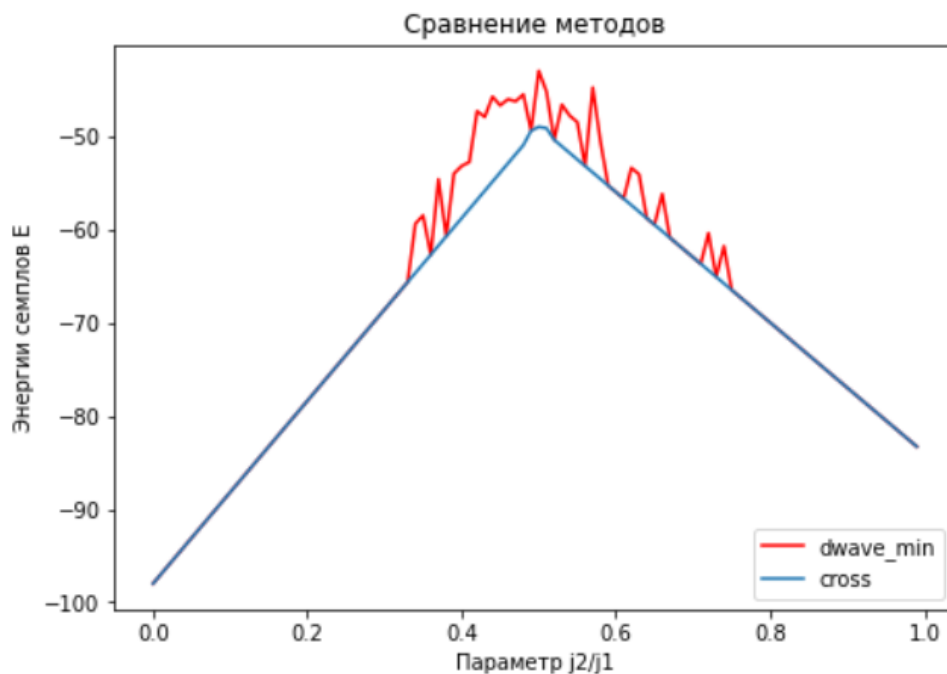
5 Поиск основного состояния через d-wave и cross-opt. Сравнение качества решения.

Взял тестовую решетку размеров 7×7 и начал искать основное состояние. Для разных параметров $\frac{j_2}{j_1}$ провел серию из 23 запусков машины и нанес полученные энергии на график. Красными точками отмечены минимумы по каждой из серии запусков.



6 Поиск основного состояния через Cross-opt. Сравнение качества решения.

Далее я вычислил энергию основного состояния через cross-opt и сравнил его с лучшим решением d-wave. Как видно из графика, имеется область $\frac{j_2}{j_1} \in [0.4, 0.55]$, в котором d-wave не смог найти основное состояние. Вероятно, это связано с тем, что во время отжига уровень возбужденного состояния касается основного. Таким образом адиабатическая теорема нарушается и d-wave перестает эффективно находить основное состояние.



7 Уровни энергии во время отжига.

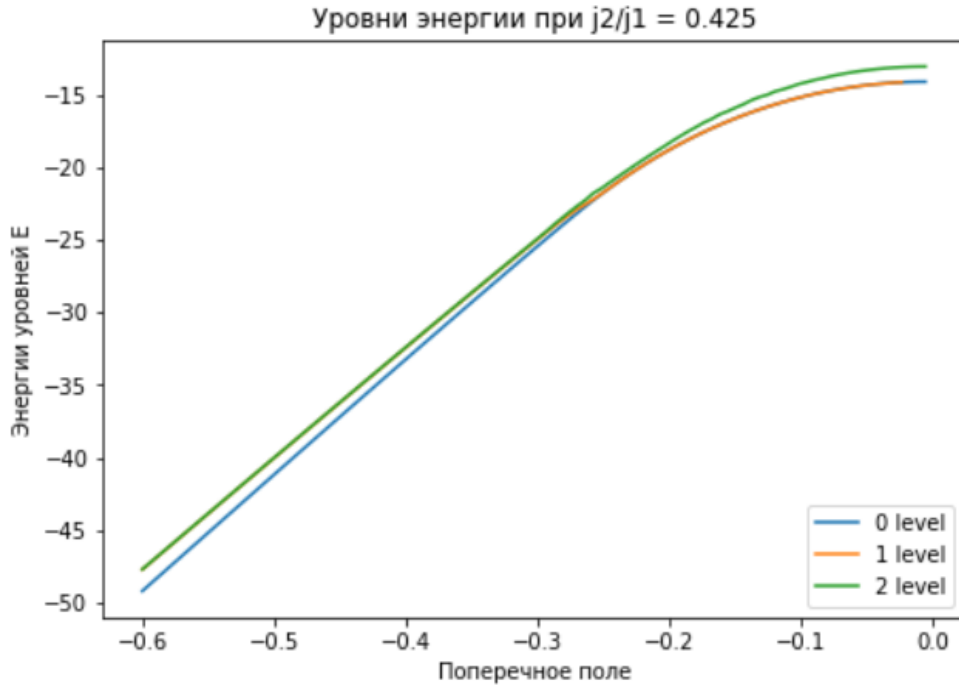
Симуляция через тензорные сети.

