УДК 004.021+519.168
DOI: 10.46548/21vek-2020-0951-0006

АНАЛИЗ ВРЕМЕННОЙ СЛОЖНОСТИ СПЕЦИАЛИЗИРОВАННОГО АЛГОРИТМА ПОИСКА МНОЖЕСТВА КРАТЧАЙШИХ ПУТЕЙ МЕЖДУ НАСЕЛЕННЫМИ ПУНКТАМИ

Костин Алексей Владимирович, начальник Управления цифрового развития, информационных технологий и связи Пензенской области
Управление цифрового развития, информационных технологий и связи Пензенской области
(440000, Россия, Пенза, улица Московская, 75, e-mail: kostin@obl.penza.net)

Аннотация. Возможности практического использования алгоритмов зависят от их характеристик, в частности, от продолжительности решения задач по выбранному алгоритму при различных значениях параметров, характеризующих размерность задачи. Цель работы – анализ временной сложности вариантов специализированного алгоритма поиска совокупности кратчайших путей между заданными множествами населенных пунктов для анализа транспортной доступности в регионе. В качестве модели для задачи поиска кратчайших путей между указанными объектами используется неориентированный или смешанный граф с заданными весами рёбер (и, необязательно, дуг), в котором множество вершин разделено на три подмножества, представляющих населенные пункты, районные центры и узлы дорожной сети. В данной постановке задачи нужно для каждой вершины, представляющей населенный пункт из заданного подмножества, ближайшую к ней вершину из заданного подмножества вершин, представляющих районные центры. Для получения априорных оценок временной сложности используются описания структуры двух вариантов алгоритма поиска путей из одной вершины, представляющей населенный пункт до множества вершин, представляющих заданные районные центры. В качестве параметров размерности задачи используются: мощность множества вершин моделирующего графа, мощности заданных подмножеств вершин, представляющих заданные районные центры, степени вершин и другие. Получены выражения для оценки временной сложности вариантов алгоритма, включающие показательную и степенную функции от мощности множества вершин моделирующего графа. Результаты экспериментального исследования программной реализации варианта алгоритма с меньшей априорной оценкой временной сложности показали, что его временная сложность является линейной функцией от мощности множества вершин, представляющих населенные пункты, при фиксированном составе графа, представляющего транспортную сеть региона.

Ключевые слова: анализ временной сложности алгоритма, асимптотические оценки, экспериментальное исследование временной сложности алгоритмов, неориентированный граф, алгоритм поиска кратчайших путей.

THE TIME COMPLEXITY ANALYSIS OF A SPECIALIZED ALGORITHM FOR FINDING A SET OF SHORTEST PATHS

Kostin Aleksey Vladimirovich, head of Administration of digital development, information technologies and communications of the Penza region
Administration of digital development, information technologies and communications of the Penza region
(440000, Russia, Penza, Moskovskaya street, 75, e-mail: kostin@obl.penza.net)

Abstract. The possibilities of practical use of algorithms depend on their characteristics, in particular, on the duration of solving problems using the chosen algorithm for different values of parameters that characterize the dimension of the problem. The work purpose – analysis of the time complexity of variants of a specialized algorithm for finding a set of shortest paths between specified sets of localities for analyzing transport accessibility in the region. As a model for the problem of finding the shortest paths between the specified objects, an undirected or mixed graph with specified weights of edges (and, optionally, arcs) is used, in which the set of vertexes is divided into three subsets representing localities, district centers, and road network nodes. In this problem statement, for each vertex that represents a locality from a given subset, the nearest vertex from a given subset of vertexes that represent district centers is required. To obtain a priori estimates of time complexity, we use descriptions of the structure of two variants of the algorithm for finding paths from one vertex representing a locality to a set of vertices representing the specified district centers. As parameters the dimension of the problem are used: the cardinality of the set of vertices of the simulating graph of the power given subsets of vertices, representing a set of district centers, the degree of each vertex and others. Expressions are obtained for estimating the time complexity of the algorithm variants, including exponential and power functions of the power of the set of vertices of the modeling graph. Experimental results the software implementation of the algorithm with a lesser a priori estimation of the time complexity has shown that its time complexity is a linear function of the power set of vertices, representing the towns, with a fixed part of the graph representing the transportation network of the region.

