Автоматика и телемеханика, № 10, 2022

© 2022 г. **Н.А. ДРАГУНОВ (nikitadragunovjob@gmail.com),
Е.В. ДЮКОВА, д-р физ.-мат. наук (edjukova@mail.ru)
(Федеральный исследовательский центр
«Информатика и управление»
Российской академии наук, Москва)**

ОБ ОДНОМ ПОДХОДЕ К РАСШИФРОВКЕ МОНОТОННОЙ ЛОГИЧЕСКОЙ ФУНКЦИИ

Рассматривается задача расшифровки двузначной монотонной функции f, определенной на k-значном n-мерном кубе. Традиционным подходом к решению данной задачи является построение оптимального по Шеннону алгоритма. Оптимальный по Шеннону алгоритм расшифровки имеет минимальную сложность в «худшем случае» (эффективен для наиболее трудного варианта задачи). Авторами предложен и исследован подход к задаче расшифровки, основанный на применении асимптотически оптимального алгоритма дуализации над произведением k-значных цепей. Асимптотически оптимальная расшифровка функции f нацелена на «типичный случай» (на типичный вариант задачи). Экспериментально выявлены условия применимости традиционного и нового подходов.

Ключевые слова: верхний ноль монотонной логической функции, нижняя единица монотонной логической функции, оптимальный по Шеннону алгоритм расшифровки, асимптотически оптимальный алгоритм расшифровки, максимальный частый элемент, минимальный нечастый элемент, дуализация над произведением k-значных цепей.

DOI: 10.31857/S0005231022100129, **EDN:** ALIJYE

1. Введение

Расшифровка двузначной монотонной функции, определенной на k-значном n-мерном кубе, — одна из важнейших задач дискретной математики. Задача формулируется следующим образом.

Положим

$$E_k^n = \{(\alpha_1, \ldots, \alpha_n) \,|\, \alpha_i \in \{0, 1, \ldots, k-1\} \text{ при } i = 1, 2, \ldots, n,\ k \geq 2\}.$$

Множество E_k^n называется k-значным n-мерным кубом. На E_k^n устанавливается частичный порядок, согласно которому элемент $\beta = (\beta_1, \ldots, \beta_n)$ из E_k^n следует за элементом $\alpha = (\alpha_1, \ldots, \alpha_n)$ из E_k^n (α предшествует β), если $\beta_i \geq \alpha_i$ при $i = 1, 2, \ldots, n$. Для обозначения того, что $\beta \in E_k^n$ следует за $\alpha \in E_k^n$, используется запись $\alpha \preccurlyeq \beta$ или $\beta \succcurlyeq \alpha$.

Функция f, определенная на E_k^n и принимающая два значения 0 и 1, называется монотонной, если для любых двух элементов α и β из E_k^n таких, что $\alpha \preccurlyeq \beta$, выполнено $f(\beta) \geq f(\alpha)$. Функция f задается при помощи некоторого оператора B, который для любого $x \in E_k^n$ выдает значение $f(x)$. Если

$f(x) = 0$, то элемент x называется нулем функции f, если же $f(x) = 1$, то элемент x называется единицей функции f. Требуется путем «минимального» числа обращений к оператору B найти множество всех нулей функции f и множество всех ее единиц.

Традиционный подход к решению задачи расшифровки основан на построении оптимального по Шеннону алгоритма (предложен В.К. Коробковым в 1965 г. в [1]). Согласно данному подходу, сложность алгоритма расшифровки следует оценивать числом обращений к оператору B в худшем случае. Это означает следующее. Пусть V — множество всех двузначных монотонных функций, определенных на E_k^n. Пусть A — некоторый алгоритм, выполняющий расшифровку функций из V. Под сложностью алгоритма A понимается максимум числа обращений к оператору B, где максимум берется по всем функциям из V. Алгоритм A называется оптимальным по Шеннону на множестве V, если его сложность минимальна среди всех алгоритмов, выполняющих расшифровку функций из V.

Оптимальный по Шеннону алгоритм расшифровки монотонной булевой функции построен Ж. Анселем в 1968 г. [2]. В 1976 г. В.Б. Алексеевым предложен алгоритм расшифровки функции из V, который является оптимальным в случае $k = 2$ и близок по сложности к оптимальному в случае $k > 2$ [3].

