

Моделирование и сложность поиска в многопроцессорных системах *

Э.Э.Гасанов, Е.Р.Ерохина
Московский государственный университет

Аннотация

При программировании поиска для многопроцессорных систем существует как минимум два подхода. Первый (сепаративный) подход предполагает разделение данных между процессорами (исполнителями) и дальнейшую независимую обработку каждым исполнителем своей части данных. Второй (кооперативный) подход предполагает совместное владение данными и совместную обработку. В работе предлагается математическая модель параллельных алгоритмов поиска и в рамках этой модели исследуется параллельное решение задач поиска с отношением поиска, являющимся отношением линейного квази порядка. В работе описан способ оптимального разделения данных при использовании сепаративного подхода для данной задачи, и показано, что в общем случае сепаративный подход не дает оптимального решения, то есть приведен пример таких задач поиска с отношением линейного квази порядка, для которых кооперативный подход дает лучшие результаты.

1 Введение

Одной из основных задач информатики является проблема сокращения времени обработки данных, и создание быстрых алгоритмов, в частности, быстрых алгоритмов поиска — это шаги на этом пути. С появлением многопроцессорных вычислительных систем возникла принципиально новая возможность сокращения времени работы алгоритмов, возникло, так называемое параллельное программирование, то есть написание программ (алгоритмов), задействующих в своей работе несколько процессоров (исполнителей). В параллельном программировании существует как минимум два подхода. Первый предполагает разделение данных между исполнителями и дальнейшую независимую обработку каждым исполнителем своей части

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (гранты 95-01-00597, 98-01-00130)

данных. Второй подход предполагает кооперативное (совместное) владение данными всеми исполнителями и сложную систему регулировки отношений между исполнителями в процессе обработки совместных данных. Первый подход мы в дальнейшем будем называть сепаративным, а второй — кооперативным. Понятно, что кооперативный подход более сложен в реализации, но поскольку он более общий, то, вообще говоря, должен давать лучшие результаты.

Задачи поиска относятся к такому типу задач, для которых сепаративный подход вполне приемлем. То есть множество объектов поиска можно раз и навсегда поделить между исполнителями, и затем при осуществлении очередного поиска каждый из исполнителей будет вести поиск в своей части данных, после чего остается только объединить результаты поиска. Возникает вопрос обеспечивает ли сепаративный подход лучшее решение для какой-либо задачи поиска. Казалось, что хорошим кандидатом на такой тип задач поиска является задача с отношением поиска, являющимся отношением линейного квазиупорядка, описанная, например, в [1]. Эта задача заключается в поиске в линейно упорядоченном множестве элементов, не больших, чем запрос. Частным примером таких задач является одномерная задача о доминировании (см. [2]).

В работе предлагается математическая модель параллельных алгоритмов поиска и в рамках этой модели исследуется параллельное решение упомянутой задачи поиска с отношением линейного квазиупорядка. В работе описан способ оптимального разделения данных при использовании сепаративного подхода для данной задачи, и показано, что в общем случае сепаративный подход не дает оптимального решения, то есть приведен пример таких задач поиска с отношением линейного квазиупорядка, для которых кооперативный подход дает лучшие результаты.

2 Основные понятия и формулировка результатов

Мы будем использовать терминологию и обозначения из работы [3]. Напомним понятие информационной сети с переключателями (ИСП) и другие необходимые понятия из [3].

Пусть X — множество запросов, причем на X определено вероятностное пространство $\langle X, \sigma, \mathbf{P} \rangle$, где σ — алгебра подмножеств множества X , \mathbf{P} — вероятностная мера на σ ; Y — множество записей (объектов поиска); ρ — бинарное отношение на $X \times Y$, называемое отношением поиска; тройку $S = \langle X, Y, \rho \rangle$ (или пятерку $S = \langle X, Y, \rho, \sigma, \mathbf{P} \rangle$, если мы хотим подчеркнуть какое именно вероятностное пространство над X используется) будем называть *типом* или иногда более развернуто *типом задач информационного поиска*; тройку $I = \langle X, V, \rho \rangle$, где V — некоторое конечное подмножество мно-

жества Y , в дальнейшем называемое *библиотекой*, будем называть задачей информационного поиска (ЗИП) типа $S = \langle X, Y, \rho \rangle$ (или ЗИП, принадлежащей типу S , и обозначать $I \in S$), и будем считать, что задача $I = \langle X, Y, \rho \rangle$ состоит в перечислении для произвольно взятого запроса $x \in X$ всех тех и только тех записей из V , которые находятся в отношении ρ с запросом x , т.е. удовлетворяют запросу x ; $O(y, \rho) = \{x \in X : x\rho y\}$ — тень записи $y \in Y$; $N_f = \{x \in X : f(x) = 1\}$, где f — одноместный предикат, определенный на X , т.е. $f : X \rightarrow \{0, 1\}$; $\chi_{y, \rho} : X \rightarrow \{0, 1\}$ такая, что $N_{\chi_{y, \rho}} = O(y, \rho)$ — характеристическая функция записи y ; F — множество символов одноместных предикатов, определенных на множестве X , называемое базовым множеством предикатов; G — множество символов одноместных переключателей, определенных на множестве X . Под переключателями будем понимать функции, область значений которых являются конечные подмножества натурального ряда.

Пару $\mathcal{F} = \langle F, G \rangle$ назовем базовым множеством.

Если n — натуральное число, а $g(x)$ — некий переключатель, то через $\xi_g^n(x)$ обозначим предикат, определенный на X , такой, что

$$N_{\xi_g^n} = \{x \in X : g(x) = n\}.$$

Обозначим

$$\widehat{G} = \{\xi_g^n : g \in G, n \in \mathbf{N}\},$$

где \mathbf{N} — множество натуральных чисел.

Мы не будем здесь приводить понятие информационной сети с переключателями (ИСП) над базовым множеством $\mathcal{F} = \langle F, G \rangle$, поскольку оно громоздко и полностью приводится в [3]. Там же можно найти понятия проводимости ребра, проводимости ориентированной цепи, функции фильтра вершины β , обозначаемой через $\varphi_\beta(x)$, функции ответа ИСП, обозначаемой через $\mathcal{J}(x)$.

Перейдем к описанию математической модели алгоритмов параллельного поиска.

Пусть нам дана некоторая ИСП U над базовым множеством \mathcal{F} , и пусть имеется некоторое число исполнителей. Исполнителя можно воспринимать как некоторый автомат, который будет обходить нашу ИСП. Будем считать, что все исполнители как автоматы одинаковы, то есть имеют одинаковое поведение, и это поведение мы опишем, описав процедуру обхода исполнителем ИСП U . Причем, будем считать, что все исполнители осуществляют обход сети одновременно, то есть в параллельном режиме. Пусть число исполнителей равно m . Прежде чем описать процедуру обхода сделаем следующие предположения.

- Будем считать, что каждому ребру сети U мы можем временно приписывать номер, который назовем *путевым*.

- Будем считать, что мы можем помечать листья сети, в которые мы уже заходили, причем в начальный момент все листья непомеченные.
- Будем считать, что каждому из исполнителей приписано некое целое число j ($1 \leq j \leq m$), которое будем называть порядковым номером исполнителя. Причем, различным исполнителям соответствуют различные номера.
- Будем считать, что помимо порядкового номера каждый исполнитель j ($j = \overline{1, m}$) имеет свой цвет C_j , с помощью которого он может окрашивать ребра и переключательные вершины сети, которые он просмотрел. Причем будем полагать, что различным исполнителям соответствуют различные цвета.
- Будем считать также, что до начала работы процедуры ребра и переключательные вершины ИСП были окрашены в некий цвет C_0 , который назовем нейтральным, причем этот цвет не совпадает ни с одним из цветов C_1, \dots, C_m .
- Положим также, что имеется некоторый цвет C_{m+1} , который не совпадает ни с одним из цветов C_0, C_1, \dots, C_m .
- Назовем множество ребер, исходящих из одной вершины, *пучком* и будем считать, что в каждом пучке определен порядок следования ребер, т.е. существует ребро первое, второе и т.д.
- Будем считать, что каждый исполнитель совершает свои действия (просмотр окрестности, перекрашивание ребер, движение к следующей вершине) за 1 такт. Но поскольку у нас сразу несколько исполнителей обходят сеть, то чтобы разрешать спорные ситуации, связанные с порядком действий, при одновременном попадании нескольких исполнителей в одну вершину, будем считать, что исполнитель с порядковым номером j начинает обход сети в момент времени $\frac{j-1}{m+1}$ такта. Это простейший способ разрешения конфликтов, при котором порядок выполнения действий автоматами определяется порядком попадания в вершину, при этом все же будем считать, что все исполнители, попавшие в одну вершину, успевают хоть и по очереди за один такт сделать свои элементарные действия.

