



темпоральных систем на основе реляционной базы данных [Электронный ресурс] / Режим доступа: <http://technomag.edu.ru/doc/441884.html> (дата обращения 19.11.2019).

7 Балдин А.В., Тоноян С.А., Елисеев Д.В., Анализ избыточности хранения темпоральных данных средствами реляционных СУБД. [Электронный ресурс] / Режим доступа: <http://engjournal.ru/catalog/it/hidden/1273.html> (дата обращения 19.11.2019).

8 Котиков П.Е., Нечай А.А., Зацепин В.А. Темпоральные базы данных и язык запросов // Наука и современность. – 2014. – №2 – С.60-68.

9 Котиков П.Е. Варианты построения темпоральных баз данных в геоинформационных системах // Научный аспект. – 2014. - №4. - С.118-120.

10 Проектирование информационной обучающей системы выбор СУБД и способа доступа к данным / О.К. Головнин, Д.Е. Егоров // Информационные технологии поддержки принятия решений: материалы конф. – Уфа-Ставрополь: УГАГУ, СКФУ, 2018. – С.116-121.

11 Храмов В.Ю., Ханов Э.Б., Мишуков А.Н. Автоматизированная система ведения темпоральной базы данных системы поддержки принятия решений по идентификации беспилотных летательных аппаратов // Энергия – XXI век. – 2018. – №3 (103). – С.85-89.

12 Матчин В.Т. Обновление темпоральной базы данных в транспортной сфере // Наука и технологии железных дорог. – 2017. - №2. - С.39-46.

13 Gupta P.K, Rishi R., Biswas R. A comparative analysis of temporal data models // International Journal of Advanced Computational Engineering and Networking, ISSN (p): 2320-2106, Volume- 1, Issue- 8, Oct-2013.

А.О. Пшеничных, Э.И. Ватутин

## ВЛИЯНИЕ ВЫБОРА НАЧАЛЬНОГО ЦВЕТА НА КАЧЕСТВО РЕШЕНИЙ ДЛЯ МЕТОДА ВЗВЕШЕННОГО СЛУЧАЙНОГО ПЕРЕБОРА ПРИ ПОИСКЕ РАСКРАСКИ ГРАФА

(Юго-Западный государственный университет, Курск)

**Аннотация.** В работе приводится описание эффекта выбора вероятности минимально допустимого или случайного цвета выбранной вершины на качество получаемых решений методом взвешенного случайного перебора поиска хроматического числа графа. Для разработанных программных реализаций приведены оценки временных затрат и скорости сходимости.

Существует большое количество практических задач, которые можно решить за полиномиальное время к задачам теории графов [1]. Одна из них – поиск оптимальной раскраски графа, которая может быть применена для разбиения графов, составления расписаний различного рода, компиляции программ, решения задач на базе латинских квадратов и пр.



Задача раскраски неориентированного графа  $G = \langle A, V \rangle$ , где  $A = \{a_1, a_2, \dots, a_N\}$  – множество вершин графа,  $N = |A|$  – число вершин,  $V = \{v_1, v_2, \dots, v_M\} \subseteq A \times A$  – множество рёбер,  $M = |V|$  – общее число рёбер, в минимальное количество цветов относится к классу  $NP$ -полных. В некоторых случаях граф может не быть полносвязным, что свидетельствует об отсутствии связей между некоторыми парами вершин, при этом граф характеризуется значением «плотности»  $d(G) = \frac{M}{N(N-1)}$  (данная характеристика является

важной в связи с тем, что качество работы эвристических методов зависит от области в некотором многомерном пространстве, одной из координат которого в задачах на графах является плотность  $d(G)$  этого графа [1, 2]). Необходимо найти такой набор цветов  $X = \{x(a_1), x(a_2), \dots, x(a_N)\}$  (раскраску) для каждой вершины графа  $a_i$ ,  $x(a_i) \in C$ , из множества цветов  $C = \{c_1, c_2, \dots, c_{\chi^*}\}$ , чтобы  $\chi^* = |C| \rightarrow \min$ , причём смежные вершины графа не могут быть раскрашены в одинаковые цвета:  $\forall v_i = (a_{i_1}, a_{i_2}) : x(a_{i_1}) \neq x(a_{i_2})$ . В случае нахождения оптимальной раскраски  $\chi^*(G) = \chi(G)$ , где  $\chi(G)$  – хроматическое число графа  $G$ , а в случае нахождения суб- или квазиоптимальной –  $\chi^*(G) \geq \chi(G)$ .