Введем понятия верхнего нуля и нижней единицы функции f, $f \in V$. Эти понятия являются центральными для рассматриваемой задачи расшифровки. Ноль функции f называется верхним, если он не предшествует никакому другому нулю этой функции. Единица функции f называется нижней, если она не следует ни за какой другой единицей этой функции. Очевидно, что для расшифровки f достаточно найти все ее верхние нули и все ее нижние единицы.

Пусть D — произвольная совокупность элементов из E_k^n, $x \in E_k^n$, $s \in [0,1]$. Элемент x называется s-частым, если доля элементов в D, следующих за x, не меньше s, иначе x — s-нечастый элемент. Элемент x называется максимальным s-частым, если x — s-частый элемент и любой другой следующий за ним элемент является s-нечастым. Элемент x называется минимальным s-нечастым, если x — s-нечастый элемент и любой другой предшествующий ему элемент является s-частым.

Пусть $X_{\max}$ и $Y_{\min}$ — множества, состоящие соответственно из всех максимальных s-частых и минимальных s-нечастых элементов множества E_k^n. На множестве E_k^n определим монотонную функцию $f_{D,s}$, которая принимает значения 0 и 1 соответственно на s-частых элементах и s-нечастых элементах этого множества. Фактически $f_{D,s}$ задается при помощи оператора B_D, который для произвольного x из E_k^n выдает значение $f_{D,s}(x)$ путем вычисления частоты встречаемости x в D. Таким образом, для расшифровки функции $f_{D,s}$ могут применяться методы поиска множеств $X_{\max}$ и $Y_{\min}$, и наоборот, для поиска $X_{\max}$ и $Y_{\min}$ применимы методы расшифровки $f_{D,s}$.

Следует отметить, что основным приложением методов поиска частых и нечастых элементов в данных, в том числе частично упорядоченных, являются вопросы построения ассоциативных правил, впервые возникшие в связи с задачей анализа потребительской корзины [4]. В машинном обучении логический анализ признаковых описаний прецедентов фактически основан на поиске в этих описаниях часто и нечасто встречающихся фрагментов [5].

В [6] анонсирована идея последовательно-совместного перечисления $X_{\max}$ и $Y_{\min}$, основанная на решении задачи дуализации над произведением k-значных цепей, и приведены результаты тестирования последовательно-совместного поиска $X_{\max}$ и $Y_{\min}$ на случайных модельных данных, показавшие его эффективность в случае, когда мощности $X_{\max}$ и $Y_{\min}$ примерно равны.

В настоящей работе проведено экспериментальное сравнение двух методов расшифровки функции $f_{D,s}$. Первый метод — это упомянутый выше алгоритм расшифровки В.Б. Алексеева. Второй метод основан на предложенной в [6] идее последовательно-совместного перечисления $X_{\max}$ и $Y_{\min}$ с применением асимптотически оптимального алгоритма дуализации над произведением k-значных цепей RUNC-M+ [7]. Задача дуализации относится к числу труднорешаемых дискретных задач и асимптотически оптимальные алгоритмы дуализации лидируют по скорости счета. На случайных модельных данных для каждого тестируемого метода расшифровки оценивалось время работы и число обращений к оператору B_D. Показано, что асимптотически оптимальная расшифровка функции $f_{D,s}$, нацеленная на типичный вариант задачи, в определенных случаях имеет лучшие результаты по сравнению с оптимальной по Шеннону расшифровкой, ориентирующейся на самый сложный вариант задачи.

2. Традиционный подход к расшифровке монотонной логической функции. Алгоритм Алексеева

В настоящем разделе рассматривается алгоритм A_{opt} расшифровки функции f из V, описанный в [3]. Как уже было отмечено во введении, этот алгоритм является оптимальным по Шеннону в случае $k = 2$ и близок по сложности к оптимальному в случае $k > 2$.

Алгоритм A_{opt} работает в два этапа. На первом этапе куб E_k^n разбивается на непересекающиеся подмножества, каждое из которых является цепью. На втором этапе выполняется расшифровка f на каждой построенной цепи с помощью хорошо известного алгоритма двоичного поиска.

Пусть $i \in \{2, \dots, n\}$, $r \in \{0, 1, \dots, k-1\}$. Положим

$$S_r^i = \{(\alpha_1, \dots, \alpha_{i-1}, r) \mid (\alpha_1, \dots, \alpha_{i-1}) \in E_k^{i-1}\}.$$

Процесс разбиения куба E_k^n на непересекающиеся цепи происходит путем последовательного построения разбиений на непересекающиеся цепи кубов меньшей размерности.

