

ПОИСК ОБЪЕКТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ С ПОМОЩЬЮ РЕШЕНИЯ ЗАДАЧИ ИНТЕРВАЛЬНОГО ПОИСКА

Лялин В.Е., Мальцев С.А., Тарануха В.П., Шишов Д.Р.

Статья посвящена оптимизации алгоритмов интервального поиска над различными базовыми множествами с использованием информационно-графовых моделей, направленной на сокращение временных затрат при восстановлении резервной информации.

Ключевые слова: интервальный поиск, графовая модель, поиск информации.

Введение

Обозначим временной отрезок резервных копий множеством точек на отрезке $[0; 1]$. Задача интервального поиска является распространенной геометрической задачей, согласно которой при подаче запроса на поиск отрезка $[a, b] \subseteq [0, 1]$ необходимо перечислить элементы множества, входящие $[a; b]$. В статье рассматриваются алгоритмы, возникающие при различных базовых множествах. В одномерной задаче интервального поиска множество записей Y_{int} есть отрезок $[0, 1]$, множество запросов X_{int} – множество всех отрезков $[u, v] \subseteq [0, 1]$, или, другими словами, множество пар точек (u, v) , таких, что $0 \leq u \leq v \leq 1$, то есть

$$X_{\text{int}} = \{x = (u, v) : 0 \leq u \leq v \leq 1\}.$$

Зададим на X_{int} вероятностное пространство $\langle X_{\text{int}}, \sigma, P \rangle$, где σ – алгебра подмножеств множества X_{int} , состоящая из всех прямоугольников и прямоугольных равнобедренных треугольников со сторонами, параллельными осям координат, и на плоскости uOv [1]. P – вероятностная мера на σ , определяемая плотностью распределения вероятностей $p(u, v)$ для любого $B \in \sigma$, то есть

$$P(B) = \int_B p(u, v) du dv.$$

Будем считать, что плотность вероятностей $p(u, v)$ определена на всем квадрате $[0, 1] \times [0, 1]$, но $p(u, v) = 0$ при $(u, v) \notin X_{\text{int}}$.

Отношение интервального поиска ρ_{int} определим соотношением $(u, v) \rho_{\text{int}} y \Leftrightarrow u \leq y \leq v$, где $(u, v) \in X_{\text{int}}$, $y \in Y_{\text{int}}$. Обозначив $S_{\text{int}} = \langle X_{\text{int}}, Y_{\text{int}}, \rho_{\text{int}}, \sigma, P \rangle$ типом одномерного интервального поиска, исследуем задачи данного типа над различными базовыми множествами.

Представление бинарного поиска с помощью графовой модели

Базовое множество характеристических функций. Рассмотрим $I = \langle X, V, \rho \rangle$: $|V| = k$ над базовым множеством $\tilde{F}_1 = \langle F_1, \emptyset \rangle$, $F_1 = \{\chi_{y, \rho_{\text{int}}} : y \in Y_{\text{int}}\}$. Напомним, что

$$\chi_{y, \rho_{\text{int}}}(x) = \begin{cases} 1, & \text{если } x \rho_{\text{int}} y; \\ 0, & \text{в противном случае.} \end{cases}$$

Отметим, что N_f имеет вид, представленный на рис. 1, а значит, $N_f \in \sigma$ для любого предиката $f \in F_1$. Кроме того, понятно, что \tilde{F}_1 полно для S_{int} . Рассмотрим следующее утверждение.

Если $I = \langle X, V, \rho \rangle$ и $\tilde{F} = \langle F, \emptyset \rangle$ такие, что

$$\forall y \in V \chi_{y, \rho} \in F, O(y, \rho) \setminus \left(\bigcup_{f \in F \setminus \{\chi_{y, \rho}\}} N_f \right) \neq \emptyset,$$

то $T(I, \tilde{F}) = |V|$. Доказательство данной теоремы приводится в [1].

