1. Помечаем вершину, из которой посылаем данные, нулем (информация первоначально находится в нем), и добавляем ее номер в очередь.
2. Извлекаем из очереди номер вершины и рассчитываем момент времени, к которому придет весь информационный пакет, если передача осуществляется из текущей вершины во все связанные с ней (с учетом скорости и расписания работы узлов). Возможны следующие варианты:
   а) Смежная вершина не имеет пометки. Помечаем вершину и заносим в очередь.
   б) Смежная вершина уже имеет метку и она меньше ожидаемого времени: Метку не изменяем. Если вершины нет в очереди, то добавляем ее.
   в) Смежная вершина уже имеет метку и она больше ожидаемого времени: Модифицируем метку. Если вершины нет в очереди, то добавляем ее.
3. Если в очереди есть вершины, то повторяем пункт 2, иначе пункт 4.
4. Двигаясь из вершины – получателя по промежуточным вершинам (используя ссылки) к начальной получим искомый путь. Пометка вершины – получателя это искомое минимальное время.

Алгоритм реализован в среде MatLab 6.1. Он может быть полезен для оптимальной маршрутизации пакетов в сети передачи данных (например, Fido-NET).

ПОИСК КРАТЧАЙШИХ ПУТЕЙ НА ГРАФАХ ПРИ УЗЛОВЫХ ОГРАНИЧЕНИЯХ НА СТРУКТУРУ ПУТИ

Дарадкех Ю.И., Ревотюк М.П., БГУИР, г.Минск

Задача поиска кратчайших путей на графах [1] здесь рассматривается с целью построения шаблона функции полиморфного класса, где определение графа и условий перемещения транспортной на сети допускают конкретизацию в рамках технологии объектно-ориентированного программирования [2].

Пусть транспортная сеть представлена нагруженным ориентированным графом $G(M,N)$, где N и M – множества вершин и дуг графа, а каждой дуге $(i,j) \in M$, $i,j \in N$, соответствует положительное вещественное число $w(i,j)$, называемое длиной дуги. Обозначим для любой дуги $(i,j)\in M$ множество допустимых вершин для развития путей из вершины j через $cont(i,j)$. Очевидно, что $cont(i,j)\in\{k|w(j,k)\geq 0\}$, а в случае отсутствия ограничений на выбор пути после прохождения дуги (i,j) – $cont(i,j)=\{k|w(j,k)\geq 0\}$, $i,j,k\in N$, $(i,j)\in M$.

Требуется найти, если существует, на множестве вершин $\{s(0)=s$, $s(1)\in cont(s,s)$, $s(2)\in cont(s(0), s(1))$,... $s(i)\in cont(s(i\text{-}2), s(i\text{-}1))$,... $t\}$ кратчайший путь от вершины s к вершине t, где $s,t\in N$.

Покажем, что искомый алгоритм может быть построен путем целенаправленной модификации алгоритма Дийкстры [1], что обосновано [3].

Рассмотрим часто встречающийся на практике случай узловых ограничений, когда для всех $s\in N$ $cont(i,j)=cont(prec(j),j))$, $(i,j)\in M$, т.е. альтернативы перемещения из любой вершины графа сети определены только входящими в эту вершину дугами кратчайшего пути в вершину j (здесь $prec(j)$ – номер начальной вершины дуги кратчайшего пути в вершину j). Такой случай удобно представить введением локальной относительно отдельных вершин раскраски дуг.

Раскрасим разными красками каждую входную дугу отдельных вершин графа. Пометим выходные дуги, доступные для продолжения пути, цветом входной дуги. В результате выходные дуги получат композиционную окраску. Количество различных красок на отдельной выходной дуге не превышает количества входных дуг соответствующей вершины графа. При построении дерева кратчайших путей необходимо будет выбирать из множества выходных дуг только дуги, содержащие пометку с цветом зафиксированной входной дуги.

Обозначим для каждой вершины $x\in N$ множество входных вершин $`x=\{i|w(i,x)\geq 0\}$ и пусть максимальное значение полустепени захода $C=max\{|`i|, i\in N\}$. Для кодирования раскраски дуг графа можно использовать отображение цветов дуг на характеристические двоичные векторы размерностью C.