На первом шаге рассматривается куб E_k^1, представляющий собой цепь согласно установленному частичному порядку.

Пусть на шаге $i-1, 2 \leq i \leq n$, куб E_k^{i-1} разбит на непересекающиеся цепи. Тем самым, очевидным образом на непересекающиеся цепи разбито каждое из множеств S_r^i, $r \in \{0, 1, \dots, k-1\}$. Далее построенные разбиения множеств $S_0^i, \dots, S_{k-1}^i$ изменяются. Сначала в S_0^i добавляются все максимальные элементы цепей из построенных разбиений для множеств $S_1^i, \dots, S_{k-1}^i$, при этом все добавленные к S_0^i элементы удаляются из множеств $S_1^i, \dots, S_{k-1}^i$. Затем аналогичная процедура проводится для измененной последовательности $S_1^i, \dots, S_{k-1}^i$ и т.д. В результате, учитывая, что $E_k^i = \bigcup_{r=0}^{k-1} S_r^i$, получается требуемое разбиение для куба E_k^i.

Построенные на первом этапе работы алгоритма цепи просматриваются в порядке не убывания их мощности. Пусть C_i — очередная цепь. Если для некоторого элемента $x \in C_i$ известно, что $x \preccurlyeq y$, где y — ноль, принадлежащий ранее просмотренной цепи C_j $(j < i)$, то x — ноль цепи C_i. Аналогично, если для некоторого элемента $x \in C_i$ известно, что $y \preccurlyeq x$, где y — единица, принадлежащая ранее просмотренной цепи C_j $(j < i)$, то x — единица цепи C_i. Таким образом, цепь C_i делится на три отрезка: сначала следуют найденные нули, затем следуют элементы, на которых значение функции f неизвестно, после чего следуют найденные единицы. Для расшифровки цепи C_i на втором отрезке запускается алгоритм двоичного поиска. При этом для определения значения функции f происходит обращение к оператору B.

Пусть $t_A(f)$ — общее число обращений к оператору B алгоритма A, выполняющего расшифровку функции f из V. Сложностью алгоритма A по Шеннону (сложностью в худшем случае) называется величина $\max[t_A(f)]$, где максимум берется по всем функциям из V. Пусть A_0 — любой алгоритм расшифровки функций из V с наименьшей сложностью. Тогда $t_{A_{\text{opt}}}(f)/t_{A_0}(f) \le 0{,}5\,(\log_2(k) + 1)$.

3. Новый подход к расшифровке монотонной логической функции. Асимптотически оптимальная расшифровка

В данном разделе описывается подход к задаче расшифровки монотонной логической функции, базирующийся на применении алгоритмов дуализации над произведением k-значных цепей.

3.1. Дуализация над произведением k-значных цепей

Задача дуализации над произведением k-значных цепей относится к числу трудно решаемых перечислительных задач дискретной математики и ставится следующим образом.

Представим множество E_k^n в виде декартова произведения n цепей мощности k, положив $E_k^n = E_1 \times E_2 \times \ldots \times E_n$, где каждое $E_i = \{0, 1, \dots, k-1\}$. Считается, что элемент $\beta = (\beta_1, \dots, \beta_n)$ из E_k^n следует за элементом $\alpha = (\alpha_1, \dots, \alpha_n)$ из E_k^n (α предшествует β), если $\beta_i \ge \alpha_i$ при $i = 1, 2, \dots, n$.

Элемент $x \in E$, $E \subset E_k^n$, называется минимальным элементом множества E, если не существует другого элемента множества E, предшествующего x. Элемент $x \in E$, $E \subset E_k^n$ называется максимальным элементом множества E, если не существует другого элемента множества E, следующего за x.

Пусть $E \subset E_k^n$. Введем обозначения: E^+ — множество всех элементов из E_k^n, каждый из которых следует хотя бы за одним элементом из E; E^- —

множество всех элементов из E_k^n, каждый из которых предшествует хотя бы одному элементу из E; $Q(E)$ — множество, состоящее из минимальных элементов множества $E_k^n \backslash E^-$ (здесь и далее обозначение $A \backslash B$ используется для разности множеств A и B); $G(E)$ — множество, состоящее из всех максимальных элементов множества $E_k^n \backslash E^+$.

Каждая из задач построения множества $Q(E)$ или $G(E)$ называется задачей дуализации над произведением k-значных цепей.

