АКСЕЛЕРАТОРЫ ПРОЦЕДУР ЛОГИЧЕСКОГО ВЫВОДА В СИСТЕМАХ ОЦЕНКИ КОМПЕТЕНЦИЙ

М.П. Ревотюк, М.К. Кароли, Т.В. Тиханович

Белорусский государственный университет информатики и радиоэлектроники
Минск, Беларусь, rmp@bsuir.by

Abstract. The problem of evaluating competencies was considered in terms of search and analysis of shortest paths on weighted graphs projection systems of formal productions that reflect the subject area. The proposed procedure for accelerate the search is based on a compact mapping of shortest paths on the arc of the graph.

Задачи оценки компетенций могут рассматриваться в терминах поиска и анализа кратчайших путей на нагруженных графах проекции систем формальных продукций, отражающих предметную область и действия экзаменуемого персонала.

Известно, что при поиске кратчайших путей на нагруженном ориентированном графе $G(N,A)$, где N – множество вершин, A – множество дуг, время построения дерева путей растет квадратично или, по меньшей мере, при тщательно построенной вычислительной схеме по закону $O(m+n\cdot\log_2 n)$, где $m=|A|, n=|N|$.

Известные приемы ускорения процесса поиска кратчайших путей, такие как целенаправленный поиск, встречный поиск, многоуровневый подход [2], базируются на ограничении локальных областей поиска. Однако для абстрактных графов систем формальных продукций проблемным является выделение таких областей.

Далее будем полагать, что построение дерева путей ведется алгоритмом Дийкстры по волновой схеме однократного просмотра дуг. Состояние поиска решения представляется массивом расстояний $D=\{D_i, i\in N\}$, а также очередью вершин, элементы которой упорядочены по значению расстояния от корня дерева.

Можно заметить, что в процессе развития дерева кратчайших путей каждая вершина окончательного дерева, как минимум, один раз будет представлена в очереди вершин. Вершины в очереди находятся до момента выбора входных дуг минимальной длины. Множество таких дуг $T_j=\{(i,j): i=\arg\min_{i,j}\{w(i,j):(i,j)\in A\}\}$, $j\in N$, можно выделить до начала поиска[1]. Известный более эффективный прием уменьшения количества просматриваемых дуг графа – каждой дуге поставить в соответствие список вершин, кратчайшие пути к которым к которым включают такую дугу[2]. Построение подобных списков требует предварительного построения всех деревьев кратчайших путей. Однако в дальнейшем процесс поиска дерева путей можно вести на разреженном графе, дуги которого ассоциированы с целевой вершиной.

Задачи с фиксацией целевой вершины часто встречаются на практике. В этом случае целесообразно организовать встречный поиск, тогда количество анализируемых дуг сокращается в два раза. Предлагается учесть ассоциации дуг с целевыми вершинами не списками, а характеристическими множествами признаков вхождения вершин в заранее выделенные подмножества вершин. При этом процедура нумерация вершин не нуждается в отражении топологии графа, а оценка потребности в дополнительной памяти – $O(m\cdot\log_2 n)$.

Обозначим исходный граф через $G^+(N^+,A^+)$, а граф с инвертированием направления дуг – $G^*(N^*,A^*)$. Очевидно, что множества дуг таких графов могут не включать дуги без ассоциаций с начальной и конечными вершинами пути.

С целью удобства организации процесса ветвления желательно использовать общую очередь, для чего множество вершин и дуг графа $G^*(N^*, A^*)$ определим так:

$$N^* = \{x^* = x^+ + n,\ x^+ \in N^+\},\ A^* = \{(x^*, y^*) = (y^+ + n,\ x^+ + n),\ (x^+, y^+) \in A^+\} \quad (1)$$

В общем случае связь сопряженных вершин x^+ и x^* пусть задается функцией

$$conj(x) = x^* \cdot (x \in G^+) + x^+ \cdot (x \in G^*),\ x \in N^+ \cup N^*. \quad (2)$$

В случае же нумерации вершин по правилу (1) связь вершин имеет вид

$$conj(x) = (x+n)\cdot(x<n) + (x-n)\cdot(x \geq n)\ x \in N^+ \cup N^* \quad (3).$$

Пусть заданы s и f – начальная и конечная вершины исходного графа G^+. Так как $N^+ \cap N^* = \varnothing$, то достаточно начать процесс ветвления из вершин $s \in G^+$ и $f^* = conj(f)$, $f^* \in G^*$. Можно показать, что остановка поиска должна соответствовать моменту фиксации постоянной пометки вершины дерева, когда сопряженная вершина уже является постоянно помеченной.

Если для некоторого дерева кратчайших маршрутов максимальное расстояние от постоянно помеченных вершин до корня есть d, то признаком постоянной пометки вершины x является условие $D_x \leq d$. В рассматриваемом случае для обоих деревьев значение d одинаково. Отсюда следует, что правило остановки можно определить на значениях текущих расстояний – $D_{conj(i)} \leq D_i$, где i – вершина графа $G(N^+, A^+)$ или графа $G(N^*, A^*)$, получающая постоянную пометку.

Таким образом, использование синхронного движения от корней деревьев не требует хранения пометок, а момент остановки совпадает с моментами постоянной пометки вершин дерева маршрутов.

В случае наличия предопределенных решений правило остановки можно определить относительно фактов пометки обеих вершин анализируемой дуги графов G^+ или G^*. Такие решения можно фиксировать по ходу поиска. Факт пометки конечной вершины дуги, имеющей постоянную пометку в инвертированном графе – достаточное условие остановки. Правило остановки здесь имеет вид $(D(y) < \infty) \wedge ((x, y) \in T_j),\ y = conj(x)$. После остановки в вершине x остается достроить маршрут до конечной вершины в исходном графе. Так как остановка может быть обнаружена в любом из встречно растущих деревьев, а результат поиска необходимо получить лишь для дерева из исходной вершины, то переход в такое дерево реализует функция $orig(x) = x \cdot (x \in G^+) + conj(x) \cdot (x \in G^*),\ x \in N^+ \cup N^*$. В случае нумерации вершин расширенного графа по правилам (1) и (2) $orig(x) = x \cdot (x < n) + (x-n) \cdot (x \geq n),\ x \in N^+ \cup N^*$.

Таким образом, построенные процедуры поиска используют для представления модели сети память в два раза увеличенного объема. Эксперименты показывают, что среднее время поиска кратчайших маршрутов между случайными парами вершин сокращается существенно более, чем в два раза. Степень сокращения зависит от количества подмножеств вершин кратчайших путей, ассоциируемых с дугами. Рассмотренный подход пригоден как для статически, так и динамически формируемых графов.

Литература

1. Ревотюк, М.П. Поглощение предопределенных решений жадными алгоритмами/ М.П. Ревотюк [и др.]//Известия Белорусской инженерной академии. – № 1(17), 2004. – С. 112 –114.

2. Holzer, M. Combining Speed-up Techniques for Shortest-Path Computations/M. Holzer, F. Schulz, D. Wagner, T. Wilhalm//ACM Journal of Experimental Algorithmics. – Vol. 10, No. 2.5, 2005. – P. 1–18.