МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тверской государственный университет»

Факультет прикладной математики и кибернетики

Направление 01.04.02 – Прикладная математика и информатика
Программа магистратуры «Системное программирование»

Отчет по итогам производственной практики по получению профессиональных умений и опыта профессиональной деятельности 2017-2018 уч. год, 3 семестр

Автор: студент 2 курса магистратуры СП
Наймушин Алексей Владимирович

Научный руководитель:
к.ф.-м.н., доцент
Карлов Борис Николаевич

Тверь – 2017

Оглавление

Введение

Минимизация пересечений ребер у графа является одним из часто используемых на практике эстетических критериев укладки графа на плоскость. Используя данный критерий можно достичь высокого уровня читабельности графа, так как большую часть его связей будет легче проследить. Кроме этого, задача минимизации пересечений напрямую связана с эффективностью той или иной укладки, хотя в большинстве случаев полностью избавиться от пересечений невозможно.

Проблема минимизации количества пересечений является комбинаторной проблемой, которая состоит в том, чтобы выбрать упорядочение вершин, с наименьшим количеством пересечений. Стоит отметить, что данная задача является NP-полной даже в случае Сигуяма-подобной укладки графа с двумя уровнями. Более того, задача по-прежнему остается NP-полной, если известно оптимальное расположение узлов для одного из уровней.

Цели работы

Главная цель данной работы и выбранной тематики моих исследований является изучение проблемы укладки графов на плоскость.

Для данной работы, были выбраны следующие цели.

- Более подробное изучение метода укладки графов на плоскость с использованием физических моделей;
- Изучение метода Сигуямы, для поуровнего отображения графа на плоскость;
- Изучение популярных эвристических методов, направленных на улучшение работы метода Сигуямы;
- Разработка собственных эвристических подходов для данного алгоритма;
- Программная реализация метода Сигуямы и предложенных эвристических подходов в программной среде разработки приложений Qt;
- Сравнение качества произведенной укладки графа методом Сигуямы, с ранее реализованным методом физических сил.

В первом пункте работы будет рассказано об укладке с применением физических сил, описаны преимущества и недостатки данного метода, будет предоставлено описание самого алгоритма.

Во втором пункте данной работы будет рассказано о методе Сигуямы и, аналогично методу с использованием физических моделей, будет описан сам алгоритм, его преимущества и недостакти.

В третьем пункте будет кратко описаны наиболее популярные эвристические методы, улучшающие качество работы самого алгоритма.

В четвертом пункте будут предложены другие возможные эвристические методы, для решения некоторых задач, возникающих в процессе работы алгоритма Сигуямы.

В пятом пункте работы будет приведен пример и описание программной реализации метода Сигуямы и эвристических подходов, описанных в четвертом пункте данной работы.

В шестом пункте данной работы будет проведено сравнение метода Сигуямы и метода с использованием физических моделей. Данное сравнение будет включать в себя такие критерии как практическое время работы методов, занимаемая область построения изображений и конечное число пересечений.

1. Укладка с применением физической модели

Одними из наиболее применяемых методов укладок графов на плоскость являются методы с применением физических моделей.

Данные методы появились в работах Тутте в 1963 году, который показал, что графы с большим количеством узлов могут быть хорошо уложены на плоскость с применением физических моделей, добавив каждому ребру различные силы и позволив системе самой достигнуть состояния равнения.

Преимущества укладки с использованием физических моделей заключаются в том, что они способны оптимизировать сразу несколько эстетических критериев, таких как максимизация симметрии или равномерное распределение вершин. Также данные методы привлекательны своей гибкостью. Можно с легкостью применять различные эвристики, улучшающие качество укладок, при этом практически не изменяя самого алгоритма.

К недостаткам данных методов, можно отнести то, что они имеют достаточно высокое время работы: $O(I * n^3)$, где I – число итераций, n – количество вершин в графе. Однако на практике, методы данного класса сходятся достаточно быстро.

Главным недостатком данных методов является проблема локального минимума. Это объясняется тем, что данные методы стремятся уменьшить общую энергию системы и очень часто найденный минимум не является глобальным минимумом функционала энергии. Это приводит к ухудшению качества укладки.

Смысл данных методов заключается в том, что для каждого ребра или узла, задаются некоторые силы, обычно согласно закону Гука, т.е. $\boldsymbol{F} = \boldsymbol{k} * \Delta\boldsymbol{l}$, где $\boldsymbol{F}$ – сила упругости, $\boldsymbol{k}$ – жесткость тела, а $\Delta\boldsymbol{l}$ – деформация тела. Затем данный алгоритм пытается минимизировать энергию полученной системы для получения самой укладки.

