Параметризованные полуопределенные релаксации и их приложения

Александр Сергеевич Подкопаев

Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Научный руководитель: н.с ИППИ РАН, к.ф.-м.н. Ю. В. Максимов

Группа 274, 29 июня 2015

Проблема

Цель исследования

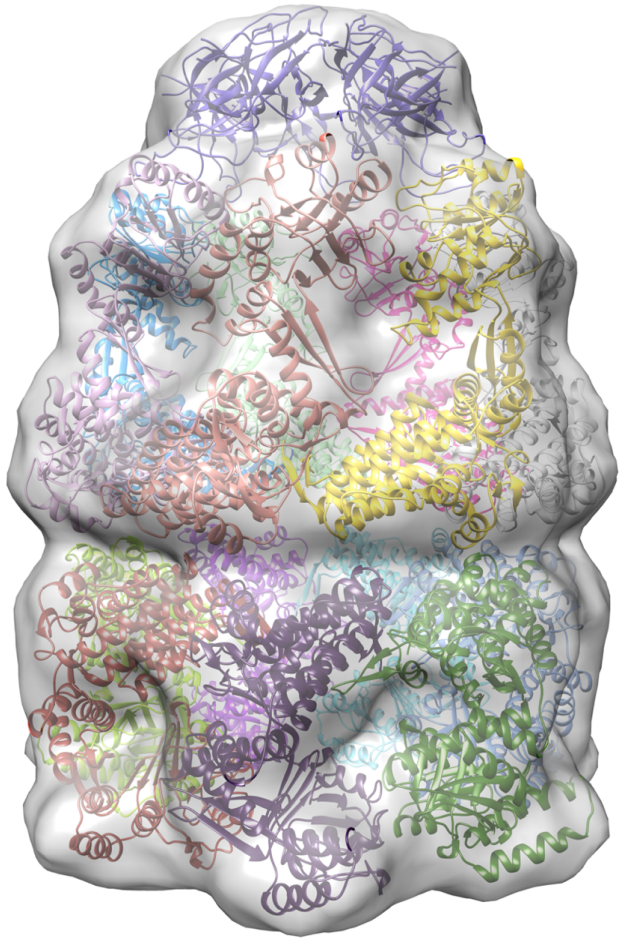
- быстрое и точное решение оптимизационных задач на примере задачи упаковки белковых молекул в комплекс

Проблема

- Появление задач, точное решение которых за разумное время вряд ли возможно

Предложение

- Применение различных релаксаций к исходной постановке



Обозначения

- В цепи n белков ($n \sim 10$), каждый белок может занимать одну из p позиций ($p \sim 100$).
- Позиция каждого i-ого белка определяется вектором $\mathbf{x}^i = [x_1^i, x_2^i, ..., x_p^i]^T$.
- N - число возможных позиций белков, составляющих один комплекс.
- q - энергия взаимодействия пары белков.
- b_0 - собственная энергия, отвечающая позиции белка.

Математическая постановка задачи

$$\underset{\mathbf{x}\in\{0,1\}^N}{\text{minimize}} \quad \mathbf{x}^T\mathbf{Q}_0\mathbf{x} + \mathbf{b}_0^T\mathbf{x}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{1}_n,$$

где

$$\mathbf{A} = \begin{bmatrix} \underbrace{1 \cdots 1}_{p_1} & \underbrace{0 \cdots 0}_{p_2} & \cdots\cdots & \underbrace{0 \cdots 0}_{p_n} \\ \underbrace{0 \cdots 0}_{p_1} & \underbrace{1 \cdots 1}_{p_2} & \cdots\cdots & \underbrace{0 \cdots 0}_{p_n} \\ \vdots \quad \ddots \quad \vdots & \vdots \quad \ddots \quad \vdots & \cdots\cdots & \vdots \quad \ddots \quad \vdots \\ \underbrace{0 \cdots 0}_{p_1} & \underbrace{0 \cdots 0}_{p_2} & \cdots\cdots & \underbrace{1 \cdots 1}_{p_n} \end{bmatrix}$$