Для программной реализации данного метода необходимо вычислить минимально возможную мощность множества цветов  $|C|$ , в которые могут быть раскрашены вершины заданного графа без нарушения условий правильной раскраски, в этом случае такие цвета будем считать допустимыми (при нарушении условий поставленной задачи – недопустимыми). Для проверки работы метода берётся некая выборка из  $K$  графов  $\Lambda = \{G_1, G_2, \dots, G_K\}$  и оценивается усреднённое качество решения  $\bar{Q} = \frac{\sum_{i=1}^K Q(G_i)}{K}$ , где  $Q(G_i)$  – качество  $i$ -го решения и равно  $\chi(G_i)$ , обычно оно больше оптимума  $\bar{Q} > Q^*$ .

В простейшей реализации метода взвешенного случайного перебора вершины окрашиваются последовательно в том порядке, в каком они указаны в графе. В процессе разработки алгоритма было предложено реализовать ещё три различных способа обхода вершин неориентированного графа:

1. В случайном порядке: из множества ещё не окрашенных вершин  $\tilde{A} \subseteq A$  выбирается случайная  $a_i$ , ей присваивается минимально допустимый цвет  $c_j$ .
2. В порядке уменьшения степеней вершин: из множества ещё не окрашенных вершин  $\tilde{A}$  выбирается такая  $a_i$ , у которой больше остальных связей с другими в матрице смежности (наибольшее



возможное значение степени вершины  $\rho(a_i) \leq N-1$   $\rho(a_i) \rightarrow \max$ , ей присваивается минимально допустимый цвет  $c_j$ .

3. В порядке возрастания количества допустимых цветов вершины: из множества ещё не окрашенных вершин  $\tilde{A}$  выбирается такая  $a_i$ , у которой имеется наименьшая мощность множества допустимых цветов  $X(a_i) \subseteq X$ , где  $|X(a_i)| \rightarrow \min$ , ей присваивается минимально допустимый цвет  $c_j$ .

Ввиду того, что при переборе в порядке уменьшения степеней вершин или же возрастания количества допустимых цветов может возникать ситуация, при которой несколько вершин будут иметь равное значение степени либо количество допустимых цветов, было решено ввести дополнительный параметр метода:  $\eta$  – вероятность выбора минимально допустимого или случайного цвета для текущей вершины.

Зависимости средневыборочного хроматического числа  $\chi^*$  и времени получения решения  $t$  от параметра  $\eta$  приведены на рис. 1 и 2. Вычислительный эксперимент проводился для  $K = |\Lambda| = 250$  случайных графов выборки  $\Lambda = \{G_1, G_2, \dots, G_K\}$  графов с псевдослучайной структурой размером  $N = 40$  вершин и плотностью  $d(G) = 0,777$ , используя количество итераций  $C_{\max} = 1000$  (здесь WRS\_M0 – порядок перебора вершин по минимуму доступных цветов с назначением 1 цвета первой вершине; WRS\_Max01 – порядок перебора вершин по степеням вершин с последовательным назначением  $N$  цветов первой вершине; WRS\_Max11 – порядок перебора вершин по степеням вершин с последовательным назначением  $N$  цветов первой вершине и применением дополнительной сортировки (при одинаковых значениях степенях вершин будет выбрана та, которая имеет больше связей с уже окрашенными)).