Теперь мы можем перейти к описанию процедуры обхода сети U . Входными данными процедуры обхода A будут информационная сеть U и запрос $x \in X$. Выходными данными — множество \mathcal{Y} записей, входящих в ответ на запрос x , и набор натуральных чисел T_1, \dots, T_m , показывающих нам сколько предикатов или переключателей было вычислено каждым из исполнителей в процессе обхода сети U на запросе x .

Сначала опишем процедуру обхода сети U на запросе $x \in X$ j -ым исполнителем ($j = \overline{1, m}$).

1. (Инициализация процесса обхода.)
 - (a) Объявим текущей вершиной корень сети U .
 - (b) Присвоим начальное значение счетчику путей $s_j = 1$.
 - (c) Присвоим начальное значение счетчику вычисленных функций $t_j = 0$.
 - (d) Перейдем к пункту 2.
2. (Прямой ход.)
 - (a) Если текущая вершина — непомеченный лист сети U , то помечаем ее и добавляем к множеству \mathcal{U} запись, принадлежащую текущей вершине.
 - (b) Если текущая вершина есть точка переключения, то:
 - i. если текущая вершина окрашена не в нейтральный цвет, то идем к пункту 3;
 - ii. окрашиваем текущую вершину в цвет C_j ;
 - iii. вычисляем значение переключателя, приписанного текущей вершине, на запросе x (пусть это значение равно n);
 - iv. увеличиваем на 1 значение счетчика вычисленных функций: $t_j = t_j + 1$;
 - v. если среди ребер, исходящих из текущей вершины, нет ребра с номером n , то переходим к пункту 3;
 - vi. ребру с номером n , исходящему из текущей вершины, приписываем путь s_j , окрашиваем его в цвет C_j , объявляем конец этого ребра текущей вершиной, присваиваем $s_j = s_j + 1$ и возвращаемся к пункту 2.
 - (c) Если текущая вершина не является точкой переключения, то:
 - i. если из текущей вершины не исходит ни одно ребро, или же среди ребер, исходящих из текущей вершины, есть ребро, окрашенное в цвет C_j , или же среди ребер, исходящих из текущей вершины, нет ни одного, имеющего нейтральный цвет, тогда переходим к пункту 3;
 - ii. выбираем первое по порядку следования нейтральное ребро;
 - iii. окрашиваем выбранное ребро в цвет C_j ;
 - iv. величину t_j увеличиваем на 1;
 - v. приписываем выбранному ребру путь s_j .
 - vi. присваиваем $s_j = s_j + 1$;
 - vii. объявляем конец просматриваемого ребра текущей вершиной;

- viii. если значение предиката, соответствующего выбранному ребру, на запросе x равно 1, то идем к пункту 2, иначе — к пункту 3.

3. (Обратный ход)

- (a) Если текущая вершина есть переключательная вершина, окрашенная в цвет C_j , то окрашиваем ее в цвет C_{m+1} .
- (b) Уменьшаем значение счетчика путевых номеров: $s_j = s_j - 1$.
- (c) Если $s_j = 0$, то присваиваем числу T_j значение счетчика вычисленных функций t_j и завершаем работу процедуры.
- (d) Если $s_j > 0$, то существует ребро, входящее в текущую вершину и имеющее путевой номер s_j и цвет C_j , выбираем его.
- (e) Окрашиваем выбранное ребро в цвет C_{m+1} .
- (f) Объявляем текущей вершиной начало выбранного ребра;
- (g) Если текущая вершина является точкой переключения, то возвращаемся к пункту 3, иначе возвращаемся к пункту 2.

Теперь описать процедуру обхода \mathcal{A} сети U на запросе x всеми исполнителями очень просто.

1. Присвоим начальное значение множеству $\mathcal{Y} = \emptyset$.
2. Запускаем по очереди в моменты времени $\frac{j-1}{m+1}$ процедуру обхода сети U на запросе x j -ым исполнителем ($j = \overline{1, m}$).
3. Считаем, что работа процедуры обхода \mathcal{A} полностью завершена тогда и только тогда, когда каждый из исполнителей завершил работу своей процедуры.

Легко убедиться, что данная процедура обхода позволяет посетить все вершины, функции фильтров которых принимают значение 1 на запросе x , и следовательно полученное на выходе процедуры множество \mathcal{Y} совпадает с ответом $\mathcal{J}(x)$ сети U на запрос x , что говорит о корректности определения процедуры обхода. На выходе процедуры обхода мы также получим множество $\{T_1, \dots, T_m\}$, где для любого $j \in \overline{1, m}$ величину T_j можно трактовать как время, затраченное исполнителем с порядковым номером j .

В принципе можно также учитывать время обратного хода, изменяя соответствующим образом значение t_j счетчика времени на шаге 3 процедуры. Но в данной работе мы рассматриваем такое приближение к реальному времени вычисления, при котором мы пренебрегаем всеми затратами времени, кроме времени вычисления предикатов и переключателей, причем считаем, что каждый из них вычисляется за время, равное одному такту.

В завершение описания процедуры обхода ИСП, заметим, что данная процедура позволяет использовать конкретную ИСП U более, чем один раз. Для повторного обхода сети U достаточно объявить цвета C_0 и C_{m+1} нейтральными и положить, что при обратном ходе каждый исполнитель должен использовать некоторый цвет C_{m+2} вместо используемого им ранее цвета C_{m+1} . Причем, цвет C_{m+2} полагаем отличным от любого из цветов C_0, C_1, \dots, C_{m+1} . Кроме того надо изменить число, которым помечаются листья, посещенные процедурой обхода.

На основе описанной выше процедуры обхода, которая по сути является описанием соответствующего сети алгоритма параллельного поиска, мы можем ввести понятие сложности ИСП.

Сначала определим понятие сложности ИСП на запросе.

Пусть нам дана некая ИСП U и произвольно взятый запрос x . Пусть \mathcal{A} — определенная ранее процедура обхода, причем, число исполнителей, реализующих процедуру обхода \mathcal{A} , равно $m \geq 1$, и пусть $\{T_1 = T_1(U, m, x), \dots, T_m = T_m(U, m, x)\}$ — множество тех самых чисел, полученных на выходе процедуры, которые указывают на число вычисленных каждым из исполнителей функций, при работе процедуры \mathcal{A} на запросе x . Тогда *сложностью* ИСП U на запросе x для m исполнителей (*m -сложностью на запросе*) назовем величину

$$T(U, m, x) = \max_{1 \leq j \leq m} T_j(U, m, x).$$

Понятие сложности ИСП введем как среднее значение сложности ИСП на запросе, взятое по множеству всех запросов.

Для этого введем вероятностное пространство над множеством запросов X , под которым будем понимать тройку $\langle X, \sigma, \mathbf{P} \rangle$, где σ — некоторая алгебра подмножеств множества X , \mathbf{P} — вероятностная мера на σ , т. е. аддитивная мера такая, что $\mathbf{P}(X) = 1$.

Справедлива следующая лемма.

Лемма 1 *Если алгебра σ содержит все множества N_f , где $f \in F \cup \hat{G}$, то для любого натурального m и для любой ИСП U над базовым множеством $\mathcal{F} = \langle F, G \rangle$ функция $T(U, m, x)$, как функция от x , является случайной величиной.*

Далее всюду будем предполагать, что мы находимся в условиях леммы 1.

Сложностью ИСП U для m исполнителей (*m -сложностью сети*) назовем величину

$$T(U, m) = \mathbf{M}_x T(U, m, x).$$

Скажем, что ИСП U разрешает ЗИП $I = \langle X, V, \rho \rangle$, если для любого запроса $x \in X$ ответ на этот запрос содержит все те и только те записи из

V , которые удовлетворяют запросу x , то есть

$$\mathcal{J}(x) = \{y \in V : xry\}.$$

Пусть нам дана некая ЗИП I и число исполнителей $m \geq 1$. Сложностью задачи I при базовом множестве \mathcal{F} для m исполнителей назовем число

$$T(I, \mathcal{F}, m) = \inf\{T(U, m) : U \in \mathcal{U}(I, \mathcal{F})\},$$

где $\mathcal{U}(I, \mathcal{F})$ — множество всех ИСП над базовым множеством \mathcal{F} , разрешающих ЗИП I .