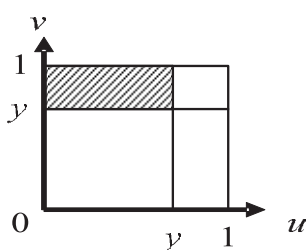


Рис. 1. Характеристическое множество вида 1

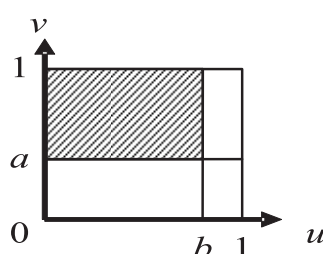


Рис. 2. Характеристическое множество вида 2

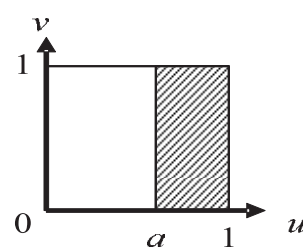


Рис. 3. Характеристическое множество вида 3

Множество $\tilde{X} = O(y, \rho) \setminus (\bigcup_{f \in F \setminus \{X_{y,\rho}\}} N_f)$ является множеством всех запросов x , таких, что $X_{y,\rho}(x)$ – единственный предикат из F , принимающий на y значение «1». Покажем, что оно не пусто. Для этого необходимо найти хотя бы один запрос, который для произвольного y попадет в множество \tilde{X} . Рассмотрим запрос (y, y) . Для любого $y \in Y_{\text{int}}$ очевидно выполняется $(y, y) \in N_{X_{y,\rho_{\text{int}}}}$, а для любого $y' \in Y_{\text{int}} : y' \neq y$ выполняется $(y, y) \notin N_{X_{y',\rho_{\text{int}}}}$, то есть запрос (y, y) принадлежит множеству \tilde{X} , а значит, $\tilde{X} \neq \emptyset$. Таким образом, $T(I, \tilde{F}) = |V| = k$.

Простое базовое множество. Обозначим

$$M_{a,b} = \{x = (u, v) \in X_{\text{int}} : u \leq b, v \geq a\}.$$

Рассмотрим случай, когда базовое множество предикатов равно

$$F_2 = \{f_{a,b} : (a, b) \in X_{\text{int}}, N_{f_{a,b}} = M_{a,b}\},$$

а базовое множество равно $\tilde{F}_2 = \langle F_2, \emptyset \rangle$. Отметим, что N_f имеет вид, представленный на рис. 2, а значит, $N_f \in \sigma$ для любого предиката $f \in F_2$ и для любой записи $y \in Y_{\text{int}}$ выполняется $X_{y,\rho_{\text{int}}} = f_{y,y}$, то есть \tilde{F}_2 полно для S_{int} .

Тогда для сложности ЗИП $I = \langle X_{\text{int}}, V, \rho_{\text{int}} \rangle$ при $k = |V| \rightarrow \infty$ справедливы следующие нижняя и верхняя оценки:

$$\sum_{y \in V} P(O(y, \rho_{\text{int}})) \leq T(I, \tilde{F}_2) \leq \sum_{y \in V} P(O(y, \rho_{\text{int}})) + O(\sqrt{k}).$$

Доказательство данного утверждения приводится в [1].

Базовое множество логарифмического поиска. Рассмотрим базовое множество предикатов $F_3 = F_2 \cup \{f_{0,a} : a \in [0, 1]\}$, где

$$\bar{f}_{0,a}(x) = \begin{cases} 0, & \text{если } u \leq a \\ 1, & \text{если } u > a \end{cases}.$$

Положим, что $\tilde{F}_3 = \langle F_3, \emptyset \rangle$. $N_{\bar{f}_{0,a}}$ имеет вид, представленный на рис. 3, а значит, $N_f \in \sigma$ для любого $f \in F_3$. Пусть V – упорядоченная библиотека мощности k ($V = \{y_1, \dots, y_k\}$, причем $0 \leq y_1 \leq \dots \leq y_k \leq 1$). Тогда для сложности ЗИП $I = \langle X_{\text{int}}, V, \rho_{\text{int}} \rangle$ при $k \rightarrow \infty$ справедлива следующая верхняя оценка:

$$T(I, \tilde{F}_3) \leq \sum_{i=1}^{k-1} P(O(y_i, \rho_{\text{int}})) + 2 \lceil \log_2 k \rceil,$$

где запись $\lceil \log_2 k \rceil$ обозначает наименьшее целое число, которое больше или равно $\log_2 k$. При

этом ИГ, для которого получена данная оценка, строится нижеописанным способом.

Возьмем произвольное ориентированное сбалансированное бинарное дерево с корнем и k листьями высоты $\lceil \log_2 k \rceil$, у которого все ребра ориентированы по направлению от корня к листьям. Обозначим правые и левые ребра бинарного дерева соответственно символами «П» и «Л». Сопоставим каждому листу слово, состоящее из символов «Л» и «П», согласно последовательности символов, приписанных ребрам, составляющим цепь из корня в рассматриваемый лист. Таким образом, исходящему из корня ребру будет соответствовать первый символ в записи, а входящему в лист – последний. Упорядочим листья согласно порядку слов, считая символ «Л» меньше символа «П». Обозначим i -ый лист α_i и сопоставим ему запись y_i ($i = 1, k$).