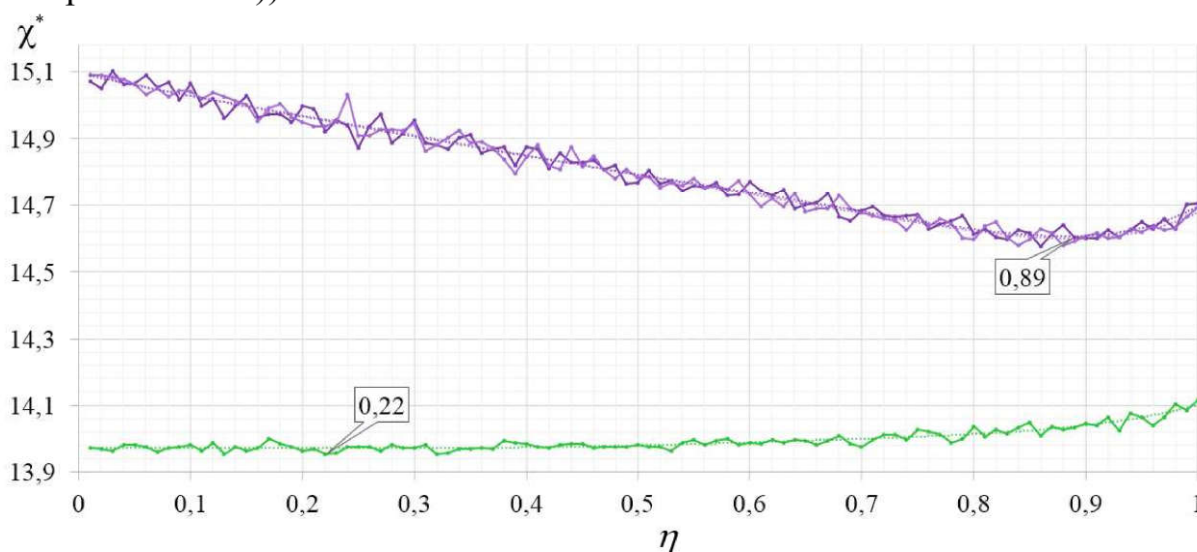


Рис. 1. Зависимость усреднённого хроматического от параметра  $\eta$



В ходе проверки получены оптимальные значения параметра для данных реализаций: для WRS\_Max01 и WRS\_Max11 равно  $\eta = 0,89$ , для WRS\_M0 случайная составляющая оказывает большее влияние на качество, чем жадная, т.о. за оптимальное значение принято  $\eta = 0,22$ . WRS\_Max01 и WRS\_Max11 практически не отличаются по качеству решения и по сравнению с WRS\_M0 дают ухудшение качества на  $\approx 4,6\%$  при полученных оптимальных значениях параметра. Для перебора по степеням вершин влияние параметра на качество решения является более выраженным, чем при переборе по минимуму допустимых цветов аналогично результатам работы [3], также выявлено, что для перебора по степеням вершин бассейн притяжения глобального оптимума параметра  $\eta$  для метода взвешенного случайного перебора не так сильно выражен, как для метода случайного перебора [3].

