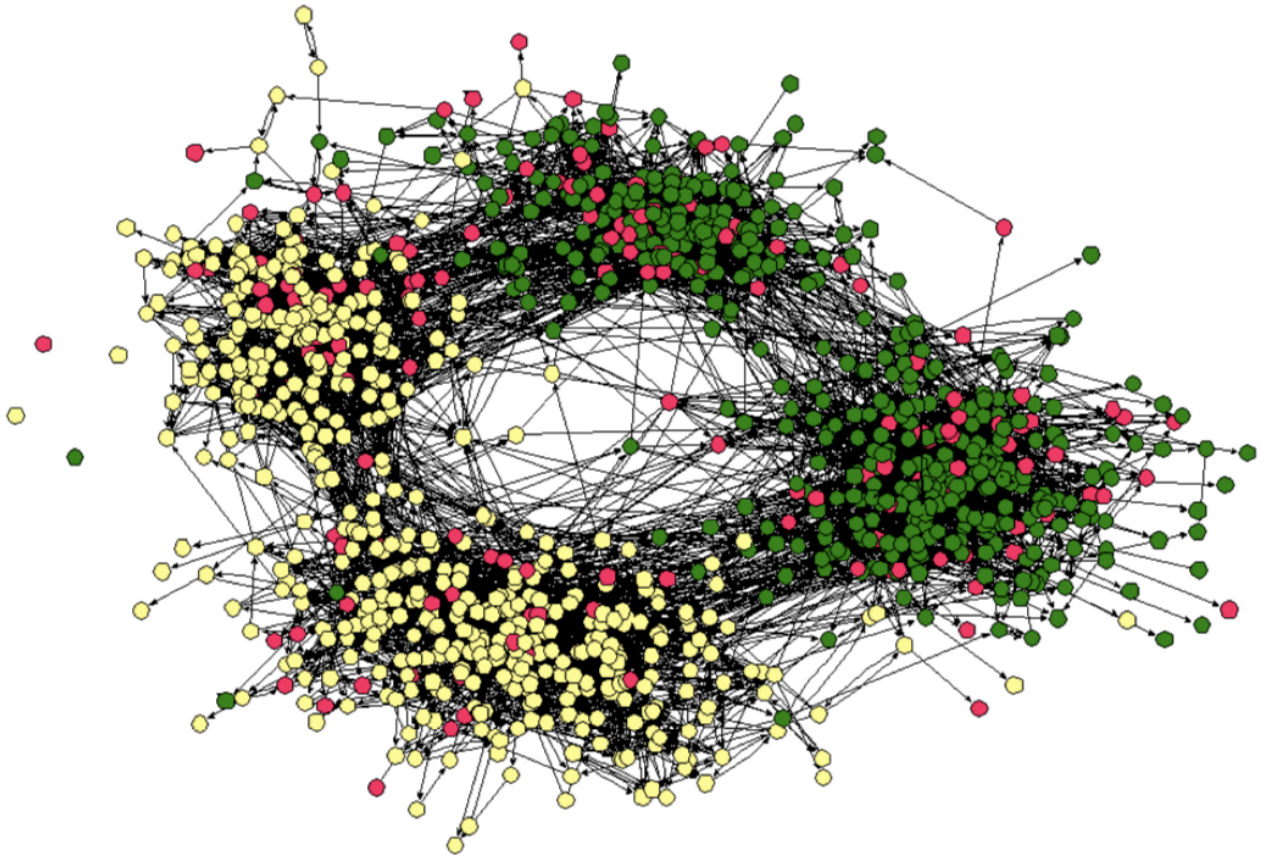
Улучшенный алгоритм для решения задачи Maximum Happy Vertices на деревьях

Беличенко Дмитрий 17.Б10-мм

Научный Руководитель: Сагунов Данил Георгиевич (JetBrains, ПОМИ РАН, СПбГУ)

Научный консультант: к. ф-м. н. Близнец Иван Анатольевич (JetBrains, ПОМИ РАН, СПбГУ)

Вступление



Социальные связи порождают сложные задачи на графах, решение которых необходимо для моделирования взаимодействия различных групп людей.