Если существует такая ИСП $U \in \mathcal{U}(I, \mathcal{F})$, что $T(U, m) = T(I, \mathcal{F}, m)$, то ИСП U будем называть *оптимальной* для ЗИП I и m исполнителей.

Пусть U — некоторая ИСП, $x \in X$ — некоторый запрос, m — некоторое натуральное число, β — некоторое предикатное ребро (некоторая переключаемая вершина) сети U . Обозначим через $n_\beta(x, m)$ число, равное 0, если в результате применения процедуры обхода с m исполнителями к сети U на запросе x ребро (вершина) β остается окрашенным (окрашенной) в нейтральный цвет, и равно j , если в процессе работы процедуры обхода с m исполнителями к сети U на запросе x перед тем как быть окрашенным (окрашенной) в цвет C_{m+1} ребро (вершина) β было окрашено (была окрашена) в цвет C_j , то есть функция, соответствующая β вычислялась j -ым исполнителем.

Скажем, что предикатное ребро (переключаемая вершина) β *m -сепаративно* (*m -сепаративна*), если для любых $x_1, x_2 \in X$ справедливо

$$(n_\beta(x_1, m) = 0) \vee (n_\beta(x_2, m) = 0) \vee (n_\beta(x_1, m) = n_\beta(x_2, m)).$$

Скажем, что ИСП U *m -сепаративна*, если любое предикатное ребро и любая переключаемая вершина сети U *m -сепаративны*.

Легко видеть, что *m -сепаративные* ИСП соответствуют параллельным алгоритмам поиска, относящимся к сепаративному подходу, при котором все данные делятся на части и при просмотре всего множества запросов каждая часть данных обрабатывается только одним исполнителем.

Обозначим через $\mathcal{U}_s(I, \mathcal{F}, m)$ — множество всех *m -сепаративных* ИСП над базовым множеством \mathcal{F} , разрешающих ЗИП I .

Если существует такая ИСП $U \in \mathcal{U}_s(I, \mathcal{F}, m)$, что $T(U, m) = \inf\{T(U, m) : U \in \mathcal{U}_s(I, \mathcal{F}, m)\}$, то ИСП U будем называть *оптимальной* для ЗИП I в классе *m -сепаративных*.

Задача поиска, в которой отношение поиска является отношением линейного квазипорядка, наверное одна из самых распространенных и, вместе с тем, самых простых задач поиска. Эта задача имеет полное и окончательное решение, опубликованное в [1].

Если X — некоторое множество, то под *отношением линейного квазипорядка* \succeq на $X \times X$ будем понимать бинарное отношение, для любых $x, y, z \in X$ удовлетворяющее условиям

- рефлексивности $x \succeq x$;
- транзитивности $(x \succeq y) \& (y \succeq z) \rightarrow (x \succeq z)$;
- связности $(x \succeq y) \vee (y \succeq x)$.

Итак, мы будем рассматривать типы задач поиска, в которых в качестве отношения поиска выступает отношение линейного квазиупорядка, а именно рассмотрим следующий тип

$$S_{lin} = \langle X, X, \succeq \rangle,$$

где X — некоторое множество, \succeq — некоторое отношение линейного квазиупорядка на $X \times X$.

Пусть $a \in X$, K_a — функция, действующая из X в $\{0, 1\}$ такая, что $N_{K_a} = \{x \in X : x \succeq a\}$. Отметим, что $K_a(x)$ есть характеристическая функция записи a . Пусть $\mathcal{K} = \{K_a(x) : a \in X\}$.

Будем рассматривать следующее базовое множество $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$. Так как оно содержит характеристические функции всех записей, то согласно [4, теорема 2] \mathcal{F}_0 полно для типа S_{lin} , то есть для любой ЗИП типа S_{lin} множество $\mathcal{U}(I, \mathcal{F}_0)$ не пусто.

Справедливы следующие теоремы.

Теорема 1 Для любого натурального числа m и любой ЗИП I типа S_{lin} существует оптимальная ИСП над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$.

Весом записи y назовем число $\|y\| = \mathbf{P}(O(y, \succeq))$.

Пусть $V = \{y_1, \dots, y_k\}$ — такое упорядоченное множество, что $y_k \succeq y_{k-1} \succeq \dots \succeq y_1$. Тогда через $R(m, V)$ обозначим функцию, принимающую вещественные значения, определяемую рекуррентно следующими соотношениями:

$$R(1, V) = \sum_{i=1}^{k-1} \|y_i\|, \quad (1)$$

$$R(m, V) = \min_{1 \leq l \leq \lfloor k/m \rfloor} \left(R(m-1, V^{(l-1)m+2}) + \sum_{i=1}^{l-1} \|y_{1+(i-1)m}\| \right), \quad (2)$$

где $V^j = \{y_j, y_{j+1}, \dots, y_k\}$ ($j = \overline{1, k}$), $[c]$ — ближайшее целое, не превышающее вещественного c .

Теорема 2 Для любого натурального числа m и любой ЗИП $I = \langle X, V, \succeq \rangle$ типа S_{lin} , такой, что $V = \{y_1, \dots, y_k\}$ — такое упорядоченное множество, что $y_k \succeq y_{k-1} \succeq \dots \succeq y_1$, существует оптимальная в классе m -сепаративных сеть над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$ и

$$T_s(I, m) \stackrel{def}{=} \inf \{T(U, m) : U \in \mathcal{U}_s(I, \mathcal{F}_0, m)\} = 1 + R(m, V),$$

где $R(m, V)$ — функция определяемая соотношениями (1), (2).

Теорема 3 *Существуют такое натуральное число m и такая ЗИП I типа S_{lin} , что любая оптимальная для ЗИП I сеть над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$ не принадлежит классу m -сепаративных.*

3 Доказательство леммы 1

Если U — ИСП над базовым множеством \mathcal{F} , $c = (\alpha, \beta)$ — некоторое ребро сети U , f_c — функция проводимости ребра c , φ_α — функция фильтра вершины α , то предикат

$$\zeta_c(x) = f_c(x) \& \varphi_\alpha(x)$$

назовем *функцией состояния ребра c* .

Если $\mathcal{C}(U) = \{c_1, \dots, c_q\}$ — произвольным образом упорядоченное множество ребер сети U , то вектор-функцию $(\zeta_{c_1}(x), \dots, \zeta_{c_q}(x))$ назовем *вектором состояния сети U* .

При фиксации запроса $x \in X$ полученный вектор $(\zeta_{c_1}(x), \dots, \zeta_{c_q}(x))$ из q -мерного единичного куба B^q будем называть *вектором состояния сети U на запросе x* .

Пусть число исполнителей $m \geq 1$ у нас фиксировано.

Нам необходимо доказать, что для любой ИСП U над базовым множеством \mathcal{F} и любого действительного числа r множество

$$\{x \in X : T(U, m, x) < r\} \in \sigma.$$

Пусть $\mathcal{R}(U), \mathcal{C}(U)$ — множества вершин и ребер сети U , соответственно.

Покажем сначала, что для любой вершины $\beta \in \mathcal{R}(U)$ справедливо $N_{\varphi_\beta} \in \sigma$.

Пусть для каждой $\beta \in \mathcal{R}(U)$ \mathcal{C}_β — множество всех ориентированных цепей сети U , ведущих из корня в вершину β ; C — некоторая цепь, c — некоторое ребро, f_c — функция проводимости ребра c . Так как по определению для любого $c \in \mathcal{C}$ $f_c(x) \in F \cup \widehat{G}$, то $N_{f_c} \in \sigma$ и

$$\varphi_\beta = \bigvee_{C \in \mathcal{C}_\beta} \bigwedge_{c \in C} f_c,$$

$$N_{\varphi_\beta} = \bigcup_{C \in \mathcal{C}_\beta} \bigcap_{c \in C} N_{f_c} \in \sigma,$$

и следовательно $N_{\zeta_c} \in \sigma$.

Пусть после некоторого упорядочения $\mathcal{C}(U) = \{c_1, \dots, c_q\}$.

Пусть $\tilde{\omega} = \{\omega_1, \dots, \omega_q\}$ — произвольный булев вектор длины q , то есть $\omega_i \in \{0, 1\}$, $i = \overline{1, q}$.