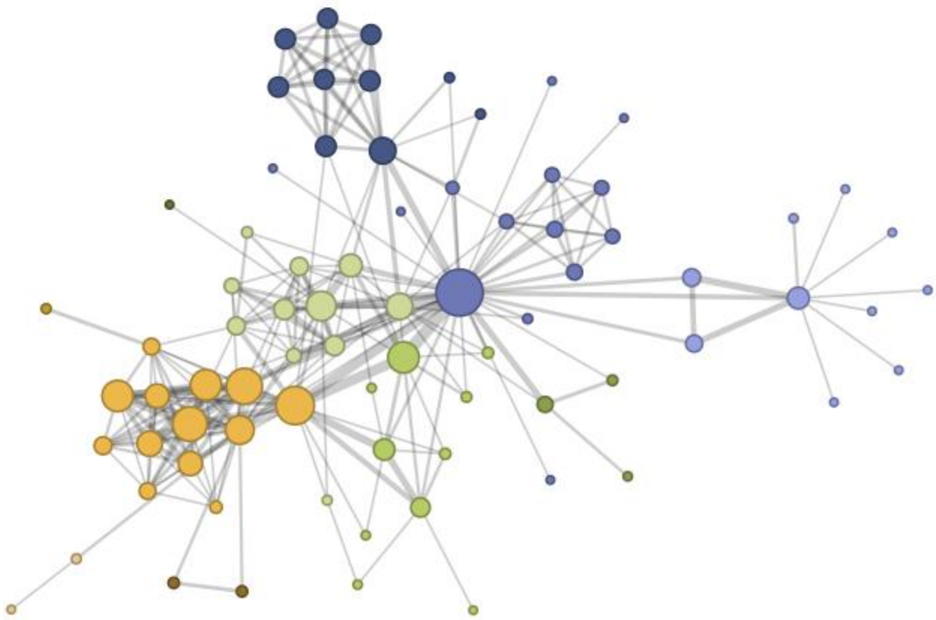

Рис. 1 Укладка с применением физической модели

Сформулируем сам алгоритм укладки с применением физической модели. Пусть на вход подается некий граф $G = (V, E)$.

1. **Инициализация**. Каждому узлу графа задается некоторая случайная точка на координатной плоскости.

2. **Итерации алгоритма**. Для каждого узла имеем векторную переменную *displ*, которая характеризует смещение узла на данной итерации. Перед началом каждой итерации, данная переменная для всех узлов равна 0.

2.1. **Вычисление отталкивающих сил**. Для каждого ребра $e = (u, v) \in E \ (u \neq v)$, вычисляем значение смещения по следующим формулам:

$$\delta = v.pos - u.pos,$$

$$v.displ = v.displ + \left(\frac{\delta}{|\delta|}\right) * f_r(|\delta|),$$

где $f_r(x) = \frac{x^2}{\sqrt{\frac{area}{|V|}}},$ $area = W * L, W \geq 0 \ , L \geq 0$ – некоторые переменные, ограничивающие площадь, в которой будет размещен граф, $|x|$ - оператор нормы векторной величины x.

2.2. Вычисление притягивающих сил. Для каждого ребра $e = (u, v) \in E$, произведем корректировку смещений его узлов u и v по следующим правилам:

$$\delta = e.v.pos - e.u.pos,$$

$$e.v.displ = e.v.displ - \left(\frac{\delta}{|\delta|}\right) * f_a(|\delta||,$$

$$e.u.displ = e.u.displ + \left(\frac{\delta}{|\delta|}\right) * f_a(|\delta||,$$

где $f_a(x) = \frac{\frac{area}{|V|}}{x}$. *u.pos* – векторная переменная, определяющая позицию узла u на плоскости.

2.3. Корректировка позиции узлов. Для каждого узла $u \in V$ произведем коррекцию позиции по следующему правилу:

$$u.pos = u.pos + \frac{u.displ}{|u.displ|}.$$

2.4. Критерий останова. Если среднеквадратичное значение всех смещений достигло некоторого порогового значения, то выходим из алгоритма, иначе переходим на пункт 2.

2. Метод Сигуямы

На практике очень часто изображения, полученные с использованием физических моделей, не всегда являются наилучшем способом отражения информации. Например, данные алгоритмы не могут явным образом отобразить иерархическую составляющую тех или иных связей, что в следствии может привести к затруднительному анализу некоторой системы, для которой и было получено изображение. В таких случаях следует использовать алгоритмы иерархической укладки графов на плоскость.

