Об одном способе получения нижних оценок сложности информационных графов

А. П. Пивоваров

В работе приведен один способ получения нижних оценок сложности информационных графов различных видов (как в худшем случае, так и в среднем). Для этого вводится понятие информационного графа общего вида, под которое подпадают большинство модификаций информационных графов. Для информационных графов общего вида вводится понятие допустимости для выходной функции. Полученные нижние оценки сложности информационного графа общего вида зависят от количества различных значений, принимаемых выходной функцией.

Ключевые слова: информационный граф общего вида, нижняя оценка сложности, моделирование поиска.

1. Введение

В работе приводится один общий способ получения нижних оценок сложности в худшем случае и среднем, являющийся обобщением результата, приведенного в [1, раздел 2.1.3]. При этом в рассмотрение попадают задачи, решаемые с помощью информационных графов различных видов включая обычные информационные графы, вычислительные информационные графы, упрощенные вычислительные информационные графы [1, 2, 3].

В работе вводится определение информационного графа общего вида (ИГОВ), под которое подпадают все вышеперечисленные виды информационных графов. При этом такие понятия, как сложность на запросе, сложность в худшем случае и в среднем полностью совпадают с соответствующими определениями для конкретных видов информационных графов. Для ИГОВ вводится определение допустимой функции. Доказываются некоторые нижние оценки сложности ИГОВ, зависящие от количества различных значений функции, допустимой для оцениваемого графа. Полученные таким образом нижние оценки могут быть использованы для любых видов информационных графов, определение которых соответствует понятию информационного графа общего вида.

Автор выражает глубокую благодарность профессору Э. Э. Гасанову за постановку задачи и помощь в работе.

2. Основные понятия и формулировка результатов

Определим понятие информационного графа общего вида. Для этого нам потребуются следующие множества:

- множество запросов X;
- *множество F одноместных предикатов*, заданных на множестве X;
- *множество G одноместных переключателей*, заданных на множестве X (*переключатели* — это функции, область значений которых является начальным отрезком натурального ряда);

Пару $\mathcal{F} = \langle F, G \rangle$ будем называть *базовым множеством.*

Определим информационный граф общего вида с точки зрения его структуры.

Пусть нам дан ориентированный граф с выделенной вершиной, которую мы назовем *корнем.*

Выделим в графе некоторые вершины и назовем их *точками переключения* (корень может быть точкой переключения).

Если β — вершина графа, то через ψ_β обозначим *полустепень исхода* вершины β.

Каждой точке переключения β сопоставим некий символ из G. Это соответствие назовем *нагрузкой точек переключения.*

Для каждой точки переключения β ребрам, из нее исходящим, поставим во взаимно однозначное соответствие числа из множества $\{\overline{1, \psi_\beta}\}$. Эти ребра назовем *переключательными*, а это соответствие — *нагрузкой переключательных ребер*.

Ребра, не являющиеся переключательными, назовем *предикатными*.

Каждому предикатному ребру сети сопоставим некоторый символ из множества F. Это соответствие назовем *нагрузкой предикатных ребер*.

Полученный нагруженный граф назовем *информационным графом общего вида* над базовым множеством $\mathcal{F} = \langle F, G \rangle$.

Определение функционирования ИГОВ.

Скажем, что предикатное ребро проводит запрос $x \in X$, если предикат, приписанный этому ребру, принимает значение 1 на запросе x; переключательное ребро, которому приписан номер n, проводит запрос $x \in X$, если переключатель, приписанный началу этого ребра, принимает значение n на запросе x; ориентированная цепочка ребер проводит запрос $x \in X$, если каждое ребро цепочки проводит запрос x; запрос $x \in X$ проходит в вершину β ИГОВ, если существует ориентированная цепочка, ведущая из корня в вершину β, которая проводит запрос x.

Результатом функционирования ИГОВ на запросе $x \in X$ считается множество вершин, в которые проходит запрос x.

Понятие ИГОВ определено. Как видно из определения, информационные графы (ИГ) [1, 3], вычислительные информационные графы (ВИГ) [2] и упрощенные вычислительные информационные графы (УВИГ) [2] являются частными случаями информационных графов общего вида (они получаются путем выделения листьев среди вершин графа и нагрузки листьев определенным образом).

Пусть β — некоторая вершина ИГОВ. Тогда обозначим $\varphi_\beta(x)$ предикат на X, такой что $\varphi_\beta(x) = 1$ для тех и только тех запросов x, которые проходят в вершину β. Предикат $\varphi_\beta(x)$ называется функцией фильтра вершины β.

Переключатель g, приписанный переключательной вершине β считаем *вычисленным* на запросе x, если запрос x проходит в вершину β. Аналогично предикат f, приписанный некоторому ребру, выходящему из предикатной вершины β, считаем *вычисленным* на запросе x, если запрос x проходит в вершину β.

