S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 16, стр. 144–144 (2021) УДК 519.172.2
DOI 10.33048/semi.2021.16.xxx MSC 68R10

СВОЙСТВА ЦИКЛОВ DFS-БАЗИСА КАРТЫ ГРАФА. ВЫДЕЛЕНИЕ ЦИКЛОВ ЯЧЕЕК КАРТЫ

B. N. ИВАНОВ

Abstract. A constructive method for forming cycles of map cells of a simple planar graph. The geometry of the *DFS-basis* cycles of the graph map explicitly included in the structure of the algorithm. The properties of the *DFS-basis* cycles of the graph map allowed us to construct root trees of the cycle nesting structure. The cell cycles were the result of traversing a given set of root trees.

Предлагается конструктивный метод формирования циклов ячеек карты простого планарного графа. Геометрия циклов *DFS-базиса* карты графа явно внесена в структуру алгоритма. Свойства циклов *DFS-базиса* карты графа позволили построить корневые деревья структуры вложенности циклов. Циклы ячеек стали результатом обхода данного множества корневых деревьев.

Keywords: DFS-basis, graph cycles.

1. Введение

Рассматриваемый метод выделения ячеек карты графа реализован как решение задачи раскраски различного рода территорий на поверхности Земли, в частности государств в рамках программного комплекса ГИС "ОКЕАН" [1]. Исходные данные в задаче раскраски территорий — это география поверхности Земли (береговая черта) и границы государств. Линии границ разбивают береговую черту на отрезки, которые дополняют границы государств. Данной модели поверхности Земли поставим в соответствие конечный простой планарный граф $G = (X, U)$, где U — ребра графа (границы государств); X —

Ivanov, B. N., Свойства циклов dfs-базиса карты графа. Выделение циклов ячеек карты.

Поступила 7 февраля 2021 г., опубликована 31 февраля 2021 г.

вершины графа (точки пересечения границ). Простой граф является неориентированным графом и любая пара его вершин соединена не более, чем одним ребром. К такому графу $G = (X, U)$ можно всегда свести рассматриваемую модель поверхности Земли, включая фиктивные вершины на границах или береговой черте.

Укладка на плоскости ребер планарного графа без самопересечений определяет разбиение такой плоскости, называемое *планарным подразбиением* или *картой* графа [2, с. 31]. Карта графа фиксирует геометрию его ребер. Сама карта складывается из простых многоугольников (подразбиений), которые будем называть *ячейками* карты. Циклы ячеек карты иначе называют «безхордовыми циклами» графа (в переводе «chordless cycle»). Каждая сторона такой ячейки может принадлежит ещё ровно одной такой же ячейке. Границы ячеек карты являются простыми циклам. К решению предлагается задача выделения циклов ячеек карты *конечного связного простого планарного графа* $G = (X, U)$. В общем случае граф $G(X, U)$ необходимо разложить на компоненты связности и уже для каждой компоненты отдельно решать обозначенную задачу. Далее полагаем, что исходный граф $G(X, U)$ является связной картой графа.

Изучение циклических структур графа является важной составляющей общего анализа структуры графа. Первый практический алгоритм перечисления всех циклов графа был опубликован в работе *Welch* [3]. Модифицированная версия алгоритма [3] была опубликована в статье *Gibbs* [4]. Работы [3] и [4] определили направление разработки алгоритмов генерации циклов графов. Было показано, что в данной задаче метод поиска с возвращениями позволяет в среднем существенно понизить сложность полного перебора и перейти к практически значимым алгоритмам.

После этого было предложено ряд других алгоритмов [5, 6] порождения *всех циклов графа*. Обзор этих алгоритмов и их относительных достоинств можно найти в обзоре [7]. Основу этих алгоритмов составляет *метод поиска в глубину* на графе [8]. Это адаптированный к структуре графа поиск с возвращениями. Оказалось, что генерацию всех циклов графа не обязательно проводить только на основе циклов базиса.

Сегодня можно встретить и параллельные алгоритмы генерации циклов в графе [9]. В большей степени такие алгоритмы представляют интерес в теоретическом плане. Авторы пытаются решать тестовые задачи большой размерности, используя параллельные вычисления. В этом случае для поиск циклов в работе [10] предлагается приближённый алгоритм, как альтернатива полному перебору. Для информационных сетей актуальна задача сетевой надежности передачи данных. Здесь первостепенное значение приобретает знание циклической структуры сети [11].

В данной работе выделение всех циклов ячеек карты выполняется на основе *DFS-базиса* циклов графа $G(X, U)$. Каждый цикл ячейки карты рассматривается как линейная комбинация циклов базиса. Предлагаемый метод является конструктивным и не обращается к перебору какого-либо множества циклов. Искомые линейные комбинации циклов базиса строятся явно на основе выделенных свойств структуры вложенности циклов *DFS-базиса* и циклов ячеек карты графа. Карта планарного графа позволила отойти от общепринятого подхода генерации циклов и явно внести геометрию карты в структуру алгоритма.

