

список популярных. Система оценки новостей будет основываться на элементе «Мне нравится». Чем больше людей будут оценивать новость, тем выше она будет находиться в списке популярных новостей.

### Список литературы

1. Электронный учебник HTML и CSS [Электронный ресурс]. – 2016. – Режим доступа: <http://htmlbook.ru/> (дата обращения: 17.03.2016);
2. Электронный учебник Javascript [Электронный ресурс]. – 2016. – Режим доступа: <http://javascript.ru/> (дата обращения: 17.03.2016);

УДК 004

## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ВИЗУАЛИЗАЦИИ ГРАФОВ

Демешко М.В., Дёмин А.Ю.

Научный руководитель: Дёмин А.Ю., к.т.н., доцент кафедры ИПС ИК ТПУ

*Национальный Исследовательский Томский политехнический университет,  
634050, Россия, г. Томск, пр. Ленина, 30  
E-mail: demeshkomaria@gmail.com*

*This article describes some algorithms of graph visualization such as circle-based, arc-based and force-based visualization. There is an overview of their advantages and disadvantages. Article contains examples of visualization and gives an explanation about using metrics for adaptive visualization.*

**Key words:** *graph, force-directed, graph-visualization, graph drawing, D3.js.*

**Ключевые слова:** *граф, силовой граф, визуализация графов.*

Графы используются повсеместно для моделирования широкого спектра различных объектов и их связей. Эти объекты могут быть частью материального мира, а могут представлять собой математическую или программную абстракцию. В каждом из этих случаев может потребоваться визуализация графа.

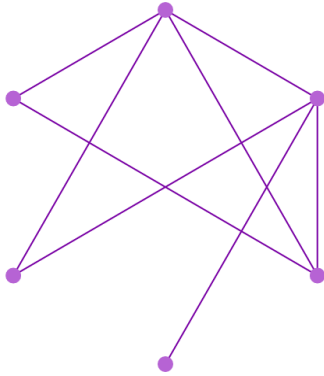
Визуализированный граф может стать мощным инструментом в задачах анализа данных. При этом задача визуализации заключается не только в выводе вершин и ребер согласно представлению графа, но и в том, чтобы этот вывод был визуальным прост для пользователя.

Можно сформулировать несколько очевидных критериев, которые раскрывают понятие «визуальной простоты»: вершины не должны накладываться друг на друга, пересечение ребер должно быть сведено к минимуму (для планарных графов – отсутствие пересечений), суммарная длина ребер должна быть минимальной, число изломов ребер должно быть минимальным, визуализация должна отражать топологические свойства графа и его симметрию. Стоит отметить, что перечисленные критерии будут несколько отличаться для различных способов отрисовки графов.

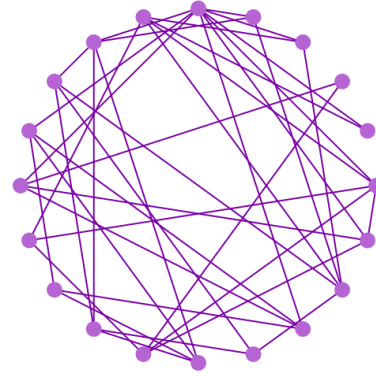
Для реализации алгоритмов визуализации, описанных ниже, использовался язык программирования JavaScript, а также библиотека D3.js.

Самым очевидным способом визуализации графов является визуализация на основе окружности. Этот способ заключается в построении вершин на окружности определенного радиуса. Ребра в таком случае являются хордами этой окружности. К преимуществам этого

способа относится простота реализации. Тем не менее, этот способ дает корректные результаты при сравнительно небольшом количестве вершин и ребер. Этот недостаток можно проиллюстрировать с помощью рис. 1 и 2.



