НАХОЖДЕНИЕ КРАТЧАЙШИХ ПУТЕЙ В ГРАФЕ МЕТОДОМ МАТРИЧНОГО УМНОЖЕНИЯ

Плотников В. В., Кресс В. Д.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Болтак С. В. – ассистент

В работе было проведено исследование одного из методов решения проблемы APSP (All Pairs Shortest Path) в алгоритмической теории графов с использованием подхода динамического программирования. Была прослежена связь решения проблемы с перемножением матриц, а также был осуществлен вывод сложности с её последующим улучшением.

Любой динамически программируемый алгоритм включает в себя следующие этапы: задание общей структуры оптимального решения поставленной задачи, рекурсивное определение оптимального решения, вычисление значения, процесс конструирования решения.

Пусть граф задан матрицей смежности $W = (w_{ij})$. Рассмотрим кратчайший путь p из вершины i в вершину j, причем p содержит не более чем m дуг графа. В случае, если $i = j$, путь p имеет нулевой вес и не содержит дуг. В случае, если $i \neq j$, проведем разложение пути p в $i \xrightarrow{p'} k \rightarrow j$, где p' – путь, содержащий не более чем m - 1 дуг графа. Таким образом, общая структура наикратчайшего пути из вершины i в вершину j может быть задана формулой:

$$\delta(i, j) = \delta(i, k) + w_{kj}, \tag{1}$$

где i, j, k – вершины графа.

Пусть l_{ij}^{m} – вес самого короткого пути из вершины i в вершину j с не более чем m дугами. Тогда справедлива следующая рекуррентная формула:

$$l_{ij}^{m} = \min_{1 \le k \le n}[l_{ik}^{\{m-1\}} + w_{kj}], \tag{2}$$

где *i*, *j*, *k* – вершины графа.

Тогда вес кратчайшего пути из вершины i в вершину j может быть задан следующим образом:

$$\delta(i, j) = l_{ij}^{\{n-1\}} = l_{ij}^{\{n\}} = l_{ij}^{\{n+1\}} = \cdots. \tag{3}$$

В алгоритме происходит последовательное вычисление матриц вида $L^{(m)} = (l_{ij}^{m})$. Для вычисления очередной матрицы используется функция $\mathrm{ESP}(L, W)$.

$$
\begin{array}{l}
\boldsymbol{function}\ ESP(L, W) \\
n = L \rightarrow getRowsCount() \\
L' = (l'_{ij}) - \text{новая матрица размером } n \times n \\
\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ n \\
\quad \boldsymbol{for}\ j = 1\ \boldsymbol{to}\ n \\
\qquad l'_{ij} = \infty \\
\qquad \boldsymbol{for}\ k = 1\ \boldsymbol{to}\ n \\
\qquad\quad l'_{ij} = \boldsymbol{min}\,(l'_{ij}, l_{ik} + w_{kj}) \\
\boldsymbol{return}\ L'
\end{array}
$$

Пользуясь повторным возведением в квадрат, можно сконструировать следующий алгоритм для решения задачи APSP с общей сложностью $\Theta(n^3 \ln n)$, где n – количество вершин графа.

$$
\begin{array}{l}
L^{(1)} = W \\
m = 1 \\
\boldsymbol{while}\ m < n - 1 \\
\qquad L^{(2m)} = \mathrm{ESP}(L^{(m)}, L^{(m)}) \\
\qquad m = 2 * m \\
\boldsymbol{return}\ L^{(m)}
\end{array}
$$

Список использованных источников:

1. *C++ Programming Robert Sedgewick - Princeton University Addison Wesley Professional Algorithms in C++, Parts 1A4: Fundamentals, Data Structure, Sorting, Searching, Third Edition – 752 p.*
2. *Introduction to algorithms / Thomas H. Cormen . . . [et al.].—3rd ed – P. 686-695.*
3. *Library of Congress Cataloging-in-Publication Data Sedgewick, Robert, 1946 Algorithms in C++/Robert Sedgewick.—3d ed.*