2. Свойства фундаментального множества циклов

Следуя [12, с. 382], введем определение базиса циклов и свяжем его с циклами ячеек карты графа. Пусть $G = (X, U)$ — конечный связный простой планарный граф. Обозначим через $M = \{G_1, G_2, \ldots, G_{n_M}\}$ множество всех подграфов $G_i = (X, U_i)$ графа G, где $U_i \subseteq U$. На множестве M определим бинарную операцию $\oplus$ так, что $\forall G_i, G_j \in M\ : G_i \oplus G_j = (X, U_i \oplus U_j)$, где символ $\oplus$ обозначает операцию симметрической разности, $U_i \oplus U_j = \{u \,|\, u \in U_i \cup U_j \ \wedge\ u \notin U_i \cap U_j\}$. Множество M с операцией $\oplus$ является абелевой группой. Единица группы — пустой подграф $O = (X, \varnothing)$. Обратным к G_k является тот же самый G_k, так как $G_k \oplus G_k = O$.

Обозначим через $Z = \{Z_1, Z_2, \ldots, Z_{n_Z}\}$, где $Z_k = (X, z_k)$ — подграфы графа G, z_k — все простые циклы графа G. Пусть $L = \{\lambda_{i_1} Z_{k_1} \oplus \lambda_{i_2} Z_{k_2} \oplus \ldots \oplus \lambda_{i_Z} Z_{k_Z}\}$ — линейная оболочка циклов множества Z, где $\lambda_{i_r} \in \{0, 1\}$, $\ 0 \cdot Z_{k_r} = O$ и $1 \cdot Z_{k_r} = Z_{k_r}$. Множество L является коммутативной группой с операцией $\oplus$ и линейным векторным пространством *циклов* над полем $P =< \{0, 1\}, \oplus, \cdot >$ [13, с. 61]. Уточним структуру, размерность и базис пространства L.

Пусть $T_0 = (X, U_0)$ — произвольное остовное дерево исходного графа $G = (X, U)$. Для дерева выполняется соотношение $|U_0| = |X| - 1$. Пусть $E = U \setminus U_0 = \{e_1, e_2, \ldots, e_{n_F}\}$ — ребра, не вошедшие в T_0. Такие ребра называют обратными. Любое ребро $e_i \in E$ порождает в T_0 ровно один *простой цикл* C_i, где $e_i \in C_i$ и $e_i \notin C_j$, если $i \neq j$. Таких циклов C_i можно составить $n_F = |U \setminus U_0| = |U| - |X| + 1$. Множество $F = \{C_1, C_2, \ldots, C_{n_F}\}$ называется *фундаментальным множеством циклов* и составляет базис пространства L [13, с. 63].

Замечание 1. *Все простые циклы графа распределяются между блоками графа. Каждый простой цикл расположен целиком в своем блоке* [14, с. 114]. *Поэтому далее полагаем, что исходный граф $G = (X, U)$ есть не вырожденный конечный блок карты графа (вырожденный блок состоит из одного ребра). Для выделения блоков можно воспользоваться алгоритмом, представленным в оригинальной работе* [8] — *модифицированный* метод поиска в глубину.

Замечание 2. *В блоке все ребра сильно циклически связаны* [14, с. 109]. *Это означает, что для любой пары ребер блока u_1, u_2 существует такая последовательность простых циклов $H_1, H_2, \ldots, H_k$, что $u_1 \in H_1$, $u_2 \in H_k$ и любая пара соседних циклов H_i, H_{i+1} имеет по крайней мере одно общее ребро.*

Приведём две теоремы из работы О. Оре [14, с. 110], на которые будем ссылаться ниже.

Теорема 1. *Два ребра сильно циклически связаны тогда и только тогда, когда существует простой цикл, содержащий оба эти ребра (следствие теоремы 2)* [14, с. 110].

Теорема 2. *Если e_1 — ребро в простом цикле H_1, e_2 — ребро в простом цикле H_2, а H_1 и H_2 имеют хотя бы две общие вершины, то существует простой цикл H, содержащий оба ребра e_1 и e_2* [14, с. 110].

Доказательство. Можно считать, что e_1 не принадлежит H_2. Тогда, проходя по циклу H_1 в обе стороны от ребра e_1, мы найдем соответственно две вершины x_1 и x_2, принадлежащие также H_2. Обозначим через $H_1(x_1, e_1, x_2)$, $H_2(x_1, e_2, x_2)$ участки циклов H_1 и H_2, содержащие соответственно ребра e_1 и e_2. Их объединение $H = H_1(x_1, e_1, x_2) \cup H_2(x_1, e_2, x_2)$ есть простой цикл, содержащий оба ребра e_1 и e_2.

Обозначим через $R_1(x_1, x_2)$ и $R_2(x_1, x_2)$ участки циклов H_1 и H_2, не вошедшие в простой цикл $H = H_1(x_1, e_1, x_2) \cup H_2(x_1, e_2, x_2)$. Если $R_1(x_1, x_2) \neq R_2(x_1, x_2)$, то их объединение $R_1(x_1, x_2) \cup R_2(x_1, x_2)$ содержит хотя бы один цикл, а, значит, $H_1 \oplus H_2$ состоит более, чем из одного простого цикла и $H \neq H_1 \oplus H_2$.

