УДК 515.14+519.6

МЕТОД ЭКВИВАЛЕНТНЫХ ДУГ ДЛЯ МИНИМИЗАЦИИ ПУТЕЙ В ЗАДАННОМ ГОМОЛОГИЧЕСКОМ КЛАССЕ

А.В. Галанин

Нижегородский госуниверситет им. Н.И. Лобачевского

al@galanin.nnov.ru

Поступила в редакцию 24.09.2013

Рассматриваются триангулированные замкнутые многообразия, реберные пути на них и группы гомологий по модулю 2. Разработан метод снижения алгоритмической сложности для алгоритма поиска минимального пути, гомологичного заданному.

Ключевые слова: симплекс, полиэдр, группа гомологий, алгоритм, минимизация.

Краткие сведения об индексной вектор-функции

В настоящем пункте мы кратко напомним понятие индексной вектор-функции, более детально данная конструкция описана в [1]. Пусть P – m-мерное триангулированное многообразие без края и $\mathrm{Ind}: H_1(P) \times H_{m-1}(P) \to \mathbf{Z}_2$ – индекс пересечения, а $[y_1], \ldots, [y_r]$ – базис группы гомологий $H_{m-1}(P)$.

Гомоморфизм $J: C_1(P) \to \mathbf{Z}_2^r$, $J = (J^1, \ldots, J^r)$, называется индексной вектор-функцией, если для произвольных $z \in Z_1(P)$ и $k \in \{1, \ldots, r\}$ имеет место равенство

$$J^k(z) = \mathrm{Ind}([z],[y_k]). \tag{1}$$

При этом для любой цепи $x \in C_1(P)$ значение $J(x)$ называется ее индексом относительно базиса $[y_1], \ldots, [y_r]$.

Имеет место [1]

Предложение 1. Если $x, x' \in C_1(P)$ и $\partial x = \partial x'$, то $J(x) = J(x')$ тогда и только тогда, когда $x \sim x'$.

Процесс построения компонент индексной вектор-функции $J: C_1(P) \to \mathbf{Z}_2^r$ относительно некоторого набора базисных $(m-1)$-мерных циклов называется индексацией.

Постановка задачи и подходы к ее решению

В работе рассматриваются прямолинейные полиэдры в $\mathbf{R}^n$. Пути предполагаются рёберными. Полиэдр предполагается двумерным, однако алгоритм может быть модифицирован для поиска минимального пути и в полиэдре размерности m. Используются группы гомологий с коэффициентами из поля $\mathbf{Z}_2$.

Дан полиэдр P, являющийся двумерным замкнутым многообразием, неотрицательная весовая функция $L: C_1(P) \to \mathbf{R}$ и путь $x_0 \in C_1(P)$ с начальной вершиной s и конечной вершиной t. Требуется найти путь с минимальным весом среди всех путей с концами s и t, гомологичных x_0.

Для решения этой задачи в [1] используется переход к накрывающему полиэдру $\hat{P}$. Его вершинами являются пары (v, i), где v – вершина исходного полиэдра P, а $i \in \mathbf{Z}_2^r$, $r = \mathrm{rank}\, H_1(P)$. Кроме того, для любого пути $x \in C_1(P)$, соединяющего вершины $a, b \in V(P)$, накрывающие пути соединяют вершины $(a, i_a), (b, i_b) \in V(\hat{P})$, удовлетворяющие равенству $i_a + i_b = J(x)$. Здесь $J: C_1(P) \to \mathbf{Z}_2^r$ – описанная выше индексная вектор-функция относительно некоторого базиса группы $H_1(P)$.

Запишем основные шаги алгоритма, разработанного в [1].

Алгоритм 1

1. Нахождение базиса группы гомологий $H_1(P)$.
2. Построение индексной вектор-функции J относительно найденного базиса.
3. Построение накрывающего полиэдра $\hat{P}$.
4. Применение алгоритма Дейкстры (см. [2]) для поиска пути с минимальным весом на накрывающем полиэдре. При этом в качестве начальной точки на $\hat{P}$ выбирается пара $(s, 0)$, а в качестве конечной – $(t, J(x_0))$.

5. Вычисление проекции найденного пути на P.