Keywords: time complexity analysis of the algorithm, asymptotic estimate, experimental study of the time complexity of algorithms, undirected graph, shortest path search algorithm.

Введение. В предыдущей публикации «Разработка алгоритма решения задачи анализа транспортной доступности» (XXI век: итоги прошлого и проблемы настоящего плюс. 2019, том 8, № 3 (47). С. 12 – 16) рассмотрена формализация задачи анализа транспортной доступности населенных пунктов, которая сводит указанную задачу анализа к поиску совокупности кратчайших путей между двумя заданными подмножествами вершин N и R неориентированного или смешанного графа $G(X, U)$ с заданными весами рёбер (и, необязательно, дуг), где X - множество вершин, U - множество рёбер, при этом:

$$N \subset X;\ R \subset X;\ X = R \cup N \cup C;\ \ C = X \setminus (R \cup N;\ R \cap N \cap C = \emptyset.$$

Для решения поставленной задачи предложен алгоритм решения на основе алгоритма Дейкстры [1 – 3]. Поскольку размерность практических задач достаточно велика (количество вершин моделирующего графа порядка 10^3, количество рёбер порядка 10^4), то требуется оценить временную сложность разработанных вариантов алгоритма с целью прогнозирования возможности их практического использования на задачах различной размерности [4]. Рассматриваемая задача относится к «переборным», для решения которых должны использоваться алгоритмы с полиномиальной временной сложностью, которые относят к эффективным [5, 6].

Анализ временной сложности алгоритмов для оценки вида функции (линейная, полиномиальная и др.) может быть выполнен по формальному представлению алгоритма (например, по блок-схеме) [7 – 10] и/или экспериментально [10,11]. При выполнении анализа временной сложности принято, что запись $f(n)=O(g(n))$ означает принадлежность функции f классу $O(g)$, т.е. значения функции f не превышают значений функции g для достаточно больших значений аргумента: $\exists n0 > 0,\ c > 0 \mid \forall n > n0,\ f(n) \le c \cdot g(n)$ [6 – 16].

В работе выполнен анализ следующих вариантов алгоритма:

● непосредственное применение алгоритма Дейкстры для поиска вершины $x_j \in R$, ближайшей к вершине $x_i \in N$: поочередный поиск кратчайшего пути $D(x_i, x_j)$ от каждой $x_i \in N$ до каждой $x_j \in R$ и последующий выбор для каждой $x_i \in N$ ближайшей вершины из подмножества R на основе значений $D(x_i, x_j)$. Этот вариант включает $|R|$ процессов «распространения волны» по графу (окрашивания вершин, смежных с ранее окрашенными [1 – 3]), после каждого из которых выполняется поиск наименьшего значения в массиве из $|R|$ значений $D(x_i, x_j)$;

● модификация алгоритма Дейкстры: поиск $|R|$ путей для каждой $x_i \in N$ за один «проход» (цикл процесса «распространения волны» по графу от $x_i \in N$ до всех $x_j \in R$), из которых выбирается путь минимальной длины. Этот вариант включает один процесс «распространения волны» по графу, но продолжительность каждого процесса «распространения волны» больше, чем в первом варианте, т. к. этот процесс от каждой $x_i \in N$ заканчивается при достижении всех $x_j \in R$.

Для получения асимптотических оценок временной сложности алгоритмов решения поставленной задачи по формальному представлению алгоритма используются:

● полный граф, включающий $|X|$ вершин;

● граф, образующий Гамильтонов цикл ($|X|$ вершин и не менее ($|X|$+1) ребер) [1, 2].

Материалы и методы исследования. *<u>Анализ временной сложности первого варианта алгоритма по формализованному описанию.</u>*

Исходные данные:

● списки вершин подмножеств N и R (при проведении исследований можно использовать порядковые номера вершин вместо названий объектов, табл. 1);

● матрица смежности (расстояний).

Для промежуточных результатов используются (табл. 1):

● массивы расстояний от вершины $x_i \in N$ до достигнутых вершин $x_j \in R$, $x_k \in N$ (и до $x_k \in C$, которые в табл. 1 не показаны) на очередном шаге алгоритма;

● массивы предков для каждой из достигнутых вершин на очередном шаге алгоритма;

● массив промежуточных результатов для записи расстояний от каждой $x_i \in N$ до каждой $x_j \in R$.