и

$$\mathbf{Q}_0 = \begin{bmatrix} [0] & [q_{12}(p_1,p_2)] & \cdots & [q_{1n}(p_1,p_n)] \\ [q_{21}(p_2,p_1)] & [0] & \cdots & [q_{2n}(p_2,p_n)] \\ \vdots & \vdots & \ddots & \vdots \\ [q_{n1}(p_n,p_1)] & [q_{n2}(p_n,p_2)] & \cdots & [0] \end{bmatrix}$$

Особенности решения

- $NP-$трудная задача.
- Основной подход – релаксация к выпуклой постановке (полуопределенная, Лагранжева).
- Классические релаксации – $O(n^3)$ времени.
- Качество решения задачи определяется размерностями и свойствами матрицы $\mathbf{Q}_0$.
- Большие размерности – нужны более эффективные подходы, учитывающие свойства $\mathbf{Q}_0$.

Полуопределенная релаксация

Свойство

$$a^T \mathbf{B} a = \mathrm{tr}(a^T \mathbf{B} a) = \mathrm{tr}(\mathbf{B} \underbrace{a a^T}_{\mathbf{A}}) = \sum_{i,j} \mathbf{B}_{ij} \mathbf{A}_{ij}$$

Полуопределенная релаксация

$$\begin{aligned} \underset{\mathbf{X} \in \mathcal{S}_+^N,\ \mathbf{x} \geq \mathbf{0}_N}{\text{minimize}} \quad & \sum_{i,j} \mathbf{Q}_{ij} \mathbf{X}_{ij} \\ \text{subject to} \quad & \mathbf{X} \succeq \mathbf{x}\mathbf{x}^T, \\ & X_{ii} = x_i, \ \ i = 1, ..., N, \\ & \mathbf{A}\mathbf{x} = \mathbf{1}_n, \end{aligned}$$

где $\mathcal{S}_+^N = \{\mathbf{X}_{N,N} : \mathbf{X} = \mathbf{X}^T \succeq 0\}$ – пространство всех симметричных положительно-полуопределенных матриц.

- Матрице энергий $\mathbf{Q}$ размера $n \times n$ сопоставим граф G с матрицей смежности $\mathbf{A}$:

$$\mathbf{A} = \begin{cases} 1, & \text{если } \mathbf{Q}_{ij} \neq 0 \\ 0, & \text{если } \mathbf{Q}_{ij} = 0 \end{cases}$$

- Строим хордальное расширение (Алгоритм 1).
- В полученном расширении ищем клики максимального размера за полиномиальное время (Алгоритм Тарьяна, Яннакакиса).
- Декомпозируем задачу на задачи "в кликах".
- Декомпозиция дает существенный выигрыш для достаточно "простых" графов.

Алгоритм 1

Основной алгоритм

1. **Вход**: граф $G = (V, E)$ и число вершин $n = |V|$;
2. $G_1 = G; E' = \emptyset$;
3. **Для** $i = 1$ *to* n:

При необходимости добавляем ребра к G_i так, чтобы все соседи вершины i стали попарно смежными, добавляем эти ребра к E', убираем вершину i и получаем граф G_{i+1};

4. **Выход**: $G = (V; E \cup E')$ - хордальное расширение графа G.

Свойство

Если размер максимальных клик невелик, то переходим к быстрому поиску минимума по подматрицам, соответствующим кликам.

Случайные графы

- n изолированных вершин $V = \{1, \ldots, n\}$
- Предлагается рассмотреть схему Бернулли, при которой вероятность возникновения ребра между любой фиксированной парой вершин в графе есть $p : p \in [0, 1]$
- Пусть все ребра в графе появляются в графе независимо.
- Вероятность реализации случайного графа $G(n, p)$ с $|E|$ ребрами есть:

$$P(G) = C_{C_n^2}^{|E|} p^{|E|} (1-p)^{C_n^2 - |E|}$$

- Вероятность p не фиксируется, а рассматривается как функция от числа вершин: $p = p(n)$

Фазовые переходы

Теорема 1 [Эрдеш-Реньи]

