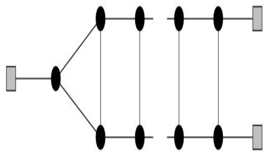
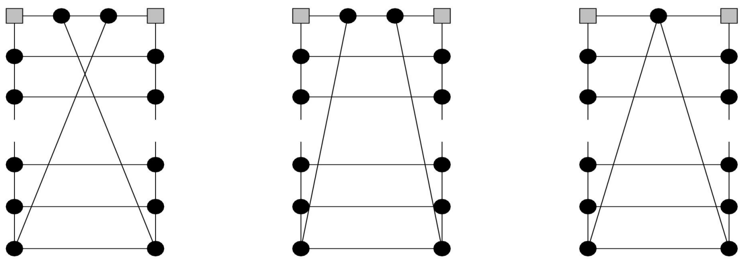


*а* *б*

Рис. 1. Минимальное вершинное расширение $P_n$

**Теорема 2.** Минимальные рёберные расширения цепей $P_n$ с вершинами двух типов, в которых белые вершины имеют степень 1, а чёрные — степень 2, содержат $m = 3k - 1$ рёбер при чётном $n = 2k$, и одно из минимальных рёберных расширений имеет вид, показанный на рис. 2,*а* (для чётного $k$) и рис. 2,*б* (для нечётного $k$). При нечётном $n = 2k + 1$ количество рёбер $m = 3k + 1$, и одно из минимальных рёберных расширений имеет вид, показанный на рис. 2,*в*.

При этом минимальные рёберные расширения, показанные на рис. 2,*а* и *б*, в то же время являются минимальными вершинными расширениями циклов с вершинами двух типов — одной белой вершиной и остальными чёрными — с количеством вершин на 2 меньшим, чем в расширениях.



*а* *б* *в*

Рис. 2. Минимальное рёберное расширение $P_n$

ЛИТЕРАТУРА

1. *Абросимов М. Б.* Минимальные $k$-расширения предполных графов // Изв. вузов. Математика. 2003. №6(493). С. 3–11.
2. *Heyes J. P.* A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. 25. No. 9. P. 875–884.
3. *Harary F. and Hayes J. P.* Edge fault tolerance in graphs // Networks. 1993. V. 23. P. 135–142.

УДК 519.7

# ДЕТЕРМИНИРОВАННАЯ РАЗМЕТКА ВЕРШИН ГРАФА БЛУЖДАЮЩИМ ПО НЕМУ АГЕНТОМ

И. С. Грунский, С. В. Сапунов

Рассматривается задача разметки вершин конечного простого связного неорграфа посредством блуждающего по нему агента. Разметка производится за один проход так, что в окрестности каждой вершины все вершины размечены разными метками.

Размеченные таким способом графы (*помеченные графы*) могут быть использованы в качестве топологических моделей операционной среды мобильных роботов [1].

Помеченным графом будем называть конечный простой связный неорграф с помеченными вершинами $G = (V, E, M, \mu)$, где $V$ — множество вершин; $E$ — множество ребер; $M$ — множество меток вершин; $\mu : V \to M$ — сюръективная функция разметки. Путем длины $k$ в графе $G$ будем называть последовательность его вершин $p = g_1, \ldots, g_k$, такую, что $(g_i, g_{i+1}) \in E$, $i = 1, \ldots, k-1$. Меткой $\mu(p)$ пути $p$, определяемой вершиной $g_1$, назовём слово $w = \mu(g_1) \ldots \mu(g_k)$. Обозначим через $L_g$ множество всех слов $w \in M^+$, определяемых вершиной $g$. Введём операцию $\star : V \times M^+ \to 2^V$: для любой $g \in V$ и любого $w \in M^+$ через $g \star w$ обозначим множество всех вершин $h \in V$, таких, что существует путь $p$ из $g$ в $h$ и $\mu(p) = w$. Множество всех вершин, находящихся от $g$ на расстоянии не больше $k$, назовём $k$-окрестностью $\Gamma_g^{(k)}$ вершины $g \in V$.

Мобильный агент $A$, находясь в вершине $g \in V$, наблюдает метки всех вершин из $\Gamma_g^{(k)}$, $k \geqslant 2$. Положим, что наименьшим наблюдением, необходимым для мобильности агента, является наблюдение $\Gamma_g^{(2)}$. Основываясь на анализе «увиденного», агент принимает решение о перемещении между смежными вершинами. Агент может заменить метку вершины другой меткой из $M$. Агент также может устанавливать в текущей вершине переносной маркер (*камень*) или подбирать его. Через $A2$ обозначим агента, наблюдающего только $\Gamma_g^{(2)}$; через $A3$ — агента, наблюдающего только $\Gamma_g^{(3)}$.