Если $k = 2$, то к построению $Q(E)$ сводится задача поиска нижних единиц монотонной булевой функции, заданной множеством нулей E, называемая задачей монотонной дуализации. В матричной формулировке монотонная дуализация — это задача построения неприводимых покрытий булевой матрицы, строками которой являются элементы из E. Эквивалентной задачей является перечисление минимальных вершинных покрытий гиперграфа.

Если $k \geq 2$ и множество E состоит из попарно несравнимых элементов, то к построению $Q(E)$ сводится поиск нижних единиц двузначной монотонной функции k-значной логики, заданной множеством верхних нулей E. Аналогично к построению $G(E)$ сводится поиск верхних нулей двузначной монотонной функции k-значной логики, заданной множеством нижних единиц E.

В теории алгоритмической сложности дискретных задач эффективность алгоритмов для перечислительных задач принято оценивать временем выполнения одного шага, т.е. временем нахождения очередного нового решения. Наиболее эффективными считаются алгоритмы с полиномиальными временны́ми оценками. Такие алгоритмы имеют временны́е оценки вида $O(N)$, где N — полином от размера входа задачи, и называются алгоритмами с полиномиальными задержками. Причем временны́е оценки даются для самой сложной индивидуальной задачи. Полиномиальные алгоритмы удалось построить для немногих частных случаев монотонной дуализации [8]. Наилучший теоретический результат получен в [9]. Предложенный в [9] инкрементальный квазиполиномиальный алгоритм монотонной дуализации имеет временную оценку вида $N^{o(logN)}$, где N — полином не только от размера входа задачи, но и от числа решений, найденных на предыдущих шагах, т.е. N — полином от размера входа и выхода задачи.

В [10] предложен подход к построению асимптотически оптимальных алгоритмов монотонной дуализации. Эти алгоритмы имеют теоретическое обоснование эффективности и показывают хорошие результаты по скорости счета. Асимптотически оптимальный алгоритм отличается от алгоритма с полиномиальной задержкой тем, что имеет лишние полиномиальные шаги. Это шаги, на которых не строятся новые решения. Основное требование: для почти всех индивидуальных задач число лишних шагов алгоритма должно быть мало по сравнению с числом всех решений задачи. При этом проверка того, является ли шаг лишним, должна происходить за полиномиальное от размера входа время. Данный подход к задаче монотонной дуализации значительно позже был продемонстрирован в работе [11] на примере задачи построения минимальных вершинных покрытий гиперграфа.

3.2. Последовательно-совместное перечисление верхних нулей и нижних единиц монотонной логической функции

Алгоритм последовательно-совместного перечисления верхних нулей $X_{\rm up}$ и нижних единиц $Y_{\rm low}$ функции f, $f \in V$, заданной при помощи оператора B, основан на решении задачи дуализации над произведением k-значных цепей. Рассматриваемый алгоритм является синтезом последовательного и совместного подходов к поиску $X_{\rm up}$ и $Y_{\rm low}$, которые подробно описаны в разделе 3.3. Метод базируется на приведенных ниже утверждениях 1–3.

Утверждение 1. Если $X \subset X_{\rm up}$, то $Q(X)$ содержит хотя бы один ноль функции f.

Доказательство. Пусть $X \subset X_{\rm up}$, $x \in X_{\rm up} \backslash X$. Из того, что x не сравним ни с одним другим элементом множества $X_{\rm up}$, следует, что $x \in E_k^n \backslash X^-$. Таким образом, в $Q(X)$ существует элемент q такой, что $q \preccurlyeq x$ и $f\,(q) = 0$.

Утверждение 2. Если $X \subset X_{\rm up}$, а элемент $y \in Q(X)$ является единицей функции f, то y — нижняя единица функции f.

Доказательство. Пусть $y \notin Q(X_{\rm up}) = Y_{\rm low}$. Тогда, так как y — единица функции f, то в $E_k^n \backslash X_{\rm up}^-$ найдется нижняя единица z такая, что $z \neq y$, $z \preccurlyeq y$. Из $(E_k^n \backslash X_{\rm up}^-) \subseteq (E_k^n \backslash X^-)$, следует, что $z \in E_k^n \backslash X^-$, что противоречит условию $y \in Q(X)$.