Каждая дуга (i,j) как элемент возможного пути является выходной для вершины i и входной для вершины j, поэтому для характеристики раскраски потребуется $`B(i,j)$ – вектор раскраски относительно входа в вершину j и $B'(i,j)$ – вектор раскраски этой дуги относительно выхода из вершины i.

Пусть для некоторой вершины $x \in N$ $`x=\{i(1),i(2),\ldots\ i(j)\}$, где $j \leq C$. Тогда компоненты вектора $`B(i(k),x)=\{`b(i(k),x,m),\ m=\overline{1,C}\}$, определяются выражением $`b(i(k),x,m)=(k=m)\&(m \leq j),\ m=\overline{1,C},\ i(k) \in `x$.

После проведения раскраски всех дуг графа относительно входа в вершины возможно определение раскраски выходных дуг каждой вершины. Обозначим для каждой вершины $x \in N$ множество выходных вершин $x'=\{i|w(x,i) \geq 0\}$ и пусть для некоторой вершины $x'=\{j(1),j(2),\ldots\ j(m)\}$ (в общем случае возможно выполнение условия $m>C$).

Для каждой дуги $(x,j(k))$ выделим подмножество $I(k) \in x'$ вершин, из которых разрешено построение участков пути $(i,x,j(k))$, $k=\overline{1,m}$, $x \in N$. Отображение множества $I(k)$ на вектор $B'(x,j(k))=\{b'(x,j(k),m),\ m=\overline{1,C}\}$, задается выражением $b'(x,j(k),m) = \cup\{`b(j,x,m),\ j \in I(k)\},\ m=\overline{1,C},\ j(k) \in x'$.

В итоге получаем описание узловых ограничений на структуру путей на графе $G(N,M)$ в виде $cont(i,j)=\{k|(w(j,k) \geq 0)\ \&\ B'(j,k)\ \&\ `B(j,k)\},\ (i,j) \in M$.

Очевидно, что требуемая для кодирования раскраски память оценивается величиной $2C|M|$, что хорошо согласуется с экономным способом представления графов в виде списка смежности вершин. Если учесть дискретность процесса построения дерева путей, то каждому листу текущего дерева на разных этапах построения дерева соответствует единственный цвет дуги. Пусть $\{clr(i,j),\ i=\overline{1,|N|},\ j=\overline{1,C}\}$ – множество двоичных векторов, элементы которого отражают цвет дуг, по которой достигнуты вершины текущего дерева. На исходном шаге построения дерева из вершины s положим $clr(s,j)=1$, а на всех других шагах при включении в дерево дуги (v,i) – $clr(i,j)=`b(v,i,j),\ j=\overline{1,C}$. Тогда для развития дерева путей вместо $`B(i,k)$ можно использовать значение $\{clr(k,j),\ j=\overline{1,C}\}$, что обеспечивает возможность работы только с выходными дугами вершины k и ассоциированными с ними в списке смежности значениями $B'(k,j)$

и $w(k,j), j\in`k$. Таким образом, введение отображения раскраски листьев текущего дерева на множество $\{clr(.)\}$ позволяет воспользоваться преимуществами спискового представления как графа $G(N,M)$ [1], так и ограничений $\{cont(.)\}$.

Перейдем, наконец, к рассмотрению алгоритма построения дерева кратчайших путей при узловых ограничениях на структуру пути.

Шаг 1. Пусть задан граф $G(N,M)$, описание узловых ограничений на множествах $\{B'(i,j)\}$, $\{`B(i,j)\}$, $(i,j)\in M$, номер исходной вершины $s\in N$. Зарезервируем память для комплектов $\{R(i)\}$, $\{clr(i,.)\}$, $i=\overline{1,|N|}$.

Шаг 2. Положим $R(s)=0$, $R(i)=\infty$, $i=1,2,\ldots s\text{-}1, s+1,\ldots |N|$, $prec(s)=s$, $N'=\{s\}$, $clr(s,j)=1, j=\overline{1,C}$.

