

## АЛГОРИТМЫ И ТЕХНОЛОГИИ ПОСТРОЕНИЯ РЕШЕНИЙ РАЗРЕЖЕННЫХ НЕДООПРЕДЕЛЁННЫХ ЛИНЕЙНЫХ СИСТЕМ В WOLFRAM MATHEMATICA

Л.А. Пилипчук, А.А. Лагуто

*Рассматриваются алгоритмы и технологии построения решений разреженных недоопределенных систем линейных алгебраических уравнений и их реализация в системе Wolfram Mathematica. Применяются структуры данных для построения базиса пространства решений системы с учетом разреженности матрицы системы и теоретико-графовых свойств базисов. Приводятся оценки сложности алгоритмов построения характеристических векторов и частного решения линейной системы.*

Рассмотрим разреженную недоопределенную систему линейных алгебраических уравнений вида:

$$\sum_{j \in I_i^+(U)} x_{ij} - \sum_{j \in I_i^-(U)} x_{ji} = b_i, \quad i \in I, \quad (1)$$

где  $S = (I, U)$  – конечный ориентированный связный граф без кратных дуг и петель,  $I$  и  $U$  – соответственно множество узлов и множество дуг графа  $S$ , определенных на прямом произведении  $I \times I$ ,  $|I| < \infty$ ,  $|U| < \infty$ ,  $I_i^+(U) = \{j : (i, j) \in U\}$ ,  $I_i^-(U) = \{j : (j, i) \in U\}$ ,  $b = (b_i, i \in I)$  – вектор правой части системы (1). Предположим, что для системы (1) выполняются условия:  $m < n$ , где  $m = |I|$ ,  $n = |U|$  и  $\sum_{i \in I} b_i = 0$ . Согласно [1] ранг матрицы системы (1) равен  $|I| - 1$ . Опорой графа  $S = (I, U)$  для системы (1) является покрывающее дерево графа  $S$  [2]. Обозначим через  $U_T \subset U$  – множество дуг покрывающего дерева графа  $S$ ,  $U_N = U \setminus U_T$ .

Пусть  $\tilde{x} = (\tilde{x}_{ij}, (i, j) \in U)$  – некоторое частное решение системы (1), построенное по правилам [1]. Тогда  $\tilde{x}_{ij} = 0$ ,  $(i, j) \in U \setminus U_T$  и общее решение системы (1) имеет вид:

$$x_{ij} = \sum_{(\tau, \rho) \in U \setminus U_T} x_{\tau\rho} \delta_{ij}^{\tau\rho} + \tilde{x}_{ij}, \quad (i, j) \in U_T. \quad (2)$$

Компоненты характеристического вектора  $\delta(\tau, \rho) = (\delta_{ij}^{\tau\rho}, (i, j) \in U)$ , порожденного дугой  $(\tau, \rho) \in U \setminus U_T$ , относительно покрывающего дерева  $(I, U_T)$  вычисляются из системы:

$$\sum_{j \in I_i^+(U_T \cup (\tau, \rho))} \delta_{ij}^{\tau\rho} - \sum_{j \in I_i^-(U_T \cup (\tau, \rho))} \delta_{ji}^{\tau\rho} = 0, \quad i \in I.$$

Согласно [1] компонента характеристического вектора  $\delta_{ij}^{\tau\rho}$  – это знак дуги  $(i, j)$  в цикле  $L(\tau, \rho)$ , порожденном дугой  $(\tau, \rho) \in U \setminus U_T$ ,  $(i, j) \in U$  и  $\delta_{ij}^{\tau\rho} = 0$ ,

$(i, j) \in U \setminus (U_T \cup (\tau, \rho))$ .

Для определения компонент характеристического вектора, соответствующих множеству дуг покрывающего дерева  $U_T$  применяется технология представления покрывающего дерева  $(I, U_T)$  в виде корневого дерева  $(I, \tilde{U}_T)$  [3], что позволяет установить биективное отображение между множеством узлов  $I$  (за исключением одного узла) и множеством дуг  $U_T$ . Технология представления корневого дерева основана на применении следующих структур:

- $pred = \{pred[i], i \in I\}$  – список предков узлов (биективное отображение).

Список определяет для каждого узла  $i$  значение предка узла  $pred[i]$  в корневом дереве, значение  $pred[i] = 0$  соответствует корню дерева.

