АНАЛИЗ И МОДЕЛИРОВАНИЕ СЛОЖНЫХ СИСТЕМ

УДК 656.2

ПОИСК КРАТЧАЙШИХ ПУТЕЙ НА СЕТЯХ СО СТРУКТУРНЫМИ ОГРАНИЧЕНИЯМИ

Батура П.М., Шешко Е.В.
УО «Белорусский государственный университет информатики и радиоэлектроники», г. Минск
Научный руководитель – Ревотюк М.П., к. т.н, доцент

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 неотрицательное вещественное число $w(i,j)<\infty$, называемое длиной дуги. Если между вершинами s и t, $s,t\in N$ существует ориентированный путь, то длиной пути будем называть сумму длин дуг, образующих путь.

Обозначим для любой дуги $(i,j)\in M$ графа $G(M,N)$ множество допустимых вершин для развития путей из вершины $j\in N$ через $cont(i,j)$, а для любой вершины $x\in N$ множество ее смежных вершин – x': $x'=\{k|w(x,k)\geq 0\}$.

Очевидно, что $cont(i,j)\subseteq j'$, а в случае отсутствия ограничений на выбор пути после прохождения дуги (i,j) – $cont(i,j)\equiv j'$, $i,j\in N$, $(i,j)\in M$.

Решаемая задача: найти, если существует, на множестве вершин $\{s_0=s$, $s_1\in cont(s,s)$, $s_2\in cont(s_0,s_1),\dots$ $s_i\in cont(s_{i-2},s_{i-1}),\dots$ $t\}$ кратчайший путь от вершины s к вершине t, где $s,t\in N$.

Содержательный смысл функции $cont(i,j)$ может иметь отношение, например, к правилам дорожного движения, определяющих альтернативы выбора дуг в узле сети с учетом параметров транспортного средства. Другой пример – задачи управления подвижными единицами промышленного транспорта, когда необходимо учесть занятость участков сети на интервалах уже реализуемых маршрутов перевозки. Введение функции $cont(i,j)$ отражает намерение разделения модели транспортной сети и модели перемещения по ее дугам.

Модель графа $G(M,N)$ обычно задана структурой смежности в форме представления *FSF* (*Forward Star Form*) [1,2]:

$$FSF=\begin{cases} H_0=0, \\ H_{i+1}=H_i+|i'|,\ i=\overline{1,n-1}; \\ Y(H_i+j)=i'(j), \\ W(H_i+j)=w(i,i'(j)),\ j=\overline{0,|i'|-1},\ i=\overline{0,n-1}. \end{cases} \qquad (1)$$

Модель перемещений – отображение дополнительных параметров дуг на множество функций $cont(i,j)$, может быть определена системой продукционных правил, например, соответствующих правилам дорожного движения. Структура (1) при этом остается без изменений.

Ограничения на выбор пути после прохождения некоторой дуги $(i, j) \in M$ формально можно рассматривать как изменение структуры графа. Отсюда следует, что искомый алгоритм может быть построен путем целенаправленной модификации алгоритма Дийкстры [2]. Общая схема такого алгоритма соответствует жадной волновой схеме построения дерева кратчайших маршрутов от заданной исходной вершины $s \in N$ до всех остальных. Случай поиска кратчайшего пути до заданной конечной вершины $f \in N$ легко учитывается введением проверки необходимости развития волны после прохождения очередной вершины.

Однако при неудачном выборе метода расстановки пометок вершин в реализации алгоритма Дийкстры [2] можно получить неверный результат [3]. Действительно, пусть дерево кратчайших путей характеризуется множеством предшествующих вершин $\{P(x),\, x \in N\}$. Значение расстояния до вершин дерева обозначим через $\{R(x),\, x \in N\}$. Состояние процесса построения дерева при отсутствии ограничений пусть отражается в очереди вершин $\{Q(x),\, x \in N\}$. Элементы очереди $Q = \{q_1, q_2, \ldots q_i\}$ в этом случае должны быть упорядочены по значениям расстояний так, что $R(q_1) \le R(q_2) \le \ldots \le R(q_i)$.