Утверждение 3. Пусть $X \subseteq X_{\rm up}$, $Y \subseteq Y_{\rm low}$. Тогда $Q(X) = Y$ в том и только в том случае, если $X = X_{\rm up}$, $Y = Y_{\rm low}$.

Доказательство. Пусть $X \subset X_{\rm up}$. Из утверждения 1 следует, что $Q(X)$ содержит хотя бы один ноль f. Однако в множестве Y нет нулей функции f, следовательно $Q(X) \neq Y_{\rm low}$. Если же $X = X_{\rm up}$, то $Q(X) = Y_{\rm low}$. Таким образом, $Q(X) = Y$ тогда и только тогда, когда $X = X_{\rm up}$, $Y = Y_{\rm low}$.

Последовательно-совместный алгоритм работает следующим образом. Положим $X_0 = \varnothing$. Строится последовательность $X_1 \subset X_2 \subset \ldots \subset X_{\rm up}$.

Шаг 1. Рассматривается множество $X_1 = \{x\}$, где x — произвольный верхний ноль f.

Шаг $i{+}1$ ($i \geq 1$). Решается задача дуализации множества $X_i \backslash X_{i-1}$. Пусть множество Z есть результат дуализации $X_i \backslash X_{i-1}$. Согласно утверждению 1, множество Z содержит хотя бы один ноль функции f. Для каждого нуля из Z находится один содержащий его верхний ноль. Все найденные верхние нули, которые не содержатся в множестве X_i, добавляются к X_i, формируя в результате множество X_{i+1}. Если же все найденные верхние нули уже содержатся в X_i, то происходит дуализация множества X_i, в результате чего формируется множество $Q(X_i)$. Если в $Q(X_i)$ нет нулей, то, согласно утверждению 3, следует, что $Q(X_i) = Y_{\rm low}$, $X_i = X_{\rm up}$, и алгоритм завершает свою работу. Иначе для каждого нуля из $Q(X_i)$ находится один содержащий его верхний ноль, который пополняет множество X_i, формируя в результате множество X_{i+1}.

3.3. Последовательный и совместный поиск X_{up} и Y_{low}

Достаточно очевидным является поиск X_{up} и Y_{low} функции f, $f \in V$, заданной при помощи оператора B, путем последовательного построения этих множеств. Сначала строится множество X_{up}, например, алгоритмом Apriori [4, 12], модифицированным на случай произведения k-значных цепей. Затем используется свойство двойственности $Q\,(X_{\mathrm{up}}) = Y_{\mathrm{low}}$. Аналогично можно сначала строить Y_{low} модифицированным алгоритмом Apriori, а затем искать X_{up} путем дуализации множества Y_{low}.

В [13] предложена идея совместного перечисления множеств $X_{\max}$ и $Y_{\min}$, которая в применении к задаче построения X_{up} и Y_{low} заключается в следующем.

Выбирается произвольный элемент q из E_k^n. Если q — ноль функции f, то ищется верхний ноль, следующий за q. Если же q — единица функции f, то ищется нижняя единица, предшествующая q. Пусть на очередной итерации построены множества $X \subseteq X_{\mathrm{up}}$ и $Y \subseteq Y_{\mathrm{low}}$. Если $X \neq \varnothing$ и $Y = \varnothing$, то выбирается любой $x \in X$ и ищется элемент q, который не предшествует x. Если $X = \varnothing$ и $Y \neq \varnothing$, то выбирается любой $y \in Y$ и ищется элемент q, который не следует за y. Если же $X \neq \varnothing$ и $Y \neq \varnothing$, то ищется элемент q, который не предшествует x и не следует за y. Затем аналогичным образом в зависимости от значения элемента q находится верхний ноль или нижняя единица функции f.

Алгоритм, основанный на описанной выше идее совместном перечислении множеств X_{up} и Y_{low}, строит вложенные последовательности: $X_1 \subset X_2 \subset \ldots \subset X_{\mathrm{up}}$ и $Y_1 \subset Y_2 \subset \ldots \subset Y_{\mathrm{low}}$.