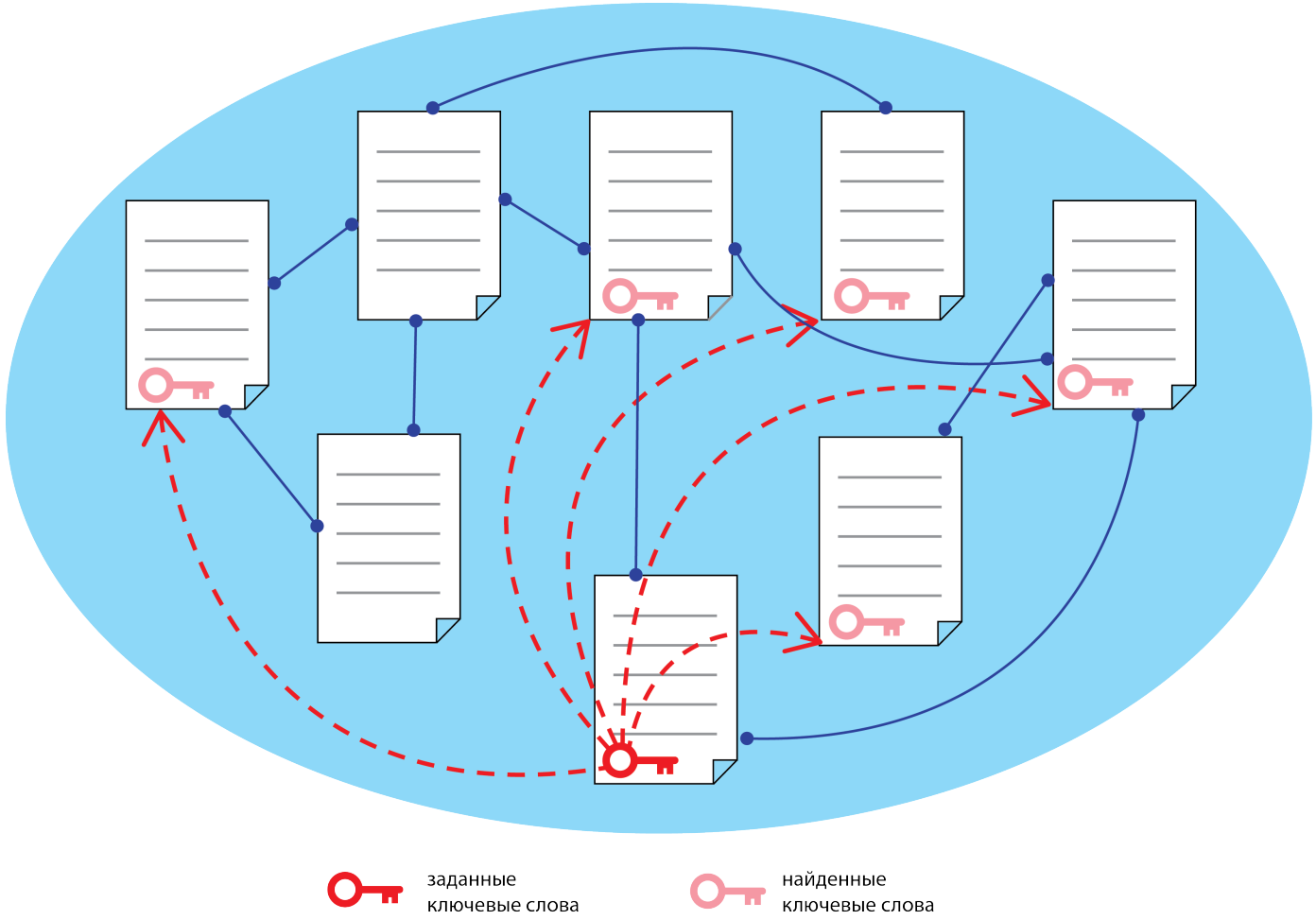
Рис. 1 Сеть дружбы американских школьников на основе данных из работы Дж. Муди

Описание задачи Maximum Happy Vertices

- Дан граф, часть вершин которого покрашена
  Необходимо выбрать цвета для оставшихся вершин, так чтобы максимизировать число счастливых вершин
- Вершина называется счастливой, если все ее соседи имеют тот же цвет, что и она сама
- Ежегодно выходит ряд статей, посвященных частным случаям MHV

Применение MHV к гомофилии

- 27700 статей (вершин)
- 352017 ссылок (ребер)
- 1214 с ключевыми словами
- Выбрав MHV как метрику удалось найти ключевые слова 14 123 статей



Древесная ширина социальных сетей

- Несмотря на огромное количество пользователей, некоторые социальные сети имеют сравнительно небольшую древесную ширину

type	Dataset name	nodes	edges	Lower width	Upper width
social	Facebook	4 039	88 234	142	247
	Enron	36 692	183 831	257	1 989
	WikiTalk	2 394 385	4 659 565	1 113	12 843
	CitHepth	34 546	420 877	469	9 498
	Stack-TCS	25 232	69 026	143	717
	Stack-Math	1 132 468	2 853 815	850	11 100
	LiveJournal	3 997 962	34 681 189	360	919 532*

Данные из исследования Silviu Maniu и Pierre Senellart (январь 2019)

MHV на деревьях

- MHV — NP трудная задача и решается для частных случаев
- MHV может быть применимо к древесной декомпозиции социальных графов с ограниченной древесной шириной
- Для того, чтобы применить MHV к древесному разложению нужен быстрый алгоритм на деревьях