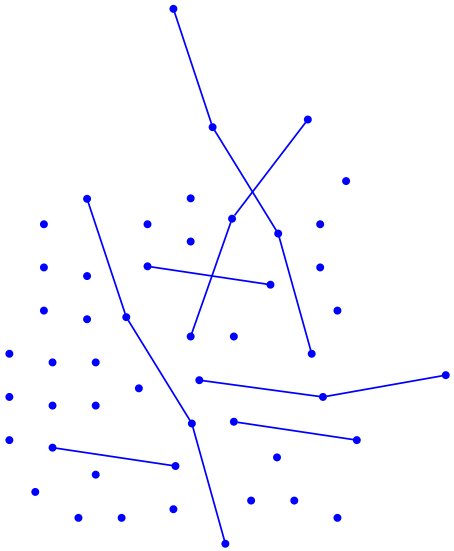
Рассмотрим модель $G(n, p)$. Пусть $p = \frac{c \ln n}{n}$. Если $c > 1$, то с вероятностью, стремящейся к единице при $n \to \infty$, случайный граф связен. Если $c < 1$, то почти всегда случайный граф не является связным.

Теорема 2 [Эрдеш-Реньи]

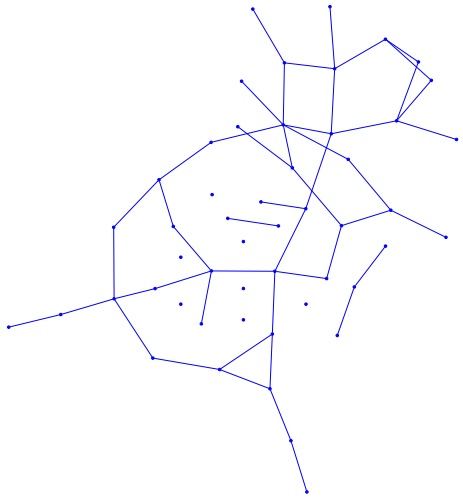
Рассмотрим модель $G(n, p)$. Пусть $p = \frac{c}{n}$. Если $c < 1$, то найдется такая константа $\beta = \beta(c)$, что с вероятностью, стремящейся к единице при $n \to \infty$, размер каждой связной компоненты случайного графа не превосходит $\beta \ln n$. Если $c > 1$, то найдется такая константа $\gamma = \gamma(c)$, что с вероятностью, стремящейся к единице при $n \to \infty$, в случайном графе есть ровно одна компонента связности размера $\geq \gamma n$.

Случайные графы

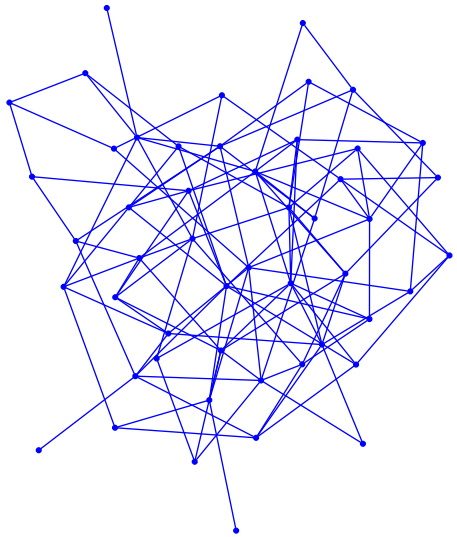
- Рассмотрим случайные графы на 50 вершинах с различными значениями p



$p = \frac{1}{2}\frac{1}{n}$



$p = \frac{3}{2}\frac{1}{n}$



$p > \frac{\ln n}{n}$

Лапласиан графа

Определение

Пусть задан граф G на n вершинах. Тогда его лапласиан определяется как:

$$L = D - A,$$

где D - матрица, на главной диагонали которой степени вершин графа, а остальные элементы - нули, а A - матрица смежности графа G.