Если $R_1(x_1, x_2) = R_2(x_1, x_2)$, то $H = H_1 \oplus H_2$. где H — простой цикл, составленный из участков $H_1(x_1, e_1, x_2)$ и $H_2(x_1, e_2, x_2)$. □

Определение 1. *Под областью цикла H будем понимать открытую область, заключенную в границах данного цикла на карте графа. Область цикла H будем обозначать через $\alpha(H)$, а её площадь через $\mu(H)$.*

Теорема 3. *Пусть карта графа $G = (X, U)$ является блоком и H есть его простой цикл. Тогда подграф $B(H) = (X_H, U_H) \subseteq G$, ограниченный циклом H, является блоком.*

Доказательство. Пусть $u, v \in U_H$ — произвольные два ребра. Тогда в графе G существует простой цикл P, для которого $u, v \in P$. Если цикл P выходит за границы области цикла H, то существует и простой цикл в графе $B(H)$, содержащий ребра u и v. Действительно, исходный граф — это карта графа, и все пересечения ребер цикла P с ребрами цикла H являются вершинами в X_H. В этом случае любую выступающая часть дуги цикла P за область цикла H заменим на часть дуги цикла H, которая замыкает выступающую часть дуги P. Новый цикл P будет простым и локализованным в границах цикла H. Значит $B(H)$ — блок. □

Определение 2. *Будем говорить, что цикл H_1 охватывает цикл H_2 или цикл H_2 вложен в цикл H_1 на карте графа, если $\alpha(H_1) \cap \alpha(H_2) = \alpha(H_2)$. Такие циклы могут иметь общие ребра и вершины.*

Определение 3. *Два цикла H_1, H_2 будем называть* смежными*, если их пересечение $H_1 \cap H_2$ состоит из одного непрерывного участка ребер. Под операцией $H_1 \cap H_2$ понимается пересечение циклов H_1, H_2.*

Свойство 1. *Если C_1 и C_2 — смежные циклы базиса, то $C_1 \oplus C_2 = H$, где H — простой цикл, составленный из двух участков циклов C_1 и C_2, после удаления из них общего участка ребер.*

Свойство 2. *Если пара циклов базиса C_1 и C_2 имеет хотя бы две общие вершины, то такие циклы смежные.*

Доказательство. Пусть $e_1 \in C_1$ и $e_2 \in C_2$ — обратные ребра соответствующих циклов $e_1 \notin C_2$ и $e_2 \notin C_1$. Циклы базиса C_1 и C_2 — простые циклы. Тогда по теореме 2 существует простой цикл H, содержащий оба ребра e_1 и e_2. По построению (см. теорему 2) данный цикл H состоит из ребер исходных циклов C_1 и C_2. Если для него $H = C_1 \oplus C_2$, то по построению такие циклы C_1 и C_2 имеют один общий непрерывный участок ребер, значит C_1 и C_2 смежные.

Предположим теперь, что $H \neq C_1 \oplus C_2$. Разложим цикл H по базису $H = \lambda_1 C_1 \oplus \lambda_2 C_2 \oplus \ldots \oplus \lambda_{n_F} C_{n_F}$. Так как $H \neq C_1$, $H \neq C_2$ и $H \neq C_1 \oplus C_2$, то существует $\lambda_r \neq 0$, $r \neq 1$ и $r \neq 2$. Значит обратное ребро $e_r \in C_r$ является ребром цикла $e_r \in H$. По построению цикл H составлен из ребер циклов C_1 и C_2. Следовательно, обратное ребро $e_r \in C_r$ принадлежит и одному из циклов C_1 или C_2. Это противоречит свойству циклов базиса, что каждый его цикл определяется ровно одним обратным ребром $e_1 \in C_1$ и $e_2 \in C_2$, где $e_1 \neq e_r$ и $e_2 \neq e_r$. Значит предположение не верное и $H = C_1 \oplus C_2$, а, значит, циклы C_1, C_2 смежные. □

Замечание 3. *Взаимное расположение каждой пары циклов базиса C_1 и C_2 такого, что если $C_1 \cap C_2 \neq \varnothing$, то либо C_1 и C_2 — смежные (свойство 2), либо C_1 и C_2 касаются только в одной вершине.*

Свойство 3. *Если C_1, C_2 — циклы базиса и $C_1 \oplus C_2$ — простой цикл, то C_1 и C_2 — смежные. Действительно, если $C_1 \oplus C_2$ — один цикл, то циклы C_1, C_2 должны пересекаться по ребрам. По свойству 2 такие циклы смежные.*

3. Свойства циклов *DFS-базиса* карты графа

В нашей задаче для построения фундаментального множества циклов обратимся к *методу поиска в глубину.* Подробное изложение данного алгоритма встречается во многих работах. Оригинальное его описание представлено в статье [15]. Адаптированный вариант алгоритма для практического применения рассматривается в работе [12, с. 385]. При порождении фундаментального множества циклов метод поиска в глубину на простом связном графе $G = (X, U)$ строит дерево поиска, которое является остовным деревом графа и называется *DFS-деревом.* Каждое обратное ребро порождает один цикл *DFS-базиса* относительно этого дерева. Для того чтобы следить за ребрами дерева, используется поиск в глубину *со стеком*, в котором хранятся все текущие вершины пройденного пути в данный момент. Когда попадаем на обратное ребро, обнаруженный цикл будет состоять из этого ребра и ребер, соединяющих вершины из верха стека до вершины обратного ребра в стеке.