Сложностью ИГОВ U на запросе x назовем число $T(U, x)$, равное сумме числа переключателей и предикатов, вычисленных в процессе обработки запроса x. Эту величину также называют временем работы U на запросе x. *Верхней сложностью* (или временем работы в худшем случае) называют величину $\hat{T}(U) = \max_{x \in X} T(U, x)$. В случае, когда на X введено вероятностное пространство, а переключатели и предикаты являются измеримыми функциями, с помощью действий, дословно совпадающих с описанными в [1], можно показать, что $T(U, x)$ является случайной величиной относительно x и можно рассматривать $T(U) = \mathbf{M}_x\, T(U, x)$ — *сложность ИГОВ U в среднем* (или время работы U в среднем).

Пусть X — некоторое множество запросов, Z — множество, которое мы будем называть множеством ответов и задана функция $A\colon X \to Z$. Пусть $X' \subset X$, обозначим $A(X') = \{A(x) : x \in X'\}$. Нас будет интересовать случай, когда область значений функции A конечна, то есть $|A(X)| < \infty$. Здесь и далее $|X|$ — мощность множества X.

Пусть дан ИГОВ U. Обозначим множество его вершин как W. Для любого запроса $x \in X$ обозначим множество достижимых вершин U на этом запросе как $W(x) = \{\beta \in W\colon \varphi_\beta(x) = 1\}$. Будем говорить, что U является *допустимым для функции* $A\colon X \to Z$, если существует такая функция $B\colon 2^W \to Z$, что для любого $x \in X$ выполнено $B(W(x)) = A(x)$. Иными словами, значение функции $A(x)$ может быть полностью восстановлено по множеству вершин, в которые прошел запрос x.

Посмотрим, как это определение связано, например, с обычными информационными графами. Пусть есть задача информационного поиска (ЗИП) $I = \langle X, V, \rho\rangle$. Пусть некоторый ИГ U решает ЗИП I. Рассмотрим ИГОВ U', получающийся из U путем удаления нагрузки листьев. Тогда U' является допустимым для функции $A\colon X \to 2^V$, определяемой соотношением $A(x) = \{y \in V : x\, \rho\, y\}$.

Сформулируем теорему, позволяющую получать нижние оценки на сложность ИГОВ, для которых известны какие-либо допустимые для них функции.

Теорема 1. *Пусть X — множество запросов, Z — множество ответов и задана функция $A\colon X \to Z$. Пусть также задано некоторое базовое множество $\mathcal{F} = \langle F, G\rangle$ и U — некоторый ИГОВ над $\mathcal{F}$, допустимый для функции A, m — максимальная полустепень исхода среди переключательных вершин U, либо 2, если таких вершин нет.*

1) Тогда для любого непустого множества $X_0 \subseteq X$ выполнено

$$\max_{x\in X_0} T(U,x) \geqslant]\log_m |A(X_0)|[\,.$$

2) Пусть на X задано вероятностное пространство и функция A и функции из $\mathcal{F}$ измеримы относительно этого пространства, $A(X) = \{z_1,\dots,z_t\}$ и $p_i = P(A^{-1}(z_i))$, причем $p_1 \geqslant \dots \geqslant p_t$. В этом случае имеют место следующие оценки на сложность в среднем:

$$T(U) \geqslant \sum_{i=1}^{|A(X)|} u_i p_i,$$

$$T(U) \geqslant \max_{1\leqslant i\leqslant |A(X)|} i\cdot p_i \cdot \left(]\log_m i[- \frac{m}{m-1}\right),$$

где $u_1 = 1$, а при $i \geqslant 2$ выполнено $u_i =]\log_m i[$.

Следствие 1. *Если выполнены условия теоремы 1, имеет место следующая оценка на сложность в худшем случае:*

$$\hat{T}(U) \geqslant]\log_m |A(X)|[.$$

В случае, если выполнено условие второй части теоремы 1 и вероятности всех ответов равны между собой, имеет место следующая оценка на среднюю сложность:

$$T(U) \geqslant\ ]\log_m |A(X)|[- \frac{m}{m-1}.$$

Доказательство следствия 1. Первое утверждение следствия получается из утверждения первой части теоремы если в качестве X_0 выбрать все множество X.

Второе утверждение следствия получается из утверждения второй части теоремы, так как $p_1 = \dots = p_t = \frac{1}{t}$.