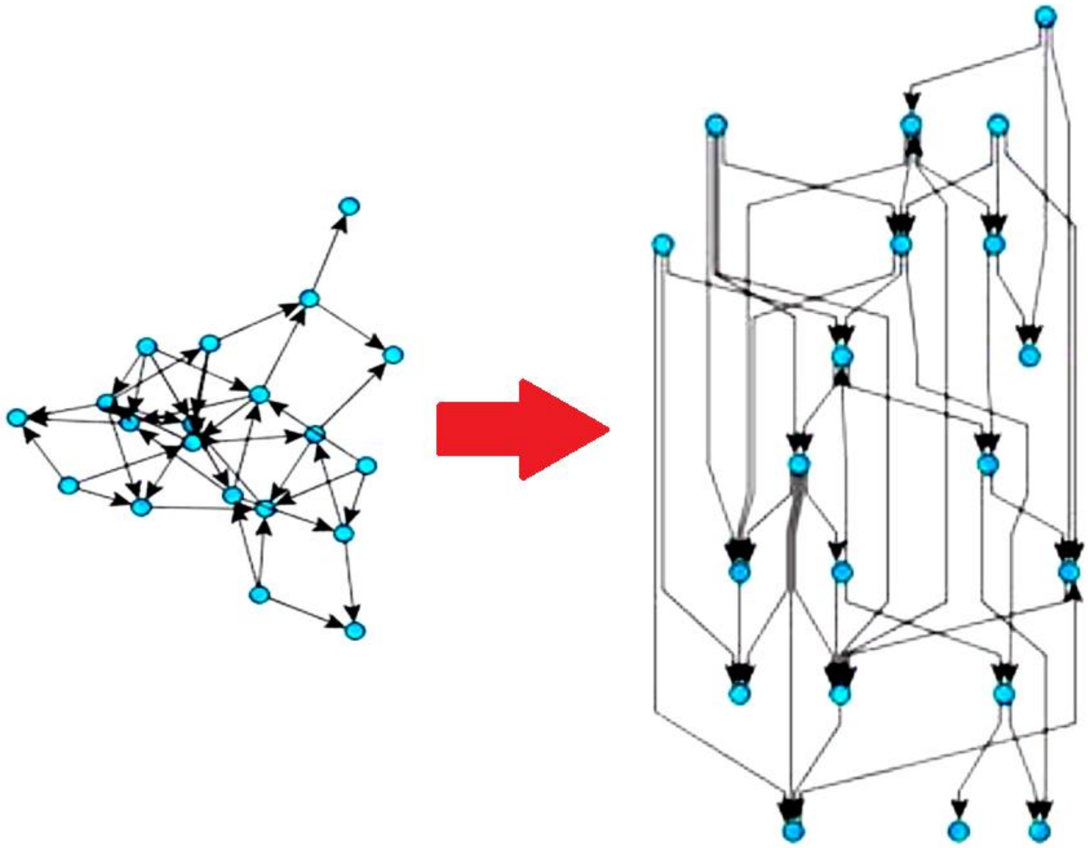

Рис. 2 Сравнение различных методов укладки графа

Одним из наиболее популярных методов послойной укладки графа на плоскость является известный метод – метод Сигуямы, названный в честь исследователя Козо Сигуяма (Kozo Sugiyama) предложившего эту методику в 1981 году. В своих работах исследователь очень часто использовал данный подход к отображению графов сетей.

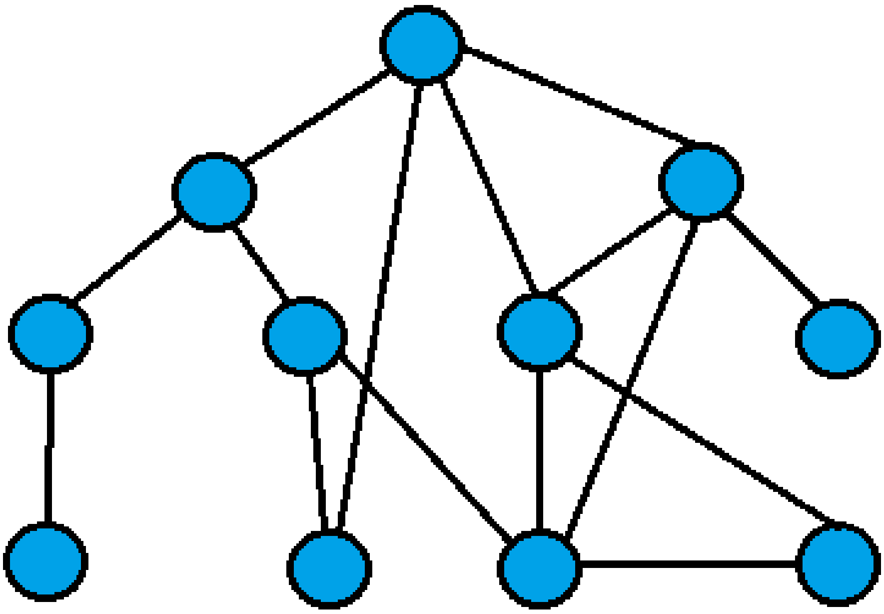

Рис. 3 Пример послойной укладки графа

Основным преимуществом данного метода является то, что свойство иерархичности вершин очень легко проследить. Также следует отметить быстроту работы данных методов и возможность обобщения методики для случая неориентированных графов.

Из недостатков можно назвать то, что обычно алгоритмы данного класса очень трудно реализовать на практике и что данный метод очень трудно обобщить для укладки в трехмерном пространстве. Еще одним недостатком алгоритма является тот факт, что поданный на вход алгоритма граф G не должен иметь циклов (т.е. таких путей в графе, которые начинаются и заканчиваются в одной вершине). Это может привести к дополнительным затратам на вычисление, отслеживая уже пройденные вершины. Для удаления циклов существует несколько эвристик, о которых будет сказано позднее.