- $dir = \{dir[i], i \in I\}$  – список направлений дуг в соответствии с биективным отображением. Список определяет для каждого узла  $i$  направление дуги  $(pred[i], i)$  корневого дерева в соответствии с дугой покрывающего дерева:  $dir[i] = 1$ , если дуга  $(pred[i], i) \in U_T$ ,  $dir[i] = -1$ , если дуга  $(i, pred[i]) \in U_T$  и  $dir[i] = 0$ , если узел  $i$  является корнем.

- $d = \{d[i], i \in I\}$  – список династического обхода корневого дерева  $(I, \tilde{U}_T)$ , которое соответствует покрывающему дереву графа  $S$ . Инвертированный список династического обхода определяет порядок решения уравнений системы линейных алгебраических уравнений (1) для построения частного решения  $\tilde{x}$ .

- $depth = \{depth[i], i \in I\}$  – список уровней узлов. Список определяет для каждого узла  $i$  длину (в дугах) от данного узла до корня корневого дерева. Если узел  $i$  является корнем, то  $depth[i] = 0$ .

Вычисление ненулевых компонент характеристического вектора  $\delta(\tau, \rho) = (\delta_{ij}^{\tau\rho}, (i, j) \in U)$ , порожденного дугой  $(\tau, \rho) \in U \setminus U_T$  относительно покрывающего дерева  $(I, U_T)$  графа  $S$ , основано на алгоритме построения базисного цикла, порождаемого дугой  $(\tau, \rho)$  [4, 5]. Для узлов  $\tau$  и  $\rho$  проверяем уровни  $depth[\tau]$  и  $depth[\rho]$ . В случае неравенства, пусть для определенности  $depth[\tau] > depth[\rho]$ , в корневом дереве  $(I, \tilde{U}_T)$  поднимаемся по единственной цепи от узла  $\tau$  к корню дерева, пока не будет достигнут некоторый узел  $v$ , для которого выполняется равенство  $depth[v] = depth[\rho]$ . Другими словами, строится следующая последовательность узлов:  $\tau, pred[\tau], pred[pred[\tau]], \dots, v$ . Далее одновременно строятся последовательности узлов в корневом дереве  $(I, \tilde{U}_T)$  от узлов  $\rho$  и  $v$ :

$$\rho, pred[\rho], pred[pred[\rho]], \dots, t,$$

$$v, pred[v], pred[pred[v]], \dots, t,$$

пока не будет достигнут общий узел  $t$ . Таким образом, будет определен базисный цикл. Одновременно с построением цикла вычисляются ненулевые

компоненты характеристического вектора  $\delta(\tau, \rho)$ , соответствующие дугам построенного цикла.

Приведем реализацию в системе *Wolfram Mathematica* процедуры *Determining* построения ненулевых компонент характеристического вектора  $\delta(\tau, \rho) = (\delta_{ij}^{\tau\rho}, (i, j) \in U)$ , порожденного дугой  $(\tau, \rho) \in U \setminus U_T$ , относительно покрывающего дерева  $(I, U_T)$  за  $O(m)$  операций в наихудшем случае. Входными параметрами процедуры являются номера узлов  $k$  и  $l$  дуги  $(k, l) \in U \setminus U_T$ , порождающей базисный цикл, и списковые структуры *pred*, *dir*, *depth*. Компоненты характеристического вектора, порожденного дугой  $(k, l)$ , сохраняются в списке *deltaC*, дуги цикла, образованного дугой  $(k, l)$  – в списке *Ucycle*.

```
Determining[k_, l_, pred_, dir_, depth_, Ucycle_] := Module[{deltaC = {}},
  Ucycle = List[];
  i = k;
  j = l;
  While[i != j,
    If[depth[[i]] > depth[[j]],
      {
        If[dir[[i]] == 1,
          AppendTo[deltaC, Subscript[x, pred[[i]] \[Mapsto] i] \[Rule] 1];
          AppendTo[Ucycle, DirectedEdge[pred[[i]], i]],

          AppendTo[deltaC, Subscript[x, i \[Mapsto] pred[[i]]] \[Rule] -1];
          AppendTo[Ucycle, DirectedEdge[i, pred[[i]]]]
        ];
        i = pred[[i]];
      },
    If[depth[[j]] > depth[[i]],
      {
        If[dir[[j]] == 1,
          AppendTo[deltaC, Subscript[x, pred[[j]] \[Mapsto] j] \[Rule] -1];
          AppendTo[Ucycle, DirectedEdge[pred[[j]], j]],

          AppendTo[deltaC, Subscript[x, j \[Mapsto] pred[[j]]] \[Rule] 1];
          AppendTo[Ucycle, DirectedEdge[j, pred[[j]]]]
        ];
        j = pred[[j]];
      },
    {
      If[dir[[i]] == 1,
        AppendTo[deltaC, Subscript[x, pred[[i]] \[Mapsto] i] \[Rule] 1];
        AppendTo[Ucycle, DirectedEdge[pred[[i]], i]],

        AppendTo[deltaC, Subscript[x, i \[Mapsto] pred[[i]]] \[Rule] -1];
        AppendTo[Ucycle, DirectedEdge[i, pred[[i]]]];
      ];
      If[dir[[j]] == 1,
        AppendTo[deltaC, Subscript[x, pred[[j]] \[Mapsto] j] \[Rule] -1];
      ];
    }
  ];
```

```

AppendTo[Ucycle,DirectedEdge[pred[[j]],j]],

AppendTo[deltaC,Subscript[x,j->pred[[j]]->1];
AppendTo[Ucycle,DirectedEdge[j,pred[[j]]]];
];
i=pred[[i]];
j=pred[[j]];
}
];
];
Return[deltaC];
]