3. Доказательство основного результата

Доказательство теоремы 1. Для каждой предикатной вершины β рассматриваемого графа U занумеруем все выходящие из β ребра натуральными числами начиная с 1. Далее, для каждого запроса $x \in X$ построим код обхода U по этому запросу x. Этот код мы обозначим как $c(x)$ и он будет представлять из себя строку в алфавите $E_m = \{0, \ldots, m-1\}$. Для построения $c(x)$ рассмотрим подграф исходного графа $U(x)$, состоящий из всех вершин U и всех тех ребер, которые проводят запрос x. Фиксируем некоторый способ обхода для таких $U(x)$. Пусть, для определенности, это будет обход в глубину (с посещением каждой вершины не более одного раза) начиная из корня. При этом порядок обхода предикатных ребер, выходящих из одной вершины определяется номерами, которые этим ребрам приписаны. Код обхода $c(x)$ будем строить последовательно начиная с пустой строки. При посещении очередной вершины β будем добавлять к уже построенной части кода новый символ (или несколько символов) по следующим правилам. Если β — переключательная вершина, которой приписан переключатель $g \in G$ и $g(x) = n$, то к уже построенной части кода будем добавлять символ $n-1$. Если же β — предикатная вершина, из которой выходят ребра $e_1, \ldots, e_r$, которым приписаны номера $1, \ldots, r$ и предикаты $f_1, \ldots, f_r$ соответственно, то к уже построенной части кода приписывается строка $c_1, \ldots, c_r$, где $c_i = f_i(x)$. В частности, если из β не выходит ни одного ребра, то к уже построенной части кода ничего не приписывается. К концу обхода графа $U(x)$ мы получим строку $c(x)$, которую и будем называть кодом обхода U по запросу x. Эти коды обладают следующими свойствами:

1) для любого запроса $x \in X$ выполнено $T(U, x) = |c(x)|$, где $|c(x)|$ — длина $c(x)$;
2) если коды обхода U по запросам x_1 и x_2 различны, то ни один из них не является префиксом другого;
3) если коды обхода U по запросам x_1 и x_2 совпадают, то совпадают и множества $W(x_1)$ и $W(x_2)$ вершин, достижимых из корня на этих запросах.

Рассмотрим множество $A(X_0)$. Пусть $A(X_0) = \{z_1, \ldots, z_s\}$, где $s = |A(X_0)|$. Так как каждое $z_i \in A(X_0)$, то существует соответствующий ему запрос $x_i \in X_0$, такой что $A(x_i) = z_i$. По условию U допустим для A. Следовательно, существует функция $B\colon 2^W \to Z$, что $B(W(x)) = A(x)$. Предположим, что для некоторых различных запросов x_i и x_j выполнено $c(x_i) = c(x_j)$. Тогда по свойству 3 имеет место $W(x_i) = W(x_j)$, но тогда получается, что $z_i = A(x_i) = B(W(x_i)) = B(W(x_j)) = A(x_j) = z_j$, что неверно. Значит, $c(x_i) \neq c(x_j)$ при $i \neq j$. Пусть $l = \max_{1 \leqslant i \leqslant s} |c(x_i)| = \max_{1 \leqslant i \leqslant s} T(U, x_i)$ дополним каждый из кодов $c(x_i)$ до длины l (например, нулями). Так как все коды $c(x_i))$ были различными и не являлись префиксами друг друга, то все дополненные до длины l коды тоже различны и их ровно s штук. Но количество различных строк длины l в алфавите E_m не может превышать m^l. Значит, $s \leqslant m^l$. Отсюда имеем $l \geqslant \log_m s$. Итого, $\max_{x \in X_0} T(U, x) \geqslant \max_{1 \leqslant i \leqslant s} T(U, x_i) = l \geqslant \log_m s = \log_m |A(X_0)|$.