Функцию разметки $\mu$ назовём детерминированной (или Д-разметкой), если для любой $g \in V$ и любых $s, t \in \Gamma_g^{(2)}$ из $s \neq t$ следует $\mu(s) \neq \mu(t)$. Граф с Д-разметкой будем называть детерминированным, или Д-графом. Далее всюду используются Д-графы. В таких графах для любой вершины $g \in V$ и любого слова $w \in M^+$ выполняется $|g \star w| \leqslant 1$, где $|g \star w| = 1$, если $w \in L_g$, и $|g \star w| = 0$ иначе. Показано, что агент $A2$, «зная» слово $w \in L_g$, такое, что $g \star w = h$, может переместиться из $g$ в $h$. Таким образом, Д-разметка графа является достаточным условием для организации на нём навигации мобильных агентов. Показано далее, что для любых $g, h \in V$, $g \neq h$, $\mu(g) = \mu(h)$, и любого $w \in L_g \cap L_h$ расстояние между $g \star w$ и $h \star w$ не меньше 4, т. е. для того, чтобы выполнить Д-разметку вершин, необходимо наблюдение их 3-окрестностей.

Задача построения Д-разметки формулируется следующим образом. Агент устанавливается в произвольную вершину априори неизвестного ему графа, все вершины которого помечены одной и той же меткой. Агент должен осуществить Д-разметку вершин этого графа, причём если вершина уже помечена агентом, то её метка в дальнейшем не изменяется.

Построение минимальной Д-разметки на основе только локальной информации о вершинах графа представляется в общем случае невозможным. Поэтому в работе рассматривается построение «жадных» алгоритмов разметки. Показано, что с помощью Д-разметки агент может восстановить исследуемый граф с точностью до изоморфизма и что минимальная Д-разметка восстановленного графа может быть получена применением к его транзитивному замыканию известных в теории графов алгоритмов правильной раскраски [2].

Предложен метод Д-разметки вершин агентом $A3$, основанный на обходе графа в ширину. При этом для неявного именования вершин используются метки путей в них из начальной вершины по дереву обхода. Разработан соответствующий полиномиальный алгоритм.

Увеличение размера наблюдаемой агентом окрестности (т. е. объёма его входной информации) приводит к увеличению сложности робота, который реализует функции агента, поэтому целесообразно рассмотреть модель с ограничениями на размер наблюдения.

Предложена модификация алгоритма разметки для системы $(A2, p_1, p_2)$, состоящей из агента $A2$ и камней двух видов: одного камня $p_1$ для обозначения текущей вершины и нескольких камней $p_2$ для обозначения непомеченных вершин из её 2-окрестности. Число камней $p_2$ не превышает максимальной степени вершин графа.

**Теорема 1.** При решении задачи построения Д-разметки вершин помеченного графа агент $A3$ и система $(A2, p_1, p_2)$ эквивалентны по вычислительной мощности.

Для графов типа $n$-цепь, $n$-веер, $n$-угольник [2] разработана модификация алгоритма разметки агентом $A2$ без использования камней и без запоминания неявных имён вершин. Показано, что разметка $n$-цепи и $n$-угольника может быть выполнена конечным автоматом.

ЛИТЕРАТУРА

1. *Dudek G. and Jenkin M.* Computational Principles of Mobile Robotics. Cambridge: Cambridge University Press, 2000.
2. *Зыков А. А.* Основы теории графов. М.: Вузовская книга, 2004.

УДК 519.1

## ОБ ИНДЕКСАХ В ДИНАМИЧЕСКОЙ СИСТЕМЕ ДВОИЧНЫХ ВЕКТОРОВ, АССОЦИИРОВАННЫХ С ОРИЕНТАЦИЯМИ ЦИКЛОВ

А. В. Жаркова

Под *конечной динамической системой* понимается пара $(S, \delta)$, где $S$ — конечное непустое множество, элементы которого называются *состояниями системы*, $\delta : S \to S$ — отображение множества состояний в себя, называемое *эволюционной функцией системы*. Каждой конечной динамической системе сопоставляется *карта* — ориентированный граф с множеством вершин $S$ и дугами, проведенными из каждой вершины $s \in S$ в вершину $\delta(s)$. Компоненты связности графа, задающего динамическую систему, называются её *бассейнами*. Каждый бассейн представляет собой контур с входящими в него деревьями. Контуры называются предельными циклами или *аттракторами*.

Основными проблемами теории конечных динамических систем являются задачи отыскания эволюционных параметров без проведения динамики. К их числу относится индекс состояния — расстояние до аттрактора того бассейна, которому принадлежит состояние. Программа [1] позволяет вычислять различные параметры динамических систем двоичных векторов, ассоциированных с некоторыми типами графов.

В данной работе предлагается алгоритм для подсчёта индексов состояний в динамической системе двоичных векторов, порожденных такими графами, как циклы. Определяется также максимальный из индексов системы заданной размерности.

На множестве $B = \bigcup\limits_{n=3}^{\infty} B^n$, где через $B^n$, $n > 2$, обозначается множество всех двоичных векторов длины $n$, рассмотрим динамическую систему $(B, \theta)$. Пусть состоянием динамической системы в данный момент времени является вектор $v \in B$. Тогда в следующий момент времени она окажется в состоянии $\theta(v)$, полученном путем одновременного применения следующих правил: I) если первой компонентой в $v$ является 0 и последней компонентой — 1, то первой компонентой в $\theta(v)$ будет 1, а последней — 0;