УСКОРЕНИЕ МНОГОКРАТНОГО ПОИСКА КРАТЧАЙШИХ ПУТЕЙ НА ГРАФАХ

М.П. Ревотюк, М.К. Кароли, О.В. Кот, В.В. Наймович
Кафедра информационных технологий автоматизированных систем,
Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: {rmp, kafitas}@bsuir.by

Предлагаются приемы ускорения многократного поиска кратчайших путей на графах, когда порядок порождаемых деревьев путей существенно меньше порядка графа. Однократная инициализация переменных состояния и выделение предопределенных решений снижает сложность поиска путей до линейной зависимости от объема сканируемого пространства.

I. Постановка задачи

Известно, что при поиске кратчайших путей на нагруженном ориентированном графе $G(N, A)$, где N – множество вершин, A – множество дуг с весовой функцией $W : A \to R^+$ время построения дерева путей одним из лучших для подобной задачи алгоритмом Дийкстры растет в первом приближении по закону $x \log_2 x$ с увеличением расстояния x от корня дерева.

Особенность алгоритма Дийкстры – однократный просмотр дуг формируемого дерева. Это отражается появлением в асимптотиках вычислительной сложности различных реализаций процедур поиска количества дуг. Например, реализация алгоритма Дийкстры с отображением очередей анализируемых вершин на кучи Фибоначчи характеризуется сложностью $O(m + n \log_2 n)$, где $m = |A|$, $n = |N|$. Отображение очереди вершин на вектор размером n позволяет снизить сложность до величины $O(m + nL)$, где L – максимальная длина дуги графа [1,2]. Однако построение дерева путей до вершины с расстоянием x от корня дерева при этом требует в среднем $x + nL$ операций. Значение nL соответствует операциям инициализации переменных состояния, которые далее будет предложено исключить.

II. Ревизия алгоритма Дийкстры

Известно, что вполне достаточным для реализации алгоритма Дийкстры представлением разреженного нагруженного графа $G(N, A)$ является структура смежности FSF (Forward Star Form) [1]. В этом случае для каждой вершины x эффективно представлено множество непосредственно достижимых смежных вершин x' , $x' = \{k|(w(x, k) \geq 0)\}$, где $w(x, k)$ – вес дуги $x \to k$, $x, k \in N$. Объем памяти для хранения структуры смежности – $O(m + n)$.

Пространство состояний поиска решения алгоритмом Дийкстры включает:

D – массив расстояний от корня дерева, $D = \{D_i, i \in N\}$;

P – массив номеров предшествующих вершин, $P = \{P(i), i \in N\}$;

Q – очередь вершин, $Q = \{Q_i, i \in N\}$, элементы которой упорядочены по текущему значению расстояния от корня дерева [1,2].

Анализ алгоритма Дийкстры показывает, что количество изменяющихся элементов пространства состояний поиска соответствует порядку итерационно создаваемого дерева кратчайших путей. Однако этот факт при кодировании алгоритма обычно не учитывается, неявно полагая степень исхода вершин графа соизмеримой со значением $n - 1$.

Процесс построения дерева кратчайших путей имеет волновой характер до исчерпания возможности его развития из исходной вершины. Алгоритм построения дерева с корнем в вершине r для любых схем организации очереди вершин имеет вид:

[H] $D_i \leftarrow \infty, P(i) = i, i \in N$;
$D_r \leftarrow 0, Q \leftarrow \{r\}$;
$Q \neq \oslash \;\; i = Q_1; Q \leftarrow Q \setminus \{i\}$;
$j \in i' \quad D_j > D_i + w(i, j)$
$D_j < \infty \;\; Q \leftarrow Q \setminus \{j\}; \;\; D_j \leftarrow D_i + w(i, j)$;
$P(j) = i; Q \leftarrow Q \cup \{i\}$;

На итерациях построения дерева последовательно выполняются операции:

- выборка вершины из очереди вершин с минимальным расстоянием от корня дерева;
- развитие дерева из выбранной вершины, когда для всех ее выходных дуг выполняется процедура релаксации с включением новых вершин в очередь.

Вершина графа может находиться в следующих последовательно фиксируемых состояниях: не рассматривалась, рассматривается и рассмотрена. Признак состояния явно обычно не используется, а его отображение проводится неявно на элементах $D = \{D_i, i \in N\}$. Все вершины графа перед построением дерева помечаются парами начальных значений $D_i \leftarrow \infty, P(i) \leftarrow i, i \in N$. Принадлежность произвольной вершины $x \in N$ дереву путей на любом этапе его построения определяется истинностью выражения $\overline{[D_i = \infty] \wedge [P(i) = i]}$. На практике именно вершины дерева путей значимы для задач управления транспортом.