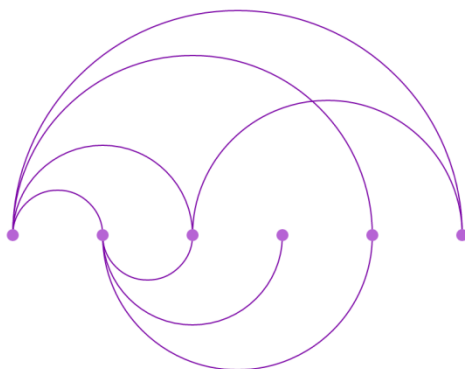
*Рис. 1. Граф, состоящий из 6 вершин и 8 ребер*



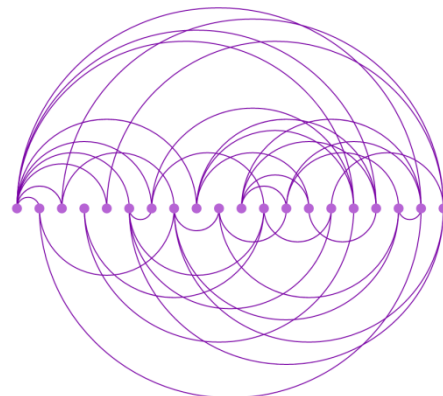
*Рис. 2. Граф, состоящий из 20 вершин и 40 ребер*

Благодаря небольшому количеству вершин и ребер граф на рис. 1 выглядит просто и структурировано. Граф на рис. 2 в свою очередь имеет множество пересечений ребер, а также близко стоящие вершины.

Следующий способ визуализации – представление ребер в качестве дуг. Этот способ подразумевает расположение вершин на одной прямой. Ребро в этом случае представляет собой полуокружность, проходящую через соответствующие вершины. Основным недостатком этого способа совпадает с недостатком визуализации на основе окружности – при большом количестве вершин и ребер граф становится визуально перегруженным. Кроме того, для уменьшения числа пересечений ребер следует осуществлять предварительный анализ графа для определения полуплоскости, в которой должна лежать полуокружность. Примеры графов, визуализированных этим способом, представлены на рис. 3 и 4.



*Рис. 3. Граф, состоящий из 6 вершин и 8 ребер*



*Рис. 4. Граф, состоящий из 20 вершин и 40 ребер*

Несмотря на большое число пересечений, результат визуализации выглядит более «разряженным» по сравнению с визуализацией на основе окружности.

Наибольшим потенциалом обладают алгоритмы, основанные на физических аналогиях [1]. Все алгоритмы, относящиеся к этому классу, моделируют граф с помощью некоей физической системы. Самыми популярными алгоритмами считаются силовой и пружинный. Силовой алгоритм описывает функцию, определяющую параметры, к которым будет стремиться система. После нескольких итераций этой функции система приходит в равновесие, которое считается удовлетворяющим ранее определенным критериям.

Для вычисления следующей позиции вершины используется метод численного интегрирования Верле [2]. С его помощью следующее положение может быть вычислено на основе текущего и прошлого, это позволяет не использовать скорость движения.

Для корректной визуализации несвязных или слабосвязанных графов используется дополнительная силовая функция, моделирующая условную гравитацию [3].

Силовые алгоритмы являются гибкими, так как обладают множеством настраиваемых параметров. Расчет определенных метрик, связанных с конфигурацией графа, позволяет подбирать эти параметры, что делает визуализацию адаптивной. Кроме того, некоторые параметры (например, длину ребер) можно задавать динамически. Но иногда даже использование статических параметров позволяет получить эстетически более правильные результаты, чем при использовании описанных ранее алгоритмов. Так, на рис. 5 представлен рассмотренный ранее граф, визуализированный с помощью силового алгоритма. Пересечения ребер здесь сведены к нулю, а расположение вершин наилучшим образом отражает их связи.

На рис. 6 и 7 приведены другие примеры использования силового алгоритма.