Для того, чтобы проверить, происходит ли вырождение основного состояния, воспользуемся тензорными сетями. Хотя аналитическое разложение гамильтониана в МРО достаточно простое, его ранг оказывается L^4 , что является недопустимо большим числом. Вместо этого приблизим его через крестовую аппроксимацию.

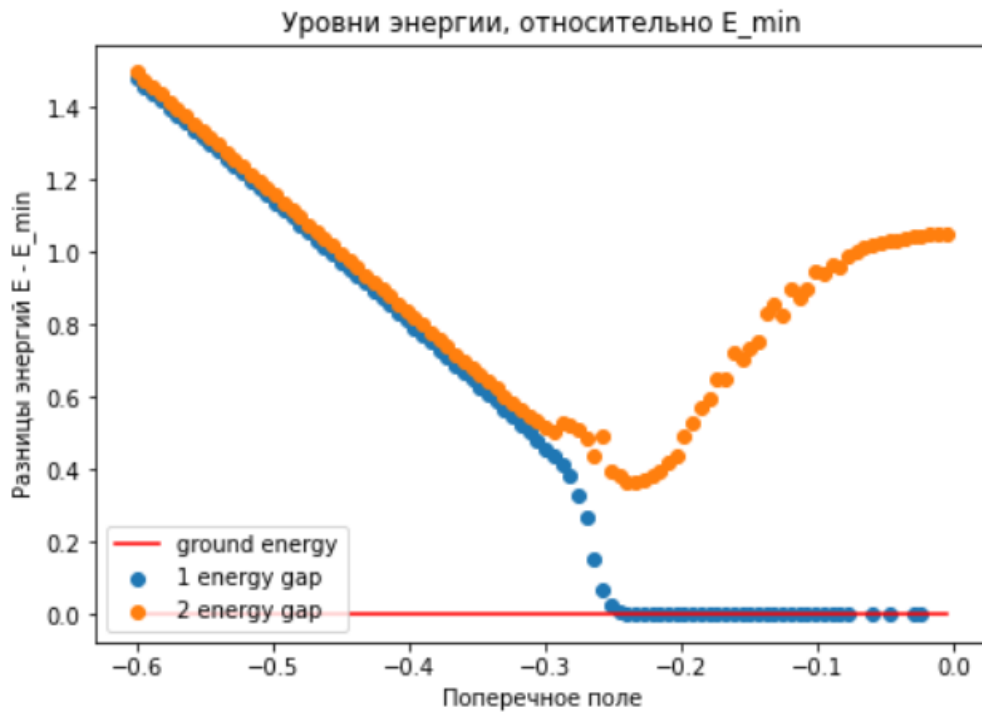
Для построения уровней энергии сгенерируем гамильтониан решетки \hat{H}_0 через крестовую аппроксимацию, гамильтониан поперечного поля $\hat{H}_{field} = \sum_i \hat{\sigma}_x^{(i)}$ можно получить аналитически с рангом 2.

Сначала находим основное состояние этого гамильтониана, добавляем к гамильтониану проектор на это состояние и находим основное состояние снова. Повторяем это столько раз, сколько уровней энергии необходимо найти.

По такому алгоритму я собрал данные о трех уровнях энергии и построил график ниже.



Смотреть на график в таком масштабе очень неудобно, поэтому дальше я построил график разниц энергии между основным и возбужденными состояниями. Как видно из этого графика, никакого касания основного состояния, способного вызвать ошибки, не наблюдается.



8 Вывод

Дивейв не просто работает хуже, чем тензорные сети, но и ломается при неопределенных обстоятельствах. Даже в случае, когда дивейв теоретически работает, его практическая реализация дает сбой. А тензорные сети работают достаточно хорошо, чтоб это проверить.

Список используемой литературы

- [1] Duzdar Argun, Irem. (2019). An Overview on Set Covering Problems With a Focus on Military Applications. 10.4018/978-1-5225-5513-1.ch003.
- [2] Glover, Fred and Kochenberger, Gary and Du, Yu. (2018). A Tutorial on Formulating and Using QUBO Models. 10.48550/ARXIV.1811.11538
- [3] Michael Dinitz. (2019). Approximation Algorithms Lectures. Local Search: Max Cut and Weighted Max Cut. 601.435/635
- [4] Alexander K. Hartmann, Heiko Rieger. (2002). Optimization Algorithms in Physics.
- [5] Zahedinejad, Ehsan and Zaribafiyani, Arman. (2017). Combinatorial Optimization on Gate Model Quantum Computers: A Survey. 10.48550/ARXIV.1708.05294