Теперь определим нагрузку ребер дерева предикатами из F_3 . Для этого рассмотрим некоторую произвольную внутреннюю вершину β . Обозначим множество записей, соответствующих листьям ветвей, растущих из вершины β , через V_β . Обозначим запись с минимальным в V_β номером как y_i , а с максимальным как y_j . Заметим, что в V_β записи идут друг за другом, то есть $\forall l \in \{i, j\} \ y_l \in V_\beta$. Через β' и β'' обозначим вершины с ребрами, исходящими из β . Пусть y'_j – запись с максимальным номером в $V_{\beta'}$. В случае, если β' – не лист, сопоставим ребру, ведущему из β в β' , предикат f_{0,y'_j} , а если лист, то – f_{y_j,y'_j} . Правому ребру сопоставим \bar{f}_{0,y'_j} , если β'' – не лист, и $f_{y'_j,y''_j}$, если β'' – лист, где y''_j – запись с максимальным номером в $V_{\beta''}$. Прделав данную операцию для всех внутренних вершин, мы определим нагрузку ребер. Теперь из всех листов α_i ($i = 1, k-1$) выпустим ребра в листья α_{i+1} и припишем этим ребрам предикаты $f_{y_{i+1},y_{i+1}}$. Полученный граф обозначим через U^1 . На рис. 4 представлен пример графа U^1 при $k = 10$.

Алгоритм поиска, соответствующий графу U^1 , состоит в следующем. С помощью бинарного поиска в отсортированной по возрастанию библиотеке за $\lceil \log_2 k \rceil$ шагов определяется самая левая запись, расположенная правее левого конца запроса либо совпадающая с ним. Далее, от найденной записи двигаясь вправо, просматриваемые записи поочередно сравниваются с правым концом запроса. При этом если оказывается, что просмотренная запись не больше правого конца, то она включается в ответ, иначе поиск завершается, т.е. этот алгоритм решения представляет собой традиционный поиск с помощью структуры,

называемой прошитым двоичным деревом. Доказано [2], что описанный алгоритм оптимален по времени поиска в худшем случае и по памяти.

Пусть, как и прежде, V – упорядоченная библиотека мощности k . Тогда для любого k существуют такие ЗИП $I = \langle X_{\text{int}}, V, \rho_{\text{int}} \rangle$ и плотность распределения вероятностей $p^k(u, v)$, что для

любого измеримого базового множества \tilde{F}_3 справедлива следующая нижняя оценка сложности включающего поиска:

$$T(I, \tilde{F}_3) \geq \sum_{y \in V_k} P(O(y_i, \rho_{\text{int}})) + \frac{(3 \log_3 k - 1)k}{2k + 1}.$$

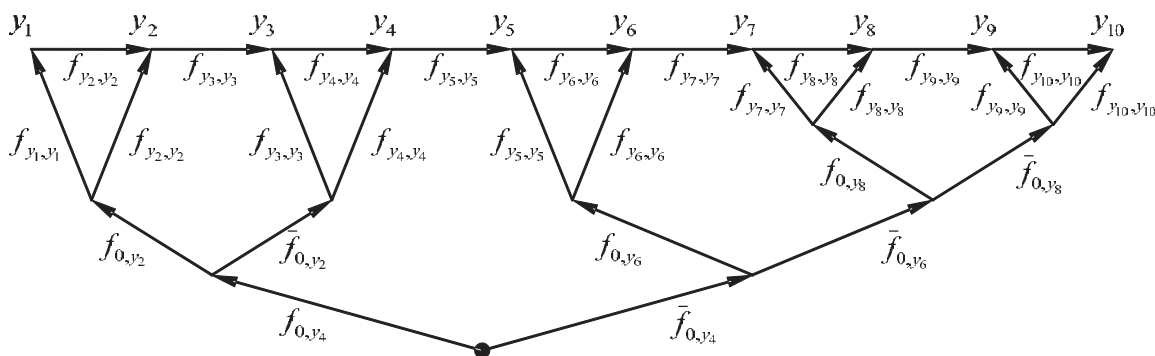


Рис. 4. Пример ИГ логарифмического поиска

Оптимизация алгоритма бинарного поиска в информационно-графовой модели

Базовое множество сверхлогарифмического поиска. Рассмотрим следующие два предиката:

$$f_{-,a} : N_{f_{-,a}} = \{(u, v) \in X_{\text{int}} : 0 \leq v - u \leq a\}$$

$$\tilde{f}_{-,a} : N_{\tilde{f}_{-,a}} = \{(u, v) \in X_{\text{int}} : (v - u > a) \vee (v - u < 0)\},$$

где $0 \leq a \leq 1$.