Построенное дерево путей – связный граф по определению. Если s – исходная вершина, а t – лист дерева путей, $s,t \in N$, то после завершения поиска кратчайший путь можно восстановить обратным движением из вершины t: $t, P(t), P(P(t)), \ldots P(\ldots P(P(t))), s$.

Отображение состояния вершин на множествах D и P вынуждает каждый раз перед построением дерева устанавливать начальные значения для всех элементов. Вычислительная сложность такой операции – $O(n)$. Очевидно, что в случае построения кратчайших путей на больших графах количество вершин дерева может оказаться существенно меньше значения n.

Учитывая последовательный характер изменения состояния вершин дерева в соответствии со значением расстояния от его корня, очевидна идея включения такой последовательности в качестве этапа построения множества деревьев. Для ее реализации необходимо обеспечить условие распознавания исходного состояния вершин новых деревьев. Это легко достигается дополнением признака состояния вершин дерева номером такого дерева.

III. Новая схема поиска путей

Предлагаемая реализация обсуждаемой идеи базируется на дополнении пространства состояния глобальной переменной e с целью идентификации вершин разных деревьев. Это позволяет выполнить инициализацию массивов D и P один раз:

[H] $e \leftarrow 0$;

$P(i) \leftarrow e, i \in N$;

Модифицированная схема алгоритма Дийкстры поиска дерева путей между вершинами r и t имеет вид:

[H]

$P(r) \leftarrow r + e; D_r \leftarrow 0$;

$Q \leftarrow \{r\}$;

$Q \neq \varnothing$ $i = Q_1; Q \leftarrow Q \setminus \{i\}$;

$j \in i'$ $i = t$ goto finish $(P(j) \leq e)$ $D_j \leftarrow D_i + w(i,j)$; $P(j) = i + e; Q \leftarrow Q \cup \{i\}$; $D_j > D_i + w(i,j)$ $Q \leftarrow Q \setminus \{j\}$)); $D_j \leftarrow D_i + w(i,j)$; $P(j) = i + e$; $Q \leftarrow Q \cup \{i\}$;

finish: $e = e + n$

Переменная e содержит начало интервала номеров вершин очередного дерева. Построение дерева приводит к изменению элементов массива P, соответствующих подвергшимся анализу вершинам графа. Предикат $[P(j) \leq e]$ определяет множество не рассмотренных вершин и соответствует состоянию $D_j = \infty, j \in N$. Однако использование такого предиката позволяет исключить повторную установку начального состояния множеств P и D. Вычислительная сложность построения дерева путей зависит лишь от количества фактически рассматриваемых дуг.

IV. Операции над очередями вершин

В момент достижения конечной вершины при поиске дерева путей между вершинами r и t в очереди вершин остаются фиксированными все листья дерева. Перед построением нового дерева возникает необходимость очистки очереди, что требует $O(n)$ операций.

Идея однократной инициализации массивов D и P может быть использована для отказа от очистки очереди Q. Для этого достаточно операцию $Q \leftarrow \{r\}$ в момент фиксации корня дерева в вершине r выполнить с проверкой условия $(r \leq e)$. Список элементов очереди не обязательно проецировать на интервал $\overline{1,n}$, если порядок дерева существенно меньше n.

Заключение

Таким образом,предложенный прием нумерации запросов на поиск кратчайших путей позволяет исключить холостые шаги инициализации переменных состояния поиска. В результате вычислительная сложность задачи для наиболее эффективных адресных схем организации очередей линейно зависит от количества фактически просмотренных дуг графа при построении дерева кратчайших путей. Размер исходного графа не влияет на количество шагов инициализации переменных состояния перед построением очередного дерева.

1. Deo, N. Shortest-path algorithms: Taxonomy and annotation /N. Deo,Deo, Chi-Yin Pang//Networks. – 1984. –Vol. 14(2). –P. 275–323.
2. Fredman,M.L. Fibonacci heaps and their uses in improved network optimization algorithms/Fredman M.L., Tarjan R.E.//J.ACM. –1987. –Vol.34(3). –P. 596-–615.
3. Demetrescu, C. Experimental analysis of dynamic all pairs shortest path algorithms/Demetrescu C., Italiano G.F.//ACM Transactions on Algorithms. – 2006. –No. 2(4). –P. 578–601.
4. Ревотюк, М. П. Быстрый поиск кратчайших путей на графах с предопределенными решениями /М. П. Ревотюк, М. К. Кароли, Н. В. Хаджинова //Доклады БГУИР. –2014. – № 4(82). – С. 73–79.