Приведем описание самого алгоритма. Пусть требуется уложить на плоскость некий ориентированный граф $G = (V, E)$. Для этого требуется выполнить следующие этапы:

1. **Проверка графа на отсутствие циклов**. Если циклы присутствуют, то их количество следует уменьшить или убрать совсем.
2. **Распределение вершин графа по слоям** таким образом, чтобы все дуги графа имели одно направление (обычно сверху-вниз).
3. **Нахождение оптимального порядка вершин для каждого слоя**, минимизирующее общее число пересечений дуг графа.
4. **Выбор координат для каждой вершины** таким образом, чтобы они отражали порядок вершин, полученный из третьего пункта алгоритма.

Стоит отметить, что наибольшую сложность в реализации алгоритма составляют первый и третий пункты алгоритма. Доказано, что нахождение набора дуг, разрезающего циклы, а также задача на нахождение оптимального порядка узлов являются NP-полными задачами.

Задачу на нахождение набора дуг, разрезающего циклы можно сформулировать следующим образом: пусть на вход подается ориентированный граф $G = (V, E)$, Требуется найти подмножество дуг $F \subset E$ такое, что граф $G' = (V, E \setminus F)$ не содержит циклов.

Было показано, что к данной задаче можно свести задачу о вершинном покрытии, которая, как известно, является NP-полной.

Сформулируем задачу о нахождении оптимального порядка вершин в слоях графа. На вход подается некий граф $G = (V, E)$, а для каждой его вершины v из V известен ее номер слоя в укладке. Требуется найти такие последовательности вершин для каждого слоя, чтобы общее число пересечений дуг в графе было минимальное. Этот факт был доказан Гери и Джонсоном в 1983 году.

Одно из ключевых преимуществ данного алгоритма это время его работы. Если принимать во внимание тот факт, что в графе $G = (V, E)$ полностью отсутствуют циклы, то оно составляет примерно $O(|V||E| \log|E|)$, однако если циклы в графах все-таки присутствуют, то суммарная сложность алгоритма стремительно возрастает до экспоненциальной.

3. Эвристики для улучшения работы метода Сигуямы

Как уже было сказано ранее, одним из важных свойств, которым должен обладать граф для его отображения с использованием метода Сигуямы, это отсутствие циклов. Также было сказано, что задача нахождения таких дуг, образующих циклы является NP-полной, что значительно увеличивает общее время работы алгоритма. Однако известно несколько эвристик, способные устранить эту проблему за разумное время.

Простейшей эвристикой в данном случае является использование алгоритма обхода графа в глубину. Можно также использовать алгоритм обхода в ширину, однако ему потребуется больше времени и памяти, чем обход в глубину.

После того как на некотором шаге обхода графа выясняется, что данная дуга приводит к зацикливанию в графе, то она удаляется.

Приведем алгоритм обхода графа в глубину.

Пусть на вход подается некий граф $G = (V, E)$. Требуется найти и удалить из множества дуг E такие дуги, которые позволят разорвать циклические пути в графе. Тогда:

1. **Инициализация**. Помечаем все вершины в графе как новые.
2. **Для всех вершин *v* из множества *V* выполняем следующее**: если вершина помечена как новая, то вызываем для нее вспомогательную процедуру *DFS*.

Опишем процедуру *DFS:*

1. **Помечаем вершину** *u* как старую.
2. **Для всякой вершины, смежной с *u*** выполняем следующее: если текущая смежная вершина *v* помечена как старая, то $E = E \setminus \{(u, v)\}$. В противном случае рекурсивно вызываем для вершины *v* процедуру *DFS*.

Время работы данного алгоритма составляет $O(|V| + |E|)$.

Другая эвристика представляет собой вариацию первой, однако после того как дуга образующая цикл найдена она не удаляется, а разворачивается в противоположном направлении, а после выполнения этапа получения координат для вершин разворачивается обратно. Например, если обнаруживается, что дуга $(u, v) \in E$ образует цикл, то она заменяется другой дугой (v, u), а на финальном этапе алгоритма ее вершины меняют местами обратно. Однако данный метод приводит к тому, что некоторые дуги графа будут идти «против течения». Это может сказаться на читаемости конечного изображения графа.

Перейдем к методам распределения вершин по слоям. В общем случае, выбор данной эвристики зависит от свойств рассматриваемой предметной области, для которой и требуется получение изображения графа. Если же распределение вершин по слоям не играет серьезной роли, то можно использовать следующие методы:

- Распределить вершины по уровням случайным образом. Однако это не гарантирует хорошего конечного результата.
- Использовать ранее упомянутые алгоритмы обхода графов. В качестве вершин, которые будут размещены на первом слое, следует выбрать вершины, в которые не входят дуги.