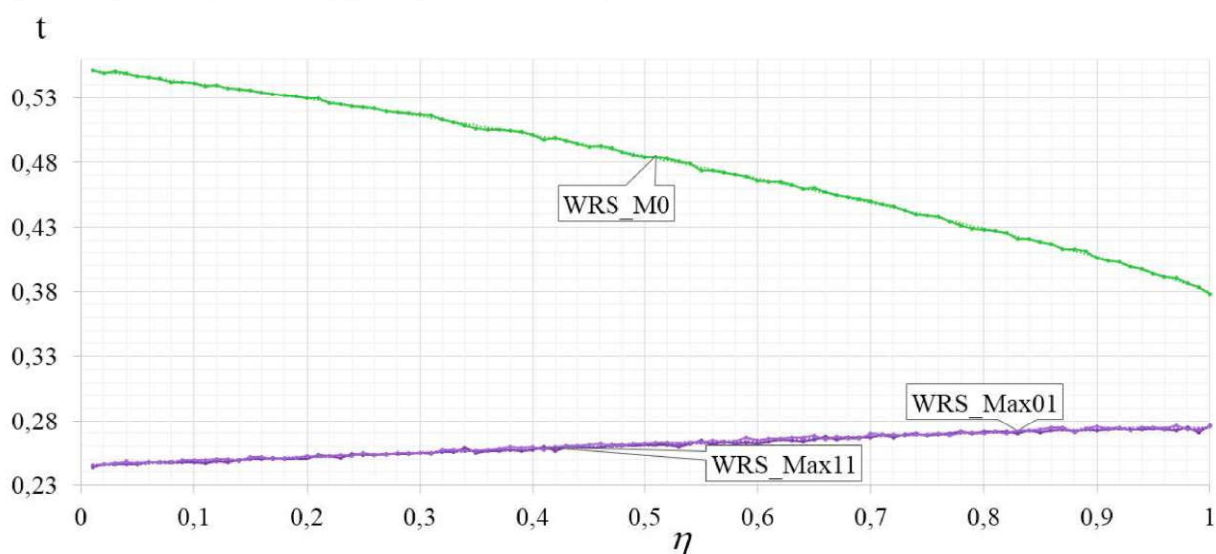


Рис. 2. Зависимость времени нахождения решения (в секундах) от параметра  $\eta$

Для перебора по минимуму цветов временные затраты снижаются в  $\approx 1,46$  раза при использовании жадного выбора по сравнению со случайным. Дополнительная сортировка при переборе по степеням вершин не оказывает влияния ни на качество получаемых решений, ни на время их генерации.

### Литература

1. Ватутин Э.И. Основы дискретной комбинаторной оптимизации. [Текст] / Ватутин Э.И., Титов В.С., Емельянов С.Г. – М.: Аргатак-Медиа, 2016. – 270 с.
2. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновлённые природой. [Текст] – М.: МГТУ им. Н.Э. Баумана, 2014. – 446 с.
3. Пшеничных А.О. О влиянии вероятности выбора минимально допустимого или случайного цвета для метода случайного перебора эвристической оценки хроматического числа графа [Текст] / Пшеничных А.О.,



Гвоздева С.Н., Панищев В.С., Вагутин Э.И // Интеллектуальные и информационные системы. – Тула: изд-во ТулГУ, 2019. – с. 59–63.

Цзе Дон, С.В. Востокин

## ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ПЛАТФОРМ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ EVEREST И MYBINDER

(Самарский университет)

На этапах исследования динамических систем применяются различные программные средства компьютерного моделирования. Этап анализа и разработки модели в форме, пригодной для интегрирования, реализуется в системах компьютерной алгебры (Maple, Mathematica, SymPy). Этап построения траекторий в фазовом пространстве выполняется с помощью пакетов численного интегрирования (Boost.Numeric.Odeint). Этап статистического анализа наблюдаемого поведения систем проводится в фреймворках анализа данных (Apache Spark). Этап представления результатов моделирования в удобном для анализа виде осуществляется при помощи пакетов визуализации (D3.js) [1].

Очевидно, что для снижения трудоемкости рабочего процесса моделирования динамических систем необходима комплексная автоматизация всех этапов исследования. Традиционно для этого использовались монолитные коммерческие системы и сервисы (Mathcad, SMath Studio). Однако в связи с интенсивным развитием web-технологий и технологий облачных вычислений стали активно развиваться новые подходы автоматизации рабочего процесса (workflow) численного моделирования (Project Jupyter, Everest Platform).

Ключевыми парадигмами проекта Jupyter являются клиент-серверная модель и графическая интерактивная среда в стиле REPL (read-eval-print loop), реализованные в форме web-интерфейса. Многооконный интерфейс JupyterLab позволяет работать сразу с несколькими языками программирования в дополнении к языку Python [2]. Публичные облачные сервисы (mybinder.org) автоматизируют развертывание экземпляров JupyterLab на бесплатных виртуальных машинах [3].

Облачная платформа в виде сервиса (PaaS) Everest [4] предназначена для композиции приложений, работающих на различных вычислительных ресурсах, с целью реализации единого рабочего процесса численного моделирования. В качестве аппаратных ресурсов приложений могут выступать физические рабочие станции, виртуальные машины, кластерные системы. При этом для ресурса не требуется выделенного IP-адреса и постоянного подключения. Ресурс может работать в конфигурации «за брэндмауэром». Платформа Everest является публично доступной.

Изложенное выше показывает, что существует техническая возможность совместного использования облачной платформы Everest и платформ проекта