Переход к накрывающему полиэдру в данном алгоритме позволяет свести рассматриваемую задачу к уже решённой задаче поиска пути с минимальным весом на графе. Однако при этом не используется информация о соответствии между вершинами исходного полиэдра и накрывающего. Это приводит к тому, что пути на накрывающем полиэдре, имеющие один образ при накрывающем отображении, минимизируются независимо друг от друга, что приводит к значительному увеличению времени работы алгоритма. Поэтому предлагается модификация алгоритма, свободная от указанного недостатка.

Заметим также, что накрывающий полиэдр $\hat{P}$ может быть построен и с помощью гомоморфизма $J : C_1(P) \to \mathbf{Z}_2^r$, не являющегося индексной вектор-функцией. В этом случае применение шагов 3–5 указанного алгоритма позволяет найти минимальный элемент среди всех путей $x \in C_1(P)$ с концами s и t, удовлетворяющих равенству $J(x) = J(x_0)$.

Метод эквивалентных дуг

Множество вершин полиэдра триангулированного многообразия P обозначим буквой V, множество рёбер – E. Пусть, как и выше, L – неотрицательная весовая функция, J – индексная вектор-функция относительно некоторого базиса группы $H_1(P)$, а граф $G = (V, E)$ представляет собой одномерный остов полиэдра P.

Определение 1. Ребро $e \in E$ назовём проиндексированным, если $J(e) \neq 0$. Обозначим множество проиндексированных рёбер буквой I.

Определение 2. Назовём вершину $v \in V$ специальной, если она совпадает с начальной или конечной вершиной минимизируемого пути или инцидентна как минимум одному индексированному ребру. Множество всех специальных вершин обозначим как $S \subset V$.

Определение 3. Назовём путь нуль-индексным, если он начинается и кончается в специальных вершинах u и v и не содержит ни одного проиндексированного ребра. Множество всех нуль-индексных путей, соединяющих вершины u и v графа G, обозначим N_{uv}.

Если $N_{uv} \neq \varnothing$, то множество всех путей из N_{uv}, имеющих минимальный вес, обозначим как $\widetilde{N}_{uv}$. Назовём элементы этого множества минимальными нуль-индексными путями. Выберем и зафиксируем путь $x_{uv} \in \widetilde{N}_{uv}$.

Построим мультиграф $G' = (V', E')$ с множеством вершин $V' = S$. Пусть $u, v \in V'$. Если существует ребро $e = [u, v] \in I$, то будем считать, что $e \in E'$. Кроме того, при $N_{uv} \neq \varnothing$ включим в E' путь $x_{uv} \in \widetilde{N}_{uv}$.

Замечание 1. По построению каждую пару вершин $u, v \in S$ могут соединять не более чем две дуги графа $G' = (V', E')$, так как в графе G может быть не более чем одно проиндексированное ребро, соединяющее u и v, а множество нуль-индексных путей $\widetilde{N}_{uv}$ заменяется не более чем на одну дугу. Следовательно,

$$|E'| \leq \frac{K^2}{2} \cdot 2 = K^2,$$

где $K = |S|$.

Обозначим символом $\Omega(G, S)$ множество путей графа G, соединяющих точки множества S, а символом $\Omega(G')$ – множество всех путей мультиграфа G'. По построению $E' \subset \Omega(G, S) \subset$ $\subset C_1(P)$. Следовательно, определено включение $i' : E' \to C_1(P)$, причем $i'(E') \subset \Omega(G, S)$.

Произвольная цепь $x' \in C_1(G')$ представляет собой формальную сумму $x' = \sum_k e'_k$ дуг $e'_k \in E'$. Положим

$$f(x') = f(\sum_k e'_k) = \sum_k i'(e'_k). \qquad (2)$$

Этим построен гомоморфизм $f : C_1(G') \to$ $\to C_1(P)$, удовлетворяющий условию $f(\Omega(G')) \subset$ $\subset \Omega(G, S)$. С его помощью посредством формул $L' = L \circ f$ и $J' = J \circ f$ определим весовую функцию $L' : \Omega(G') \to \mathbf{R}$ и гомоморфизм $J' : C_1(G') \to \mathbf{Z}_2^r$. Значения последнего на одномерных цепях мультиграфа $G' = (V', E')$ договоримся по-прежнему называть их индексами.