Характеристические множества $N_{f_{-,a}}$ и $N_{\tilde{f}_{-,a}}$ представлены на рис. 5 и рис. 6 соответственно.

Пусть базовое множество предикатов равно:

$$F_4 = F_3 \cup \{f_{-,a}, a \in [0, 1]\} \cup \{\tilde{f}_{-,a}, a \in [0, 1]\}.$$

Понятно, что $N_f \in \sigma$ для любого $f \in F_4$.

Положим $\tilde{F}_4 = \langle F_4, \emptyset \rangle$. Пусть $I = \langle X_{\text{int}}, V, \rho_{\text{int}} \rangle$ – ЗИП типа S_{int} . Построим ПИГ U^2 над базовым множеством $\tilde{F}_4 = \langle F_4, \emptyset \rangle$, разрешающий данную ЗИП.

Упорядочим записи в библиотеке $V = \{y_1, \dots, y_k\}$ по возрастанию ($y_1 \leq \dots \leq y_k$). Обозначим $m = \lceil \log_2 k \rceil$, то есть максимальное целое число, не превосходящее $\log_2 k$. Пусть $S = \{s_1, \dots, s_m\}$, где $s_i = \frac{i}{m+1}$, $i = 1, m$. Для каждого s_i ($i = 1, m$) найдем целое число l_i – номер ближайшей к s_i записи из V , меньшей чем s_i , а в случае отсутствия такой записи примем $l_i = 0$.

Аналогичным образом, как и для базового множества логарифмического поиска, построим для V граф U^1 . Перестроим его в U^2 по следующему алгоритму.

Объявим корень U^1 как обычную внутреннюю вершину и обозначим как β_1 . Проведем в β_1 ребро с началом β_0 , которое объявим корнем нового графа. Для этого ребра определим предикат $f_{-, \frac{1}{m+1}}$ из F_4 . Далее выпустим из корня еще одно ребро с предикатом $\tilde{f}_{-, \frac{1}{m+1}}$ и построим ориентированное сбалансированное бинарное дерево с m вершинами и корнем β_2 , ребра которого направлены от β_2 к конечным вершинам. Как и в случае логарифми-

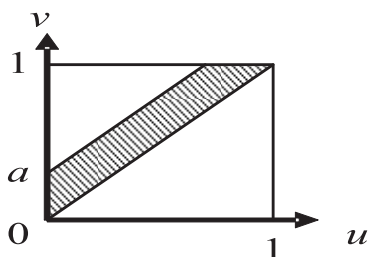


Рис. 5. Характеристическое множество вида 4

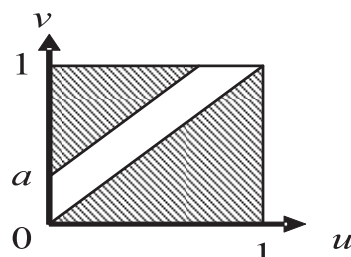


Рис. 6. Характеристическое множество вида 5

ческого поиска, введем понятия левого и правого ребер, что также порождает отношение порядка концевых вершин. В связи с этим пронумеруем концевые вершины по возрастанию и обозначим их через $\beta'_1, \dots, \beta'_m$. Таким образом, вершине β'_1 будет соответствовать слово (Л, Л, ..., Л), а вершине β'_m – (П, П, ..., П). Нагрузка ребер предикатами определяется следующим образом. Возьмем произвольную внутреннюю вершину β . Пусть β'_j – концевая вершина с наибольшим номером в ветви, исходящей из вершины с входящим левым ребром из β . Исходящему из β левому ребру сопоставим предикат f_{0,s_j} , а правому – \bar{f}_{0,s_j} . Полная нагрузка ребер определяется после применения данной операции всех внутренних вершин дерева.

В графе U^1 листья обозначены как $\alpha_1, \dots, \alpha_k$, и им приписаны записи y_1, \dots, y_k соответственно. Построим еще k листьев $\alpha'_1, \dots, \alpha'_k$ и припишем каждому α'_i запись y_i . Теперь для каждого $i \in \{2, k\}$ из листа α'_i проведем ребро в α'_{i-1} и припишем ребру $(\alpha'_i, \alpha'_{i-1})$ предикат f_{y_{i-1}, y_i} . Далее если $l_i \neq 0$, то для каждого $i \in \{1, m\}$ из β'_i выпустим ребро в α'_{l_i} и припишем ему предикат $f_{y_{l_i}, y_i}$. При этом в случае $l_i < k$ из β'_i построим еще одно ребро к листу α'_{l_i+1} и припишем ему предикат $f_{y_{l_i+1}, y_{l_i+1}}$. После этого удалим все предикатные ребра, не являющиеся частью цепи из корня в каком-либо из листьев, вместе с их началами.