Таблица 1 – Таблица для представления промежуточных результатов (подмножества вершин N и R, массивы кратчайших расстояний от вершины $x_i \in N$ до вершин $x_k \in F_2$)

Данные вершин множества N			Данные вершин множества R		
Имена $x_i \in N$	Расстояния между $x_i \in N$ (массив D_N)	Предки x_k	Имена $x_j \in R$	Расстояния от $x_i \in N$ до $x_j \in R$ (массив D_R)	Предки для $x_j \in R$
x_1	$d(x_k, x_1)$	x_l	$x_{R,1}$	$d(x_1, x_{R,1})$	
x_2	$d(x_k, x_2)$	x_m	$x_{R,2}$	$d(x_2, x_{R,2})$	
...	…	...	...	...	…
x_i	0	0	$x_{R,j}$	$d(x_i, x_{R,j})$	
…	…	…	…	…	…

Вариант 1 алгоритма поиска кратчайшего пути от заданной вершины $x_i \in N$ до ближайшей из множества R следующие составные части:

1. Цикл по массиву R: $j=1$ (номер вершины, до которой надо найти кратчайший путь из вершины $x_i \in N$).

2. Инициализация массивов предков, расстояний, фронтов.

3. Цикл до достижения до заданной вершины:

3.1. $F_1=F_2$, $F_2=\emptyset$.

3.2. Цикл формирования следующего фронта F_2 (определение соседних вершин для всех $x \in F_1$).

3.3. Цикл вычисления длины пути от x_i до каждой из вершин $x_k \in F_2$ и выбор ближайшей вершины.

3.4. Если целевая вершина x_j не достигнута, то переход к п. 3.1, иначе – к следующему пункту.

4. Запись в массив промежуточных результатов строки (x_i, x_j, $d(x_i, x_j)$).

5. Переход к следующей целевой вершине: $j=j+1;$ если $j \le |R|$, то перейти к п. 2, иначе – к следующему.

6. Выбор для x_i в массиве промежуточных результатов строки с наименьшим значением $d(x_i, x_j)$, в ко-

торой записан номер (идентификатор) ближайшей к xi вершины $x_j \in R$ (циклическая процедура поиска наименьшего числа в массиве длиной $|R|$).

На основе анализа описания алгоритма 1 получено общее выражение для оценки продолжительности решения задачи, в котором индексы обозначают этапы (шаги) алгоритма:

$$T_{A1}=T_1[T_2+T_3(T_{3.1}+T_{3.2}+T_{3.3})+T_4+T_5]+T_6. \qquad (1)$$

Определяем зависимость составляющих T_{A1} от характеристик графа и условий задачи.

Цикл №1 выполняется $r=|R|$ раз, тогда $T_1=rt_1$, где t_1 – продолжительность операций, связанных только с циклом 1 (без учета внутренних циклов). Продолжительность выполнения шагов 2, 4, 5 может быть принята постоянной, не зависящей от характеристик графа.

Цикл 3 (до достижения целевой вершины) выполняется для произвольного графа в среднем k_{cp} раз (среднее количество ребер в пути от $x_i \in N$ до $x_j \in R$), тогда $T_3=k_{cp}\,t_3$; $k_{cp}=\frac{1}{|R|}\sum_{i=1}^{|R|}k_i$, где k_i - количество ребер в пути от $x_i \in N$ до ближайшей $x_j \in R$.

Если рассматривать теоретически возможные модели, количество повторений цикла 3 составит:

- для полного графа 1 раз,
- для Гамильтонова графа максимум ($|X|$-1) раз.

Количество повторений внутренних циклов (3.1 и 3.2) зависит от степеней вершин, поэтому здесь необходимо определить значение этого показателя. Поскольку рассматривается связный граф, то минимальное значение степени вершины равно 2 (для Гамильтонова графа), максимальное ($|X|$-1) для полного графа, среднее значение зависит от предметной области (обозначим ρ_{cp}). Значит фронт F_2 составляют:

- в Гамильтоновом графе 1 вершина;
- в полном графе ($|X|$-1) вершин.