Шаг 3. Если $N'=\varnothing$, то построение дерева кратчайших путей завершено и его полностью характеризуют комплекты $\{R(.)\}$, $\{prec(.)\}$.

Шаг 4. Пусть вершина $v=argmin\{R(i),\ i\in N'\}$ имеет направленные дуги к вершинам из множества $v'=\{j(k),\ k=\overline{1,m(v)}\}$. Кратчайший путь (s,v) уже известен, поэтому положим $N':=N'\setminus\{v\}$, $m=0$.

Шаг 5. Если все выходные дуги вершины v просмотрены, т.е. $m=m(v)$, то возврат к шагу 3, иначе – $m:=m+1$.

Шаг 6. Если дуга $(m,j(m))$ не может быть продолжением пути из вершины v, т.е. $\cup\ \{clr(v,i)\cap`b(v,j(m),i),\ i=\overline{1,C}\}=\varnothing$, то возврат к шагу 5.

Шаг 7. Длина пути $(s,j(m))$ через вершину v: $r=R(v)+w(v,j(m))$.

Шаг 8. Если $R(j(m))<r$, то возврат к шагу 5, так как найденный ранее путь в вершину $j(m)$ предпочтительнее.

Шаг 9. Если $R(j(m))=\infty$ то $N':=N'\cup\{j(m)\}$.

Шаг 10. Фиксация описания нового пути в вершину $j(m)$:

$$R(j(m))=r,\ prec(j(m))=v;\ clr(j(m),j)=`b(v,j(m)),j),\ j=\overline{1,C}.$$

Шаг 11. Возврат к шагу 5 для продолжения ветвления дерева путей.

Таким образом, схема порождения дерева кратчайших путей с учетом ограничений на их структуру подобна схеме безусловной оптимизации, реализуемой в алгоритме Дийкстры [1]. Очевидно, что вычислительная сложность представленного алгоритма строго не хуже варианта безусловной оптимизации, для которого оценка сложности – $O(|N|^2)$ [1,3].

Литература. 1. Свами М., Тхуласираман К. Графы, сети и алгоритмы: Пер. с англ.– М.: Мир, 1984. – 455 с. **2.** Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2001. – 386 с. **3.** Ревотюк М.П. Поиск кратчайших путей со структурными ограничениями на графах неоднородных транспортных сетей. – Мн.: МРТИ, 1990. – 16 с. – Деп. в ВИНИТИ 08.06.90, № 3244-В90.

ОБ ОДНОМ АЛГОРИТМЕ ИСПОЛЬЗОВАНИЯ ВЕЙВЛЕТОВ ПРИ ОБРАБОТКЕ МЕДИЦИНСКОЙ ИНФОРМАЦИИ

Дубровина О.В., БНТУ, Минск

Вейвлет-преобразования (непрерывное и дискретное) находят широкое применение в различных научных и прикладных исследованиях. Данное направление оформилось как отдельная математическая дисциплина в середине 80-х годов XX века (см., например, [2, 5]). Одно из перспективных направлений использования вейвлет-преобразований – обработка различного рода сигналов, содержащих медицинскую информацию (см. [1]).

Целью данной работы является разработка методики применения непрерывного и дискретного вейвлет-преобразования при исследовании звуковых сигналов, полученных при кардиотокографии (см., например, [3]). Для этого необходимо ввести интегральное вейвлет-преобразование, рассмотреть различные возможности выбора вейвлетов и только затем перейти к обработке сигналов на практике моделируется непрерывной кривой, хотя, как правило, заключенная в нем информация носит дискретный характер. Поэтому необходимо дискретизировать как сигнал, так и преобразование. Предлагается процедура дискретизации, позволяющая в дальнейшем использовать стандартные программные средства.

Приведем формальные определения. Пусть $x(t) \in L_2(R)$ – некоторая функция, в дальнейшем называемая сигналом. Интегральным вейвлет-преобразованием сигнала $x(t)$ называется преобразование вида

$$W_\psi(a,b)(x) = w(a)\int_{-\infty}^{\infty} x(t)\overline{\psi\left(\frac{t-b}{a}\right)}dt,$$