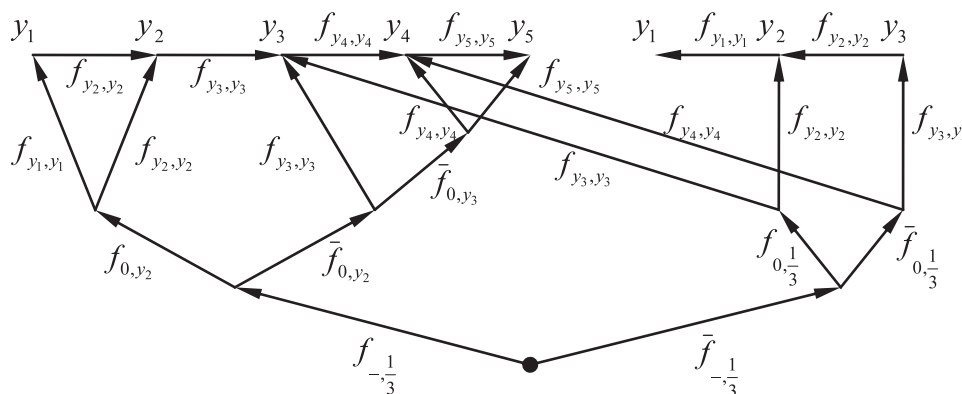


Рис. 7. Пример ИГ сверхлогарифмического поиска

Полученный граф обозначим через U^2 . На рис. 7 представлен пример ИГ U^2 для библиотеки $V = \{y_i = \frac{2i-1}{2k}, k = 1, 5\}$.

Дадим неформальное описание алгоритма сверхлогарифмического поиска. Пусть нам дан запрос $x = (u, v) \in X_{\text{int}}$. Для начала вычислим длину интервала x . Если $v - u < 1/([\log_2 k] + 1)$, то ответ будем искать по методу, описанному в алгоритме логарифмического поиска (левое ребро, исходящее из корня на рис. 7), в противном случае за время $O(\log_2 \log_2 k)$ найдем в множестве S самую левую точку s_i , попадающую в интервал x , затем по ссылке l_i перейдем к ближайшей слева к s_i записи библиотеки V и проверим, попадет ли она в x . Если попадет, то справа налево проверяем на попадание в x все следующие записи. Затем слева направо начиная с записи по ссылке $l_i + 1$ проверяем на попадание в x все оставшиеся записи.

Таким образом, в первом случае помимо времени на перечисление ответа мы тратим время $O(\log_2 k)$, а во втором – $O(\log_2 \log_2 k)$. Для данного ИГ была доказана верхняя оценка сложности

ти [3]. Пусть плотность распределения вероятностей $p(u, v)$, определяющая вероятность P на алгебре σ , ограничена сверху некоторой константой c . Пусть $I = \langle X_{\text{int}}, V, \rho_{\text{int}} \rangle$ – произвольная ЗИП типа S_{int} . Тогда справедлива следующая верхняя оценка сложности алгоритма включающего поиска над базовым множеством \tilde{F}_4 :

$$T(I, \tilde{F}_4) \leq \sum_{y \in V} P(O(y, \rho_{\text{int}})) + 2 \log_2 \log_2 |V| [6 + 2c].$$

Мгновенное решение. Пусть

$$G_1 = \{g_{-,m}(u, v) = \max(1, \lfloor u \cdot m \rfloor)\},$$

$$G_2 = \left\{ g_{\leq, a}(u, v) = \begin{cases} 1, & \text{если } u \leq a; \\ 2, & \text{если } u > a; \end{cases} \quad a \in [0, 1] \right\},$$

$$G_3 = \left\{ g_{-,m}(u, v) = \begin{cases} 1, & \text{если } 0 \leq v - u < \frac{1}{m}; \\ 2 & \text{в противном случае} \end{cases} \right\},$$

$$\tilde{F}_5 = \langle F_1, G_1 \cup G_2 \cup G_3 \rangle,$$

где F_1 – базовое множество характеристических функций.