Тогда:

- для Гамильтонова графа $T_{Гг,3.1}=t_{3.1}$; $T_{Гг,3.2}=t_{3.2}$; $T_{Гг,3.3}=0$;
- для полного графа $T_{пг,3.1}=(|X|-1)t_{3.1}$; $T_{пг,3.2}=(|X|-1)t_{3.2}$; $T_{пг,3.3}=(|X|-1)t_{3.3}$ или $T_{пг,3.3}=rt_{3.3}$ (если имеется проверка $x_j \in R$).

Продолжительность выполнения этапа 3.1 для произвольного графа (перепись вершин из F_2 и в $F1$) и 3.2 зависит от количества вершин во фронте F_2, которое, в свою очередь, зависит от степеней вершин (ρ) в предыдущем фронте и количества альтернативных путей между парами вершин в F_1 и F_2, при этом нужно учитывать, что эти этапы выполняются последовательно. Минимальное значение, равное 1, будет для Гамильтонова графа; для полного графа этот этап не выполняется, т. к. цикл 3 выполняется только один раз, а на первом шаге $|F_2|=0$. Для произвольного графа следует оценить наибольшее количество вершин в F_2, обозначим $|F_2, max|$; тогда $T_{3.1} \le |F_{2,max}|t_{3.1}$.

Значение $|F_2, max|$ можно оценить следующим образом:

- примем, что при распространении волны по графу количество вершин на следующем шаге (формировании следующего фронта) определяется по закону геометрической прогрессии $|F_{2,max}| \le \rho_{cp}^{K-1}$ (при $|F_1|=1$ на первом шаге), где K - номер шага при распространении волны; ρ_{cp} – средняя степень вершины;
- наибольшее суммарное количество вершин, которые включались в состав F_1 и F_2 до следующего ($K+1$)-го шага, должно быть не меньше $|X|/2$, т.к. на последующих шагах это количество будет сокращаться. Тогда из выражения для суммы членов геометрической прогрессии получим:

$$\frac{|X|}{2} \le \frac{\rho^K-1}{\rho-1}; \quad K \cong \frac{1}{log_2\rho}[log_2|X|+log_2(\rho-1)-1].$$

Приняв $\rho=\rho_{cp}=4$ получим для первого варианта алгоритма номер шага распространения волны, при котором количество вершин в массиве F_2 наибольшее: $K_{1,max} \cong \log_2{}^{|X|/2}$; тогда

$$T_{3.1,max} \le 2^{log2\,|X|/2}\,t_{3.1}. \qquad (2)$$

Если, как указано в преамбуле, количество вершин в транспортной задаче порядка 103 - 104, то приняв $|X|=216$, получим $K_{max}=8$ и, соответственно, $|F_{2,\ max}|=214$. Среднее значение можно оценить, как $K_{cp}=(K_{min}+K_{1,max})/2$ при $K_{min}=\rho_{cp}$. Используя указанные выше числовые оценки ρ_{cp} и $K_{1,max}$ получим $K_{cp}=6$.

Наибольшее значение продолжительности выполнения шага 3.2 ($T_{3.2}$) можно оценить, используя оценку значения $|F_2,\ max|$, т. е. можно принять, что $T_{3.2} \le |F_{2,max}|t_{3.2}$.

Количество повторений цикла 3.3 зависит от количества вершин во фронтах (массивах) F_1 и F_2:

$n_{3.3}=|F_1|\cdot|F_2|$.

Приняв, что количество вершин на следующем, K-м, шаге определяется по закону геометрической прогрессии, получим:

$n_{3.3} \le \rho^{K-1}\rho^{K-2}=\rho^{2K-3}$.

Подставив в (1) выражения для составляющих оценки продолжительности решения задачи, получим:

- для произвольного графа:

$$T_{A1}=r\{t_1[C_2+k_{cp}t_3(2^{\wedge log2|X|/2})(t_{3.1}+t_{3.2})+\rho_{cp}{}^{2K1,max-3}t_{3.3})+C_4+C_5]+t_6\}; \qquad (3)$$

- для Гамильтонова графа:

$$T_{A1,Гг}=rt_1[C_2+(|X|-1)t_3(t_{3.1}+t_{3.2})+C_4+C_5];$$

- для полного графа:

$$T_{A1,пг}=rt_1[C_2+(|X|-1)t_3(t_{3.1}+t_{3.2}+t_{3.3})+C_4+C_5].$$

Выполнение этапа 6 (поиск минимального $d(x_i, x_j)$ для полного и для Гамильтонова графа не требуется в силу свойств графов (в обоих случаях результат выполнения этапов 1 – 5 позволяет получить единственный результат).