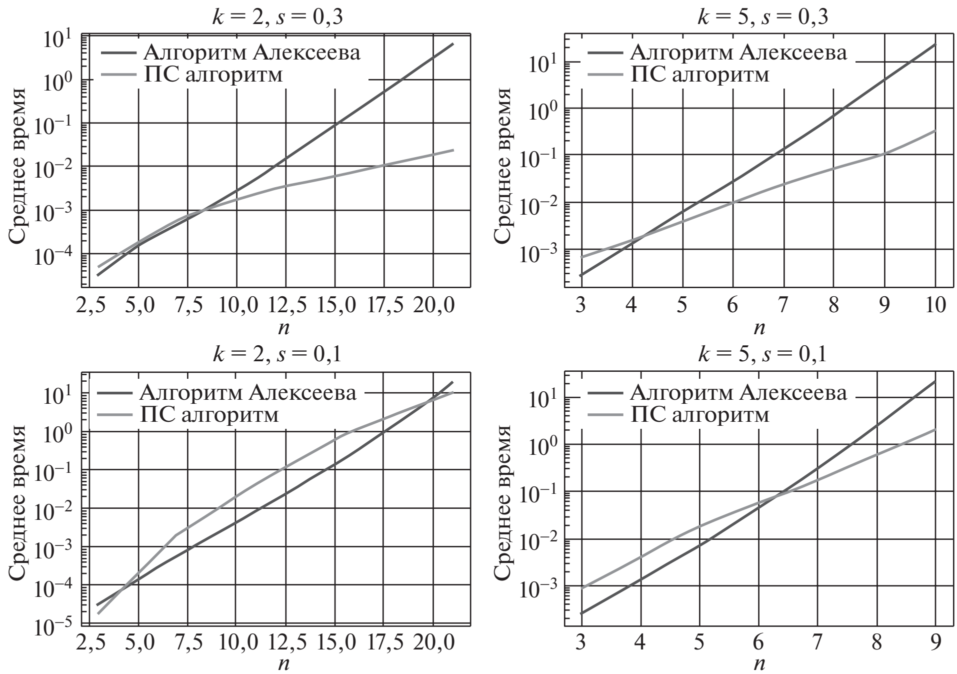
Шаг 1. $X_1 = \{x\}$, $Y_1 = \{y\}$, где x и y находятся модифицированным алгоритмом Apriori.

Шаг $i+1$ $(i \geq 1)$. Строится либо $Q(X_i)$, либо $G(Y_i)$. Пусть построено множество $Q(X_i)$. Если $Q(X_i)$ не содержит нулей функции f, то, согласно утверждению 3, оно совпадает с множеством Y_{low} (в этом случае $X_i = X_{\mathrm{up}}$ и алгоритм заканчивает работу). Согласно утверждению 2, каждая единица из множества $Q(X_i)$ является нижней единицей. Эти единицы пополняют множество Y_i, формируя в результате множество Y_{i+1}. Для каждого нуля из $Q(X_i)$ находится один содержащий его верхний ноль элемент путем последовательного увеличения текущего нуля согласно заданному порядку, который затем пополняет множество X_i, формируя в результате множество X_{i+1}.

В [6] приведены результаты тестирования последовательного, совместного и последовательно-совместного поиска $X_{\max}$ и $Y_{\min}$ на случайных модельных данных. Согласно этим результатам, последовательно-совместное перечисление наиболее эффективно, когда мощности множеств $X_{\max}$ и $Y_{\min}$ примерно равны, иначе более эффективным является последовательное перечисление. Наихудшие показатели у совместного перечисления множеств $X_{\max}$ и $Y_{\min}$.

4. Эксперименты

Проведено экспериментальное сравнение двух алгоритмов расшифровки функции $f_{D,s}$, описанных в разделах 3.2 и 3.3. Алгоритмы реализованы



Зависимость времени работы алгоритмов от n в секундах (при различных k и s).

на языке C++. При реализации последовательно-совместного перечисления верхних нулей и нижних единиц функции $f_{D,s}$ использовался асимптотически оптимальный алгоритм над произведением k-значных цепей RUNC-M+ [7].

Эксперименты проведены для случайных множеств D, $D \subset E_k^n$, с числом элементов $m = 100$. Данные выбирались из равномерного распределения. Результаты усреднялись по 20 независимым запускам.

Из представленных на рисунке графиков следует, что при $s = 0{,}3$ и $n > 5$ независимо от значения k последовательно-совместный алгоритм (ПС алгоритм) работает быстрее алгоритма Алексеева. Алгоритм Алексеева более эффективен при $s = 0{,}1$, $k = 2$, но при этом время его работы растет быстрее с ростом n. В случае $k = 5$, $n > 7$ последовательно-совместный алгоритм на порядки быстрее алгоритма Алексеева.