Пусть

$$X_{\tilde{\omega}} = \{x \in X : \zeta_{c_i}(x) = \omega_i, i = \overline{1, q}\},$$

то есть это множество таких запросов, вектор состояния сети на которых равен $\tilde{\omega}$.

Очевидно, что для любых $x_1, x_2 \in X_{\tilde{\omega}}$ справедливо

$$T(U, m, x_1) = T(U, m, x_2) \stackrel{def}{=} T_{\tilde{\omega}}.$$

Нетрудно видеть, что если $\tilde{\omega}_1 \neq \tilde{\omega}_2$, то $X_{\tilde{\omega}_1} \cap X_{\tilde{\omega}_2} = \emptyset$, причем $\bigcup_{\tilde{\omega} \in B^q} X_{\tilde{\omega}} = X$.

Легко заметить, что

$$X_{\tilde{\omega}} = \bigcap_{i=1}^q \{x \in X : \zeta_{c_i}(x) = \omega_i\},$$

причем для любого $i \in \{\overline{1, q}\}$

$$\{x \in X : \zeta_{c_i}(x) = \omega_i\} = \begin{cases} N_{\zeta_{c_i}}, & \text{если } \omega_i = 1 \\ X \setminus N_{\zeta_{c_i}}, & \text{если } \omega_i = 0 \end{cases} \in \sigma,$$

значит, $X_{\tilde{\omega}} \in \sigma$.

Возьмем произвольное действительное число r .

Обозначим

$$\Omega_r = \{\tilde{\omega} : T_{\tilde{\omega}} < r\}.$$

Тогда

$$\{x \in X : T(U, m, x) < r\} = \bigcup_{\tilde{\omega} \in \Omega_r} X_{\tilde{\omega}} \in \sigma.$$

Таким образом, лемма доказана.

4 Существование оптимальной сети

Если $a, b \in X$ и $a \succeq b$, то будем говорить, что a не меньше, чем b , и b не больше, чем a .

Если $a, b \in X$ и $a \succeq b$ и $b \succeq a$, то будем говорить, что a и b равны и писать $a = b$.

Отношение равенства "=" делит все множество X на классы эквивалентности. Выберем произвольно из каждого класса по одному элементу и назовем *представителем класса*. Понятно, что если $a = b$, то $K_a(x) \equiv K_b(x) \equiv K_c(x)$, где $c \in X$ — представитель класса, которому принадлежат a и b .

Если $a, b \in X$ и $a \succeq b$ и $a \neq b$, то будем писать $a \succ b$ и говорить, что b предшествует a , или b меньше a , или a больше b .

Если $a, b \in X$ и $b \succeq a$, то обозначим

$$[a, b) = \{x \in X : b \succ x \& x \succeq a\}$$

и

$$[a, b] = \{x \in X : b \succeq x \& x \succeq a\}.$$

Если $M \subseteq X$, то минимальным элементом множества M назовем такой элемент $x \in M$, что для любого $x' \in M$ $x' \succeq x$.

Через x_{min} обозначим представителя класса минимальных элементов множества X , а через x_{max} обозначим представителя класса эквивалентности таких a , что для любого $x \in X$ $a \succeq x$.

Далее всюду будем считать, что элементы x_{min} и x_{max} существуют, хотя если в X таких элементов нет, например, в случае открытого множества, то в качестве x_{min} и x_{max} можно взять граничные точки, поскольку далее x_{min} и x_{max} используются только как обозначения нечто такого, не меньше чего (соответственно не больше чего) любой элемент из X .

Так как для любых $f, g \in \mathcal{K}$ справедливо $f \vee g \in \mathcal{K}$ и $f \& g \in \mathcal{K}$, то для любого ребра c произвольной сети U над базовым множеством \mathcal{F}_0 функция состояния ребра c $\zeta_c \in \mathcal{K}$.

Назовем ИСП U над базовым множеством \mathcal{F}_0 *правильно нагруженной сетью*, если для любого ребра c сети U нагрузка ребра c

$$[c] = \zeta_c = K_a,$$

где a — представитель соответствующего класса эквивалентности.

Лемма 2 *Для любого натурального числа t и любой ИСП U над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$ существует такая правильно нагруженная сеть U' , что $T(U, t, x) \equiv T(U', t, x)$ и функционирование сетей U и U' совпадает.*

Доказательство: Пусть нам дана сеть U . Заменяем нагрузку каждого ребра c сети U на функцию $K_a \equiv \zeta_c$, где a — представитель класса элементов b , для которых $K_b \equiv \zeta_c$. Полученную сеть обозначим через U' . По определению U' — правильно нагруженная сеть. Поскольку произведенная замена нагрузочных функций не меняет функций фильтра вершин сети, то очевидно, что функционирование сетей U и U' совпадает.

Поскольку вектора состояния сетей U и U' совпадают (нагрузка каждого ребра сети U' определялась как функция состояния ребра сети U), то очевидно, что для любого запроса $x \in X$

$$T(U, t, x) \equiv T(U', t, x).$$

Тем самым лемма доказана.

Пусть U — произвольная правильно нагруженная ИСП над базовым множеством \mathcal{F}_0 . Пусть $\{K_{a_1}, \dots, K_{a_q}\}$ — множество нагрузочных функций сети U , упорядоченное так, что $a_q \succeq a_{q-1} \succeq \dots \succeq a_1$. Тогда упорядоченное множество $\{a_1, \dots, a_q\}$ назовем *определяющим множеством* правильно нагруженной сети U .

Легко видеть, что так как сеть U правильно нагруженная, то для любого $i \in \{1, q-1\}$ $a_i \neq a_{i+1}$ и значит $a_q \succ a_{q-1} \succ \dots \succ a_1$.

Лемма 3 Для любого натурального числа m и любой правильно нагруженной ИСП U над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$, если $\{a_1, \dots, a_q\}$ — определяющее множество сети U и $a_0 = x_{min}$, $a_{q+1} = x_{max}$, то для любого $i \in \{\overline{r_1}, \overline{r_2}\}$, где

$$r_1 = \begin{cases} 1, & \text{если } a_1 = x_{min} \\ 0, & \text{если } a_1 \neq x_{min} \end{cases},$$

$$r_2 = \begin{cases} q-1, & \text{если } a_q = x_{max} \\ q, & \text{если } a_q \neq x_{max} \end{cases},$$

и для любых $x_1, x_2 \in [a_i, a_{i+1})$ справедливо

$$T(U, m, x_1) = T(U, m, x_2).$$

Доказательство: Возьмем произвольное число $i \in \{\overline{r_1}, \overline{r_2}\}$ и два произвольных запроса $x_1, x_2 \in [a_i, a_{i+1})$. Так как для любого $j \in \{\overline{r_1}, i\}$ $K_{a_j}(x_1) = K_{a_j}(x_2) = 1$ и для любого $j \in \{i+1, q\}$ $K_{a_j}(x_1) = K_{a_j}(x_2) = 0$, а других нагрузочных функций в сети U нет, то векторы состояния сети на запросах x_1 и x_2 совпадают, следовательно процедура обхода будет вести себя одинаково на запросах x_1 и x_2 и, значит,

$$T(U, m, x_1) = T(U, m, x_2).$$

Тем самым лемма доказана.

Пусть $I = \langle X, V, \succeq \rangle$ — ЗИП типа S_{lin} . Тогда упорядоченное множество $\{a_1, \dots, a_r\}$ ($a_r \succ a_{r-1} \succ \dots \succ a_1$) представителей классов эквивалентности записей из V назовем *опорным множеством* задачи I .

Понятно, что $r \leq |V|$.

Если I — произвольная ЗИП типа S_{lin} , то правильно нагруженную ИСП $U \in \mathcal{U}(I, \mathcal{F}_0)$, определяющее множество которой совпадает с опорным множеством ЗИП I , назовем *приведенной сетью задачи I* .