Рассмотрим вершины $u, v \in S$ и вектор $j \in \mathbf{Z}_2^r$. Обозначим символами $\Omega(G, u, v, j)$ и $\Omega(G', u, v, j)$ множества путей графов G и G' соответственно, соединяющих вершины u и v и имеющих индекс j. Согласно построению функции J' имеет место включение

$$f(\Omega(G', u, v, j)) \subset \Omega(G, u, v, j). \qquad (3)$$

Также введем обозначения $M(G, u, v, j)$ и $M(G', u, v, j)$ для множеств путей с минимальным весом из $\Omega(G, u, v, j)$ и $\Omega(G', u, v, j)$ соответственно.

Теорема 1. Имеет место включение $f(M(G',u,v,j)) \subset M(G,u,v,j)$.

Доказательство. Произвольный путь $x \in \Omega(G,S)$ представляет собой сумму $x=[v_0v_1]+[v_1v_2]+\ldots+[v_{n-1}v_n]$, где $v_i \in V$, $[v_{i-1}v_i] \in E$ для всех $i=1,\ldots,n$, а $v_0, v_n \in S$. Допустим, что путь $x^*=[v_p v_{p+1}]+\ldots+[v_{q-1}v_q]$ составлен из максимального набора идущих подряд ребер пути x, не лежащих в I. Тогда $J(x^*)=J([v_p v_{p+1}])+\ldots+J([v_{q-1}v_q])=0$. Если $p=0$, то $v_p=v_0 \in S$ по выбору пути x. При $p>0$ ребро $[v_{p-1}v_p]$ пути x принадлежит множеству I в силу максимальности x^*. Но тогда $v_p \in S$ по определению множества специальных вершин. Аналогично проверяется, что и $v_q \in S$. Таким образом, $x^* \in N_{v_p v_q}$.

Итак, доказано, что любой путь $x \in \Omega(G,S)$ может быть представлен в виде суммы $x=\sum_k x_k$, где либо $x_k=e \in I$, либо $x_k \in N_{uv}$ для некоторых $u,v \in S$. Положим

$$F(x)=F(\sum_k x_k)=\sum_k F(x_k), \qquad (4)$$

где $F(x_k)=e$ в первой из указанных ситуаций и $F(x_k)=x_{uv}$ – во второй, причем x_{uv} – зафиксированный ранее путь из множества $\widetilde{N}_{uv}$. Этим определено отображение $F:\Omega(G,S)\to\Omega(G')$.

По построению $F(x_k)=x_{uv}$ для всех $\widetilde{N}_{uv}$. Таким образом, $J\circ f\circ F=J$ и потому $J'\circ F=J$. Отсюда следует, что

$$F(\Omega(G,u,v,j)) \subset \Omega(G',u,v,j). \qquad (5)$$

Заметим также, что $L(f(F(x_k)))=L(x_k)$ при $x_k=e\in I$ и $L(f(F(x_k)))\le L(x_k)$ при $x_k \in N_{uv}$ для некоторых $u,v\in S$. Поэтому для любого $x=\Omega(G,S)$

$$L((f\circ F)(x)) \le L(x). \qquad (6)$$

Рассмотрим далее путь $x \in M(G,u,v,j)$ и положим $x'=F(x)$ и $y=f(x')$. Согласно (5) $x'\in\Omega(G',u,v,j)$, а в силу (3) $y\in\Omega(G,u,v,j)$. Из (6) также следует, что $L(y)\le L(x)$. Таким образом, вес пути y не превышает минимального веса пути из множества $\Omega(G,u,v,j)$. А это значит, что $y\in M(G,u,v,j)$.

Пусть, наконец, z' – произвольный элемент множества $M(G',u,v,j)$. Так как z' имеет минимальный вес среди всех путей из $\Omega(G',u,v,j)$, а построенный выше путь $x'=F(x)$ принадлежит $\Omega(G',u,v,j)$, то $L'(z')\le L'(x')$. Положим $z=f(z')$. Тогда $L(z')=L(f(z'))=L'(z')\le L'(x')=L(f(x'))=L(y)$. Кроме того, согласно (3) $z\in\Omega(G,u,v,j)$. Отсюда и из включения $y\in M(G,u,v,j)$ следует, что $z\in M(G,u,v,j)$. Теорема доказана.

Следствие. Пусть $x_{G'}$ – минимальный путь индекса $J'(x_{G'})=J(x_0)$ в графе G', соединяющий вершины s и t. Тогда его образ $x_G=f(x_{G'})$ гомологичен x_0 в P и имеет наименьший вес среди таких путей.