Известный алгоритм поиска пути $s \to f$ на основе очереди вершин [2]:

```
for x∈N { // Инициализация  результата
 R(x)=∞;  P(x)=x;
}
R(s)=0; Q=s; // Фиксация корня дерева
do { // Организация ветвления
 x=Q--;
 if (x==f) break; // Прерывание цикла при достижении цели
 r=R(x);
 for (y∈x')
  if (R(y)<r+w(x,y)) {
   if (R(y)<∞) Q-=y;
   R(y)=r+w(x,y), P(y)=x;
   Q+=y;
  }
} while (Q≠∅);
```

На первый взгляд, в таком алгоритме достаточно заменить выражение $y \in x'$ на $y \in cont(P(x), x)$, и задача поиска кратчайших путей с ограничениями решена. Но характерная особенность рассматриваемой задачи – зависимость процесса расширения дерева путей от предыстории. Например, граф с ограничениями может включать цикл, когда пропущенные дуги некоторой вершины могут стать продолжением пути из последующих вершин пути. Другой пример – возврат в начальную вершину дуги после разрешенного правилами движения разворота. Как итог, решение о включении вершины в дерево кратчайших путей после ее выборки из очереди не является окончательным, хотя итерации поиска будут завершены.

Альтернативы развития путей, отражающие реальные ограничения, можно учесть перестройкой исходного графа [3]. Однако переход к поиску кратчайших путей на новом графе без учета ограничений требует не только расширения исходного графа, но и уточнения интерпретации получаемых результатов. Неудобства реализации подобного приема становятся практически неприемлемыми в многозадачной среде сервера обслуживания запросов, параметризованных относительно вида ограничений.

Вместе с тем можно исключить потребность перестройки исходного графа, если отображать процесс построения дерева кратчайших путей не на очередь вершин, а на очередь дуг $\{Q(x,y),\,(x,y)\in M\}$. Позиция дуги (x,y) в очереди дуг пусть, подобно очереди вершин, соответствует потенциалу ее конечной вершины $L(x,y),\,(x,y)\in M$. Просматриваемая дуга, помещенная в очередь, изменит свое состояние лишь один раз в момент выборки из очереди. Это не только упрощает структуру представления очереди (достаточно использования линейного списка), но и позволяет учесть различные условия развития дерева из любой вершины или дуги.

Алгоритм волновой схемы поиска пути $s \to f$ на основе очереди дуг [3]:

```
for (x∈N) { // Инициализация области данных результата
 R(x)=∞,  P(x)=x;
}
R(s)=0;
for ((x,y)∈A) { // Инициализация представления дерева дуг
 L(x,y)=∞, P(x,y)=location(x,y);
}
for (y∈s') { // Начальный узел дерева дуг
 L(s,y)=w(s,y), Q+=(s,y);
}
while (Q≠∅) {  // Расширение дерева дуг
 (x,y)=Q--; r=L(x,y), R(y)=r,  P(y)=x;
 if (y==f) break; // Прерывание цикла при достижении цели
 for ((z∈y')&&(y∈cont(y,z))
  if (L(y,z)<∞) {
   L(y,z)=r+w(y,z), P(y,z)=location(x,y);
   Q+=(y,z);
  }
 }
```

Недостаток алгоритма поиска на основе очереди дуг – дополнительная память порядка $O(|M|)-O(|N|)$. Однако свобода назначения ограничений на единственном экземпляре графа позволяет решать задачи поиска оптимальных путей перемещения разнородных подвижных единиц в реальном времени.

Список цитированных источников

1. Ревотюк, М.П. Реляционные модели задач оптимизации управления на интерпретируемых сетях/М.П.Ревотюк, З.А.Чан//Проблемы проектирования и производства радиоэлектронных средств: Сб. Материалов III Межд. НТК: в 2-х томах. – Новополоцк: ПГУ, 2004. – Том 2. – С. 139-141.
2. Кормен, Т. Алгоритмы: построение и анализ: [пер. с англ.] / Кормен Т., Лейзерсон Ч., Ривест Р. – М.: МЦМНО, 2002. – 960 с.
3. Ревотюк, М.П. Поиск кратчайших путей на графах полиморфных сетей с ограничениями / М.П.Ревотюк, И.Ю.Дарадкех, В.А. Кирейчук // Известия Белорусской инженерной академии. – 2004. – № 1(17)/1. – С. 126-129.