Перейдем к рассмотрению наиболее популярного определения порядка вершин на уровнях - метод барицентров. Его смысл заключается в том, что положение вершины на плоскости определяется как среднее арифметическое координат ее дочерних вершин. Таким образом, координаты вершины u можно вычислить по следующей формуле:

$$avg(u) = \frac{1}{\deg(u)} \sum_{v \in N(u)} x(v),$$

где $\deg(u)$ – степень вершины, $N(u) = \{V | (u, v) \in E\}$ – множество вершин, для которых есть ребро из u, а $x(v)$ – координата вершины v. Если же после определения координаты двух вершин слоя совпадают, то к координате одной из них прибавляют небольшое значение, чтобы отдалить вершины друг от друга.

После этого, сами вершины следует упорядочить в соответствии с их барицентрами. Для этого можно воспользоваться любым методом сортировки, например методом быстрой сортировки.

Стоит отметить огромную эффективность данного метода. Время вычисления барицентра одной вершины пропорционально её степени, поэтому барицентры всех вершин можно вычислить за линейное время, а сложность вычислений быстрой сортировки – $O(|V|log|V|)$.

4. Эвристика для уменьшения пересечений дуг в графе

Рассмотрим несколько возможных эвристических методов, которые могут помочь в уменьшении общего количества пересекающихся дуг.

В большинстве случаев после выполнения этапа нахождения порядка вершин в слоях, остаются некоторые пересечения, от которых можно легко избавиться «на глаз». Данный подход был разработан с целью уменьшения таких пересечений в конечном изображении.

Идея данной эвристики заключается в том, чтобы для каждой пары пересекающихся дуг попробовать поменять местами позиции их родителей.

Сформулируем псевдо-алгоритм данной эвристики. Для удобства положим, что все ребра в графе пронумерованы как $e_1, e_2, \dots e_n$.

1. **Для *i* = *1* до *n***:
2. **Для *j* = *i* + *1* до *n***:
3. **Если ребра e_i и e_j пересекаются**, то поменяем местами координаты вершин родителей данных ребер.
4. **Если общее число пересечений в графе увеличилось**, то возвращаем вершины на исходные позиции.

Теоретическое время работы такой эвристики составляет $O(n^2)$, однако на практике данный алгоритм работает достаточно быстро и эффективно. Эта эвристика позволяет устранить большое количество пересечений.

Приведем пример работы данной эвристики. Пусть имеется некоторый граф, к которому был применен метод укладки Сигуямы без сортировки вершин на уровнях.

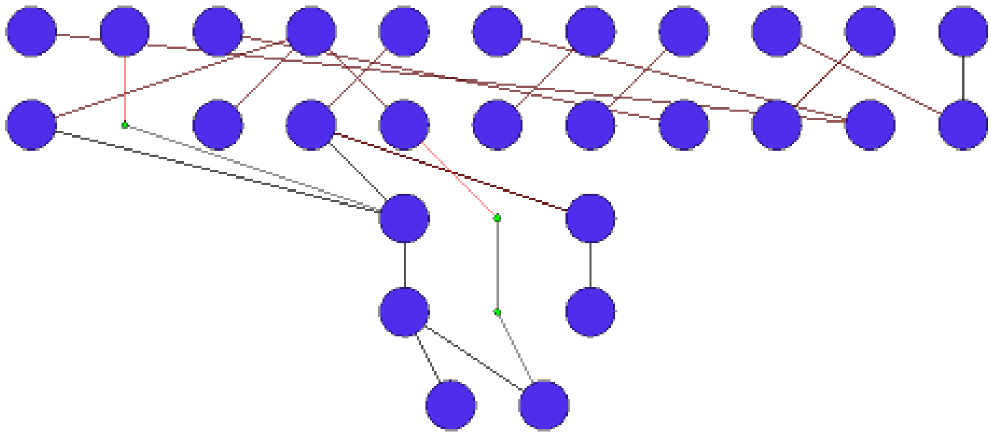

Рис. 4 Пример изображения графа без использования эвристики

Можно посчитать, что данное изображение графа содержит 21 пересечение дуг. Применим наш эвристический метод:

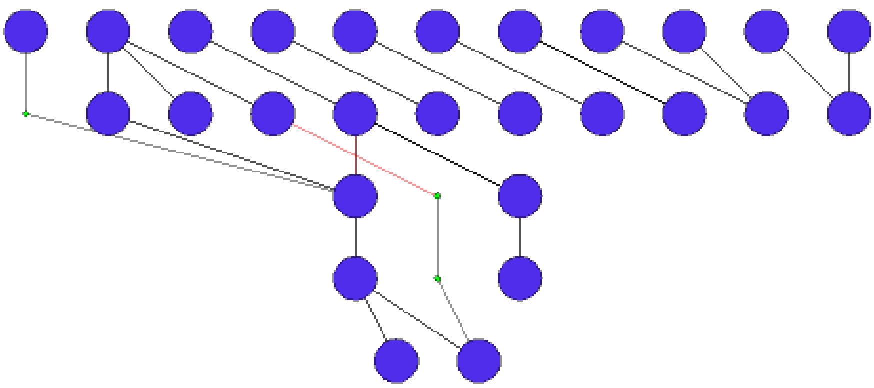

Рис. 5 Пример работы эвристики

Как можно заметить, общее число пересечений уменьшилось до одного.