Свойства

- L обладает свойством диагонального доминирования, то есть $\forall i \in \{1, \ldots, n\}$:

$$L_{ii} \geq \sum_{j=1, j \neq i}^{n} |L_{ij}|$$

Случайное прореживание

Пусть $G(n, p)$ соответствует L, где $p = \frac{1+\varepsilon}{n}$. Рассмотрим задачу:

$$\underset{\mathbf{X} \succeq 0, d(\mathbf{X}) = \mathbf{1}_n}{\text{maximize}} \quad \text{tr}(L\mathbf{X}) \tag{1}$$

$$\underset{\eta : T(\eta) \succeq L}{\text{minimize}} \quad \max_{x_i^2 = 1} \left(\mathbf{x}^T T(\eta) \mathbf{x} \right) \tag{2}$$

Предложение

Применить случайное "прореживание" к графу G: удалять ребра из G с вероятностью q, где $q : p(1-q) < \frac{1}{n}$.

Мотивация

В результате "прореживания" G с "гигантской" компонентой связности бьется на блоки (малые компоненты связности), а задача сводится к задаче минимизации по отдельным блокам.

Вычислительный эксперимент

Цель эксперимента:

Проверка работоспособности алгоритмов на реальных данных, сравнение времени и точности их работы.

Данные:

- 379 матриц энергий из PDB (Protein Database Bank)
- Случайные графы $G(n, p)$ и соответствующие L

Алгоритмы для первой части эксперимента:

- *MC* (Monte – Carlo)
- *SDP* (Полуопределенная релаксация)
- *Chordal SDP* (Полуопределенная релаксация + хордальное замыкание)

Максимальные клики

Матрицы были разбиты на группы по количеству вершин:

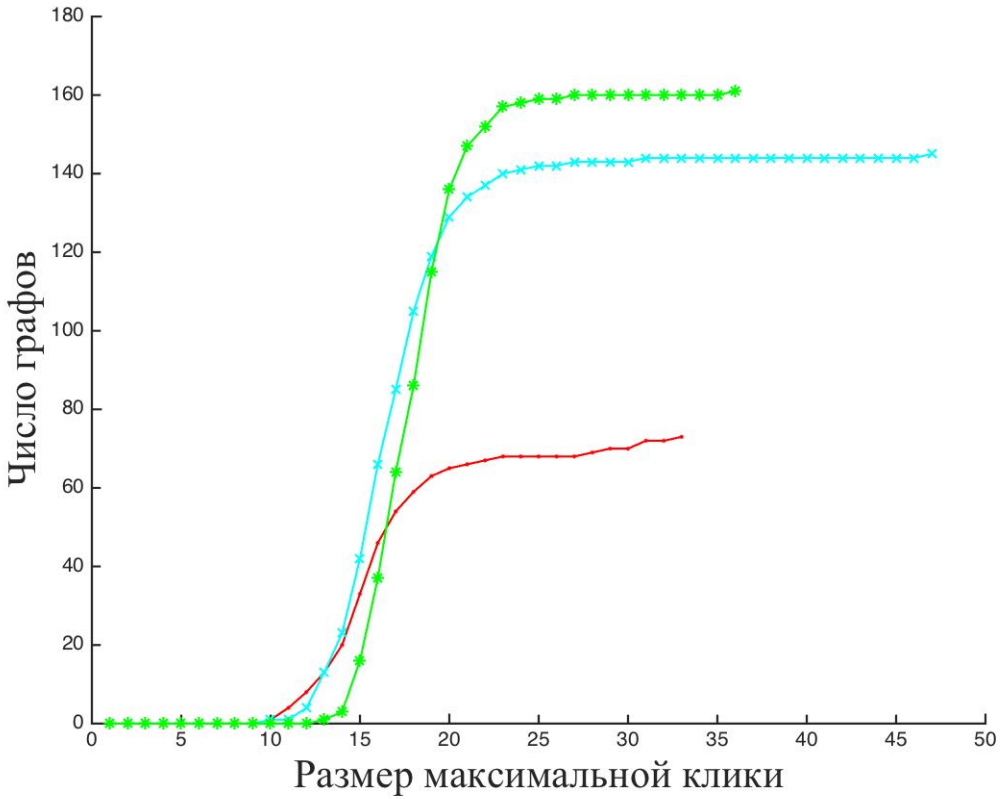
- до 500 – 73 матрицы
- 500-1000 – 145 матриц
- больше 1000 – 161 матрица