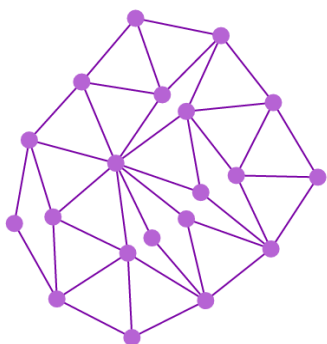


Рис. 5. Граф, состоящий из 20 вершин и 40 ребер

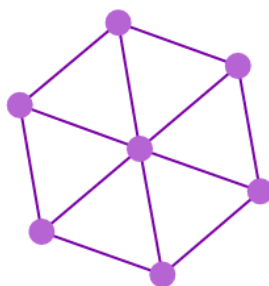


Рис. 6. Граф, состоящий из 7 вершин и 12 ребер

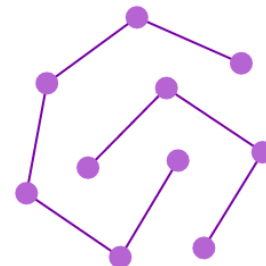


Рис. 7. Граф, состоящий из 10 вершин и 8 ребер

Примером метрики, которую следует использовать для получения адаптивной визуализации, может служить метрика бинарного дерева. В данном случае метрика представляет собой логическую переменную, которая будет принимать истинное значение в случае, если визуализируемый граф является деревом. Для осуществления такой проверки достаточно установить соотношение между числом ребер и вершин. В бинарном дереве количество ребер меньше количества вершин на единицу. Кроме того, должно соблюдаться условие связности. Для проверки этого условия необходимо обойти граф с помощью *DFS* или *BFS*.

На рис. 8 представлен граф до применения специального построения для бинарного дерева. На рис. 9 показан тот же граф, визуализированный как бинарное дерево. Такой подход позволяет наглядно показать топологию графа.

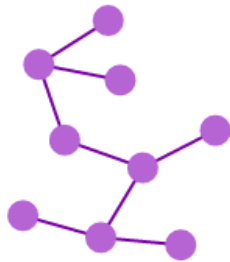


Рис. 8. Граф, визуализированный обычным силовым алгоритмом

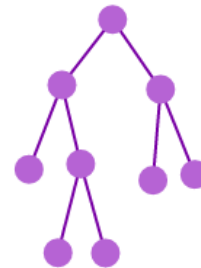


Рис. 9. Граф, визуализированный силовым алгоритмом для деревьев

### Список литературы

1. Апанович З.В. Методы визуализации информации при помощи графов. Часть 2. Методы визуализации ориентированных и неориентированных графов. – Новосибирск, НГУ, 2009 (Электронный учебник).
2. Jakobsen T. Advanced Character Physics. – IO Interactive, Copenhagen, 2003.
3. Пупырев С.Н., Тихонов А.В. Визуализация динамических графов для анализа сложных сетей // Моделирование и анализ информационных систем. – 2010. – № 1.

УДК 004

## КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИЗВЛЕЧЕНИЯ ГИДРОМЕТЕОРОЛОГИЧЕСКИХ ДАННЫХ

Ерофеева Е.Л., Попов В.Н.

Научный руководитель: Попов В.Н., к.т.н., доцент кафедры ИПС ИК ТПУ

*Национальный Исследовательский Томский политехнический университет,*

*634050, Россия, г. Томск, пр. Ленина, 30*

*E-mail: elizaveta.erofeeva@gmail.com*

*This article describes a Web-based application for automated extracting of unstructured hydrometeorological data from sites on the Internet that provide such information. The software makes it easier to receive weather reports, including encoded in the KN-01, as well as to work with them.*

**Key words:** *hydrometeorological data, web-application, code operational data.*

**Ключевые слова:** *гидрометеорологические данные, web-приложение, код оперативной передачи данных.*

Наблюдение за климатом нашей планеты ведется на протяжении нескольких веков, по причине того, что своевременная и качественная гидрометеорологическая информация играет важную роль в обеспечении защиты жизни и имущества граждан, экономике регионов. Информацию о погодных условиях предоставляют гидрометеостанции, предоставляя количественные данные о текущем состоянии атмосферы, например, температура воздуха, ско-