Другой возможной эвристикой является использование гибридного метода. Его смысл заключается в использовании сразу нескольких эвристических подходов к уменьшению числа пересечений. Например, вначале воспользовавшись методом барицентров для нахождения начального расположения вершин на уровнях, а затем улучшить полученный результат, применив указанный выше эвристический метод.

5. Программная реализация метода Сигуямы

Одной из поставленных целей данной работы являлось программная реализация метода Сигуямы и предложенных эвристических подходов в программной среде Qt.

В реализованной программе, для хранения графов в памяти, было выбрано его представление в виде списков смежности, т.е. когда каждой вершине графа соответствует список всех ее детей. Это позволяет достичь минимального количества времени, требуемое на проведение обхода графа в глубину.

В качестве эвристики для распределения вершин по уровням, была выбрана эвристика обхода графа в глубину. Также, данная эвристика используется для удаления дуг, приводящих к циклам. Все это выполняется в рамках одного прохода, что также ускоряет общее время получения изображения графа.

Для определения порядка вершин на каждом уровне графа, используется предложенная гибридная эвристика, с использованием метода барицентров и метода перестановки родителей вершин.

Координаты вершинам на уровнях задаются согласно полученным спискам уровней графа, где порядковый номер каждой вершины умножается на некоторый коэффициент.

На следующем рисунке можно увидеть несколько примеров работы программы.

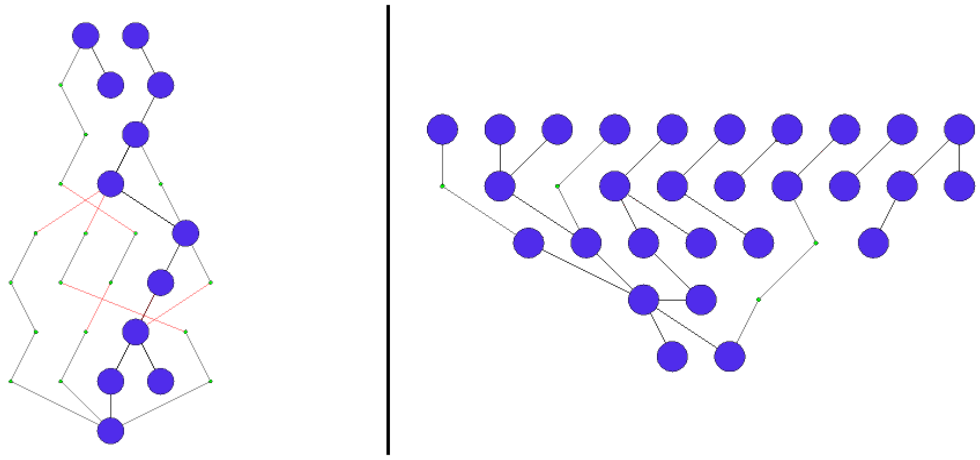

Рис. 6 Примеры работы программы

На практике, данный метод работает достаточно эффективно как для графов малого размера, так и на достаточно больших и сложных графах. Ниже можно видеть график зависимости времени работы алгоритма от количества вершин в графе. Величина времени работы измеряется в секундах