```

Характеристические векторы  $\delta(\tau, \rho)$ , порожденные дугами  $(\tau, \rho) \in U \setminus U_T$  относительно покрывающего дерева  $(I, U_T)$ , строятся процедурой *ComputeCharacteristicVectors*. Входными параметрами процедуры являются списки  $lUn$ ,  $lUt$  множеств дуг  $U_N$  и  $U_T$  соответственно. В результате работы процедуры создается список всех характеристических векторов, каждый из которых получен процедурой *Determining*.

```

ComputeCharacteristicVectors[lUn_, lUt_] := Module[{cv = {}},
dN0 = Subscript[x, #1] -> 0 & /@ lUn;
For[i = 1, i <= Length[lUn], i++,
arc = lUn[[i]];
dN = dN0; dN[[i]] = Subscript[x, arc] -> 1;
dTc = Determining[arc[[1]], arc[[2]], pred, dir, depth, Ucycle];
listUt0 = Complement[lUt, Ucycle];
Clear[Ucycle];
dT0 = Subscript[x, #1] -> 0 & /@ listUt0;
dGeneral = Join[dN, Sort[Join[dTc, dT0]]];
AppendTo[cv, {dGeneral}];
];
Return[cv];
];
delta = Flatten[ComputeCharacteristicVectors[listUn, listUt], 1]
deltaT = Transpose[delta][[n-m+2;]];

```

Алгоритм построения компонент вектора  $\tilde{x} = (\tilde{x}_{ij}, (i, j) \in U)$  – частного решения системы (1), соответствующих дугам покрывающего дерева  $(I, U_T)$ , основан на применении инвертированного списка династического обхода  $d = \{d[i], i \in I\}$  корневого дерева  $(I, \tilde{U}_T)$ . Инвертированный список  $d$  определяет порядок решения уравнений системы (1) таким образом, что каждое уравнение содержит только одну неизвестную компоненту вектора  $\tilde{x}$ . Компоненты  $\tilde{x}_{\tau\rho}$  вектора  $\tilde{x}$ , соответствующие дугам  $(\tau, \rho) \in U \setminus U_T$ , полагаем равными нулю. Число операций указанного алгоритма пропорционально числу дуг покрывающего дерева  $(I, U_T)$ .

Приведем реализацию в системе *Wolfram Mathematica* процедуры *ComputeParticularSolution* построения компонент вектора  $\tilde{x}$  частного решения