Для каждой группы были построены хордальные расширения и вычислены размеры максимальных клик:

Размер клики	<12	12-16	16-20	>20
Менее **500** вершин	**4**	**29**	**32**	**8**
500-1000 вершин	**1**	**41**	**87**	**16**
более **1000** вершин	**0**	**16**	**120**	**25**

Максимальные клики

- По оси абсцисс - фиксированные размеры
- По оси ординат - число графов, размер максимальных клик которых не превосходит это фиксированное число:

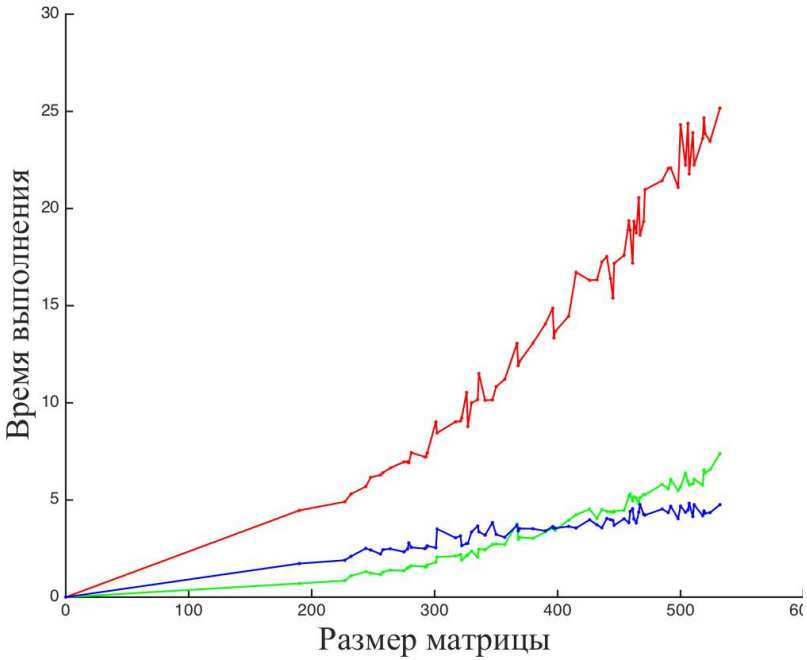


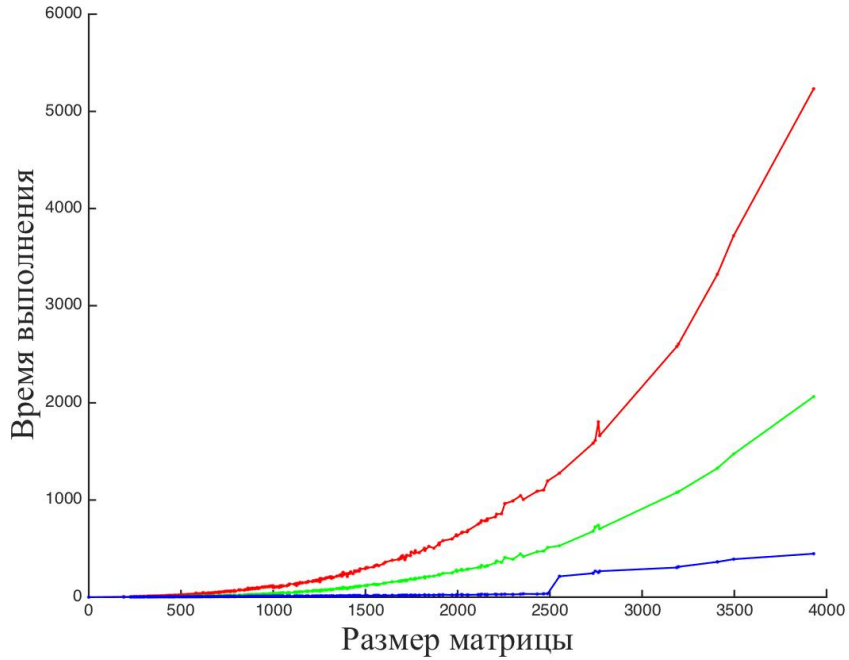
Число вершин графа

- не более 500
- от 500 до 1000
- свыше 1000

Время работы алгоритмов

После вычисления максимальных клик в графах сравнивались зависимости времени работы алгоритмов *SDP*, *Chordal SDP*, *MC* от размеров матриц.





