Методы повышения устойчивости эволюционного поиска больших графов

Логойда Роман Васильевич

Специальность: «Искусственный интеллект и машинное обучение»

Научный руководитель: к.т.н. Никитин Николай Олегович

Верхнеуровневая задача

- Ориентированный граф без петель и весов
- Граф может быть как плотным, так и разреженным
- Размерность графа неизвестна

Задача: найти граф решая задачу оптимизации по заданным метрикам

Проблематика

Чем больше размерность графа:

- Увеличивается время вычисления
- Увеличивается требуемое место под хранение в памяти
- Увеличивается пространство поиска
  - Проблема изоморфных графов

И это при прямом кодировании графа из n вершин до $n^2 - n$ ребёр

Проблематика

Рассмотрим матрицу смежностей (как вариант прямого кодирования) порядка n. Вариантов для поиска: 2^{n^2-n}

Исключим изоморфные графы: A000273

Оценка снизу: $\frac{2^{n^2-n}}{n!}$

https://oeis.org/A000273

Прикладное значение

Проект **GOLEM** от ITMO AIM.club

- Поиск графов по заданным метрикам
- Используется прямое кодирование (граф как ссылочная структура)
- Поиск больших графов (в рамках задачи ≥ 40) затрудняется деградацией
  - Рост размерности без улучшения метрики
  - Увеличение времени вычисления

Мотивировка

- Пересмотреть прямое кодирование
  - Использовать заданные метрики
  - Непрямое кодирование
- Использовать информацию о Fitness landscape
- Рассмотреть подзадачу для DAG (AutoML, NAS, BN, ...)

Идея 1: вектор инвариантов

Неспецифичные метрики на графах, как правило, используются информацию об инвариантах:

- (n вершин, m рёбер)
- Вектор степеней вершин
- Собственные числа, определитель матрица смежностей

Идея 1: вектор инвариантов

Плюсы:

- Уменьшенная размерность решения
- Элементы кодировки интерпретируемы

Минусы:

- Нужно проверять допустимость такого вектора
- Сильная привязка к искомым метрикам

Вывод:
В рамках общей задачи теряется гибкость фреймворка GOLEM.
Более не рассматривается

Идея 2: матрица смежностей v.2

Можно модифицировать матрицу смежностей:

- Задать значение матрицы через формулу с параметром p

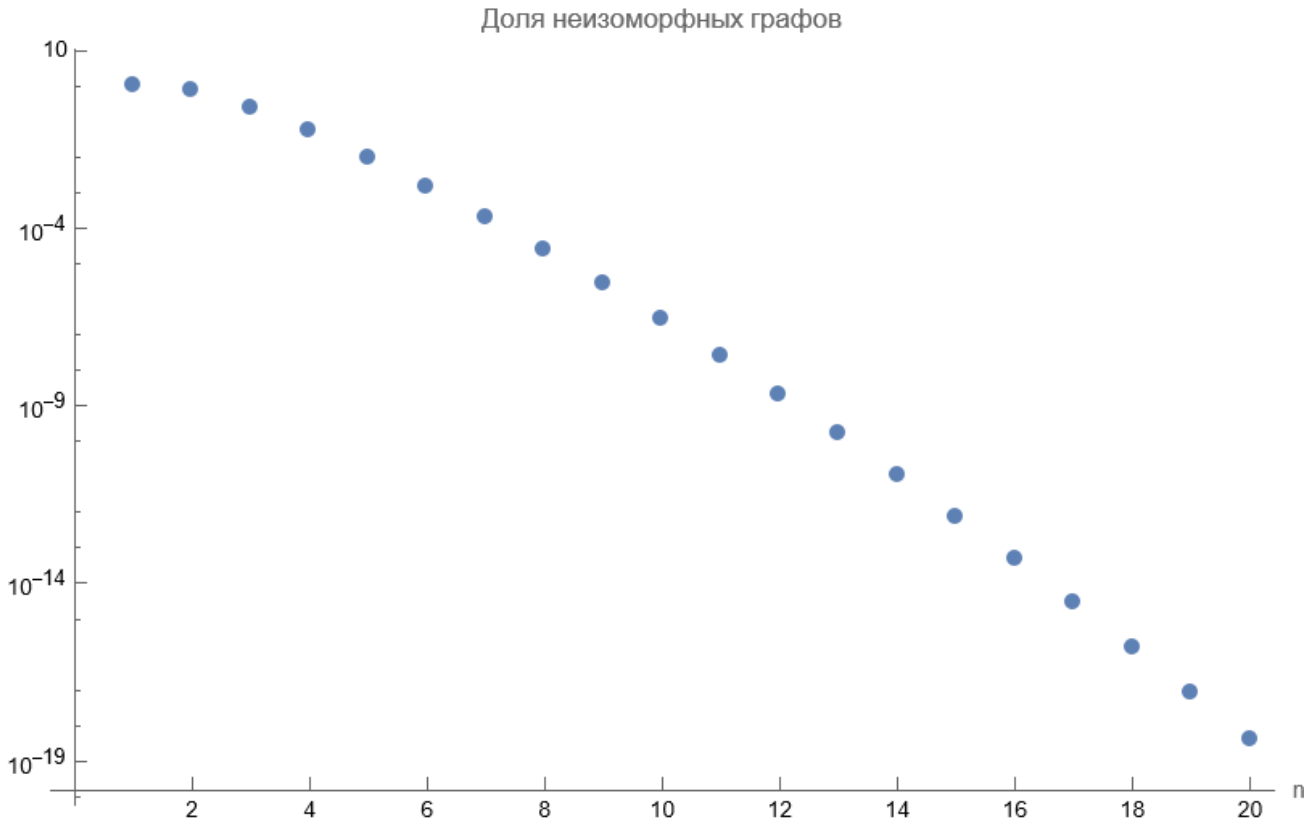
$$a_{i,j} = f(i, j, p)$$

- Задать значение матрицы через вероятностное распределение
- Матрица как блочная матрица с эволюцией блоков

http://dx.doi.org/10.7148/2011-0410-0416 (2011)

Идея 2: матрица смежностей v.2

Можно оценить долю неизоморфных графов: $\frac{1}{n!}$ (см. Проблематика)



Идея 2: матрица смежностей v.2

Плюсы:

- Уменьшенное хранение в памяти

Минусы:

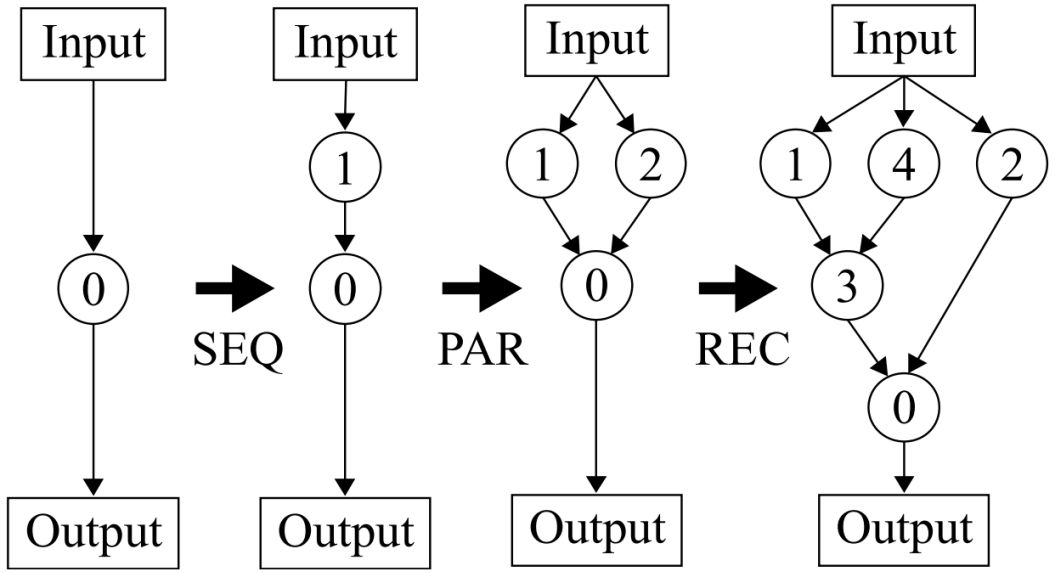
- Потеря в точности
- Остаются остальные недостатки м. с.