системы (1). Входными параметрами процедуры являются список  $lUn$  дуг множества  $U_N$ , вектор правой части  $b = (b_i, i \in I)$  системы (1) и списковые структуры  $pred$ ,  $d$ ,  $dir$ .

```

ComputeParticularSolution[ $lUn$ _,  $b$ _,  $pred$ _,  $d$ _,  $dir$ _]:=Module[{},
   $xpN0$ =Subscript[Overscript[ $x$ , ~], #1]→0&/@ $lUn$ ;
   $xpUt$ =List[];
   $xp$ =ConstantArray[0, $m$ ];
  For[ $k$ = $m$ , $k$ >1, $k$ --,
     $i$ =dinast[[ $k$ ]];
     $xp[[i]]$ +=- $dir[[i]]b[[i]]$ ;
     $xp[[pred[[i]]]]$ += $dir[[i]]dir[[pred[[i]]]]xp[[i]]$ ;
  ];
  For[ $i$ =1, $i$ <= $m$ , $i$ ++,
     $d$ = $dir[[i]]$ ;
    If[ $d$ ==0,Continue[]];
    If[ $d$ ==1, $arc$ = $pred[[i]]$ ↦ $i$ ];
    If[ $d$ ==-1, $arc$ = $i$ ↦ $pred[[i]]$ ];
    AppendTo[ $xpUt$ ,Subscript[Overscript[ $x$ , ~], $arc$ ]→ $xp[[i]]$ ];
  ];
   $xpGeneral$ =Join[ $xpN0$ ,Sort[ $xpUt$ ]];
  Return[ $xpGeneral$ ];
];

```

Для построения покрывающего дерева  $(I, U_T)$  и корневого дерева  $(I, \tilde{U}_T)$  используется встроенная функция *DepthFirstScan* системы *Wolfram Mathematica*. Данная функция задает обход в глубину графа  $S$  и позволяет одновременно построить следующие списковые структуры представления корневого дерева с корнем в заданном узле:  $pred = \{pred[i], i \in I\}$ ,  $depth = \{depth[i], i \in I\}$ ,  $d = \{d[i], i \in I\}$ . Список  $dir = \{dir[i], i \in I\}$  направлений дуг определяется в соответствии с биективным отображением:  $dir[i] = 1$ , если дуга  $(pred[i], i) \in U_T$ ,  $dir[i] = -1$ , если дуга  $(i, pred[i]) \in U_T$  и  $dir[i] = 0$ , если узел  $i$  является корнем.

Приведем реализацию в системе *Wolfram Mathematica* процедуры *BuildSpanningTreeForGraph* для построения покрывающего и корневого деревьев и определения списковых структур:  $pred$ ,  $depth$ ,  $d$ ,  $dir$ . Входным параметром процедуры является исходный граф  $S$ .

```

BuildSpanningTreeForGraph [ $g$ _]:=Module[{ $s$ ={},  $root$ =tRoot},
   $pred$ =ConstantArray[0, $m$ ];
   $depth$ =ConstantArray[0, $m$ ];
   $dir$ =ConstantArray[0, $m$ ];
   $d$ ={};

  DepthFirstScan[UndirectedGraph[ $g$ ],  $root$ ,
    {"FrontierEdge" → Function[ $e$ , {AppendTo[ $s$ ,  $e[[1]]$ ] ↦  $e[[2]]$ ],
     $pred[[e[[2]]]]$ = $e[[1]]$ ,
     $depth[[e[[2]]]]$ =1+ $depth[[e[[1]]]]$  }},
    "PrevisitVertex"→Function[ $u$ ,AppendTo[ $d$ , $u$ ]]}

```

```

];
s=Reverse[s];
listUt={};
For[k=1,k<=Length[s],k++,
  arc=s[[k]];
  If[MemberQ[listU,arc],
    {dir[[arc[[2]]]]=1;AppendTo[listUt,arc]},
    {dir[[arc[[2]]]=-1;AppendTo[listUt,Reverse[arc]]}
  ];
];
Return[s];
];

```

Общее решение разреженной недоопределенной системы (1) строится согласно формуле (2) в виде суммы общего решения однородной системы, порожденной системой (1), и некоторого частного решения системы (1), построенного по правилам [1]. Приведем реализацию в системе *Wolfram Mathematica* построения общего решения *xgeneral* системы (1), где *xpart* – построенное частное решение. Общим решением однородной системы, порожденной системой (1), является линейная комбинация характеристических векторов [1].

```

xpart=ComputeParticularSolution[listUn, b, pred, d, dir];
xgeneral=Table[Subscript[x,listUt[[j]]]→(xpart[[n-m+1+j]][[2]]+Sum[deltaT[[j,i]][[2]]
Subscript[x,listUn[[i]]],{i,1,n-m+1}},{j,m-1}];

```

Тестирование приведенных алгоритмов было проведено на небольшом числе входных файлов, в которых задавались графы различной размерности. Тестовые данные были сгенерированы отдельной программой, использующей генератор случайных чисел [6].