Справедливо следующее утверждение [4]. Пусть ЗИП $I = \langle X_{\text{int}}, V, \rho_{\text{int}} \rangle$ – одномерная задача интервального поиска над базовым множеством \tilde{F}_5 , причем $|V| = k$. Тогда если плотность распределения вероятностей $p(u, v)$, определяющая вероятность P на алгебре σ , ограничена сверху некоторой константой c , то существует граф U_0 объема

$$Q(U) \leq 4k - 1 + 6c \lceil \log_2 k \rceil,$$

для которого

$$\sum_{y \in V} P(O(y, \rho_{\text{int}})) \leq T(I, \tilde{F}_5) \leq \sum_{y \in V} P(O(y, \rho_{\text{int}})) + 5.$$

Построим ИГ U_0 , на котором достигается данная оценка сложности. Возьмем точку β_0 и сделаем ее корнем графа. Выпустим из корня 2 ребра – левое и правое. Обозначим через β_1 конец левого ребра, а через β_2 – конец правого. Пусть $m = 2c \lceil \log_2 k \rceil$. Зададим для корня переключатель $g_{-,m}(u, v)$ из G_3 . Сопоставим левому и правому ребрам значения «1» и «2» соответственно.

Из вершины β_1 выпустим дерево бинарного поиска D для библиотеки V , как описано далее. Возьмем бинарное сбалансированное дерево с k листьями высоты $\lceil \log_2 k \rceil$ с ребрами, направленными от корня к листьям. Если k нечетно, то в этом дереве существует внутренняя вершина, из которой исходит одно ребро, ведущее в лист, а другое – во внутреннюю. Из ребра, ведущего из данной вершины в лист, выпустим одно ребро, его конец объявим листом, а текущую вершину – внутренней. В полученном дереве сопоставим листьям слева направо в порядке возрастания элементов из V .

Объявим все внутренние вершины дерева D с ребрами, ведущими к другим внутренним вер-

шинам, вершинами переключения и припишем 1 левому исходящему из них ребру и 2 – правому, а самим вершинам припишем переключатель $g_{\leq, a}(u, v)$, где a – максимальная запись, достижимая из вершины, в которую ведет левое ребро из данной вершины.

Все входящие в листья ребра дерева D объявим предикатными и каждому такому ребру, ведущему в лист с записью a , припишем предикат $\chi_{a, \rho_{\text{int}}} \in F_1$. Для дальнейшего удобства назовем множество переключательных ребер дерева D левой переключательной частью, а множество предикатных ребер – левой предикатной частью.

Листья, которым соответствует запись y_i , обозначим через α_i ($i = \overline{1, k}$). От каждого листа α_i ($i = \overline{1, k-1}$) выпустим ребро, ведущее в α_{i+1} , и припишем ему предикат $\chi_{y_{i+1}, \rho_{\text{int}}} \in F_1$. Множество этих ребер назовем правосторонней концевой цепью.

Теперь из β_2 выпустим $m-1$ ребер и пронумеруем их числами от 1 до $m-1$, объявим β_2 точкой переключения с переключателем $g_{-,m}(u, v) \in G_1$. Обозначим β'_i – конец ребра, исходящего из β_2 и имеющего номер i . Введем множество номеров $S = \{s_1, \dots, s_{m-1}\}$, такое, что s_i – номер записи из V такой, что y_{s_i} – запись, ближайшая слева к точке $\frac{i}{m}$ ($i = \overline{1, m-1}$), а если такой записи не существует, то $s_i = 0$.

Возьмем k новых точек, объявим их листьями и обозначим α'_i ($i = \overline{1, k}$). Каждому листу α'_i ($i = \overline{1, k}$) припишем запись y_i . Таким образом, каждая запись y_i приписана двум листьям – α_i и α'_i . Из каждого листа α'_i ($i = \overline{2, k}$) выпустим ребро, ведущее в лист α'_{i-1} , и припишем ему предикат $\chi_{y_{i-1}, \rho_{\text{int}}} \in F_1$. Это множество назовем левосторонней концевой цепью [1].

Теперь для каждой вершины β'_i , если $s_i \neq 0$, то из β'_i выпустим ребро, ведущее в α'_{s_i} и припи-