Задачи

- Произвести анализ существующих решений задачи MHV на деревьях
- Улучшить алгоритм решения MHV на деревьях
- Проверить, влечет ли улучшение теоретической оценки времени работы улучшение производительности на практике

Известные результаты:

N. R. Aravind, Subrahmanyam Kalyanasundaram Anjeneya Swami Kare, Lauri Juho. Algorithms and hardness results for happy coloring problems. Springer (IWOCA 2016)
(Алгоритм на деревьях)

Angsheng Li Peng Zhang. Algorithmic Aspects of Homophyly of Networks. (2018)

Agrawal A. On the parameterized complexity of happy vertex coloring.
Intenational Workshop on Combinatorial Algorithms. pp. 103–115. Springer (2017)

Zhang P. Xu Y. Jiang T. Li A. Lin G. Miyano E. Improved approximation algorithms for the maximum happy vertices and edges problems.
Algorithmica 80(5), 1412–1438 (2018)

Алгоритм на деревьях за $O(nk\log(k))$

- Метод динамического программирования
- Для каждого поддерева было подсчитано две функции
  - Максимальное количество счастливых вершин, при условии, что корень поддерева счастливый и он имеет фиксированный цвет
  - Максимальное количество счастливых вершин, при условии, что корень поддерева несчастливый и он имеет фиксированный цвет
- В силу такого выбора функций их пересчет невозможен никак, кроме как за $O(nk\log(k))$, так как для пересчета второй функции на для каждого поддерева требуется сортировка

Идея альтернативного алгоритма

- Метод динамического программирования
- Избавимся от медленной функции, для несчастливых вершин
- Для каждого поддерева вычислим две функции:
  - F(v,c) — Максимальное количество счастливых вершин, при условии, что корень поддерева — v счастливый и он имеет цвет c
  - G(v,c) — Максимальное количество счастливых вершин, при условии, что корень поддерева — v любой (счастливый или несчастливый) и он имеет цвет c

Ход решения

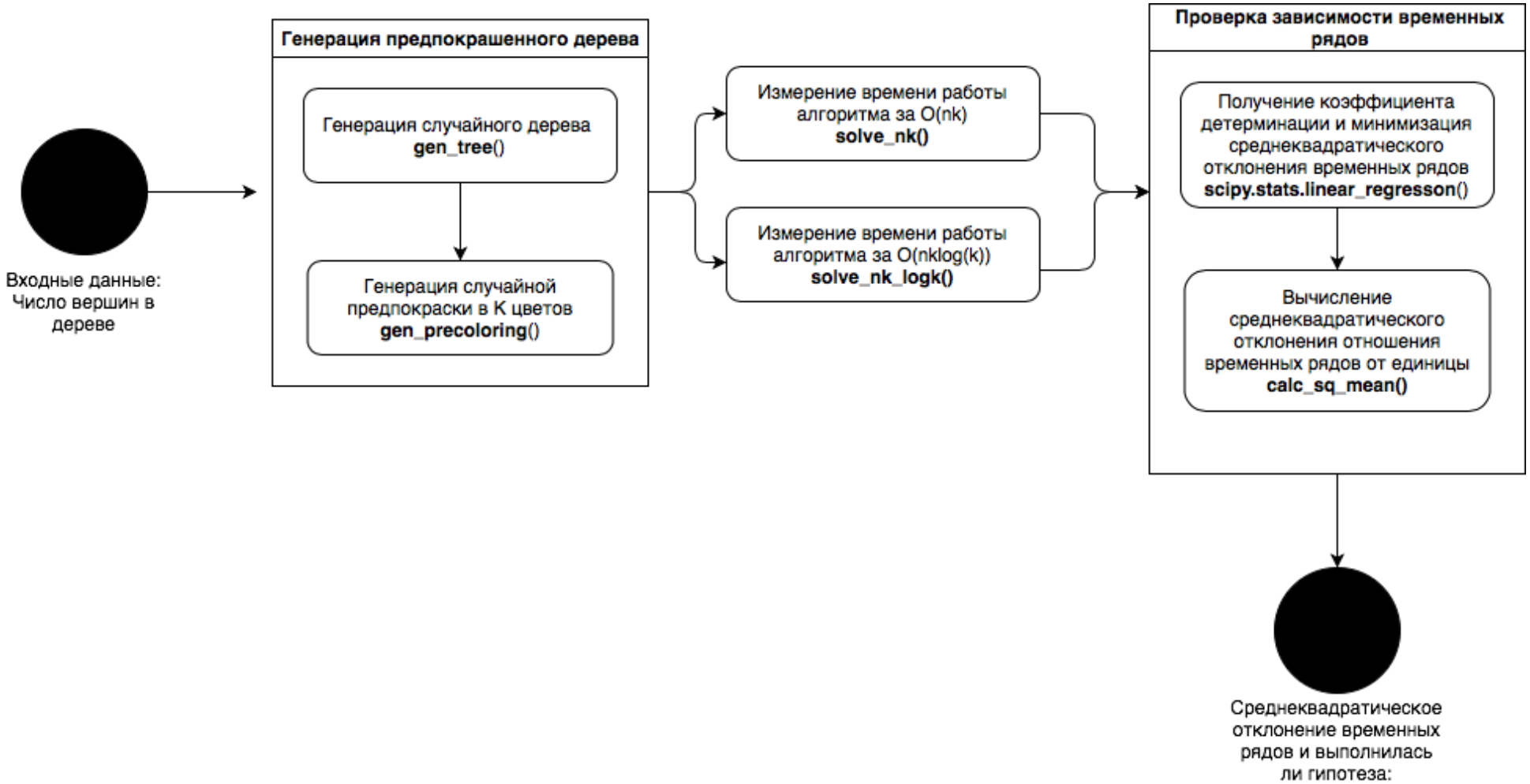
- Подсчет значений осуществляется в процессе Post-order обхода дерева, то есть на момент пересчета значений в каждой вершине значения в ее детях уже посчитаны
- Вместо сортировки на каждом этапе достаточно вычислять максимальные значения функции F(v,c), что возможно в силу выбранного порядка обхода дерева
- При помощи данной оптимизации удалось получить оценку на суммарное время работы $O(nk)$