Пусть *t1* – среднее время работы описанного алгоритма построения общего решения разреженной недоопределенной системы (1), *t2* – среднее время работы алгоритма решения системы (1) для заданной опоры с использованием встроенной функции *Solve* системы *Wolfram Mathematica*. Среднее время работы указанных алгоритмов для графов различной размерности приведено в таблице 1.

Таблица 1

Среднее время работы построения решений разреженной недоопределенной системы

Время работы алгоритма (сек)	Размерность графа $S = (I, U)$		
	$ I  = 100,  U  = 300$	$ I  = 150,  U  = 400$	$ I  = 200,  U  = 500$
<i>t1</i>	0.377	0.83	1.848
<i>t2</i>	36.74	68.99	142.12
<i>t2 / t1</i>	97.46	83.12	76.9

Рассмотрим численный пример построения общего решения разреженной недоопределенной системы линейных алгебраических уравнений следующего вида:

$$\begin{cases} x_{1,3} + x_{1,5} - x_{2,1} - x_{6,1} = 7, \\ x_{2,1} + x_{2,6} - x_{5,2} = 4, \\ -x_{1,3} = -1, \\ -x_{6,4} = -7, \\ x_{5,2} + x_{5,6} - x_{1,5} = -2, \\ x_{6,1} + x_{6,4} - x_{2,6} - x_{5,6} = -1, \end{cases} \quad (3)$$

где  $S = (I, U)$  (рисунок 1),  $I = \{1, 2, 3, 4, 5, 6\}$ ,  $U = \{(1,3), (1,5), (2,1), (2,6), (5,2), (5,6), (6,1), (6,4)\}$ .

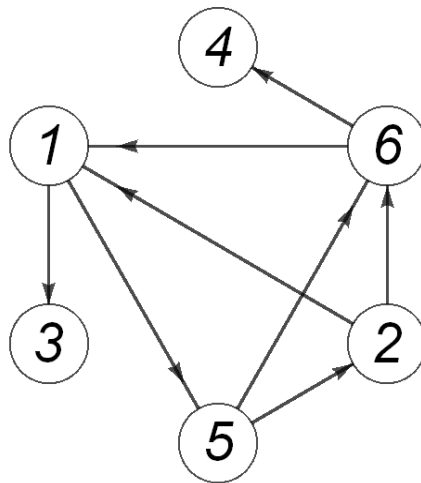


Рис.1. Граф  $S = (I, U)$

На рисунке 2 представлено покрывающее дерево  $(I, U_T)$  графа  $S$  (слева), где  $U_T = \{(1,3), (1,5), (2,1), (5,6), (6,4)\}$ , и соответствующее корневое дерево  $(I, \tilde{U}_T)$  с корнем в узле 6 (справа),  $U_N = U \setminus U_T = \{(2,6), (5,2), (6,1)\}$ .

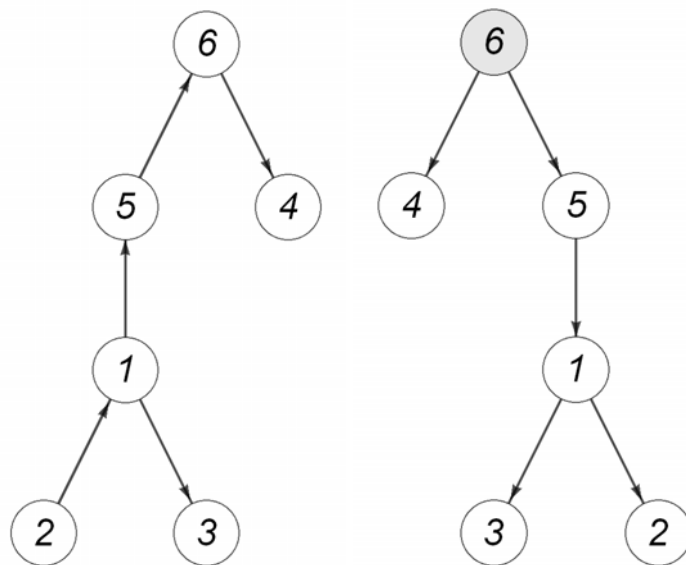


Рис.2. Покрывающее дерево  $(I, U_T)$  графа  $S = (I, U)$  (слева), и соответствующее корневое дерево  $(I, \tilde{U}_T)$  с корнем в узле 6 (справа)

В таблице 2 приведены структуры представления корневого дерева, изображенного на рисунке 2.