Таблица 1. Среднее число обращений к оператору B_D, $m = 100$

n; k; s	Алгоритм Алексеева	ПС алгоритм
5; 3; 0,1	**111**	529
5; 3; 0,3	**56**	343
5; 20; 0,3	25 484	**3039**
10; 3; 0,1	**2709**	10 172
10; 3; 0,3	**323**	3111
10; 5; 0,1	45 528	**31 791**
15; 3; 0,3	**1142**	13 940

Как видно из табл. 1, в случае небольших значений k, независимо от значения n, наилучший результат по числу обращений к оператору B_D показывает алгоритм Алексеева. Заметим, что при небольших значениях k сложность этого алгоритма почти оптимальна. При значениях $k \geq 5$ лучший результат по числу обращений к оператору B_D показывает последовательно-совместная расшифровка функции $f_{D,s}$.

5. Заключение

Исследованы актуальные вопросы уменьшения временных затрат, возникающие при логическом анализе частично упорядоченных данных. Разработан и исследован новый подход к задаче расшифровки двузначной монотонной функции, определенной на k-значном n-мерном кубе, основанный на последовательно-совместном перечислении верхних нулей и нижних единиц этой функции. Экспериментальное исследование предлагаемого подхода проведено с использованием авторской идеи последовательно-совместного поиска максимальных частых и минимальных нечастых элементов произведения k-значных цепей, базирующейся на решении задачи дуализации над произведением k-значных цепей. Показана целесообразность применения асимптотически оптимальных алгоритмов дуализации над произведением k-значных цепей для рассматриваемой задачи расшифровки монотонной логической функции.

СПИСОК ЛИТЕРАТУРЫ

1. *Коробков В.К.* О монотонных функциях алгебры логики // Сб. Проблемы кибернетики. Вып. 13. М.: Наука, 1965. С. 5–28.
2. *Ансель Ж.* О числе монотонных булевых функций n переменных // Кибернетич. сб. Нов. сер. Вып. 5. М.: Мир, 1968. С. 53–57.
3. *Алексеев В.Б.* О расшифровке некоторых классов монотонных многозначных функций // Журн. вычисл. матем. и матем. физики. 1976. Т. 16. № 1. С. 189–198.

   Alekseev V.B. Deciphering of some classes of monotonic many-valued functions // Zh. vychisl. Mat. mat. Fiz. 1976. V. 16. No. 1. P. 189–198.
4. *Agrawal R., Imielinski T., Swami A.* Mining association rules between sets of items in large databases // Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. 1993. P. 207–216.
5. *Журавлев Ю.И., Рязанов В.В., Сенько О.В.* Распознавание. Математические методы. Программная система. Практические применения. М.: ФАЗИС, 2006. Т. 176.
6. *Драгунов Н.А., Дюкова Е.В.* Поиск минимальных нечастых и максимальных частых наборов в частично упорядоченных данных // Математические методы распознавания образов: Тезисы докладов 9-й Всероссийской конференции с международным участием, 2019. С. 10–12.
7. *Дюкова Е.В., Масляков Г.О., Прокофьев П.А.* Дуализация над произведением цепей: асимптотические оценки числа решений // ДАН. 2018. Т. 483. № 2. С. 130–133.
8. *Johnson D.S., Yannakakis M., Papadimitriou C.H.* On general all maximal independent sets // Inform. Proc. Lett. 1988. V. 27. Iss. 3. P. 119–123.

9. *Fredman M.L., Khachiyan L.* On the complexity of dualization of monotone disjunctive normal forms // J. Algorithms. 1996. No. 21. P. 618–628.

10. *Дюкова Е.В.* Об асимптотически оптимальном алгоритме построения тупиковых тестов // ДАН СССР. 1977. Т. 233. № 4. С. 527–530.

    Djukova E.V. On an asymptotically optimal algorithm for constructing irredundant tests // DAN SSSR. 1977. V. 233. No. 4. P. 423–426.

11. *Murakami K., Uno T.* Efficient algorithms for dualizing large-scale hypergraphs // Discr. Appl. Math. 2014. V. 170. P. 83–94.

12. *Aggarwal C.* Frequent Pattern Mining. Springer International Publishing. Switzerland, 2014.

13. *Elbassioni K.* On Finding Minimal Infrequent Elements in Multidimensional Data Defined Over Partially Ordered Sets. 2014. arXiv:1411.2275.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 01.02.2022

После доработки 27.03.2022

Принята к публикации 29.06.2022