

Адаптивный рандомизированный алгоритм выделения сообществ в графах

Тимофей Проданов

timofey.prodanov@gmail.com

Санкт-Петербургский Государственный Университет
Кафедра Информатики

Научный руководитель: д.ф.-м.н., проф. Границин О. Н.
Рецензент: Ерофеева В. А.

Санкт-Петербург
2015

1998 г. Воттс, Строгатц — начало изучения сложных сетей¹

Графы с неправильной, сложной структурой

Применение:

- эпидемиология
- биоинформатика
- поиск преступников
- социология
- изучение структуры и топологии интернета

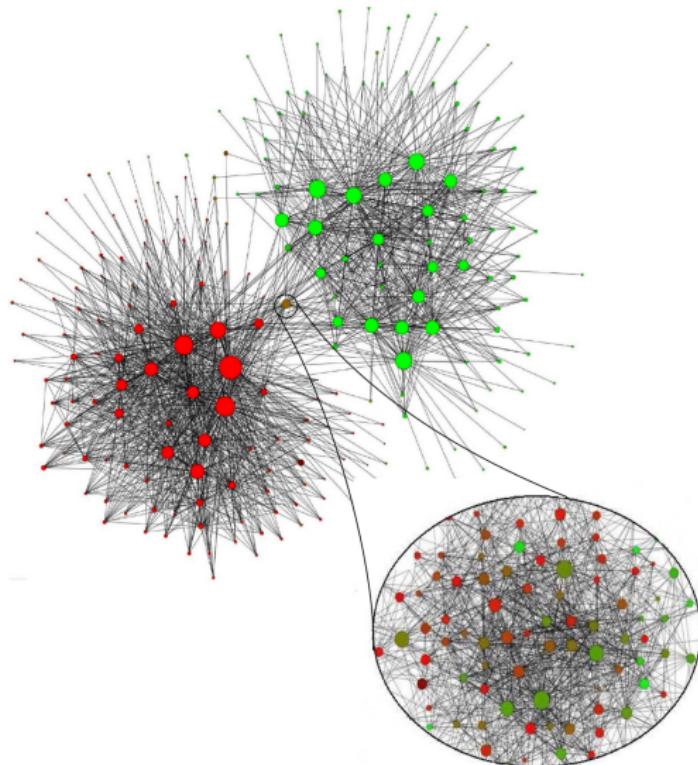
Усложнён анализ теорией графов или статистический анализ

Сообщества — тесно связанные группы узлов

Поиск таких групп — выделение сообществ, кластеризация

¹Watts and Strogatz. [Collective dynamics of “small-world” networks.](#)

Пример разбиения на сообщества



Blondel et al. [Fast unfolding of communities in large networks](#).

Алгоритмы выделения сообществ

2004 г. Модулярность² Q

2010 г. Рандомизированный жадный алгоритм³ RG :

- k случайных сообществ и их соседи
- Лучшая пара соседей

2012 г. Схема кластеризации основных групп графа⁴ $CGGC$:

1. *s начальных алгоритмов*
2. *Финальный алгоритм* распределяет неопределившиеся узлы

RG как начальный и финальный алгоритм

²Newman and Girvan. [Finding and evaluating community structure in networks.](#)

³Ovelgonne and Geyer-Schulz. [Cluster cores and modularity maximization.](#)

⁴Ovelgonne and Geyer-Schulz. [A comparison of agglomerative hierarchical algorithms for modularity clustering.](#)

Постановка задачи адаптации параметров

Качество работы RG и $CGGC$ зависит от их параметров
Для каждого графа — свои хорошие параметры

Адаптивные алгоритмы, приспосабливающиеся к входным
данным

Цель:

Хорошие результаты на большем количестве графов

Применение стохастического градиентного спуска *SPSA* к *RG* и *CGGC*

Стохастический градиентный спуск *SPSA*:

- Разбиение алгоритма на $2n$ шагов
- Каждый шаг новый параметр
- Новые значения параметра подбираются в зависимости от предыдущих шагов

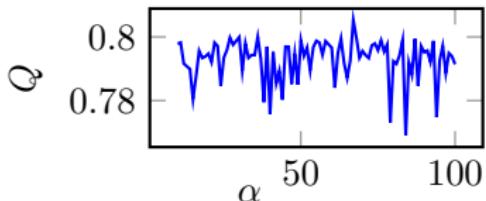
Адаптивный рандомизированный жадный алгоритм *ARG*:

- Шаги по σ итераций
- После окончания шага k меняется
- Функция качества $f(\mu, k) = -\alpha(\ln \mu_n^- - \beta \ln k_n^-)$,
 μ — медиана прироста модулярности
- \hat{k}_{n-1} — текущая оценка, её возмущение:
 $k_n^- = \max\{\hat{k}_{n-1} - d, 1\}$ и $k_n^+ = \hat{k}_{n-1} + d$
- Следующая оценка:

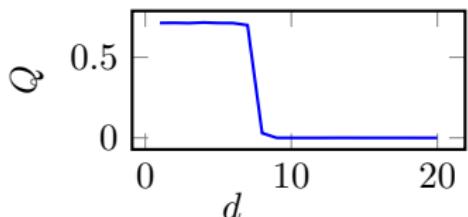
$$\hat{k}_n \leftarrow \max \left\{ 1, \left[\hat{k}_{n-1} - \frac{f_n^+ - f_n^-}{k_n^+ - k_n^-} \right] \right\}$$

Исследование работоспособности при разных параметрах

- Чувствительность α

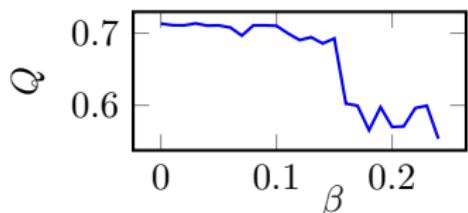


- Размер возмущения d

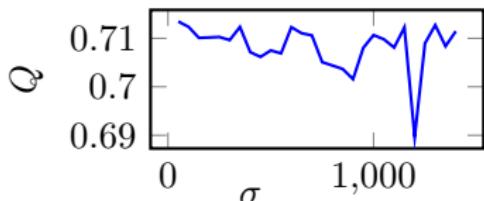


- Значимость времени β :

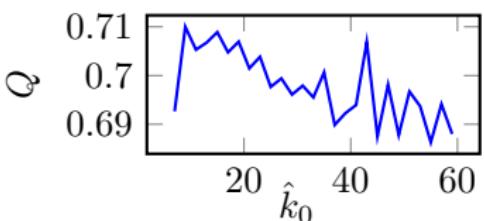
Модулярность:



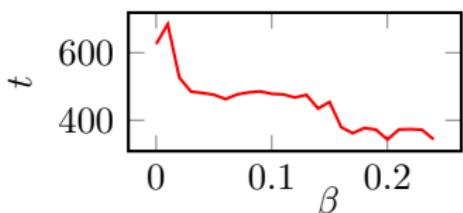
- Количество итераций в шаге σ



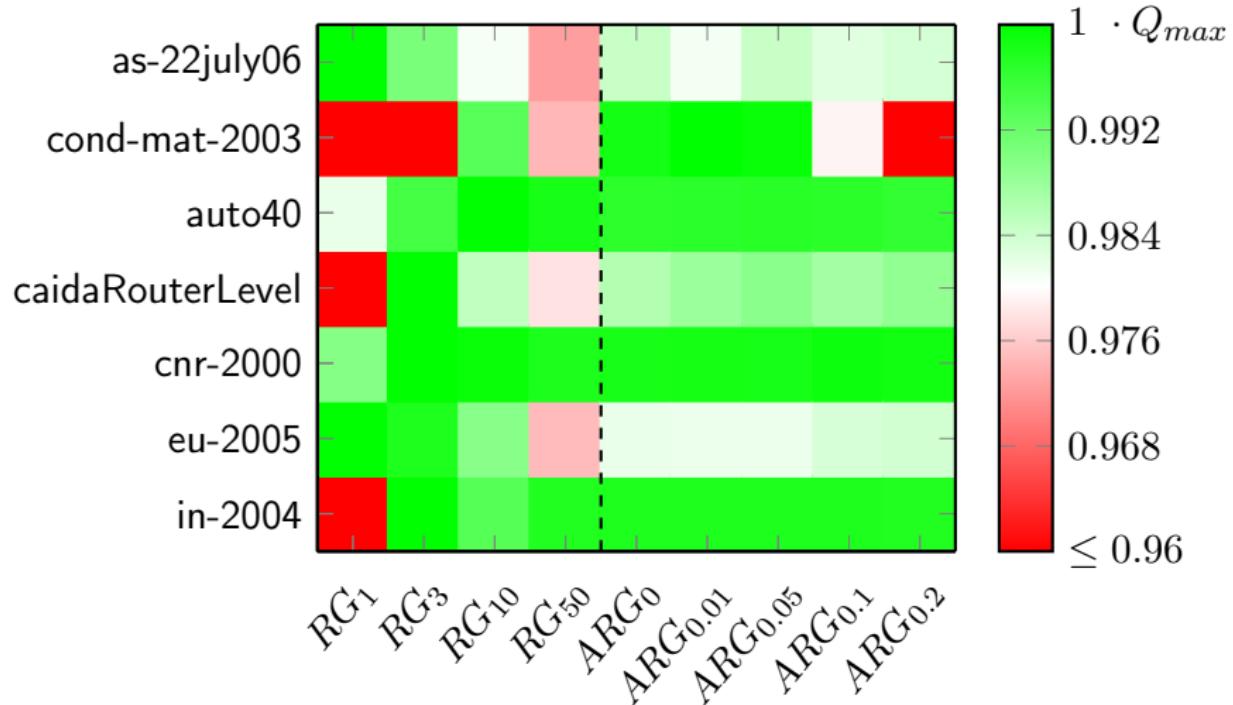
- Начальная оценка \hat{k}_0



Время:



Сравнение RG и ARG



$ACGGC$ схожа с $CGGC$:

1. l начальных алгоритмов:

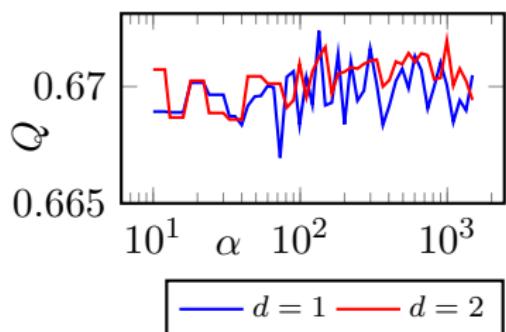
- RG_k с разными k
- k подстраиваются

2. Из l начальных разбиений выбирается несколько лучших

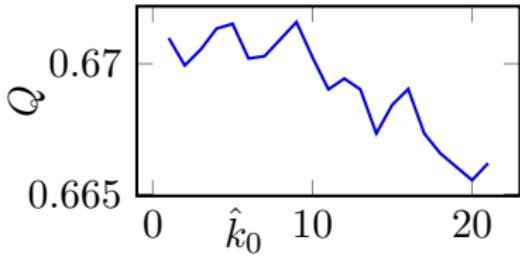
3. Неопределившиеся узлы разбиваются по сообществам
финальный алгоритм

Параметры ACGGC

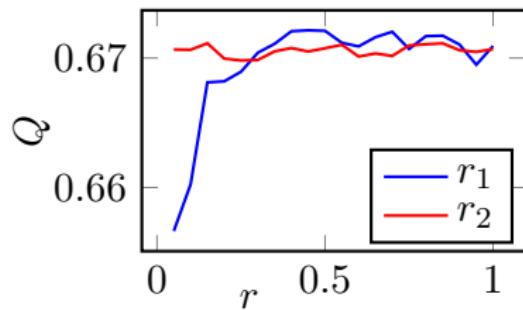
- Размер возмущения d ,
Чувствительность α