Определение 4. *Пусть $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ — циклы базиса, смежные участки циклов которых не пересекаются. Для данной совокупности циклов определим* неориентированный граф смежности $J_v = (X_v, U_v)$. *Вершинами графа $v_i \in X_v$ являются циклы $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$. Ребро $(v_i, v_j) \in U_v$, если циклы C_{v_i}, C_{v_j} смежные. Пусть $J_v = (X_v, U_v)$ имеет структуру дерева. Тогда циклы $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ будем называть* смежными в совокупности. *Дерево — неориентированный связный ациклический граф* [14, с. 77]. *В частности, дерево не имеет петель и кратных ребер.*

Определение 5. *Два графа смежности J_v и J_w будем называть* несмежными*, если каждая пара циклов $C_{v_i} \in J_v$ и $C_{w_j} \in J_w$ является несмежной.*

Свойство 4. *Результатом слияния циклов базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$, смежных в совокупности, будет один простой цикл $L_v = C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_k}$.*

Доказательство. Действительно, слияние простых циклов $C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_k}$ можно получить в результате обхода, например сверху-вниз, графа смежности $J_v = (X_v, U_v)$, соответствующего указанным циклам. Положим вначале обхода цикл слияния L_v равным C_{v_k} — начало обхода дерева. Каждый раз, приходя в вершину C_{v_j}, будем выполнять слияние $L_v = L_v \oplus C_{v_j}$. Вершина C_{v_j} смежна ранее пройденной вершине $C_{v_i} \in J_v$. Слияние смежных простых циклов

$C_{v_i} \oplus C_{v_j}$ даёт один простой цикл. А так как границы смежных циклов C_{v_j} не пересекаются, то и результатом операции $L_v = L_v \oplus C_{v_j}$ будет один простой цикл. □

Свойство 5. *Любые три цикла DFS-базиса C_1, C_2, C_3, попарно смежные друг с другом, имеют общий участок ребер. На карте графа, по крайней мере, один из данных циклов будет вложен в другой.*

Доказательство. Смежные циклы *DFS-базиса* при их формировании размещаются в стеке циклов последовательно (см. алгоритм в работе [12, с. 385]). Пусть порядок размещения циклов в стеке C_1, C_2, C_3. (i) Общий участок в стеке первого и второго циклов является началом второго цикла. (ii) Общий участок в стеке второго и третьего циклов является началом третьего цикла. (iii) Общий участок в стеке первого и третьего циклов является началом третьего цикла. Из (i), (ii), (iii) заключаем, что участок ребер начало третьего цикла является общим для всех циклов.

На *карте графа* компоновка трёх циклов базиса при общей у них границы возможна, если, по крайней мере, один из данных циклов будет вложен в другой. □

Свойство 6. *Пусть циклы DFS-базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$, смежные участки циклов которых не пересекаются, определяют связный граф смежности $J_v = (X_v, U_v)$. Тогда $J_v = (X_v, U_v)$ имеет структуру дерева.*

Доказательство. Предположим, что в графе смежности $J_v = (X_v, U_v)$ существует цикл $(C_{r_1}, C_{r_2}, \ldots, C_{r_m}, C_{r_1})$, для цикла должно выполняться условие $m \geqslant 3$. В данном цикле каждая пара соседних циклов является смежной.

Смежные циклы *DFS-базиса* при их формировании размещаются в стеке циклов последовательно (см. алгоритм в работе [12, с. 385]). Пусть в стеке циклы имели исходный порядок $(C_{r_1}, C_{r_2}, \ldots, C_{r_m}, C_{r_1})$. Общий участок в стеке первого и второго циклов является началом второго цикла. Общий участок в стеке второго и третьего циклов является началом третьего цикла и т. д. Общий участок в стеке первого и m-го циклов является началом m-го цикла. Отсюда заключаем, что участок ребер начало m-го цикла является общим для всех циклов. Так как $m \geqslant 3$, то циклы C_1, C_2, C_m — попарно смежные. Из свойства 5 заключаем, что у них есть общий участок ребер. Это противоречит условию, что общие участки смежных циклов не пересекаются. Значит связный граф смежности J_v имеет структуру дерева. □

Свойство 7. *Пусть циклы DFS-базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ карты графа попарно не вложены друг в друга. Тогда граф смежности $J_v = (X_v, U_v)$ для данной совокупности циклов имеет структуру дерева.*

Доказательство. Так как циклы $C_{v_1}, C_{v_2}, \ldots, C_{v_k}$ попарно не вложены друг в друга, то их смежных участки циклов не пересекаются. В противном случае среди исходной совокупности циклов карты графа будут вложенные циклы, что противоречит условию. Из свойства 6 заключаем, что *граф смежности* J_v имеет структуру дерева. □

Свойство 8. *Пусть структура циклов $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ карты графа DFS-базиса, попарно не вложенных друг в друга, представляется совокупностью* *деревьев смежности $J_{t_1}, J_{t_2}, \ldots, J_{t_k}$. Тогда каждый граф J_{t_j} является отдельным блоком.*

Доказательство. Каждый граф J_{t_j} является блоком, так как он составлен из попарно смежных циклов базиса (см. замечание 2). Покажем, что графы J_{t_i} и J_{t_j} являются различными блоками. Если предположить, что графы смежности J_{t_i} и J_{t_j} составляют один блок, то они должны иметь, по крайней мере, две общие вершины $x1$ и $x2$ [14, с. 111].