Действительно, по условию $x_{G'}\in M(G', s,t,J(x_0))$. Отсюда по теореме 1 следует, что путь $x_G=f(x_{G'})$ имеет наименьший вес среди всех путей x графа $G=(V,E)$, соединяющих вершины s и t, имеющих индекс $J(x)=J(x_0)$. Согласно предложению 1 для всех таких путей $[x]=[x_0]$ в $H_1(P)$.

Определение 4. Пусть V' – множество вершин графа G', E' – множество его рёбер, а $J':C_1(G')\to\mathbf{Z}_2^r$ – построенный выше гомоморфизм. Положим $\hat V=V'\times\mathbf{Z}_2^r$. Элементы $\hat u=(u,\xi)\in\hat V$ и $\hat v=(v,\eta)\in\hat V$ назовём соседними и включим ребро $[\hat u\hat v]$ в список $\hat E$, если выполнены условия:

(A1) существует ребро $e=[uv]\in E'$;

(A2) $\eta=\xi+J'(e)$.

Граф, состоящий из множества вершин $\hat V$ и множества ребер $\hat E$, обозначим как $\hat G'$.

Определение 5. Определим отображение $\pi:\hat G'\to G'$ следующим образом: $\pi_V((u,\xi))=u$, $\pi_E([\hat u\hat v])=[\pi_V(\hat u)\pi_V(\hat v)]$, $\pi=(\pi_V,\pi_E)$ для u, $v\in V$, $[\hat u\hat v]\in\hat E$.

Предложение 2. Для любого пути $x=[u_0u_1]+[u_1u_2]+\ldots+[u_{p-1}u_p]$ графа G' и произвольного вектора $\xi_0\in\mathbf{Z}_2^r$ существует единственный путь $\hat x=[\hat u_0\hat u_1]+[\hat u_1\hat u_2]+\ldots+[\hat u_{p-1}\hat u_p]$ графа $\hat G'$, обладающий свойствами:

(C1) $\hat u_0=(u_0,\xi_0)$;

(C2) $\pi(\hat x)=x$.

Предложение 3. Пусть $x=[u_0u_1]+[u_1u_2]+\ldots+[u_{p-1}u_p]$ и $y=[v_0v_1]+[v_1v_2]+\ldots+[v_{q-1}v_q]$ – пути графа G', идущие из вершины $u_0=v_0$ в вершину $u_p = v_q$, $\hat{x} = [\hat{u}_0\hat{u}_1] + [\hat{u}_1\hat{u}_2] + \ldots + [\hat{u}_{p-1}\hat{u}_p]$ и $\hat{y} = [\hat{v}_0\hat{v}_1] + [\hat{v}_1\hat{v}_2] + \ldots + [\hat{v}_{q-1}\hat{v}_q]$ – пути в $\hat{G}'$, накрывающие пути x и y соответственно и имеющие общее начало $\hat{u}_0 = \hat{v}_0$. Тогда $\hat{u}_p = \hat{v}_q$ в том и только в том случае, если $J'(x) = J'(y)$.

Предложения 2 и 3 доказываются аналогично предложениям 10.2 и 10.3 из [3] соответственно.

Предложения 2 и 3 позволяют использовать граф $\hat{G}'$ для поиска минимального элемента среди путей x' графа G', соединяющих вершины s и t и имеющих индекс $J'(x') = J(x_0)$. В результате мы получаем новый алгоритм для решения поставленной задачи.

Алгоритм 2

1. Нахождение базиса $[y_1],\ldots,[y_r]$ группы гомологий $H_1(P)$.
2. Вычисление индексной вектор-функции $J : C_1(P) \to \mathbf{Z}_2^r$ относительно базиса $[y_1],\ldots,[y_r]$.
3. Построение мультиграфа $G' = (V', E')$.
4. Построение с помощью гомоморфизмов $f : C_1(G') \to C_1(P)$ и $J' = J \circ f$ накрытия $\pi : \hat{G}' \to G'$.
5. Поиск пути $\hat{x}'$ с минимальным весом $\hat{L}'(\hat{x}')$, $\hat{L}' = L \circ f \circ \pi$, на накрывающем графе $\hat{G}'$, соединяющий вершины $(s,0)$ и $(t, J(x_0))$.
6. Вычисление проекции $x' = \pi(\hat{x}')$ найденного пути на G'.
7. Восстановление соответствующего x' пути $x = f(x')$ на G.