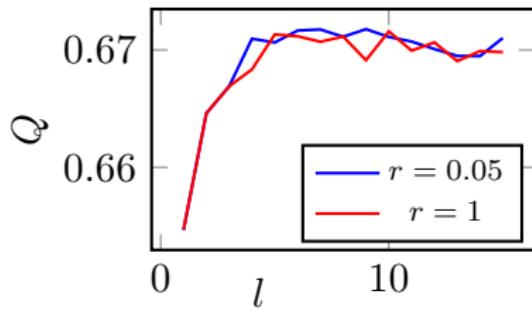
- Начальная оценка \hat{k}_0



- Доля хороших начальных
разбиений r

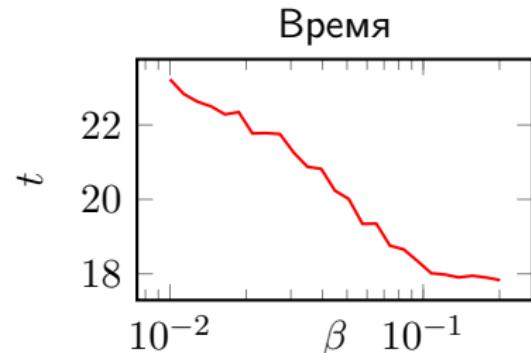
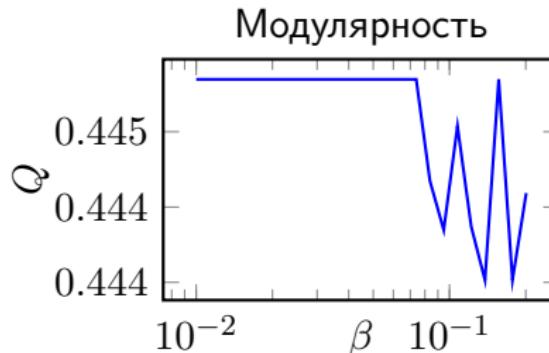


- Количество шагов l

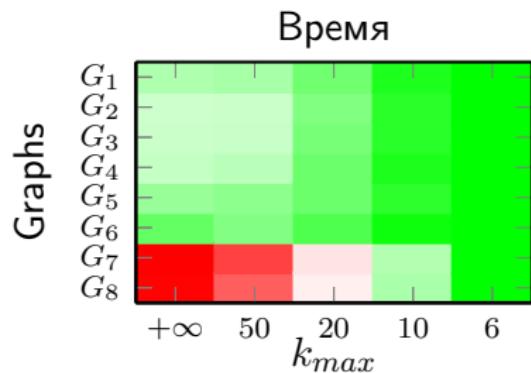
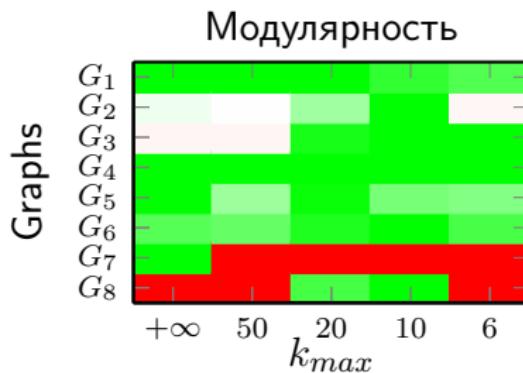


Механизмы снижения времени работы

- Увеличение значимости времени:



- Установка ограничения на максимальную оценку k_{max} :



Сравнение $CGGC$ и $ACGGC$

	$ACGGC^I$	$ACGGC^{II}$	$CGGC_{10}^{10}$	$CGGC_3^{10}$	$CGGC_{10}^3$
karate	0.417242	0.417406	0.415598	0.396532	0.405243
dolphins	0.524109	0.523338	0.521399	0.523338	0.522428
chesapeake	0.262439	0.262439	0.262439	0.262439	0.262370
adjnoun	0.299704	0.299197	0.295015	0.292703	0.290638
polbooks	0.527237	0.527237	0.527237	0.526938	0.526784
football	0.603324	0.604266	0.604266	0.599537	0.599026
celegans	0.439604	0.438584	0.435819	0.436066	0.432261
jazz	0.444739	0.444848	0.444871	0.444206	0.444206
netscience	0.907229	0.835267	0.724015	0.708812	0.331957
email	0.573333	0.573409	0.571018	0.572667	0.567423
polblogs	0.424107	0.423208	0.422901	0.421361	0.390395
pgpGiantCompo	0.883115	0.883085	0.882237	0.882532	0.880340
as-22july06	0.671249	0.670677	0.666766	0.669847	0.665260
cond-mat-2003	0.744533	0.750367	0.751109	0.708775	0.413719
caidaRouterLevel	0.846312	0.855651	0.851622	0.858955	0.843835
cnr-2000	0.912762	0.912783	0.912500	0.912777	0.912496
eu-2005	0.938292	0.936984	0.935510	0.936515	0.936420
in-2004	0.979844	0.979771	0.979883		

В рамках работы

- Проанализированы современные методы выделения сообществ
- Предложен ARG
 - Исследованы параметры
 - Проведено сравнение с RG
- Представлена $ACGGC$
 - Проанализированы параметры
 - Описаны механизмы снижения времени работы
 - Предложена итеративная версия
 - Исследована комбинация из $ACGGCi$ и $CGGCI$
 - Проведено сравнение с $CGGC$