Лемма 4 Пусть I — произвольная ЗИП типа S_{lin} . Пусть U — произвольная ИСП над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$, разрешающая ЗИП I . Тогда для любого натурального числа m существует такая приведенная ИСП U' задачи I , что для любого $x \in X$

$$T(U', m, x) \leq T(U, m, x).$$

Доказательство: Согласно лемме 2 существует правильно нагруженная сеть U'' , функционирование которой совпадает с функционированием сети U и $T(U, m, x) \equiv T(U'', m, x)$. Пусть $\{b_1, \dots, b_q\}$ — определяющее множество сети U'' . Предположим, что в этом множестве существует элемент b_j , не принадлежащий опорному множеству задачи I . Выделим все ребра сети U'' , которым соответствует нагрузочная функция K_{b_j} . Рассмотрим отрезки $[b_{j-1}, b_j)$ и $[b_j, b_{j+1})$. Если какого-то из этих отрезков нет, то он и не рассматривается. Согласно лемме 3 для любого $x \in [b_{j-1}, b_j)$ $T(U'', m, x)$ является константой, которую обозначим через t_{j-1} , и для любого $x \in [b_j, b_{j+1})$ $T(U'', m, x)$ также является константой, которую обозначим через t_j . Если $t_{j-1} \leq t_j$, то заменим в сети U'' нагрузку выделенных ребер на $K_{b_{j+1}}$, иначе на $K_{b_{j-1}}$. Полученную сеть обозначим U''' .

Так как b_j не принадлежит опорному множеству ЗИП I , то проводимость выделенных ребер не определяла ни одну из функций фильтра листьев, следовательно после произведенной замены функционирование сети сохранится, то есть сеть U''' по-прежнему разрешает задачу I .

Покажем, что для любого $x \in X$ $T(U''', m, x) \leq T(U'', m, x)$.

Понятно, что для любых запросов $x \notin [b_{j-1}, b_{j+1})$ $T(U''', m, x) = T(U'', m, x)$, так как для этих запросов x вектора состояния сетей U''' и U'' совпадают. Если у выделенных ребер нагрузка была заменена на $K_{b_{j+1}}$, то для любого $x \in [b_{j-1}, b_{j+1})$ вектор состояния сети U''' будет совпадать с вектором состояния сети U'' при $x \in [b_{j-1}, b_j)$, то есть если $t_{j-1} \leq t_j$, то для любого $x \in [b_{j-1}, b_{j+1})$ $T(U''', m, x) = t_{j-1}$. Если у выделенных ребер нагрузка была заменена на $K_{b_{j-1}}$, то для любого $x \in [b_{j-1}, b_{j+1})$ вектор состояния сети U''' будет совпадать с вектором состояния сети U'' при $x \in [b_j, b_{j+1})$, то есть если $t_{j-1} > t_j$, то для любого $x \in [b_{j-1}, b_{j+1})$ $T(U''', m, x) = t_j$. Откуда следует, что для любого $x \in X$ $T(U''', m, x) \leq T(U'', m, x)$.

Отметим, что элемент b_j не принадлежит определяющему множеству сети U''' . Тем самым мы избавились от одного "лишнего" элемента. Если в определяющем множестве сети U''' есть элемент, не принадлежащий опорному множеству задачи I , то действуем по отношению к нему точно так же, как к элементу b_j . И таким образом за конечное число шагов мы придем к сети $U^{(r)}$, которая разрешает задачу I , определяющее множество $U^{(r)}$ покрывается опорным множеством задачи I , и для любого $x \in X$ $T(U''', m, x) \leq T(U'', m, x)$.

Осталось показать, что любой элемент a , принадлежащий опорному множеству ЗИП I , принадлежит определяющему множеству сети $U^{(r)}$.

Возьмем произвольный элемент a из опорного множества ЗИП I . Так как $U^{(r)}$ разрешает ЗИП I , то в сети $U^{(r)}$ существует такой лист α , что ему приписана запись y , равная a , и $\varphi_\alpha(y) = 1$. Следовательно так как $N_{\varphi_\alpha} \subseteq N_{K_\alpha}$, то $\varphi_\alpha(x) \equiv K_\alpha(x)$. Значит существует ребро, ведущее в α , через которое в α проходит запрос y , отсюда поскольку сеть $U^{(r)}$ правильно нагруженная сразу следует, что этому ребру приписан предикат K_a . Следовательно a принадлежит определяющему множеству сети $U^{(r)}$.

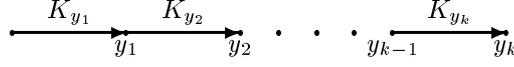


Рис. 1: Правильная цепочка

Тем самым лемма доказана.

Лемма 5 Если U такая правильно нагруженная ИСП над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$, что $\{a_1, \dots, a_q\}$ — ее определяющее множество, $l \in \{1, q\}$ — такой номер, что

$$(\mathbf{P}([a_l, x_{max}] > 0) \& (l = q \vee \mathbf{P}([a_{l+1}, x_{max}] = 0)),$$

и r — число ребер сети U , нагрузка которых принадлежит множеству $\{K_{a_1}, \dots, K_{a_l}, K_{a_{l+1}}\}$ (если $l = q$, то последний элемент отбрасывается), то для любого натурального числа m

$$T(U, m) \geq \mathbf{P}([a_l, x_{max}]) \cdot \left\lceil \frac{r}{m} \right\rceil.$$

Доказательство: Так как U — правильно нагруженная сеть, то ребра, которым приписаны функции $K_{a_{l+1}}$ (если $l < q$) исходят из вершин, функции фильтра которых принадлежат множеству $\{K_{a_1}, \dots, K_{a_l}\}$. Следовательно для любого запроса $x \in [a_l, x_{max}]$ нагрузки по крайней мере r ребер сети должны быть вычислены. А поскольку имеется m исполнителей, то для любого запроса $x \in [a_l, x_{max}]$ по крайней мере один исполнитель вычислит $\left\lceil \frac{r}{m} \right\rceil$ функций, то есть для любого запроса $x \in [a_l, x_{max}]$

$$T(U, m, x) \geq \left\lceil \frac{r}{m} \right\rceil.$$

Откуда

$$T(U, m) \geq \mathbf{P}([a_l, x_{max}]) \cdot \left\lceil \frac{r}{m} \right\rceil.$$

Тем самым лемма доказана.

Если $V = \{y_1, \dots, y_k\}$, где записи упорядочены следующим образом: $y_k \succeq y_{k-1} \succeq \dots \succeq y_1$, то цепочку нагруженных ребер, изображенную на рисунке 1, назовем *правильной цепочкой* для множества V .

Доказательство теоремы 1.

Возьмем произвольную ЗИП $I = \langle X, V, \succeq \rangle$ типа S_{lin} . Пусть $\{a_1, \dots, a_q\}$ — ее опорное множество.

Если $\mathbf{P}([a_1, x_{max}]) = 0$, то правильная цепочка для множества V разрешает ЗИП I и оптимальна, поскольку ее m -сложность равна 1, а сложность

любой сети всегда не меньше 1, поскольку из корня исходит по крайней мере одно ребро, и хотя бы одну функцию всегда надо вычислить.

Рассмотрим теперь случай, когда $\mathbf{P}([a_1, x_{max}]) > 0$.

Чтобы показать существование оптимальной сети для ЗИП I , покажем, что инфимум в выражении

$$T(I, \mathcal{F}_0, m) = \inf \{T(U, m) : U \in \mathcal{U}(I, \mathcal{F}_0)\} \quad (3)$$

берется по конечному множеству.

Согласно лемме 4 для любой сети над базовым множеством \mathcal{F}_0 , разрешающей ЗИП I , существует приведенная сеть не большей сложности, разрешающая ЗИП I . Поэтому инфимум в выражении (3) можно брать по множеству сетей, нагрузка ребер которых берется из множества $\{K_{a_1}, \dots, K_{a_q}\}$.

Существует такой номер $l \in \{\overline{1, q}\}$, что

$$(\mathbf{P}([a_l, x_{max}]) > 0) \& (l = q \vee \mathbf{P}([a_{l+1}, x_{max}]) = 0).$$

Обозначим

$$t = \begin{cases} l, & \text{если } l = q \\ l + 1, & \text{если } l < q \end{cases}.$$

Рассмотрим ЗИП $I' = \langle X, V', \succeq \rangle$, где V' — есть множество записей из V , равных элементам из $\{a_1, \dots, a_t\}$. Возьмем произвольную ИСП U_0 над базовым множеством \mathcal{F}_0 , разрешающую ЗИП I' .

Согласно лемме 5 любая правильно нагруженная сеть, разрешающая ЗИП I' , нагрузка ребер которой берется из множества $\{K_{a_1}, \dots, K_{a_t}\}$, и имеющая более

$$r_0 = \frac{T(U_0, m) \cdot m}{\mathbf{P}([a_l, x_{max}])}$$

ребер будет иметь сложность большую, чем $T(U_0, m)$. Отсюда следует, что инфимум в выражении (3), но для ЗИП I' , нужно брать по множеству правильно нагруженных сетей, нагрузка ребер которых берется из множества $\{K_{a_1}, \dots, K_{a_t}\}$, и имеющих не более r_0 ребер. Но таких сетей конечное число, следовательно существует оптимальная сеть, разрешающая ЗИП I' .

