

```
= '{$_POST['sims']}', `pereezd` = '{$_POST['pereezd']}', `koldetey` = '{$_POST['koldetey']}',
`kid` = '{$_POST['kid']}', `priemdetey` = '{$_POST['priemdetey']}', `socznach` =
 '{$_POST['socznach']}', `rassylka` = '{$_POST['rassylka']}', `databegin` =
 '{$_POST['databegin']}', `dataend` = '{$_POST['dataend']}', `inet` = '{$_POST['inet']}', `tel` =
 '{$_POST['tel']}', `showtel` = '{$_POST['showtel']}', `name` = '{$_POST['name']}', `surname` =
 '{$_POST['surname']}', `fathename` = '{$_POST['fathename']}', `adres` = '{$_POST['adres']}',
 `pasport` = '{$_POST['pasport']}', `hobbi` = '{$_POST['hobbi']}', `invalid` = '{$_POST['invalid']}',
 `invalid_desc` = '{$_POST['invalid_desc']}', `ogranich` = '{$_POST['ogranich']}', `target` =
 '{$_POST['target']}' WHERE `id` = '{$_POST['oldid']}";
```

```
print("<center><font class=\"good\"><a href=\"index.php?do=founum&num={$id}\">Абонент
№ {$id} успешно изменён.</a></font></center><br>");
$user=getabonent($who); }
```

Приведенный выше код потребовал усовершенствования путем применения стандартных и авторских подходов.

В результате нами реализованы некоторые подходы для защиты php кода и баз данных. Используются как базовые, так и дополнительные алгоритмы шифрования. Например, хэширующий алгоритм md5, для создания цифровых подписей, позволяющих однозначно идентифицировать отправителя. В PHP для указанного алгоритма существует специальная функция. Другие подходы по защите информации баз данных малых предприятий будут продемонстрированы в ходе доклада.

Литература

1. Охримчук, В.В. Организация работы малого предприятия с использованием автоматизированной системы управления «Брачное агентство» / В.В. Охримчук // Материалы II международной студенческой конференции «Студенческая наука – будущее государства» (Пинск, 25 марта 2008 года). – Пинск: Изд-во Полеский ГУ. – 2008. – Ч.2. – С. 103.
2. Охримчук, В.В. Информационные технологии WEB-программирования базы данных управления малым предприятием // Материалы IV международной научно-практической студенческой конференции «Содружество наук. Барановичи 2008» (Барановичи, 21 мая 2008 года). – Барановичи: Изд-во РИО БарГУ. – 2008. – Ч.1. – С. 42.
3. Соколов, А. Защита от компьютерного терроризма: справочное пособие / А. Соколов, О. Степанюк. – Арлит.: БХВ-Петербург, 2002. – С. 408 – 409.

УДК 519.171.2

НЕКОТОРЫЕ СПОСОБЫ РИСОВАНИЯ ГРАФОВ НА ПЛОСКОСТИ

Райченок С.В.

УО «Брестский государственный технический университет», г. Брест

Определение проблемы. Нарисовать граф красиво - это проблема, которая постоянно возникает в приложениях, таких как отображение файла или каталога деревьев, диаграммах в схемотехнике. Алгоритм должен таким образом отражать структуру графа, чтобы зритель мог лучше понять его.

Определим несколько более жестких критериев, чем понятие «красота», оценки качества рисования графов: минимизация числа пересечений, ограничение свободного места, минимизация изломов минимизация общей длины ребер.

На основе вышеперечисленных критериев, сформулируем задачу следующим образом: описать алгоритм, переводящий матрицу смежности вершин в графовое представление, оценить степень его эффективности. Желательна максимальная адаптация алгоритма для визуализации плоских трехсвязных графов.

Решение задачи

Физические модели. В качестве алгоритмов нахождения изображения графа будем использовать алгоритмы методов физических аналогий. Были применены следующие аналогии. Будем рассматривать граф как систему тел с силами, взаимодействующих между телами. Каждая конкретная раскладка графа определяет одно из состояний этой системы. Введем понятие энергии системы, задача сводится к нахождению состояния с минимальной энергией.

Пружинный алгоритм. Граф рассматривается как система тел с силами, взаимодействующими между телами, например, считая вершины графа телами, а ребра пружинами. Задача сводится к задаче о нахождении расположения тел с локально минимальной энергией - конфигурацию равновесия сил, в которой каждое тело занимает такую позицию, что сумма всех сил равна нулю. Модернизируем алгоритм, представим, что тела, соединенные пружинами являются телами, несущими электронный заряд одинаковой величины, что позволит не приближаться вершинам слишком близко друг к другу (возможно и применение механической аналогии - связи всех вершин друг с другом дополнительными наборами пружин).