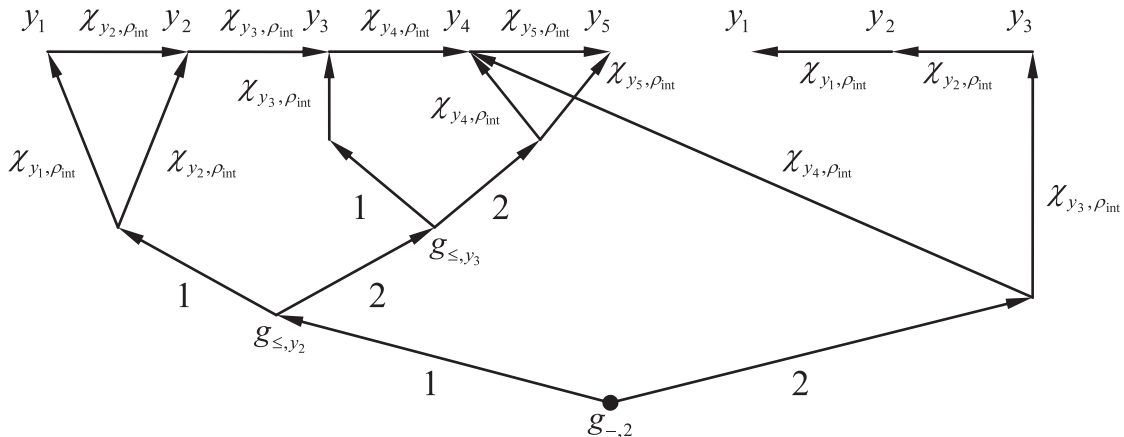


Рис. 8. Пример ИГ U^0

шем ему предикат $\chi_{y_{s_i}, \rho_{m_i}} \in F_1$; если $s_i < k$, то из β'_i выпустим ребро, ведущее в α'_{s_i+1} , и припишем ему предикат $\chi_{y_{s_i+1}, \rho_{m_i}} \in F_1$. Правой предикатной частью назовем множество ребер, исходящих из вершин β'_i ($i = 1, m-1$).

Граф, состоящий из корня β_0 , левой переключательной части, левой предикатной части, правосторонней концевой цепи, левосторонней концевой цепи и правой предикатной части, и будет искомым ИГ U_0 . Заметим, что из полученного графа можно удалить все предикатные ребра, не являющиеся частью цепи из корня в один из листьев, вместе с вершинами, являющимися их началами, а также все переключательные вершины, из которых исходит одно ребро, вместе с данным ребром, заменив рассматриваемую переключательную вершину концом исходящего из него ребра.

На рис. 8 представлен пример ИГ U^0 для библиотеки $V = \{y_i = \frac{2i-1}{2k}, k = 1, 5\}$ и $c = 0,5$. Заметим, что в этом случае $m = 2$, а значит, переключатель $g_{.m}$ может принимать только значение «1». Таким образом, мы убираем из рассматриваемого графа этот переключатель вместе с исходящим из него ребром.

Дадим неформальное описание алгоритма, приведенного выше.

Пусть нам дано множество $V = \{y_1, \dots, y_k\}$, в котором мы должны произвести поиск. Упорядочим это множество по возрастанию. В случае если оценка сверху с функции плотности распределения вероятностей появления запросов уже определена, то m возьмем как $m = 2c \lceil \log_2 k \rceil$, если же константа c неизвестна, то в качестве нее можно взять любое число, например, $c = 2$. Затем для V построим множество номеров $S = \{s_1, \dots, s_{m-1}\}$, описанное выше. Теперь поиск по произвольно взятому запросу производится следующим образом.

Сначала определяется длина запроса x . Если $x < \frac{1}{m}$, то во множестве V с помощью бинарного поиска находится запись, ближайшая к точке u справа. Далее от найденной записи просматрива-

ются все записи из V слева направо и сравниваются с точкой v , пока очередная запись меньше v . Таким образом, производится $\lceil \log_2 k \rceil$ действий, не считая перечисления ответа.

Если же $v - u \geq \frac{1}{m}$, то, применяя функцию $g_{.m}$, получаем номер j точки $\frac{j}{m}$, входящей в интервал $[u, v]$. Далее просматриваем и сравниваем с u все записи из V справа налево начиная с записи с номером s_j . Как только очередная запись окажется меньше u , с записи с номером $s_j + 1$ просматриваются и сравниваются с правым концом запроса (точкой v) слева направо записи из V до тех пор, пока очередная запись не станет больше v . В данном случае, кроме перечисления ответа, мы производим 4 лишних действия (сравнение $v - u$ с $\frac{1}{m}$, вычисление переключателя $g_{.m}$, движение по правосторонней и левосторонней концевой цепи).

Заметим, что параметр m определяется так, что средняя сложность первого случая не больше 1 при известной оценке сверху плотности распределения вероятностей запросов и не превышает некоторой константы, если эта оценка неизвестна.

Литература

1. Гасанов Э.Э., Кудрявцев В.Б. Теория хранения и поиска информации. М.: Физматлит, 2002. – 288 с.
2. Тиори Т., Фрай Дж. Проектирование структур баз данных. М.: Мир, 1985. Кн. 1 – 287 с.; Кн. 2 – 320 с.
3. Гасанов Э.Э. О сложности поиска в базах данных // Искусственный интеллект. Межвузовский сборник трудов. Саратов: Изд. СУ, 1993. – С. 41-56.
4. Гасанов Э.Э., Ерохин А. А. О быстром в среднем решении n -мерной задачи интервального поиска // Методы и системы технической диагностики. Тезисы X МК по проблемам теоретической кибернетики. Саратов: Изд. СУ, 1993. – С. 48-49.