Рассмотрим остовное дерево T_0 графа после удаления из него обратных ребер. Удаление обратных ребер разрушит только одну сторону в каждом из циклов базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ и сохранит переход по другой стороне цикла от одного его смежного участка до другого по ребрам графов J_{t_i} и J_{t_j} (обратные ребра не могут находиться в смежной части циклов базиса). Следовательно, после удаления обратных ребер вершины x_1 и x_2 останутся связанными в J_{t_i} и J_{t_j} двумя различными маршрутами, которые составят цикл. Это противоречит тому, что в дереве T_0 нет циклов. Следовательно, каждый граф смежности J_{t_j} является отдельным блоком или J_{t_i} и J_{t_j} есть один блок. □

Свойство 9. *Пусть C_v — произвольный цикл DFS-базиса карты графа и структура вложенных в него циклов базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ представляется совокупностью графов смежности $J_{t_1}, J_{t_2}, \ldots, J_{t_k}$, каждый из которых не смежный с другим. Тогда выполняется:*

1). В каждом графе J_{t_j} существует единственный цикл базиса $C_{t_j} \in J_{t_j}$ смежный циклу C_v.

2). Общими вершинами графов J_{t_j} могут быть только вершины цикла C_v.

3). Циклы C_v и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ являются смежными в совокупности.

Доказательство. Доказательство условия 1). Исходный граф является блоком, тогда и подграф $B(C_v)$, ограниченный простым циклом C_v, будет блоком (см. теорему 3). *Циклы базиса C_v и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ есть локальный базис для циклов блока $B(C_v)$.* Все рёбра циклов блока $B(C_v)$ являются ребрами циклов C_v и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ и распределяются между циклом C_v и графами J_{t_j}. В противном случае локальный базис циклов будет не полным.

Из свойства 8 имеем, что все графы J_{t_j} — различные блоки и цикл C_v является блоком. Подграф $B(C_v)$ есть блок, составляющими его являются блоки J_{t_j} и C_v. Это возможно, если каждый из блоков J_{t_j} и C_v будет иметь, по крайней мере, две общие вершины [14, с. 111].

Пусть J_{t_j} и C_v имеют две общие вершины x_1 и x_2. Если x_1 и x_2 из одного цикла $C_{t_j} \in J_{t_j}$, то свойство 2 утверждает, что циклы базиса C_{t_j} и C_v — смежные.

Предположим теперь, что x_1 и x_2 из разных циклов графа J_{t_j}. Обозначим через $T_0(C_v)$ остовное дерево блока $B(C_v)$, после удаления обратных рёбер. Граф J_{t_j} имеет структуру дерева (свойство 7) и составлен из попарно смежных циклов базиса. После удаления обратных ребер, граф J_{t_j} останется *связным.* Действительно, обратные ребра не входят в смежные участки циклов, а удаление обратного ребра разрушит только одну сторону цикла и сохранит переход по другой стороне от одной смежной части цикла до другой. Итак, вершины x_1 и x_2 остаются связанными в J_{t_j} после удаления обратных ребер.

Так как C_v есть *цикл* базиса, то после удаления из него обратного ребра, вершины x_1 и x_2 остаются связанными по его ребрам. Следовательно, вершины

x_1 и x_2 будут связаны двумя различными маршрутами, а в остовном дереве $T_0(C_v)$ будет цикл. Значит предположение не верное, вершины x_1 и x_2 должны быть из одного цикла базиса.

Доказательство условия 2). Из доказательства (1) имеем, что графы J_{t_i} и J_{t_j} смежные с циклом C_v. Предположим, что графы J_{t_i} и J_{t_j} имеют хотя бы одну общую вершину x_0, отличную от вершин цикла C_v. Удаление обратных ребер в блоке $B(C_v)$ не нарушит связности графов J_{t_i} и J_{t_j} и будет существовать маршрут от y_1 по ребрам J_{t_i} через x_0 по ребрам J_{t_j} в y_2, где $y_1 \neq y_2$ — вершины смежных участков графа J_{t_i} с циклом C_v и графа J_{t_j} с циклом C_v.

Удаление обратного ребра в цикле C_v не нарушит и его связности — останется переход по нему от y_1 до y_2, а в остовном дереве $T_0(C_v)$ будет цикл. Следовательно, предположение не верное, и графы J_{t_i} и J_{t_j} не могут иметь общих вершин, кроме вершин на цикле C_v.

Доказательство условия 3). Из доказательства (1) и (2) имеем, что графы смежности J_{t_j} попарно не смежные между собой и смежные с циклом C_v. В каждом J_{t_j} существует только один цикл $C_{t_j} \in J_{t_j}$ смежный циклу C_v. Следовательно, циклы C_v и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ смежные в совокупности. □

Свойство 10. *Пусть C_v — произвольный цикл DFS-базиса карты графа и перечень вложенных в него циклов базиса $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$. Тогда цикл*

$$S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n} \tag{1}$$

определяет одну ячейку карты графа, локализованной в цикле C_v.