Анализ показал, что продолжительность решения задачи по варианту 1 алгоритма зависит от мощности множеств X и R при фиксированной матрице смежности и, соответственно, при фиксированных значениях статистических характеристик графа (ρ_{cp}, k_{cp}, $K_{1,max}$).

Анализ временной сложности второго варианта алгоритма по формализованному описанию. Исходные данные и формы представления промежуточных результатов одинаковы с первым вариантом.

Вариант 2 алгоритма поиска кратчайшего пути от заданной вершины $x_i \in N$ до ближайшей из множества R:

1. Инициализация массивов предков, расстояний, фронтов.

2. Цикл до достижения всех вершин $x_j \in R$:

2.1. $F_1=F_2$, $F_2=\varnothing$.

2.2. Цикл формирования следующего фронта F_2 (определение соседних вершин для всех $x \in F_1$).

2.3. Если $F_2=\varnothing$, то переход к п. 3, иначе – к следующему.

2.4. Цикл вычисления длины пути от x_i до каждой из вершин $x_k \in F_2$ и выбор кратчайшего до каждой из вершин $x_k \in F_2$; запись в массив D_N, D_R, D_C и в соответствующие массивы предков (для $x_i \in N$, $x_j \in R$, $x_k \in C$).

2.5. Если все вершина $x_j \in R$ достигнуты, то переход к п. 3, иначе – к п. 2.1.

3. Выбор для x_i в массиве промежуточных результатов строки с наименьшим значением $d(x_i, x_j)$, в которой записан номер (идентификатор) ближайшей к x_i вершины $x_j \in R$ (циклическая процедура поиска наименьшего числа в массиве длиной $|R|$).

На основе анализа описания алгоритма 2 получено общее выражение для оценки продолжительности решения задачи, в котором индексы обозначают этапы алгоритма:

$$T_{A2}=T_1+T_2(T_{2.1}+T_{2.2}+T_{2.3}+T_{2.4}+T_{2.5})+T_3. \quad (4)$$

Определяем зависимость составляющих T_{A2} от характеристик графа и условий задачи.

Количество повторений операций в цикле до достижения всех вершин $x_j \in R$ существенно зависит от общего количества вершин в графе и от их связности (общего количества ребер/дуг и степеней вершин); в общем случае эта величина определяется как

$$k_{2,max} = \max\{k_i\}, i = \overline{1,|R|}, \quad (5)$$

где k_i – количество ребер/дуг от $x_i \in N$ до $x_j \in R$.

Тогда $T_2=k_{2,max}t_2$, где t_2 – продолжительность операций, связанных только с циклом 2 (без учета внутренних циклов). Продолжительность выполнения этапов 1 и 2.3 может быть принята постоянной, не зависящей от характеристик графа. Продолжительность выполнения этапов 2.1 и 2.2 одинакова с продолжительностью шагов 3.1 и 3.2 в первом варианте. Продолжительность выполнения этапа 2.4 может быть оценена по аналогии с продолжительностью шага 3.3 в первом варианте. Продолжительность выполнения этапа 2.5 (проверка наличия вершины-предка для каждой из $x_j \in R$) пропорциональна $|R|$.

Цикл 3 (циклическая процедура поиска наименьшего числа в массиве длиной $|R|$) выполняется для произвольного графа $|R|$ раз. Подставив в (4) выражения (2), (5) получим:

$$T_{A2}=t_1+k_{2,max}t_2[2^{log2|X|/2}(t_{2.1}+t_{2.2})+t_{2.3}+ \rho_{cp}^{2K_{2,max}-3}\, t_{2.4}+rt_{2.5}]+rt_3. \quad (6)$$

Сравнивая (3) и (6) можно отметить, что на продолжительность решения задачи значение $|X|$ влияет примерно одинаково в обоих вариантах алгоритма, а значение $|R|$ оказывает большее влияние в первом варианте алгоритма. Для сравнения надо принять значения констант, выполнить преобразования (3) и (6), и задать числовые значения.