Новыми в данном алгоритме являются шаги 3 и 7. Наиболее существенна задача построения графа G', потому распишем метод ее решения подробнее.

Первый шаг реализуется процедурой *build_new_graph*(*s*,*t*), в ней передаются начальная (*s*) и конечная (*t*) вершины пути. Затем производится вычисление минимальных значений весовой функции L для всех пар специальных вершин графа G, при этом начальная вершина s минимизируемого пути рассматривается первой.

Внутри процедуры используется очередь *EQ*, в которую помещаются очередные специальные вершины.

```
procedure build_new_graph(s, t)
    ENQUEUE(EQ, s)
    G' ← G' ∪ {s}
    repeat
        u ← POP(EQ)
        make_new_graph_edges(EQ, u, t)
    until EQ ≠ ∅
```

Функция *make_new_graph_edges*(*EQ*, *u*, *t*) для заданной вершины *u* находит минимальные нуль-индексные пути до всех специальных вершин (в том числе конечной вершины *t*). Для их вычисления используется алгоритм Дейкстры (см. [2]) с запретом пересечения индексированных рёбер. Граф считается заданным списком смежности *Adj*.

В случае, если во время работы алгоритма попалась ещё не рассмотренная специальная вершина v, она добавляется в очередь *EQ* на проверку. Граф G' пополняется новой вершиной и ребром, эквивалентным минимальному нуль-индексному пути между *u* и *v*.

Для специальной вершины, инцидентной индексированному ребру, производятся следующие действия: вершина w, инцидентная тому же ребру, добавляется в V'; индексированное ребро добавляется в E'.

Заметим, что начальная вершина в *make_new_graph_edges*(*EQ*, *u*, *t*) не рассматривается как специальная. Так как граф G не ориентированный и начальная вершина ставится в очередь первой, это не приводит к потере дуг в G', инцидентных ей.

В массиве *visited* хранятся булевские значения, обозначающие посещённые вершины, массив *d* хранит текущее минимальное значение весовой функции на пути до вершины *v*, массив *from* используется для сохранения предыдущей вершины в графе предшествования. Флаг *visited* выставляется, если вершина *w* является специальной (согласно определению 2). Функция *process_ null_index_path* вызывается для сохранения нуль-индексного пути, эквива-лентного новой дуге. Также в функциях *make_ new_graph_edges* и *process_null_index_path* заполняется таблица значений для функций *J'* и *F*.

```
procedure make_new_graph_edges (EQ, u, t)
    for each v ∈ G
    visited[v] ← false
    d[v] ← ∞
    ENQUEUE(Q, u, 0)
    d[u] ← 0
    from[u] ← u
    while Q ≠ ∅
    v ← POP(Q)
    if visited[v] then
    continue
    special ← v = t
```

for each $w \in Adj[v]$
$e \leftarrow [v,w]$
$edge\_weight \leftarrow L(v,w)$
$path\_weight \leftarrow d[v] + edge\_weight$
$idx \leftarrow J(e)$
if $idx \neq 0$ then
$special \leftarrow true$
if $w \notin V'$ then
$V' \leftarrow V' \cup \{w\}$
ENQUEUE(EQ, w)
if $e \notin E'$ then
$E' \leftarrow E' \cup \{[v,w]\}$
$J'(e) \leftarrow idx$
$F(e) \leftarrow \{e\}$
else
if $path\_weight < d[w]$ then
ENQUEUE(Q, w, $path\_weight$)
$d[w] \leftarrow path\_weight$
$from[w] \leftarrow v$
if $special$ then
$process\_zero\_index\_path\ (EQ, u, v, from)$

Процедура $process\_zero\_index\_path\ (EQ, u, v)$ добавляет в граф G' дугу, эквивалентную кратчайшему нуль-индексному пути между u и v. Кратчайший путь берётся из графа предшествования, сохранённого в массиве *from*. Если вершина v ещё не содержится в G', то она будет добавлена в него.

procedure process_zero_index_path (EQ, u, v, from)
if $v \notin V'$ then
$V' \leftarrow V' \cup \{v\}$
ENQUEUE(EQ, v)
$path \leftarrow \varnothing$
if $u \neq v$ then
$t \leftarrow v$
while $from[t] \neq t$ do
$path[t] \leftarrow path[t] \cup [from[t], t]$
$t \leftarrow from[t]$
$E' \leftarrow E' \cup \{[u, v]\}$
$J'([u,v]) \leftarrow 0$
$F([u,v]) \leftarrow path$

Шаги 1, 2 и 4–6 аналогичны соответствующим процедурам алгоритма 1. На шаге 7 применяется формула (2).