Если $l < q$, то есть если ЗИП I' отличается от ЗИП I , то возьмем произвольную оптимальную сеть U' , разрешающую ЗИП I' . Выберем в ней некоторый лист, которому соответствует запись из V' , равная a_{l+1} . Выпустим из этого листа сети правильную цепочку для множества $V \setminus V'$. Эта цепочка будет иметь нулевую сложность, а полученная сеть разрешать ЗИП I . Так m -сложность полученной сети равна сложности оптимальной сети U' , то полученная сеть будет оптимальной для ЗИП I .

Тем самым теорема 1 доказана.

5 Оптимальная сеть в классе сепаративных

Если m — некоторое натуральное число, I — некоторая ЗИП типа S_{lin} и сеть $U \in \mathcal{U}(I, \mathcal{F}_0)$ такая, что для любой ИСП $U' \in \mathcal{U}(I, \mathcal{F}_0)$ и любого $x \in X$

$$T(U, m, x) \leq T(U', m, x),$$

то скажем, что сеть U *равномерно m -минимальная* для ЗИП I .

Лемма 6 *Для любой ЗИП $I = \langle X, V, \succeq \rangle$ типа S_{lin} правильная цепочка для множества V является равномерно 1-минимальной сетью.*

Доказательство: Возьмем произвольную ЗИП $I = \langle X, V, \succeq \rangle$ типа S_{lin} .

Нижняя оценка. Пусть $\{a_1 \dots, a_q\}$ опорное множество задачи I . Пусть l_i ($i = \overline{1, q}$) — количество записей из V , равных a_i . Обозначим $a_0 = x_{min}$. Рассмотрим функцию

$$T_0(x) = \begin{cases} 1 + \sum_{j=1}^i l_j, & \text{если } x \in [a_i, a_{i+1}), i = \overline{0, q-1} \\ |V|, & \text{если } x \in [a_q, x_{max}]. \end{cases}$$

Покажем, что для любой ИСП $U' \in \mathcal{U}(I, \mathcal{F}_0)$ и любого $x \in X$

$$T(U', 1, x) \geq T_0(x).$$

Рассмотрим произвольную ИСП $U' \in \mathcal{U}(I, \mathcal{F}_0)$. Согласно лемме 4 существует такая приведенная сеть $U'' \in \mathcal{U}(I, \mathcal{F}_0)$, что любого $x \in X$

$$T(U'', 1, x) \leq T(U', 1, x).$$

Так как U'' приведенная сеть, то ее определяющее множество имеет вид $\{a_1 \dots, a_q\}$.

Возьмем произвольное число $i \in \{\overline{0, q-1}\}$ и произвольный запрос $x \in [a_i, a_{i+1})$. Так как U'' разрешает ЗИП I , то такой запрос должен дойти до листьев записи из V , приписанные которым, предшествуют a_{i+1} (эти листья назовем выделенными).

Таких записей будет $\sum_{j=1}^i l_j$. Поскольку в каж-

дый выделенный лист запрос проходит по некоторому ребру, то должен быть подсчитан предикат, приписанный этому ребру. Следовательно долж-

но быть подсчитано по крайней мере $\sum_{j=1}^i l_j$ предикатов. Кроме того должно

быть хотя бы одно ребро, исходящее из вершин, функция фильтра которых равна 1 на запросе x , и ведущее в невыделенную вершину, поскольку в противном случае не будет ребра, через которое можно пройти к листьям,

которым соответствуют записи из V , равные a_{i+1} . Это добавляет еще одну 1 к числу вычисленных функций. Таким образом при $x \in [a_i, a_{i+1})$

$$T(U'', 1, x) \geq 1 + \sum_{j=1}^i l_j.$$

Возьмем произвольный запрос $x \in [a_q, x_{max}]$. Такой запрос должен дойти до всех записей из V . А поскольку к каждой записи мы попадем по некоторому ребру, а всего записей $|V|$, то при $x \in [a_q, x_{max}]$

$$T(U'', 1, x) \geq |V|.$$

Таким образом мы показали, что при любых $x \in X$

$$T(U', 1, x) \geq T(U'', 1, x) \geq T_0(x).$$

Верхняя оценка. Простой проверкой легко убедиться, что правильная цепочка для множества V (обозначим ее через U_0) разрешает ЗИП I и $T(U_0, 1, x) \equiv T_0(x)$.

Отсюда с учетом нижней оценки имеем, что правильная цепочка для множества V является равномерно 1-минимальной сетью для ЗИП I .

Тем самым лемма доказана.

Лемма 7 *Для любого натурального числа t и любой t -сепаративной ИСП U над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$ существует такая t -сепаративная правильно нагруженная сеть U' , что $T(U, t, x) \equiv T(U', t, x)$ и функционирование сетей U и U' совпадает.*

Доказательство: Для доказательства леммы достаточно убедиться, что правильно нагруженная сеть U' , построенная для сети U в доказательстве леммы 2, является t -сепаративной. А это действительно так, поскольку для любого $x \in X$ вектора состояния сетей U и U' совпадают.

Тем самым лемма доказана.

Лемма 8 *Пусть I — произвольная ЗИП типа S_{lin} . Пусть U — произвольная t -сепаративная ИСП над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$, разрешающая ЗИП I . Тогда для любого натурального числа t существует такая t -сепаративная приведенная ИСП U' задачи I , что для любого $x \in X$*

$$T(U', t, x) \leq T(U, t, x).$$

Доказательство: Доказательство леммы аналогично доказательству леммы 4 и надо только убедиться, что построенная в доказательстве леммы

4 сеть U''' является m -сепаративной. В самом деле вектора состояния сети U''' и m -сепаративной сети U'' совпадают на всем X , кроме отрезка $[b_{j-1}, b_{j+1})$, но поскольку вектор состояния сети U''' на этом отрезке совпадает с вектором состояния сети U'' на одном из отрезков $[b_{j-1}, b_j)$ или $[b_j, b_{j+1})$, на котором сохраняется свойство m -сепаративности, то сеть U''' остается m -сепаративной.

Тем самым лемма доказана.

Лемма 9 *Для любого натурального числа m и любой ЗИП I типа S_{lin} существует оптимальная в классе m -сепаративных сеть над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$.*

Доказательство: Доказательство леммы аналогично доказательству теоремы 1, но только вместо лемм 2 и 4 надо воспользоваться леммами 7 и 8.

Тем самым лемма доказана.

Систему подмножеств V_1, \dots, V_n некоторого множества V назовем *разбиением* V , если $V_1 \cup V_2 \cup \dots \cup V_n = V$ и для любых $i, j \in \{\overline{1, n}\}$ из того, что $i \neq j$ следует $V_i \cap V_j = \emptyset$.

Если V_1, \dots, V_n разбиение некоторого множества V , то ИСП, составленную из n правильных цепочек для множеств V_1, \dots, V_n соответственно, растущих из одного общего корня, назовем *правильной n -метелкой разбиения* V_1, \dots, V_n .

Лемма 10 *Для любого натурального числа m и любой ЗИП $I = \langle X, V, \succeq \rangle$ типа S_{lin} существует такое разбиение множества V , что правильная метелка этого разбиения будет оптимальной в классе m -сепаративных сетью над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$ для задачи I .*

Доказательство: Возьмем произвольную ЗИП $I = \langle X, V, \succeq \rangle$ типа S_{lin} . Согласно леммам 8 и 9 для I существует приведенная оптимальная в классе m -сепаративных сеть, которую обозначим через U .

Обозначим через U_i ($i \in \overline{1, m}$) подсеть сети U , составленную только из ребер, которые хотя бы на одном запросе из X окрашиваются процедурой обхода i -ым исполнителем. Поскольку сеть U m -сепаративна, то реберное пересечение подсетей U_i невозможно, но подсети U_i могут пересекаться по вершинам. Предположим, что некоторые подсети U_{i_1}, \dots, U_{i_l} пересекаются по вершине α . Продублируем вершину α l раз (если это лист, то дублируем с нагрузкой) по одной на каждую подсеть U_{i_1}, \dots, U_{i_l} , и тем самым ликвидируем это вершинное пересечение подсетей. Такую операцию проделаем для всех вершин пересечения кроме корня и в результате получим, что подсети U_1, \dots, U_m будут пересекаться только в корне. Нетрудно убедиться, что полученная сеть разрешает ЗИП I и сложность ее такая же, что и у исходной.