Доказательство. Из свойства 9 имеем, что циклы C_v и $C_{v_1}, C_{v_2}, \ldots, C_{v_n}$ — смежные в совокупности. Граф смежности $J_v = (X_v, U_v)$ данной совокупности имеет структуру дерева (свойство 6). Результатом обхода данного дерева будет один цикл слияния $S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n}$ (свойство 4 и 9). Структурно цикл S_v есть цикл C_v, из которого удалили все вложенные в него циклы *DFS-базиса.* Значит S_v есть цикл ячейки карты графа. Предположение, что остаток S_v может оказаться объединением двух ячеек карты или более, нарушит условие полноты циклов *DFS-базиса.* Циклы *DFS-базиса* должны охватывать все циклы S_v ячеек карты графа. В противном случае базис будет не полным. Тогда из формулы (1) заключаем, что число ячеек карты графа совпадает с числом циклов базиса. □

4. Корневые деревья структуры вложенности циклов базиса

Введем на множестве циклов *DFS-базиса* карты графа $F = \{C_1, C_2, \ldots, C_{n_F}\}$ бинарные отношения, с целью определения новых структур на этом множестве, которые обеспечат доступ к перечню циклов $C_1, C_2, \ldots, C_{n_F}$ в соответствии с их структурой вложенности, с целью выделения циклов ячеек карты графа по формуле (1).

Определение 6. *Отношение $H_1 \subset H_2$ будет обозначать вложенность цикла H_1 в цикл H_2.*

Определение 7. *Определим на множестве циклов $F = \{C_1, C_2, \ldots, C_{n_F}\}$ бинарное отношение $(\prec)$ охватывающего соседства. Отношение $C_i \prec C_j$ выполняется, если $C_i \subset C_j$ и $\forall C_k \neq C_j \ : (C_i \subset C_k \Rightarrow C_j \subset C_k)$. Это означает, что C_j — наименьший охватывающий цикл для C_i.*

Поставим в соответствие структуре вложенности циклов базиса $F = \{C_1, C_2, \ldots, C_{n_F}\}$ *ориентированный граф циклов* $G_F = (X_F, U_F)$, где $X_F = \{C_1, C_2, \ldots, C_{n_F}\}$ — множество вершин; U_F — множество ребер, ребра представляются парами вершин. Так ориентированное ребро $u = (C_i, C_j) \in U_F$, если для циклов C_i и C_j выполняется отношение $C_i \prec C_j$. Направленное ребро (C_i, C_j) ориентировано от вложенного цикла C_i к наименьшему охватывающему его циклу C_j $(C_i \to C_j)$. Практическая реализация отношения $C_i \prec C_j$ не вызывает трудностей. В качестве наименьшего охватывающего цикла C_j для цикла C_i назначается тот, для которого выполняется условие $\min\limits_{k=\{1,2,\ldots,n_F\}} \{\, \mu(C_k) \,|\, C_i \subset C_k \,\}$, где $\mu(C_k)$ — площадь, охватываемая циклом C_k. Не вызывает сложностей и проверка вложенности циклов $C_i \subset C_k$. Для каждого цикла C_i в векторе счётчиков $r_1, r_2, \ldots, r_F$. будем фиксировать его уровень вложенности, где r_i — количество циклов базиса C_k, которые охватывают данный цикл C_i. Эффективные практические приемы проверки вложенности циклов $C_i \prec C_j$ и вычисления значений r_i уровней вложенности циклов C_i можно найти работе [16].

Введённое отношение охватывающего соседства $(\prec)$ позволяет выполнить представление *корневых деревьев циклов* графа $G_F = (X_F, U_F)$ ориентированным реберным списком $Q_1, Q_2, \ldots, Q_{n_F}$, где элемент $Q_i = j$, если выполняется отношение $C_i \prec C_j$. Для циклов C_i, которые не имеют охватывающих циклов, значение $Q_i = 0$. Такими вершинами являются корни деревьев графа $G_F = (X_F, U_F)$. Корни деревьев определяются по вектору счётчиков $r_1, r_2, \ldots, r_F$. Для корней значения счётчиков $r_i = 0$. *Корневые деревья циклов* позволяют выполнить перебор циклов базиса C_i по уровням их вложенности как сверху-вниз, так и обратно. Эффективность применения корневых деревьев циклов можно посмотреть в работе [17] на примере решения задачи выделения циклонических образований и их траекторий по структуре вложенности изолиний.

В описании алгоритмов для представления неориентированного графа циклов $G_F = (X_F, U_F)$ будет использоваться его структура смежности $Adj[x]$ — множество вершин, смежных с данной вершиной $x \in X_F$. Процедура преобразования реберного списка $Q_1, Q_2, \ldots, Q_{n_F}$ в структуру смежности $Adj[x]$ представлена в алгоритме 1.