Time	Reference	Year
$O(nk\log(k))$	Aravind, Kalyanasundaram, Swami Kare	2016
$O(nk)$	Belichenko	2019

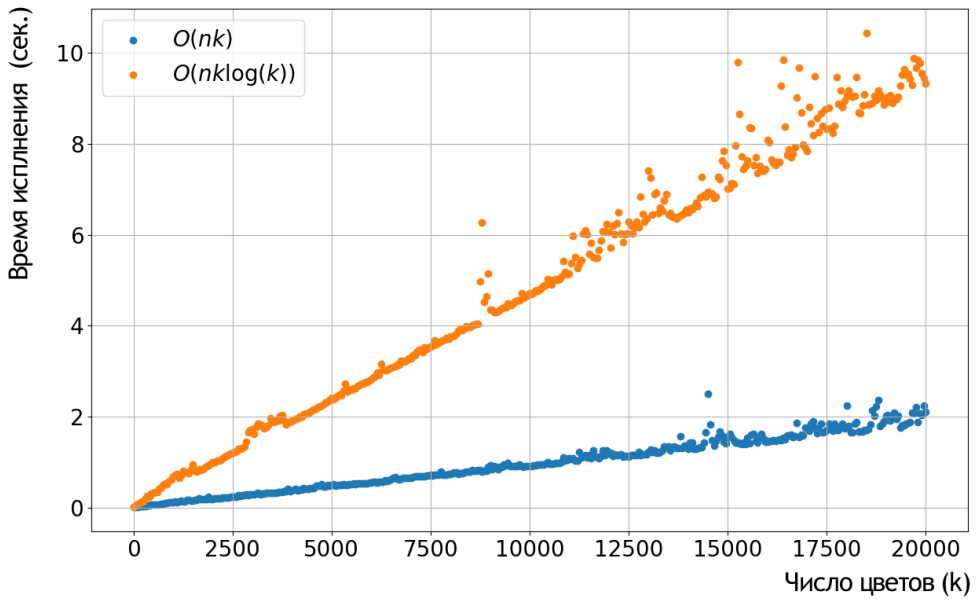
Состоятельность алгоритма

- В ходе работы были доказаны две теоремы:
  - Алгоритм описанный выше корректно вычисляет значения функций F(v,c) и G(v,c), согласно их определениям, для любых возможных значений (v,c)
  - Алгоритм описанный выше корректно вычисляет значения функций F(v,c) и G(v,c) для всех возможных значений (v,c) за O(nk)

Архитектура тестирующей системы



Результаты эксперимента



Гипотеза: $B_k = CA_k \log(k) + \alpha$

Отношение временных рядов: $T_k = \dfrac{B_k}{CA_k \log(k)}$

$n = 25000$

$R^2 = 0.962$

$C = 1.681$

$\sigma = 0.027$

Результаты экспериментов

n	R^2	σ
25000	0,962	0,027
20000	0,962	0,029
15000	0,958	0,031
10000	0,967	0,033
5000	0,959	0,039

Итоговые результаты

- Были проанализированы существующие решения задачи MHV на деревьях
- Был разработан алгоритм с асимптотической сложностью O(nk)
- Было проверено, что улучшение теоретической оценки влечет улучшение производительности на практике