В результате, сила, действующая на тело p , определяется по формуле (1):

$$F(p) = \sum_{u=(p,q) \in E} f_u + \sum_{(p,q) \in V^2} g_{(p,q)}, \quad (1)$$

где f_u - сила растяжения, действующая на вершину p из-за пружины (p,q) , а $g_{(p,q)}$ - это сила отталкивания, существующая между частицами p и q . По закону Фуке f_u пропорциональна разности между расстоянием от p и до q и длиной пружины с минимальной энергией; а сила $g_{(p,q)}$ следует обратному квадратичному закону. Получаем формулы (2), (3) для расчета силы в двумерном (плоском случае):

$$\sum_{u=(p,q) \in E} k_u^{(1)} (d(p,q) - l_u) \frac{x_p - x_q}{d(p,q)} + \sum_{(p,q) \in V^2} \frac{k_{(p,q)}^{(2)}}{(d(p,q))^2} \frac{x_p - x_q}{d(p,q)}, \quad (2)$$

$$\sum_{u=(p,q) \in E} k_u^{(1)} (d(p,q) - l_u) \frac{y_p - y_q}{d(p,q)} + \sum_{(p,q) \in V^2} \frac{k_{(p,q)}^{(2)}}{(d(p,q))^2} \frac{y_p - y_q}{d(p,q)}, \quad (3)$$

где l_u - естественная (с нулевой энергией) длина пружины, $k_u^{(1)}$ - коэффициент жесткости пружины, а $k_{(p,q)}^{(2)}$ - коэффициент отталкивания между p и q .

Для подсчета оптимального состояния системы будем использовать градиентный метод. У градиентного метода есть существенный недостаток – метод может свалиться в локальный экстремум, не достигнув глобального экстремума. Оптимизируем алгоритм, добавив в него силу притяжения к центру области, на которой происходит рисование. Так же первоначальное положение системы будем выбирать не случайно, а располагать вершины по кругу.

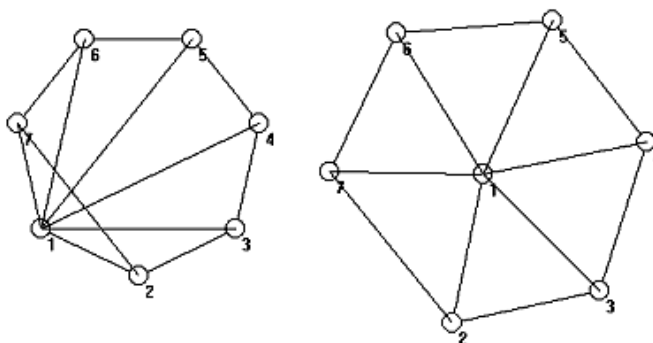


Рисунок 1 – Первый (слева) и последний шага (справа) работы алгоритма

При тестировании рисование графа с 7 вершинами (рис. 1) заняло время – 153 (мкс), что является удовлетворительным результатом. Однако были обнаружены и слабые стороны данного алгоритма – он не всегда способен минимизировать количество пересечений ребер, также необходим подбор начальных параметров элементов системы для определения размеров, которые займет представление графа.

Метод прожига. Задача минимизации энергии не может быть решена аналитически в общем случае. Применим алгоритм, который моделирует процесс отжига системы. Температуру системы (степень подвижности вершин) будем характеризовать числом T . Со временем система «остывает», т.е. T уменьшается. В процессе одной итерации каждая вершина делает попытку переместиться на произвольное расстояние $r \leq T$ в случайном направлении. Если энергия нового состояния меньше энергии старого, то система переходит в это новое состояние.