FIND THE OBJECT THE INFORMATION SYSTEM BY SOLVING THE PROBLEM OF INTERVAL SEARCH

Ljalin V.E., Maltzev S.A., Taranucha V.P., Shishov D.R.

The article is devoted to optimizing algorithms of interval search on different base set using information-graph models designed to reduce time-consuming when you restore a backup of information.

Keywords: interval search, graph model, the search for information.

Лялин Вадим Евгеньевич, д.т.н., д.э.н., профессор, Заслуженный изобретатель РФ, заведующий Кафедрой математических технологий в нефтегазовом машиностроении (МТНМ) Ижевского государственного технического университета им. М.Т. Калашникова (ИжГТУ). Тел. 8-912-767-30-00. Email: velyalin@mail.ru

Мальцев Сергей Андреевич, к.т.н., ведущий инженер ЗАО «Группа Компаний Старт» (г. Москва). Тел. 8-929-682-90-19. E-mail: rshavas@mail.ru

Тарануха Владимир Прокофьевич, к.т.н., доцент, заведующий Кафедрой конструирования радиоэлектронной аппаратуры ИжГТУ. Тел. 8-341-250-40-22. Email: kra_dept@istu.ru

Шишов Дмитрий Родионович, аспирант Кафедры МТНМ ИжГТУ. Тел. 8-912-447-51-40. E-mail: d1men27@mail.ru

УДК 621.391.266

ФИЗИЧЕСКИЕ НОСИТЕЛИ КОМПЛЕКСНОЗНАЧНЫХ КОДОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Григорьевых Е.А., Хафизов Р.Г.

Представлено два способа формирования физического носителя комплекснозначных (КЗН) кодовых последовательностей: амплитудно-фазовая модуляция и в виде полигармонического сигнала. Исследована помехоустойчивость физических носителей.

Ключевые слова: КЗН-сигналы, равномерный энергетический спектр, передача данных.

Введение

В современной технике многоканальной связи при передаче дискретных сообщений широкое применение находят системы с ортогональными индивидуальными сигналами, перекрывающимися во времени и по частоте, разделяемые в месте приема посредством согласованных пассивных фильтров или эквивалентных им активных корреляционных схем [1]. Они же и обеспечивают оптимальный прием на фоне шума. При этом актуальной является проблема устранения корреляционного шума. Если в процессе обработки сигналов обеспечить их согласование с фильтром, то требование равномерности энергетического спектра остается единственным для полного подавления корреляционных шумов при минимальном уровне флуктуационных шумов [2]. В связи с этим возникает проблема формирования кодовых последовательностей с равномерным энергетическим спектром.

В работе [2] рассматриваются вопросы синтеза и анализа КЗН кодирующих последовательностей и методов их обработки. В частности, показано, что сигнал, обладающий равномерным энергетическим спектром и дельтовидной циклической автокорреляционной функцией, в общем случае является КЗН.

Кодовые последовательности с равномерным энергетическим спектром являются лишь математической моделью реального сигнала и не могут непосредственно использоваться в системах передачи и извлечения информации. Реальным физическим носителем сигнала в пространстве может быть гармоническое колебание, какие-либо параметры которого изменяются по закону формирования элементов кодовой последовательности с равномерным энергетическим спектром. Способ кодирования обуславливает алгоритм формирования и обработки сигнала и выделения из него элементарных векторов кодирующей последовательности, аппаратную реализацию системы и, в конечном счете, ее эффективность. От выбора физического носителя зависит, насколько полно будут реализованы полезные свойства КЗН кодовой последовательности

КЗН кодовые последовательности

Сущность проводимых над сигналами операций в месте приема заключается в определении количественной меры его близости с эталонными сигналами. Мера близости между двумя сигналами определяется величиной их скалярного произведения. Поэтому операция скалярного произведения является базовой при решении задач обработки сигналов.

Скалярное произведение векторов z_1 и z_2 в комплексном линейном пространстве C задается выражением

$$\eta_C = (z_1, z_2) = \eta_{1,C} + i\eta_{2,C} = \eta_E + i\eta_{2,C}, \quad (1)$$

где η_E – скалярное произведение векторов в действительном линейном пространстве E . Скалярное произведение в пространстве C включает