Алгоритм 1. Преобразование реберного списка в структуру смежности

```
Procedure CreateAdj( Q, Adj );
   for x ∈ X_F do begin
      Adj[x] = ∅; {Начальный пустой список для x}
      if Q_x ≠ 0 then Adj[x] = {Q_x}; {Ребро, выходящее из x}
      for v ∈ X_F do begin
         if Q_v = x then Adj[x] = Adj[x] ∪ {v}; {Ребра, входящие в x}
      end;
   end;
end.
```

Замечание 4. *Корневая же структура деревьев графа G_F должна быть сохранена и в структуре смежности $Adj[x]$. Это будет выполнено, если в перечне X_F корни деревьев будут встречаться раньше своих вершин. Для этого* *необходимо упорядочить вектор структуры смежности $Adj[\,]$ по возрастанию счётчиков вложенности циклов $r_1, r_2, \ldots, r_{_F}$. Если $r_x \leqslant r_y$, то данные для вершины x в векторе $Adj[\,]$ должны располагаться раньше, чем данные для вершины y.*

5. Формальный алгоритм выделения циклов ячеек карты графа

В основе алгоритма лежит свойство 10 циклов *DFS-базиса*. Для построения циклов S_v ячеек карты графа необходимо перебрать все циклы *DFS-базиса* C_v, удаляя из каждого вложенные в него циклы базиса C_{v_j} по формуле $S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n}$ (свойство 10).

Вынесем из общего алгоритма 3 реализацию процедуры слияния циклов (свойство 4). Сама процедура слияния $Join(W, S_v)$ представлена в алгоритме 2. Параметрами процедуры выступают множество указанных циклов $W = \{C_v, C_{v_1}, C_{v_2}, \ldots, C_{v_n}\}$ и результирующий цикл слияния S_v. Начальное значение цикла полагаем равным пустому циклу $S_v = O$. Чтобы результатом слияния $S_v = S_v \oplus C_w$, где $C_w \in W$, на каждом шаге был ровно один цикл, необходимо, чтобы цикл C_w был смежным с текущим циклом слияния S_v.

Согласно свойству 4 это возможно, если перебор циклов $C_w \in W$ выполнить одним из обходов дерева смежности $J_w = (X_w, U_w)$ для совокупности циклов $W = \{C_v, C_{v_1}, C_{v_2}, \ldots, C_{v_n}\}$. В данном случае воспользуемся обходом сверху-вниз. Процедура обхода сверху-вниз заимствована из работы [8], в основе которой лежит обход графа в глубину.

Вначале процедуры формируется структура смежности $Adj[x]$ представления *структуры деревьев* графа смежности $J_w = (X_w, U_w)$. Рекурсивная процедура $Depth(x)$ выполняет обход сверху-вниз поддерева с корнем в вершине $x \in X_w$. Результатом обхода графа смежности $J_w = (X_w, U_w)$ будет цикл слияния S_v — ячейка карты графа.

Алгоритм 2. Слияние множества W смежных в совокупности циклов

Procedure $Join(W, S_v)$**;**
  $X_w = \varnothing$**;**
  for $i = 1$ **to** $|W|$ **do** $X_w = X_w \cup \{i\}$**;** *{Номера циклов множества W}*
  for $C_x \in W$ **do begin** *{Формирование $Adj[x]$ дерева $J_w = (X_w, U_w)$}*
    $Adj[x] = \varnothing$**;** *{Начальный пустой список $Adj[x]$}*
    for $C_y \in W$ **do begin**
      if $(x \neq y) \wedge (C_x \cap C_y \neq \varnothing)$ **then begin**
        $Adj[x] = Adj[x] \cup \{y\}$**;** *{Расширить список $Adj[x]$}*
      end;
    end;
  end;
  $z \in X_w$**;** *{Начало обхода графа смежности $J_w = (X_w, U_w)$}*
  $S_v = C_z$**;** *{Начальный цикл слияния}*
  $Depth(z)$**;** *{Обход поддерева с корнем в z}*
end.
Procedure $Depth(x)$**;** *{Обход сверху-вниз поддерева с корнем в x}*
  for $y \in Adj[x]$ **do begin** *{Перебор смежных вершин}*
    $S_v = S_v \oplus C_y$**;** *{Слияние циклов}*
    $Depth(y)$**;** *{Обход поддерева с корнем в y}*
  end;

end.

Расчёт циклов S_v ячеек карты графа выполняется по формуле $S_v = C_v \oplus C_{v_1} \oplus C_{v_2} \oplus \ldots \oplus C_{v_n}$ (свойство 10). Последовательность вычисления циклов S_v имеет существенное значение. В рассматриваемой задаче перебор циклов C_v наиболее оптимально выполнить обходом снизу-вверх *корневых деревьев* циклов базиса (см. раздел 4). Суть данного обхода составляет процедура, заимствованная из работы [8]. В этом случае обработке цикла C_v будет предшествовать перебор вложенных в него циклов C_{v_j}. Данный подход реализован в алгоритме 3 формирования циклов ячеек карты графа.

Замечание 5. *В алгоритме 3 исходный граф циклов $G_F = (X_F, U_F)$ задаётся структурой смежности $Adj[x]$, формирование её рассматривалось в алгоритме 1. Корневая же структура деревьев графа G_F должна быть сохранена и в структуре смежности $Adj[x]$. При обходе графа G_F это будет выполнено, если в перечне X_F корни деревьев будут встречаться раньше своих вершин (см. замечание 4).*

Чтобы отличить уже пройденные вершины, вводится вектор меток вершин $Mark[x]$, значения которых равно 1 для пройденных вершин и 0 — для не пройденных. Цикл C_x и вложенные в него циклы формируют множество W смежных в совокупности циклов. Слияние циклов множества W процедурой $Join(W, S_x)$ дает цикл S_x ячейки карты графа. Все выделенные циклы S_x составят искомое множество $S = \{S_1, S_2, \ldots, S_{n_F}\}$ ячеек карты графа.