Таблица 2

Структуры представления корневого дерева

Структуры представления корневого дерева	Номер узла					
	1	2	3	4	5	6
$pred[i]$	5	1	1	6	6	0
$depth[i]$	2	3	3	1	1	0
$dir[i]$	-1	-1	1	1	-1	0
$d[i]$	2	3	4	6	1	5

В таблице 3 приведены характеристические векторы, порожденные дугами множества  $U_N = \{(2,6), (5,2), (6,1)\}$ .

Таблица 3

Характеристические векторы, порожденные дугами множества  $U_N$

Характеристические векторы	Компоненты характеристических векторов, соответствующие дугам графа							
	(2,6)	(5,2)	(6,1)	(1,3)	(1,5)	(2,1)	(5,6)	(6,4)
$\delta(2,6)$	1	0	0	0	-1	-1	-1	0
$\delta(5,2)$	0	1	0	0	1	1	0	0
$\delta(6,1)$	0	0	1	0	1	0	1	0

В таблице 4 представлены компоненты вектора  $\tilde{x}$  частного решения системы (3). Порядок решения уравнений системы (3) определяет инвертированный список династического обхода  $\{4,3,2,1,5,6\}$ .

Таблица 4

Частное решение системы (3)

Частное решение	Компоненты вектора частного решения $\tilde{x}$ , соответствующие дугам графа							
	(2,6)	(5,2)	(6,1)	(1,3)	(1,5)	(2,1)	(5,6)	(6,4)
$\tilde{x}_{ij}$	0	0	0	1	10	4	8	7

Таким образом, общее решение системы (3) имеет следующий вид:

$$x_{1,3} = 1,$$

$$x_{1,5} = 10 - x_{2,6} + x_{5,2} + x_{6,1},$$

$$x_{2,1} = 4 - x_{2,6} + x_{5,2},$$

$$x_{5,6} = 8 - x_{2,6} + x_{6,1},$$

$$x_{6,4} = 7.$$

Таким образом, на основании результатов исследования теоретико-графовых свойств базиса пространства решений разреженной недоопределенной системы (1) и применения алгоритмов и технологий работы с корневыми деревьями, получен эффективный алгоритм построения общего решения (2) системы (1).



## Список литературы

1. Pilipchuk, L.A. Sparse Linear Systems and Their Applications. / L.A. Pilipchuk.– Minsk: BSU, 2013. – 235 p.
2. Габасов, Р. Методы линейного программирования: в 3 ч. / Р. Габасов, Ф.М. Кирилова. – Минск: БГУ, 1980. – Ч. 3: Специальные задачи. – 368 с.
3. Пилипчук, Л.А. Применение конструктивных методов декомпозиции для решения одной нелинейной задачи сетевой оптимизации / Л.А. Пилипчук. // Веснік Гродзенскага дзяржаўнага ўніверсітэта імя Янкі Купалы. Серыя 2. Матэматыка. Фізіка. Інфарматыка, вылічальная тэхніка і кіраванне. – 2015. – №2 (192). С. 54-61.
4. Йенсен, П. Потокное программирование / П. Йенсен, Д. Барнес. – М.: Радио и связь, 1984. – 392 с.
5. Ahuja, R.K. Network Flows: Theory, Algorithms, and Applications / R.K. Ahuja, T.L.Magnanti, J.B. Orlin. – New Jersey: Prentice Hall, 1993. – 864 p.
6. Таранчук, В.Б. Основы программирования на языке Wolfram: учеб. материалы для студентов фак. прикладной математики и информатики спец. 1-31 03 04 «Информатика» / В.Б. Таранчук. – Минск: БГУ, 2015. – 49 с.

*Пилипчук Людмила Андреевна, доцент кафедры компьютерных технологий и систем факультета прикладной математики и информатики Белорусского государственного университета, кандидат физико-математических наук, доцент, pilipchuk@bsu.by*

*Лагута Анна Андреевна, ассистент кафедры компьютерных технологий и систем факультета прикладной математики и информатики Белорусского государственного университета, lagutoa@yahoo.com*