Вывод:

Не стоит для дальнейшего рассмотрения

Идея 3: непрямое кодирование

Кодирование фенотип = генотип в случае графов не использует накапливаемую информацию о признаках графа.
Предлагается использовать клеточное кодирование



Размер кодировки в худшем случае: $|E|$(Список рёбер)

Идея 3: клеточное кодирование

Оригинальный подход имеет ряд недостатков:

- Операторы не отображают высокоуровневые признаки графа
- Возникновение интронов
- Bloating

Идея 3: клеточное кодирование

Ряд работ, где использование непрямого кодирования используется для накапливания полезной информации в индивидах:

- **XC-NAS (2023)** [https://doi.org/10.48550/arXiv.2312.07760]
- **Simple Genetic Operators are Universal Approximators of Probability Distributions (2022**) [https://doi.org/10.48550/arXiv.2202.09679]
- **A Flexible Variable-length Particle Swarm Optimization Approach to Convolutional Neural Network Architecture Design (2021)** [https://ieeexplore.ieee.org/document/9504716]
- **Evolutionary NAS with Gene Expression Programming of Cellular Encoding(2020)** [https://doi.org/10.48550/arXiv.2005.13110]

Идея 3: клеточное кодирование

Предложение:

- Добавить высокоуровневые признаки – граф как конструктор
  - Добавить цикл
  - Добавить регулярный граф
  - Задать степень вершине
  - …
- Без операторов удаления – исключение интронов
- Логическая оптимизация по шаблонам
- Генетические операторы:
  - Мутация операторов
  - Скрещивание с проверкой допустимости

Идея 3: клеточное кодирование

Задача:

- NSGA2
- Пространство поиска – 80
- Размерность целевого графа – 60
- Начальная популяция – 20
- 10 независимых запусков

framework		deap_de		deap_ce		golem_de
dominance	deap_ce	golem_de	deap_de	golem_de	deap_de	deap_ce
gen						
10	17	14	0	1	0	4
20	21	25	0	2	0	7
30	27	31	4	5	0	5
40	21	33	9	15	0	13
50	35	47	17	21	0	16
60	37	46	25	27	5	29
70	42	35	36	34	52	41
80	46	54	42	39	92	49
90	51	64	78	45	130	56
100	49	67	94	44	140	59
120	45	33	114	53	153	68
140	29	20	121	49	180	72
160	18	9	145	42	180	74
180	14	8	152	44	180	67
200	12	10	156	41	180	71

медленный старт непрямой кодировки и GOLEM (накопление размерности)

на больших итерациях непрямая кодировка имеет больше недоминируемых решений чем прямая

Идея 3: клеточное кодирование

Плюсы:
- Использование признаков графов обеспечивает лучшую сходимость чем при прямом кодировании
- Выигрыш по расходу памяти

Минусы:
- Нельзя просто так задать операторы, нужны проверки
- *Предположение* – выигрыш в пространстве поиска с некоторой размерности нивелируется (ожидается примерно с ~300 вершин)

Вывод:
Относительно GOLEM не удалось достичь успеха, но относительно прямого кодирования – да

Идея 4: графовые эмбеддинги

Перед эволюционным поиском обучается автоэнкодер:

$$enc: ind \rightarrow emb$$

$$dec: emb \rightarrow metric$$

POC: обучение автоэнкодера заняло больше времени, чем достижение GOLEM приемлемых результатов (<= 100 вершин)

https://doi.org/10.48550/arXiv.2008.12646

Дальнейшие действия

- Использовать информацию о Fitness landscape
- Исследовать применимость графовых эмбеддингов

Спасибо за внимание