Алгоритм 3. Формирование множества S циклов ячеек карты графа

```
S = ∅; {Начальное пустое множество ячеек карты графа}
for v ∈ X_F do Mark[v] = 0; {Метки не пройденных вершин}
for v ∈ X_F do if Mark[v] = 0 then begin {Поиск начала обхода}
   Root(v); {Обход корневого дерева с корнем в вершине v}
end;
Procedure Root(x); {Обход дерева снизу-вверх с корнем в x}
   Mark[x] = 1; {Вершина пройдена}
   W = ∅; {Начальное пустое множество циклов для C_x}
   for v ∈ Adj[x] do begin {Перебор смежных вершин}
      if Mark[v] = 0 then begin {Поиск не пройденной вершины}
         Root(v); {Обход поддерева с корнем в v}
         W = W ∪ {C_v}; {Добавить вложенный цикл C_v}
      end;
   end;
   W = W ∪ {C_x}; {Добавить цикл C_x — корень дерева}
   Join(W, S_x); {S_x — результат слияния циклов множества W}
   S = S ∪ {S_x}; {Добавить цикл ячейки S_x к карте S графа}
end.
```

6. Заключение

В работе предложен конструктивный метод формирования циклов ячеек карты простого планарного графа. Геометрия циклов *DFS-базиса* карты графа явно внесена в структуру алгоритма. Доказанные свойства циклов *DFS-базиса* карты графа позволили построить корневые деревья структуры вложенности циклов. Формирование циклов ячеек карты графа — это результат обхода указанных корневых деревьев. Формализованная часть алгоритма и его структуры данных рассмотрены на уровне, близком к практическому их использованию.

References

[1] B. N. Ivanov, *Solution of the Optimal Ship Route Problem in the Framework of the OKEAN Geoinformation System*, Vychisl. Metody Programm., **13**:1 (2012), 226–234.

[2] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer, Heidelberg, 1985; Mir, Moscow, 1989.

[3] J. Welch, *A Mechanical Analysis of the Cyclic Structure of Undirected Linear Graphs*, J. Assoc. Comput. Mech., **13** (1966), 205–210.

[4] N. W. Gibbs, *A Cycle Generation Algorithm for Finite Undirected Linear Graphs*, J. Assoc. Comput. Mech., **16** (1969), 564–568.

[5] R. Tarjan, *Enumeration of the Elementary Circuits of a Directed Graph*, SIAM J. Comput., **2**:3 (1973), 211–216.

[6] D. B. Jonson, *Finding All the Elementary Circuits of a Directed Graph*, SIAM J. Comput., **4**:1 (1975), 77–84.

[7] P. Mateti and N. Deo, *On Algorithms for Enumerating All Circuits of a Graph*, SIAM J. Comput., **5**:1 (1976), 90–99.

[8] R. Tarjan, *Depth-First Search Linear Graph Algorithms*, SIAM J. Comput., **1**:2 (1972), 146–160.

[9] F. Mahdi, M. Safar, and K. Mahdi, *Detecting Cycles in Graphs Using Parallel Capabilities of GPU*, in Digital Information and Communication Technology and Its Applications (Springer, Berlin, 2011), **167**, pp. 193–205.

[10] T. Kavitha, K. Mehlhorn, and D. Michail, *New Approximation Algorithms for Minimum Cycle Bases of Graphs*, Algorithmica, **59**:4 (2011), 471–488.

[11] J. L. Pfaltz, *Chordless Cycles in Networks*, in Proc. IEEE 29th Int. Conf. on Data Engineering Workshops (IEEE Press, New York, 2013), pp. 223–228.

[12] E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms. Theory and Practice*, PrenticeHall, Englewood Cliffs, 1977; Mir, Moscow, 1980.

[13] B. E. Alekseev and V. A. Talanov, Graphs. *Computational Models. Structures*, Lobachevsky State Univ. of Nizhni Novgorod, Nizhni Novgorod, 2005.

[14] O. Ore, *Theory of Graphs*, AMS Press, Providence, 1962; Nauka, Moscow, 1980.

[15] K. Paton, *An algorithm for finding a fundamental set of cycles of a graph*, Comm. ACM, **12**:9 (1969), 514–518.

[16] B. N. Ivanov, *Enclosure Structures of Equal Level Line Fields in the Gradient Filling Problem*, Vychisl. Metody Programm, **7**:2 (2006), 30–40.

[17] B. N. Ivanov, it Geometric approach to solving the problem of constructing the trajectories of cyclones and anticyclones, Vychisl. Metody Programm, **15**:2 (2014), 370–382.

Boris Nikolaevich Ivanov
Far Eastern Federal University,
School of Natural Sciences,
Department of Algebra, Geometry and Analysis,
st. Suhanova, 8,
690091, Vladivostok, Russia
E-mail address: ivanov.bn@dvfu.ru