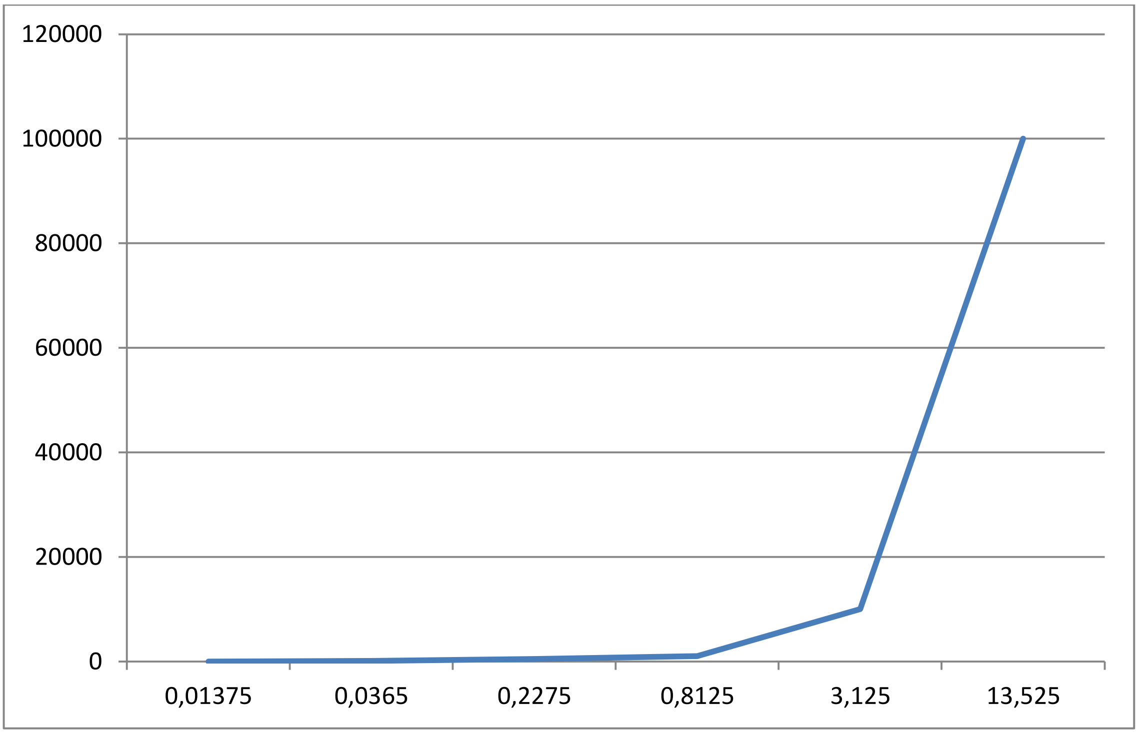

Рис. 7 График зависимости времени работы алгоритма от размеров графа

6. Сравнение метода Сигуямы и метода укладки с применением физических моделей

Хотя два метод Сигуямы и метод с применением физических моделей являются двумя совершенно разными подходами к решению задачи укладки графов на плоскость, оптимизируя различные эстетические критерии качества укладки, проведем их сравнение по нескольким критерием. В качестве критериев для сравнения, мною были выбраны следующие:

1. Время работы;
2. Трудности в реализации;
3. Размер занимаемой области уложенного графа;
4. Число пересечений;
5. Симметричность укладки вершин.

Время работы. Для определения времени работ алгоритма, были случайным образом сгенерированы графы различных сложностей на 10, 100, 1000, 10000 и 10000 вершин. Были рассмотрены как полно связные структуры графов (когда для любой пары вершин в графе u и v, найдется такая дуга $(u, v) \in E$), так и не полно связные.

Ниже приведены графики зависимости времени работ алгоритмов от сложности структуры графов.

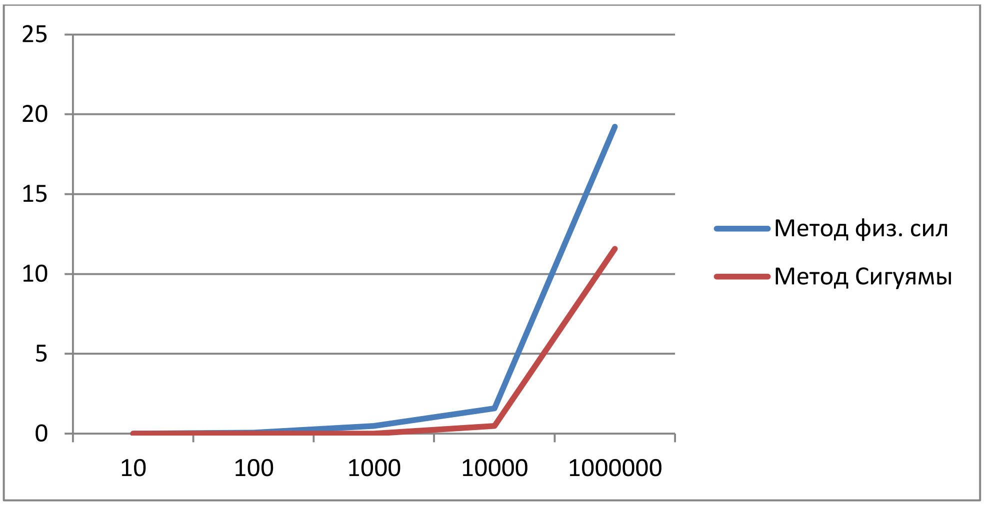

Рис. 8 График времени, затраченного на получения изображений с использованием метода физических сил и метода Сигуямы

Опираясь на полученные результаты можно сделать вывод о том, что метод Сигуямы работает на много быстрее метода физических сил. Данный факт можно объяснить тем, что метод физических сил является итеративным методом и для его сходимости может потребоваться достаточно большое число итераций. Кроме этого, время одной итерации в худшем случае занимает $O(n^2)$. Алгоритм с использованием физических сил можно сильно ускорить, добавив дополнительное ограничение на число итераций алгоритма. Однако от этого может существенно ухудшиться качество конечной укладки.

Трудности в реализации. Основное преимущество метода укладки с использованием физических моделей является его интуитивность и простота описания самого алгоритма. Алгоритм достаточно просто реализовать и использовать как для использования различных эвристических методов, так и для построения изображений графов, например, с целью анализа данных. Метод Сигуямы в этом плане проигрывает, так как для его реализации требуется протестировать множество эвристических подходов для каждого этапа алгоритма, прежде чем алгоритм начнет выдавать адекватные результаты и за разумное время. Однако эти трудности полностью оправдывают себя, так как конечный результат получается намного нагляднее и информативнее.