Приняв в (3) и (6) значения констант равными 1, получим

$$T_{A1} = r\{[k_{cp}\left(2^{log_2|X|/2}\left(2+\rho_{cp}^{2K_{1,max}-3}\right)+3\right]+1\}; \quad (7)$$

$$T_{A2} = k_{2,max}\left[2^{1+\frac{log_2|X|}{2}}+\rho_{cp}^{2K_{2,max}-3}+r+1\right]+r+1 \quad (8)$$

В обоих алгоритмах наибольшую долю в продолжительности выполнения занимает вычисление и выбор ближайших вершин (цикл 3.3. в первом варианте и цикл 2.4 – во втором). Анализ (7) и (8) показывает, что в обоих выражениях присутствуют показательная и степенная функции, однако в (7) присутствует произведение этих функций, а в (8) – сумма, следовательно, временная сложность второго варианта алгоритма существенно меньше.

Выполнив вычисления по (7) и (8) при постоянных значениях r, ρ_{cp}, k_{cp}, $k_{2,max}$ и при различных значениях $|X|$ получили числовые оценки зависимости T_{A1} и T_{A2} от $|X|$ и r, которые показывают, что эти зависимости с высокой степенью достоверности аппроксимируются степенными функциями (коэффициент детерминации $R_2 \geq 0,9999$):

$T(|X|)_{A1}=10^6|X|^{4,4962}$; $T(|X|)_{A2}=8455,1|X|^{3,9223}$.

При этом значение показателя функции $T(|X|)_{A2}$ для второго варианта алгоритма меньше на величину 0,5739, чем для первого, и коэффициент при основании ($|X|$) меньше на 0,992•10^6, что указывает на существенную разницу оценок продолжительности решения задачи. Отношение T_{A1} к T_{A2} является возрастающей функцией от $|X|$, и при изменении $|X|$ от 100 до 500 изменяется от 132 до 335, а при изменении $|X|$ от 1000 до 5000 от 474 до 1060 единиц.

Следовательно, временная сложность разработанных вариантов алгоритма выражается $T(n)_{A1}=O(n^{C1})$, $T(n)_{A2}=O(n^{C2})$, где $n=|X|$, C_1 и C_2 – константы (при фиксированных значениях остальных параметров).

<u>*Планирование эксперимента.*</u> Проведению экспериментального исследования временной сложности алгоритмов предшествует планирование, которое включает определение факторов, влияющих на продолжительность решения задачи с использованием разработанных вариантов алгоритма и вариативность их значений с учетом свойств алгоритмов [10 – 11]. Состав влияющих факторов зависит от назначения алгоритма и состава исходных данных. Факторами, влияющими на продолжительность решения рассматриваемой задачи, являются:

- общее количество вершин в графе $G(X, U)$;
- общее количество ребер (дуг) в графе $G(X, U)$ и значения их весов;
- количество вершин в подмножествах N и R;
- значения степеней вершин.

Рассматриваемые в данной работе алгоритмы должны применяться к графовой модели дорожной сети региона, которая в обозримом интервале времени может изменяться относительно немного, поэтому варьирование значениями матрицы смежности графа $G(X, U)$ можно исключить, и ограничиться изменением количества вершин в подмножествах N и R.

Эксперимент выполнен с программной реализацией второго варианта алгоритма, как более производительного. При фиксированном значении $|X|$ продолжительность работы алгоритма статистически будет линейно зависеть от величины N, т.к. увеличение этого значения соответственно увеличивает количество процессов решения задачи. Величина r в (8) входит, как аддитивная составляющая, умножаемая на длину кратчайшего пути, значение которой зависит от связности вершин графа и является величиной случайной.

Цель эксперимента – получить точечные и интервальные оценки зависимости продолжительности решения задачи (T_{ij}, T_{min}, T_{ave}, T_{max}) поиска кратчайших путей (ПКП) между двумя подмножествами вершин смешанного графа от количества вершин в подмножествах N и R. Изменение значений $|N|$ и $|R|$ будет выполняться в пределах, соответствующих Пензенской области.