Исходя из проблем, обнаруженных в ходе реализации пружинного алгоритма, в качестве основного критерия визуализации примем наименьшее количество пересечений дуг графа. Это и будет энергией системы. За температуру примем то расстояние, на которое может переместиться вершина. Начальное значение температуры выбирается исходя из следующего условия нахождения всех вершин в пределах видимости, поэтому самое большое расстояние, на которое вершина может переместиться, оставаясь видимой, будет равно наименьшей из величин Высота, Ширина (их выберем исходя из теоремы, которая гласит, что каждый плоский трехсвязный граф имеет выпуклый рисунок на сетке размерами $(N-2) * (N-2)$). Это необходимо для тех вершин, которые могут попасть за границу видимого прямоугольника. Для каждой вершины генерируем случайные координаты в пределах этого прямоугольника. Вычисляем значение энергии, количество пересечений дуг графа при такой раскладке графа. Далее, если энергия системы уменьшилась, мы запоминаем новое положение вершины и пробуем сгенерировать новые координаты для другой вершины. С полным проходом вершин температура системы уменьшается, т.е. расстояние, на которое можно двигать вершины, тоже уменьшается.

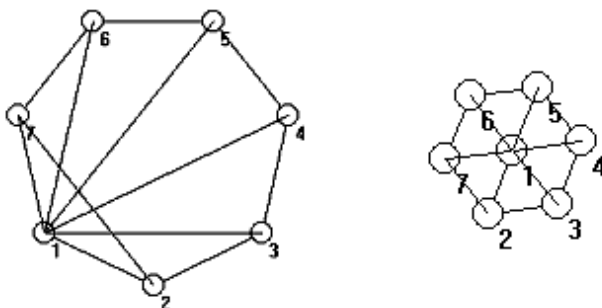


Рисунок 2 – Первый (слева) и последний (справа) шага работы алгоритма

При тестировании рисование графа с 7 вершинами (рис. 2) заняло время – 219747 (мкс), что является плохим результатом. На основе полученных данных мы получаем хорошие результаты, но этот алгоритм вряд ли применим к большим графам (более 20-30 вершин) из-за своей трудоемкости.

Аналитические модели. Был разработан следующий аналитический алгоритм рисования графа:

1) Построение треугольного графа методом смещений (shifting) де Фрайсекса. Он основан на понятии канонического упорядочения вершин графа, располагает вершины планарного графа в целочисленных узлах системы координат, так что дуги представляются прямолинейными отрезками. Вершины раскладываются по порядку в соответствии со своими номерами, и при добавлении новой вершины уже разложенные могут смещаться так, чтобы избежать пересечения дуг

2) Производится ортогонализация графа путем ассоциации графа с потоковой сетью, где в качестве потоков представлены числа изгибов, либо размеры углов, образованных в вершинах граней. Вычисляется поток с минимальной стоимостью в сети и вычисляется ортогональное представление графа, ассоциированные с потоком с минимальной стоимостью.

3) Производится компактификация графа путем построения вертикальных и горизонтальных потоков ассоциированных для каждого из выделенных сегментов графа. Вычисляются потоки минимальной стоимости горизонтальный и вертикальный – и устанавливается длина каждого сегмента, равная потоку в соответствующей дуге.

Данный алгоритм будет реализован в ходе выполнения курсовой работы по предмету «Прикладные интеллектуальные системы и системы принятия решений».

Литература

1. Ioannis G. Tollis, Giuseppe Di Battista, Peter Eades, Roberto Tamassia. Graph Drawing: Algorithms for the Visualization of Graphs. – New Jersey, 1999.
2. T. Nishizeki, M. Rahman. Planar Graph Drawing. – Danvers, 2004.
3. M. Kaufmann, D. Wagner. Drawing Graphs Methods and Models. – Berlin, 2001.
4. M. Rahman T. Nishizeki. Orthogonal Drawings of Plane Graphs Without Bends, Journal of Graph Algorithms and Applications. – Vienna. – 2003. – P. 335.

УДК 004.6

ТЕСТИРОВАНИЕ ЗАДАЧ ОЛИМПИАДНОГО ХАРАКТЕРА

Рыжков А.А.

*УО «Брестский государственный университет им. А.С.Пушкина», г. Брест
Научный руководитель – Силаев Н.В., доцент*

В данном сообщении обсуждается проблема создания программного комплекса автоматического тестирования решений задач олимпиадного характера.

Нам видится, что интерфейс системы тестирования может представляться в двух формах – в стиле веб-интерфейса, а также в стиле менее распространённом на практике – через комплекс программ «клиент-сервер». Мы предлагаем для рассмотрения тестирование через веб-интерфейс.

Тестирование через веб-интерфейс – это тестирование, которое может быть реализовано на языках php, ASP .Net, perl и др., сам же модуль сервера, обрабатывающий результаты, может представлять собой консольное приложение или WindowsForm приложение.