Размер занимаемой области. В отличие от метода Сигуямы, метод с использованием физических сил имеет параметры, регулирующие размеры области необходимой для построения изображения укладки графа. Для метода Сигуямы можно попытаться уменьшить конечную область, путем использования более жадных методов определения координат для вершин, которые будут стараться размещать вершины как можно ближе друг к другу. Однако от этого может измениться число пересекающихся дуг в изображении и, соответственно, ухудшится читаемость графа.

Число пересечений. Для определения качества конечных изображений, полученных указанными методами, аналогично методики сравнения затраченного на укладку времени, случайным образом были сгенерированы графы различной сложности на 10, 100, 1000 и 10000 вершин. После этого было взято среднее число пересечений для каждого случая. Данный способ позволяет дать среднюю оценку для числа пересечений, которые могут быть допущены при использовании метода укладки с применением физических сил и метода Сигуямы.

Ниже приведена таблица допущенных пересечений для алгоритмов:

Кол-во вершин	Метод физ. сил	Метод Сигуямы
10	0	0
100	4	11
1000	14	25
10000	74	97

Таблица 1 Среднее число допущенных пересечений

Как можно заметить, число пересечений для метода Сигуямы с выбранным набором эвристических методов, указанных в четвертом пункте данной работы, работает несколько хуже, чем метод укладки с применением физических моделей. Данный результат можно улучшить, попробовав применить другие эвристические подходы для сортировки вершин на слоях.

Симметричность укладки вершин. Преимуществом метода укладки с применением физических моделей является то, что конечный результат изображения имеет достаточную симметрию вершин. Более того, сами вершины достаточно равномерно распределены по всей области, выделенной для размещения изображения графа. Данное свойство симметрии определяет то, насколько изображение, полученное для конечного пользователя, будет ему наглядно.

Стоит отметить, что для укладки с использованием метода Сигуямы, большая часть графа может находиться только в одной части используемой области. Однако данное свойство напрямую зависит от выбранных эвристик сортировки и определения координат, которые будут более равномерно и симметрично распределять вершины по всей области укладки графа.

Заключение

В процессе выполнения данной работы был изучен метод Сигуямы для укладки графов на плоскость. Также более детально был изучен метод укладки графов с использованием физических моделей. Были рассмотрены популярные эвристики, направленные на улучшение качества конечной укладки графов. Были предложены несколько других возможных эвристик. Кроме этого, в рамках данной работы, была выполнена программная реализация метода Сигуямы и предложенных эвристик.

В дальнейшем планируется изучение алгоритма Хопкрафта-Тарьяна для проверки графов на планарность, провести исследование на тему нахождения наилучшей укладки графа с фиксированным положением узлов, а также провести анализ и сравнение между собой ранее изученных методов укладки графов на плоскость

Список литературы

1. Касьянов В. Н., Евстригнеев В.А., Графы в программировании: обработка, визуализация и применение. – СПб., БХВ-Петербург, 2003 – 1104 с.. ил.
2. Bondy J.A., Murty U.S.R., Graph Theory with Applications, North Holland, Amsterdam, 1976
3. Eades P., Complexity Issues in Drawing Directed Graphs, Proc. Int. Workshop on Discrete Algorithms and Complexity, pp. 9{15, Fukuoka, Japan, 1989.
4. Ferrari D., Mezzalira L., On Drawing a Graph with the Minimum Number of Crossings, Technical Report n. 69-11, Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, 1969.
5. Gansner E.R., Koutsofos E., North S.C., and Vo K.P., A Technique for Drawing Directed Graphs, IEEE Trans. on Software Engineering, vol. 19, no. 3, pp. 214-230, 1993.
6. Johnson D.S., The NP-Completeness Column: an Ongoing Guide," J. of Algorithms, vol. 3, no. 1, pp. 89-99, 1982.
7. Johnson D.S., The NP-Completeness Column: an Ongoing Guide," J. of Algorithms, vol. 5, no. 2, pp. 147-160, 1984.
8. Lin X., Analysis of Algorithms for Drawing Graphs, PhD thesis, Department of Computer Science, University of Queensland, 1992.
9. Sugiyama K., Tagawa S., Toda M., Methods for Visual Understanding of Hierarchical Systems, IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-11, no. 2, pp. 109- 125, 1981.
10. Sugiyama K. and Toda M., Structuring Information for Understanding Complex Systems: A Basis for Decision Making, FUJITSU Scientific and Technical Journal, vol. 21, no. 2, pp. 144-164, 1985.
11. Thomassen C., Planarity and Duality of Finite and Infinite Planar Graphs, J. Combinatorial Theory, Series B, vol. 29, pp. 244-271, 1980.
12. Vaucher J., Pretty Printing of Trees, Software Practice and Experience, vol. 10, no. 7, pp. 553-561, 1980.

13. Wetherell C. and Shannon A., Tidy Drawing of Trees, IEEE Trans. on Software Engineering, vol. SE-5, no. 5, pp. 514-520, 1979.
14. Woods D., Drawing Planar Graphs, Ph.D. dissertation (Technical Report STAN-CS-82-943), Computer Science Dept., Stanford Univ., 1982.