Теперь, если некоторая запись y приписана нескольким листьям, принадлежащим разным подсетям U_{i_1}, \dots, U_{i_l} , то выберем среди них одну такую, лист с записью y которой имеет функцию фильтра, равную K_y . Такая сеть обязательно найдется. У остальных сетей листья с записью y объявим обычными вершинами и снимем с них нагрузку y . Такую операцию проделаем для всех записей приписанных разным подсетям и в результате получим сеть с тем же функционированием и сложностью, в которой каждая запись из V приписана только одной из подсетей U_1, \dots, U_m . Поэтому, если через V_i ($i = \overline{1, m}$) обозначить записи, приписанные листьям подсети U_i , то система V_1, \dots, V_m образует разбиение множества V .

Если x — некоторый запрос из X , то обозначим через $T_i(x)$ количество вычисленных i -ым исполнителем предикатов при подаче на вход процедуры обхода запроса x . Легко видеть, что $T_i(x)$ — это 1-сложность сети U_i . Если через $T_i^0(x)$ обозначить 1-сложность правильной цепочки для множества V_i , то согласно лемме 6 для любого $x \in X$ $T_i(x) \geq T_i^0(x)$. Отсюда если, через U_0 обозначить правильную m -метелку разбиения V_1, \dots, V_m , то получим, что

$$T(U, m, x) = \min_{1 \leq i \leq m} T_i(x) \geq \min_{1 \leq i \leq m} T_i^0(x) = T(U_0, m, x).$$

Отсюда поскольку сеть U оптимальная в классе m -сепаративных для ЗИП I , то и сеть U_0 оптимальная в классе m -сепаративных для ЗИП I .

Тем самым лемма доказана.

Подмножество вершин правильной метелки некоторого разбиения назовем *ярусом высоты h* , если от корня до каждой вершины яруса ведет цепь из h ребер.

Высотой ИСП назовем длину самой длинной ориентированной цепи, исходящей из корня ИСП.

Лемма 11 Пусть U — правильная m -метелка некоторого разбиения V_1, \dots, V_m . Пусть h — высота ИСП U . Пусть b_i ($i = \overline{1, h-1}$) — минимальный элемент множества записей, приписанных листьям i -го яруса, из которых исходят ребра, и $b_0 = x_{min}$. Тогда

$$T(U, m, x) = \begin{cases} 1 + i, & \text{если } x \in [b_i, b_{i+1}), i = \overline{0, h-2} \\ h, & \text{если } x \in [b_{h-1}, x_{max}] \end{cases}$$

и

$$T(U, m) = 1 + \sum_{i=1}^{h-1} \|b_i\|.$$

Доказательство: Сначала отметим, что поскольку метелка правильная и в каждой цепочке записи идут в порядке возрастания, то $b_h \geq b_{h-1} \geq \dots \geq b_1$.

Возьмем произвольное число $i \in \{\overline{0, h-2}\}$ такое, что $[b_i, b_{i+1}) \neq \emptyset$. Возьмем произвольный запрос $x \in [b_i, b_{i+1})$. Поскольку $x \geq b_i$, то запрос дойдет

до вершины, которой приписана запись b_i (если $i = 0$, то до корня). С другой стороны $b_{i+1} \succ x$, следовательно x либо не пройдет ни в какую вершину $(i + 1)$ -го яруса, либо пройдет только в те из них, из которых не исходят ребра. Значит при $x \in [b_i, b_{i+1})$ $T(U, m, x) = 1 + i$.

И наконец, если $x \succeq b_{h+1}$, то все h предикатов некоторой самой длинной цепи будут вычислены и значит $T(U, m, x) = h$.

Отсюда следует

$$\begin{aligned}
T(U, m) &= \mathbf{M}_x T(U, m, x) = \\
&= h \cdot \mathbf{P}([b_{h-1}, x_{max}]) + \sum_{i=0}^{h-2} (1 + i) \cdot \mathbf{P}([b_i, b_{i+1})) = \\
&= \sum_{i=0}^{h-1} \left(\mathbf{P}([b_{h-1}, x_{max}]) + \sum_{j=i}^{h-2} \mathbf{P}([b_j, b_{j+1})) \right) = \\
&= \sum_{i=0}^{h-1} \mathbf{P}([b_i, x_{max}]) = 1 + \sum_{i=1}^{h-1} \|b_i\|.
\end{aligned}$$

Тем самым лемма доказана.

Зафиксируем число исполнителей $m \geq 1$.

Будем считать, что в правильной m -метелке правильные цепочки занумерованы в порядке следования ребер в пучке корня метелки. Этот номер определяет какой исполнитель обрабатывает цепочку. Легко видеть, что от перемены нумерации цепочек функция m -сложности метелки на запросе не изменится. (В общем случае для произвольных ИСП это утверждение не верно, поскольку работа процедуры обхода существенно зависит от порядка следования ребер в пучках.) Поэтому мы всюду в дальнейшем будем считать, что цепочки в правильной метелке занумерованы в порядке возрастания длин цепочек. (Длина цепочки — это количество ребер в ней.)

Пусть нам дана произвольная правильная m -метелка. Занумеруем ее листья, начиная с 1, следующим образом:

- листья, принадлежащие ярусу с меньшей высотой имеют меньшие номера,
- внутри одного яруса листья занумерованы в порядке возрастания номеров цепочек, которым принадлежат листья.

Напомним, что в правильной метелке все вершины кроме корня являются листьями. Пример нумерации листьев правильной 4-метелки изображен на рисунке 2.

Пусть U — правильная m -метелка высоты h , тогда последовательность чисел $(i_1, i_2, \dots, i_{h-1})$ назовем *кодом m -метелки U* , если i_j — есть минимальный номер вершины j -го яруса, из которой исходит ребро ($j = \overline{1, h-1}$).

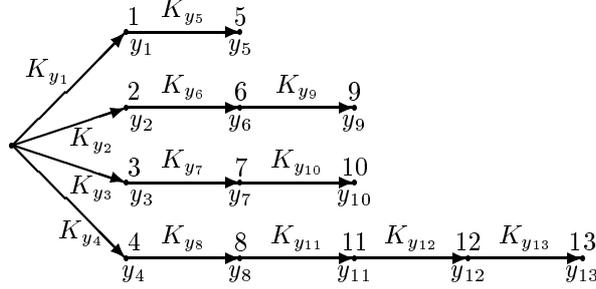


Рис. 2: Правильная 4-метелка с номерами листьев

Например, правильная m -метелка, изображенная на рисунке 2, имеет код $(1, 6, 11, 12)$.

Пусть U — правильная m -метелка некоторого разбиения библиотеки $V = \{y_1, \dots, y_k\}$, где $y_k \succeq y_{k-1} \succeq \dots \succeq y_1$, тогда U назовем простой m -метелкой, если для любого $i \in \{1, k\}$ листу с номером i соответствует запись y_i .

Например, если $y_{13} \succeq y_{12} \succeq \dots \succeq y_1$, то правильная m -метелка, изображенная на рисунке 2, является простой.

Лемма 12 Пусть t — произвольное натуральное число и U_0 — простая t -метелка с кодом $(i_1, i_2, \dots, i_{h-1})$ разбиения V_1, \dots, V_m библиотеки V . Тогда для любой правильной t -метелки U с тем же кодом $(i_1, i_2, \dots, i_{h-1})$ того же разбиения V_1, \dots, V_m библиотеки V и для любого запроса $x \in X$ справедливо

$$T(U_0, t, x) \leq T(U, t, x).$$

Доказательство: Обозначим $y_{i_0} = x_{min}$. Пусть $V = \{y_1, \dots, y_k\}$, где $y_k \succeq y_{k-1} \succeq \dots \succeq y_1$, тогда согласно лемме 11

$$T(U_0, t, x) = \begin{cases} 1 + j, & \text{если } x \in [y_{i_j}, y_{i_{j+1}}), j = \overline{0, h-2} \\ h, & \text{если } x \in [y_{i_{h-1}}, x_{max}]. \end{cases} \quad (4)$$

Возьмем произвольную правильную t -метелку U с кодом $(i_1, i_2, \dots, i_{h-1})$ разбиения V_1, \dots, V_m библиотеки V . Пусть y_{l_j} — минимальный элемент множества записей, приписанных листьям j -го яруса, из которых исходят ребра, в t -метелке U ($j = \overline{1, h-1}$). Пусть $y_{l_0} = x_{min}$. Тогда согласно лемме 11

$$T(U, t, x) = \begin{cases} 1 + j, & \text{если } x \in [y_{l_j}, y_{l_{j+1}}), j = \overline{0, h-2} \\ h, & \text{если } x \in [y_{l_{h-1}}, x_{max}]. \end{cases} \quad (5)$$

Поскольку вдоль каждой цепочки сети U записи идут в порядке возрастания, то для любого $j \in \{1, h-1\}$ $i_j \geq l_j$. Отсюда с учетом выражений (4), (5) получает утверждение леммы.