Докажем вторую часть теоремы. По условию теоремы $A(X) = \{z_1, \ldots, z_t\}$. Обозначим $X_i = A^{-1}(z_i)$ $(1 \leqslant i \leqslant t)$. Рассмотрим $C_i = \{c(x) \mid x \in X_i\}$ — множество всех кодов обхода U по запросам из X_i. Этих кодов конечное число и каждое из множеств C_i не является пустым. Пусть $C_i = \{c_{i1}, \ldots, c_{ik_i}\}$. Для каждого из этих кодов c_{ij} $(1 \leqslant i \leqslant t,\ 1 \leqslant j \leqslant k_i)$ рассмотрим множество соответствующих им запросов $X_{ij} = \{x \in X_i\colon c(x) = c_{ij}\}$. Используя определения кодов и измеримость базового множества $\mathcal{F}$ несложно показать, что X_{ij} являются событиями. Пусть $p_{ij} = P(X_{ij})$, $p_i = P(X_i)$. Так как $X_i = \sqcup_{j=1}^{k_i} X_{ij}$, то очевидно $p_i = \sum_{j=1}^{k_i} p_{ij}$. По определению X_{ij} для любого запроса $x \in X_{ij}$ код обхода U по этому запросу есть c_{ij}. По свойству 1 из этого следует, что $T(U, x) = |c_{ij}|$ для любого $x \in X_{ij}$. Следовательно, для сложности имеем: $T(U) = \sum_{i=1}^{t} \sum_{j=1}^{k_i} |c_{ij}| p_{ij}$. Обозначим как c_i тот из элементов C_i, который имеет наименьшую длину. Продолжая оценку для $T(U)$ имеем: $T(U) \geqslant \sum_{i=1}^{t} \sum_{j=1}^{k_i} |c_i| p_{ij} = \sum_{i=1}^{t} |c_i| p_i$. При этом точно известно, что все c_i различны. Переставим последовательность кодов $c_1, \ldots, c_t$ в порядке возрастания их длин. Полученную последовательность обозначим $c_{(1)}, \ldots, c_{(t)}$. Очевидно, что $\sum_{i=1}^{t} |c_i| p_i \geqslant \sum_{i=1}^{t} |c_{(i)}| p_i$. Покажем, что $|c_{(i)}| \geqslant \log_m i$. Действительно, если бы выполнялось $|c_{(i)}| < \log_m i$, то мы бы имели i различных строк в алфавите E_m, которые не могут быть префиксами друг друга и длины которых строго меньше $\log_m i$, что невозможно. Учитывая, что длина кода является целым числом, и код не может быть пустым, имеем $|c_{(i)}| \geqslant \max(1,]\log_m i[) = u_i$. Последнее равенство вытекает из определения u_i. Итого, получаем $T(U) \geqslant \sum_{i=1}^{t} u_i p_i$. Покажем, как вторая оценка на $T(U)$ вытекает из этой. $T(U) \geqslant \sum_{i=1}^{t} u_i p_i \geqslant \max_{1\leqslant i\leqslant t} \sum_{j=1}^{i} u_j p_j \geqslant \max_{1\leqslant i\leqslant t} p_i \sum_{j=1}^{i} u_j$. Оценим снизу величину $\sum_{j=1}^{i} u_j$. Для этого посмотрим, как ведет себя последовательность u_j: первые m элементов этой последовательности равны 1, затем m^2-m элементов равны 2, следующие m^3-m^2 элементов равны 3 и так далее. Подсчитаем оцениваемую сумму в частном случае, когда верхняя граница суммирования является степенью числа m. Пусть фиксировано целое число $d \geqslant 2$. Тогда $\sum_{j=1}^{m^d} u_j = 1 \cdot m + 2\cdot(m^2-m) + 3\cdot(m^3-m^2) + \cdots + d\cdot(m^d - m^{d-1}) = dm^d - m^{d-1} - \cdots - m$. При $i \leqslant m$ очевидно имеет место $\sum_{j=1}^{i} u_j = i$. Если же $i > m$, то возьмем $d =]\log_m i[\geqslant 2$. В этом случае выполнено $m^d \geqslant i$, но $m^{d-1} < i$, а значит $m^d < im$. Оценим сумму $\sum_{j=1}^{i} u_j = \sum_{j=1}^{m^d} u_j - \sum_{j=i+1}^{m^d} u_j = dm^d - m^{d-1} - \cdots - m - d\cdot(m^d - i) = di - \frac{m^d-m}{m-1}$. Легко убедиться, что неравенство $\sum_{j=1}^{i} u_j \geqslant di - \frac{m^d-m}{m-1}$ (где $d =]\log_m i[$) выполняется не только при $i > m$, но и при $i \leqslant m$. Продолжим цепочку неравенств: $\sum_{j=1}^{i} u_j \geqslant di - \frac{m^d-m}{m-1} > di - \frac{m^d}{m-1} > di - \frac{im}{m-1} = i\left(d - \frac{m}{m-1}\right) = i\left(]\log_m i[- \frac{m}{m-1}\right)$. Подставляя эту оценку в неравенство $T(U) \geqslant \max_{1\leqslant i\leqslant t} p_i \sum_{j=1}^{i} u_j$, получаем $T(U) \geqslant \max_{1\leqslant i\leqslant t} i \cdot p_i \cdot \left(]\log_m i[- \frac{m}{m-1}\right)$. Теорема доказана.

Список литературы

[1] Гасанов Э. Э., Кудрявцев В. Б. Теория хранения и поиска информации. М.: ФИЗМАТЛИТ, 2002.

[2] Пивоваров А. П. Моделирование вычислительных задач информационного поиска // Интеллектуальные системы. Т. 14, вып. 1–4. 2010. С. 237–260.

[3] Пивоваров А. П. Поиск представителя в задаче о метрической близости // Интеллектуальные системы. Т. 12, вып. 1–4. 2008. С. 333–350.