Оценка сложности

Введём обозначения: $N = \max(N_0, N_1, N_2)$, где N_i – количество симплексов размерности i в исходном полиэдре; $K = |S|$. При построении графа G' для каждой вершины $u \in S$ используется алгоритм Дейкстры для графа G с N_0 вершинами и N_1 рёбрами, сложность которого $O(N_0^2 + N_1)$. Следовательно, шаг 3 алгоритма имеет сложность

$$O\big(K(N_0^2 + N_1)\big) = O(KN^2), \qquad (7)$$

так как $N_0, N_1 \leq N$.

Для построения накрывающего графа $\hat{G}'$ и последующего поиска на нём минимального пути используется граф G', содержащий $|V'| = K$ вершин. К графу $\hat{G}'$ вновь применяется алгоритм Дейкстры. Этот алгоритм имеет сложность

$$O(|\hat{V}|^2 + |\hat{E}|) = O((K2^r)^2 + |E'|2^r) = O(K^2 2^{2r}), \qquad (8)$$

так как $|\hat{V}| = |V'|2^r$, $|\hat{E}| = |E'|2^r$ по построению $\hat{G}'$ и $|E'| \leq K^2$ по замечанию 1.

Вычисление базиса группы $H_1(P)$ и индексация относительно него могут быть выполнены с помощью алгоритмов, разработанных в [3] и [4]. Первый из них имеет сложность $O(Nr)$, а второй – $O(N \log N)$. Остальные шаги в силу их простоты на окончательную оценку сложности алгоритма повлиять не могут.

Отсюда, из (7) и (8) получаем сложность $O(Nr + N\log N + KN^2 + K^2 2^{2r}) = O(KN^2 + K^2 2^{2r})$. Заметим, что если применять алгоритм Дейкстры к графу $\hat{G}$, накрывающему G, как это делается в алгоритме 2 из [1], то сложность составит $O(N^2 2^{2r})$. Поэтому существенный выигрыш от применения нового варианта алгоритма может быть получен в ситуации, когда количество специальных вершин $K = |S|$ намного меньше параметра $N = \max(N_0, N_1, N_2)$.

Работа выполнена при финансовой поддержке Минобрнауки РФ в рамках государственного задания на оказание услуг в 2012–2014 гг. подведомственными высшими учебными заведениями (шифр заявки 1.1907.2011) и федеральной целевой программы «Научные и научно-педагогические кадры инновационной России», контракт №14.B37.21.0361.

Список литературы

1. Lapteva A.V. and Yakovlev E.I. Index Vector-Function and Minimal Cycles // Lobachevskii Journal of Mathematics. 2006. V. 22. P. 35–46.

2. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 2001. 960 с.

3. Яковлев Е.И. Вычислительная топология. Н. Новгород: Изд-во ННГУ, 2005. 214 с.

4. Lapteva A.V., Yakovlev E.I. Minimal 1-Cycles Generating a Canonical Basis of 2-Manifold's Homology Group // International Journal of Pure and Applied Mathematics. 2006. V. 31. № 4. P. 555–570.

THE METHOD OF EQUIVALENT EDGES FOR PATH MINIMIZATION IN A GIVEN HOMOLOGY CLASS

A.V. Galanin

Closed triangulated manifolds, their edge paths and 2-module homology groups are considered. A new method has been developed to reduce complexity for the shortest path algorithm which is homologous to the given one.

Keywords: simplex, polyhedron, homology group, algorithm, minimization.

References

1. Lapteva A.V. and Yakovlev E.I. Index Vector-Function and Minimal Cycles // Lobachevskii Journal of Mathematics. 2006. V. 22. P. 35–46.

2. Kormen T., Lejzerson Ch., Rivest R. Algo-ritmy: postroenie i analiz. M.: MCNMO, 2001. 960 s.

3. Jakovlev E.I. Vychislitel'naja topologija. N. Novgorod: Izd-vo NNGU, 2005. 214 s.

4. Lapteva A.V., Yakovlev E.I. Minimal 1-Cycles Generating a Canonical Basis of 2-Manifold's Homology Group // International Journal of Pure and Applied Mathematics. 2006. V. 31. № 4. P. 555–570.