Тем самым лемма доказана.

Доказательство теоремы 2.

Возьмем произвольное натуральное число m , которое будет определять число исполнителей. Возьмем произвольную ЗИП $I = \langle X, V, \succeq \rangle$ типа S_{lin} , такую, что $V = \{y_1, \dots, y_k\}$ — упорядоченное множество, такое, что $y_k \succeq y_{k-1} \succeq \dots \succeq y_1$,

Согласно лемме 9 для ЗИП I существует оптимальная в классе m -сепаративных сеть над базовым множеством $\mathcal{F}_0 = \langle \mathcal{K}, \emptyset \rangle$.

Согласно леммам 10 и 12 оптимальную в классе m -сепаративных сеть надо искать в множестве простых m -метелок. Используя этот факт, докажем индукцией по числу m , что

$$T_s(I, m) = \inf \{T(U, m) : U \in \mathcal{U}_s(I, \mathcal{F}_0, m)\} = 1 + R(m, V),$$

где $R(m, V)$ — функция определяемая соотношениями (1), (2).

Базис индукции. $m = 1$.

Согласно лемме 6 правильная цепочка U множества V будет оптимальной для ЗИП I . Согласно лемме 11

$$T_s(I, 1) = T(U, 1) = 1 + \sum_{i=1}^{k-1} \|y_i\| = 1 + R(1, V).$$

Индуктивный переход. Пусть для любого натурального $l < m$ и любой ЗИП $I' = \langle X, V', \succeq \rangle$ типа S_{lin} справедливо

$$T_s(I', l) = 1 + R(l, V').$$

Докажем справедливость утверждения для числа m и ЗИП I .

Поскольку оптимальную в классе m -сепаративных сеть надо искать в множестве простых m -метелок, будем перебирать простые m -метелки в зависимости от длины первой цепочки. Поскольку ее длина самая минимальная, то значит она может изменяться в пределах от 1 до $\lfloor k/m \rfloor$.

Возьмем произвольное число $j \in \{1, \lfloor k/m \rfloor\}$. Пусть U_j — простая m -метелка для множества V , у которой длина первой цепочки равна j , и которая имеет минимальную m -сложность среди всех простых m -метелок для множества V , у которых длина первой цепочки равна j . Согласно лемме 11 ее m -сложность равна

$$T(U_j, m) = T(U'_j, m-1) + 1 + \sum_{i=1}^{l-1} \|y_{1+(i-1)m}\|,$$

где $T(U'_j, m-1)$ — $(m-1)$ -сложность подсети U'_j , которая состоит из $m-1$ подцепочек всех цепочек, кроме первой, начинающихся с $(j-1)$ -го яруса сети U_j . Здесь берется $(m-1)$ -сложность потому, что обрабатывать эту подсеть U'_j будет уже $m-1$ исполнитель, поскольку первый исполнитель завершает на этом свою работу. Поскольку U_j имеет минимальную m -сложность

среди всех простых m -метелок для множества V , у которых длина первой цепочки равна j , то нетрудно заметить, что

$$T(U'_j, m-1) = T_s(I^j, m-1) - 1,$$

где $I^j = \langle X, V^{(j-1)m+2}, \succeq \rangle$, и $V^{(j-1)m+2} = \{y_{(j-1)m+2}, y_{(j-1)m+3}, \dots, y_k\}$. Откуда согласно предположению индукции получим

$$T(U_j, m) = 1 + R(m-1, V^{(j-1)m+2}) + \sum_{i=1}^{j-1} \|y_{1+(i-1)m}\|.$$

Откуда

$$\begin{aligned} T_s(I, m) &= \min_{1 \leq j \leq \lfloor k/m \rfloor} T(U_j, m) = \\ &= 1 + \min_{1 \leq j \leq \lfloor k/m \rfloor} \left(R(m-1, V^{(j-1)m+2}) + \sum_{i=1}^{j-1} \|y_{1+(i-1)m}\| \right) = \\ &= 1 + R(m, V). \end{aligned}$$

Тем самым теорема 2 доказана.

6 Неоптимальность сепаративных сетей

В данном разделе мы докажем теорему 3, то есть приведем ситуацию, в которой оптимальная сеть не является сепаративной.

Пусть $X = Y = [0, 1]$ — отрезок вещественной прямой. Пусть отношение поиска есть отношение " $>$ " для вещественных чисел. Пусть на X задана равномерная вероятность, то есть вероятность отрезка равна его длине. Тогда для любого $y \in X$

$$\|y\| = \mathbf{P}(O(y, >)) = 1 - y.$$

Пусть число исполнителей $m = 2$. Рассмотрим некоторую библиотеку $V = \{y_1, \dots, y_6\}$, в которой записи упорядочены по возрастанию, то есть $y_1 < y_2 < \dots < y_6$. Рассмотрим ЗИП $I = \langle X, V, > \rangle$. Согласно теореме 2 сложность $T_s(I, 2)$ оптимальной в классе 2-сепаративных сети равна максимуму из трех чисел:

$$T_1 = 1 + \|y_2\| + \|y_3\| + \|y_4\| + \|y_5\| = 5 - y_2 - y_3 - y_4 - y_5,$$

$$T_2 = 1 + \|y_1\| + \|y_4\| + \|y_5\| = 4 - y_1 - y_4 - y_5,$$

$$T_3 = 1 + \|y_1\| + \|y_3\| = 3 - y_1 - y_3.$$

Рассмотрим сеть U^* , изображенную на рисунке 3, где порядок следования ребер в пучке корня следующий: первым идет ребро, ведущее в y_4 , вторым — в y_2 , и последним — в y_1 .

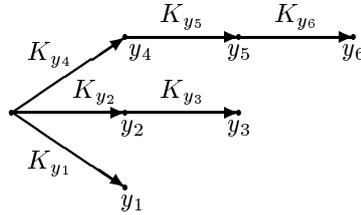


Рис. 3: Сеть U^*

Очевидно, что сеть U^* разрешает ЗИП I .

Применив процедуру обхода, легко проверить, что $T(U^*, 2, x) = 2$ при $x \in [0, y_4)$, и $T(U^*, 2, x) = 3$ при $x \in [y_4, 1]$. Отсюда

$$T(U^*, 2) = 3(1 - y_4) + 2(y_4 - 0) = 3 - y_4.$$

Согласно процедуре обхода при $x \in [0, y_5)$ ребро, ведущее из корня в вершину с записью y_1 , просматривается первым исполнителем, а при $x \in [y_5, 1]$ — вторым. Следовательно сеть U^* не является 2-сепаративной.

Возьмем теперь $y_1 = 0.05$, $y_2 = 0.1$, $y_3 = 0.15$, $y_4 = 0.8$, $y_5 = 0.9$, $y_6 = 1$. Тогда $T_1 = 3.05$, $T_2 = 2.25$, $T_3 = 2.8$, $T(U^*, 2) = 2.2$. Откуда следует, что для такой задачи I 2-сепаративная сеть не может быть оптимальной.

Тем самым теорема 3 доказана.

Список литературы

- [1] Гасанов Э. Э. Оптимальные информационные сети для отношений поиска, являющихся отношениями линейного квази порядка. *Конструкции в алгебре и логике*. Изд-во Тверского гос. ун-та, Тверь, 1990, 11-17.
- [2] Ли Д., Препарата Ф. Вычислительная геометрия. Обзор. *Кибернетический сб.* (1987) **24**, 5-96.
- [3] Гасанов Э. Э. Мгновенно решаемые задачи поиска. *Дискретная математика* (1996) **8**, N 3, 119-134.
- [4] Гасанов Э. Э. Об одной математической модели информационного поиска. *Дискретная математика* (1991) **3**, N 2, 69-76.