Схема экспериментального исследования:

1. Заполнить матрицу смежности; задать количество наборов данных из подмножеств N и R: i_{max}, j_{max}.
2. Задать первый набор данных из подмножества N; $i=1$; Задать значение $|N|=n_i$.
3. Задать первый набор данных из подмножества R; $j=1$; $|R|=r$.
4. Зафиксировать момент времени начала решения задачи с помощью исследуемого алгоритма ($T_{н,ij}$).
5. Выполнить решение задачи с помощью исследуемого алгоритма.
6. Зафиксировать момент времени окончания решения задачи ($T_{к,ij}$).
7. Вычислить продолжительность процесса решения задачи (T_{ij}) и записать в таблицу результатов: $T_{ij}=T_{к,ij}-T_{н,ij}$.
8. $j=j+1$; если $j<j_{max}+1$, то задать следующий набор данных из подмножества R и перейти к п. 4, иначе – к следующему.
9. $i=i+1$; если $i<i_{max}+1$, то задать следующий набор данных из подмножества X и перейти к п. 3, иначе – к следующему.
10. Выполнить обработку полученных результатов (дисперсионный и регрессионный анализ [17 – 19]).

Результаты исследования. При проведении эксперимента состав и мощность подмножества N изменялись случайным образом, количество вершин в подмножестве R изменялось систематически от 1 до 29 с шагом 1, состав вершин в подмножестве R – произвольно. В таблице 2 приведен фрагмент первичных данных, (значения T_{ij}), а также – оценка выборочного среднего (T_{ave}) для различных пар подмножеств N_{ij} и R_{ij}, наименьшие и наибольшие значения для каждого значения $|N|$.

Таблица 2 - Продолжительность решения задачи T_{ij} (сек.) в зависимости от $|N|$ и $|R|$

Количество вершин N	Количество вершин R				T_{min}	T_{ave}	T_{max}	T^{*}_{ave}	T_{ave} - T^{*}_{ave}	%% отклон.
	1	2	…	29						
60	30,890	30,929	…	31,987	26,219	29,984	33,369	26,4134	3,570	11,91
94	50,638	46,420	…	45,210	41,254	45,108	50,638	42,8790	2,229	4,94
186	86,612	85,236	…	91,820	81,971	89,818	97,268	87,4331	2,385	2,66
224	102,308	109,992	…	110,590	102,308	108,772	111,617	105,836	2,936	2,7
280	123,367	127,343	…	128,269	120,617	126,108	138,587	132,956	-6,848	-5,43
348	157,176	158,957	…	165,198	153,133	161,814	174,347	165,887	-4,073	-2,52
389	168,605	173,741	…	185,526	168,605	179,929	188,687	185,743	-5,814	-3,23
454	204,261	202,248	…	210,624	197,459	210,233	229,013	217,221	-6,988	-3,32
687	400,928	388,048	…	328,121	318,152	344,467	400,928	330,059	14,408	4,18
1469	741,493	675,080	…	716,154	640,428	706,961	750,005	708,769	-1,808	-0,26

Продолжительность решения в проведённым эксперименте (на базе ЭВМ *HP Proliant DL380 64 GB RAM/ 2xCPU 8 Cores Intel Xeon, 3.06 GHz*) составила примерно от 0,5 (при наименьшем значении $|N|$) до 12,5 минут (при наибольшем значении $|N|$). Выполнен регрессионный анализ влияния фактора $|N|$ на продолжительность решения задачи, результаты которого показывают, что зависимость $T(|N|)$ носит линейный характер: $T_{ave}{}^{*}=a_1|N|+a_0$, где a_1=0,4843; a_0=–2,6436; следовательно, $T(m)_{A2}=O(m)$. Величины абсолютной разности между выборочным средним (T_{ave}) и значением, вычисленным по уравнению регрессии, а также отклонение указанных величин в процентах приведены в таблице 1. При фиксированном N_i и изменении R_i значения T_{ij} меняются случайным образом, регрессионная зависимость между $|R|$ и T_{ij} отсутствует. Отношение T_{ave}/N показывает, что независимо от количества вершин N средняя продолжительность операций на одну вершину мало изменяется, что соответствует результатам теоретического анализа временной сложности варианта 2 алгоритма: $Min(T_{ave}/|N|)$ = 0,4504; $Ave(T_{ave}/|N|)$=0,47718; $Max(T_{ave}/|N|)$=0,5014(сек./1 верш.).