Время испонения SDP, Chordal SDP, Monte-Carlo

Результаты первого эксперимента

Сравнение времени и точности работы алгоритмов на выборке *Protein Database Bank* (379 матриц):

- *Chordal SDP* – быстрее и точнее обычного *SDP* (быстрее, в среднем, в 3.23 раза).
- *Chordal SDP* на 11% матриц быстрее *MC*. Метод быстрее на матрицах размера 200-300.
- *Chordal SDP* точнее *MC* более, чем на 5%, на 3% матриц, на которых он работает быстрее.

Эксперимент со случайным прореживанием

Описание

- Рассматривались $G(n, p_1) : p_1 = \frac{2}{n}, \frac{3}{n}, \frac{4}{n}, \frac{5}{n}, \frac{6}{n}$
- В результате случайного "прореживания" получался $G(n, p_2) : p_2 = \frac{1}{2n}, \frac{1}{4n}, \frac{1}{6n}, \frac{1}{8n}$
- $n = 150, 160, \ldots, 200$

Результаты

- При $p_2 = \frac{1}{2n}$ уже наблюдались блоки достаточно малого размера для быстрого и достаточно точного решения задачи.
- С уменьшением p_2 – рост числа блоков, уменьшение их размеров $\Rightarrow$ рост скорости решения задачи, падение точности.

Эксперимент со случайным прореживанием

p_1	$\frac{2}{n}$				$\frac{3}{n}$			
p_2	$\frac{1}{2n}$	$\frac{1}{4n}$	$\frac{1}{6n}$	$\frac{1}{8n}$	$\frac{1}{2n}$	$\frac{1}{4n}$	$\frac{1}{6n}$	$\frac{1}{8n}$
Точность, %	91	83,7	82,1	80,4	90,5	82,6	81,5	78,9
$\frac{t_{\text{прореженное}}}{t_{\text{без прореживания}}}$	0,2	0,18	0,18	0,17	0,22	0,19	0,18	0,18

- **В ячейке "Время" указано отношение времени решения прореженной задачи к времени решения исходной**

Выводы

- Прореживание до значения p_2 чуть меньшего $\frac{1}{n}$ эффективно
- Прореживание до меньших значений p_2 влечет несущественное улучшение скорости в сравнении с существенными потерями в качестве решения

Заключение

- Проанализированы алгоритмы *SDP*, *SDP + Chordal*, *MC* и метод случайного "прореживания" с точки зрения скорости и точности их работы.
- Продемонстрирована эффективность предложенных методов при решении оптимизационных задач, в частности, задачи прогнозирования структур белков.

Публикации

Подкопаев А. С., Карасиков М. Е., Максимов Ю. В. Прогнозирование структур белков методами полуопределенного программирования // «Труды МФТИ» Т.7, №4 (28) (2015) С. 66–73

Литература

Хордальное разложение

- E. de Klerk, Exploiting special structure in semidefinite programming: A survey of theory and applications // European Journal of Operational Research Vol. 201. 2010 – P. 1 - 10
- R. Grone, C. R. Johnson, E. M. Sa, H. Wolkowicz. Positive definite completions of partial hermitian matrices // Linear algebra and its applications. Vol. 58. 1984. – P. 109–124.

Эффективный поиск максимальных клик

- D. Rose, G. Lueker, R.E. Tarjan, Algorithmic aspects of vertex elimination on graphs // SIAM Journal on Computing. 1976. – P. 266–283.

Случайные графы

- Райгородский А. М. Модели случайных графов // МЦНМО, 2011 — 136 с.