Заключение. Выполненный по структурному представлению анализ двух вариантов алгоритма поиска совокупности кратчайших путей между двумя заданными подмножествами вершин N и R неориентированного или смешанного графа показал, что в обоих вариантах алгоритма наибольшую долю в продолжительности выполнения занимает вычисление промежуточных значений кратчайших путей и выбор ближайших вершин. Получены выражения для оценки временной сложности, включающие показательную и степенную функции от мощности множества X и указывающие на существенное преимущество второго варианта алгоритма.

Результаты экспериментального исследования программной реализации варианта алгоритма с меньшей априорной оценкой временной сложности показали, что временная сложность рассмотренного варианта алгоритма поиска $|N|$ кратчайших путей до всех вершин $x_j \in R$ при фиксированном составе других характеристик графа является линейной функцией от $|N|$.

СПИСОК ЛИТЕРАТУРЫ:

1. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978. 432 с.
2. Майника Э. Алгоритмы оптимизации на сетях и графах. М.: Мир, 1981. 325 с.
3. Макконнелл Дж. Анализ алгоритмов. Активный обучающий подход. М.: Техносфера, 2009. 416 с.
4. Поляков И. В., Чеповский А. М., Чеповский А. А. Алгоритмы поиска путей на графах большого размера // Фундамент. и прикл. матем. 2014, том 19, выпуск 1. С. 165–172.
5. Коварцев А. Н., Даниленко А. Н. Алгоритмы и анализ сложности: учебник. Самара: изд-во Самарского университета, 2018. 128 с.
6. Кузюрин Н. Н., Фомин С. А. Эффективные алгоритмы и сложность вычислений [Электронный ресурс] // URL: https://www.studmed.ru/kuzyurin-nn-fomin-sa-effektivnye-algoritmy-i-slozhnost-vychisleniy_826c09f2f83. html (дата обращения: 15.02.2020).
7. Шефер О. В. Алгоритмы и анализ сложностиЛекции-Практика.pdf [Электронный ресурс]// Корпоративный портал Томского политехнического университета. Учебная работа: [сайт]. URL: https:// portal.tpu.ru/ SHARED/s/SHEFER/Study/Tab8 (дата обращения: 15.02.2020).
8. Абрамов С. А. Лекции о сложности алгоритмов. М.: МЦНМО, 2009. 256 с.
9. Ульянов М. В. Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ: Учебное пособие. М.: Наука. Физматлит, 2007. 376 с.
10. Алгоритмы: построение и анализ / Кормен Т. Х. [и др.]. М.: Издательский дом «Вильямс», 2011. 1296 с.
11. Построение и анализ алгоритмов. Семинар 6 Анализ трудоемкости алгоритмов / НИУ «Высшая школа экономики». - [Электронный ресурс] – URL: https://www.hse.ru/data/2012/07/20/1257633567/Семинар 6 Анализ трудоемкости алгоритмов в2 весна 2012.pdf.
12. Cherkassky B.V., Goldberg A.V., Radzik T. Shortest paths algorithms: Theory and experimental evaluation // Math. Prog. - Springer Science+Business Media. 1996. Vol. 73, Iss. 2. pp. 129 – 174.
13. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. М.: Издательский дом «Вильямс», 2001. 384 с.
14. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 1989. 360 с.
15. Иванов Б. Н. Дискретная математика. Алгоритмы и программы. Расширенный курс. М.: Известия, 2011. 512 с.
16. Кувайскова Ю. Е. Алгоритмы дискретной математики: учебное пособие. Ульяновск: УлГТУ, 2017. 99 с.
17. Дрейпер Н. Р., Смит Г. Прикладной регрессионный анализ. М.: Диалектика, 2007. 911 с.
18. Фёрстер Э., Рёнц Б. Методы корреляционного и регрессионного анализа: Руководство для экономистов. М.: Финансы и статистика, 1983 г. 304 с.
19. Коновалов Ю.В. Статистическое моделирование с использованием регрессионного анализа: методические указания. М.: Изд-во МГТУ им. Н.Э. Баумана, 2013. 73 с.

Статья поступила в редакцию 06.11.2020
Статья принята к публикации 11.12.2020