

УДК 519.1

МЕТОД ВИЗУАЛИЗАЦИИ МЕТАГРАФА

Е.С. Штогрин^а, А.С.Кривенко^а

^а Национальный технический университет Украины «Киевский политехнический институт», Институт телекоммуникационных систем, Киев, Украина, L_Shtogrina@mail.ru

Проведен анализ методов визуализации разных типов графов. Сделан вывод об актуальности разработки метода визуализации метаграфов. Описываются проблемы размещения узлов, принадлежащих и не принадлежащих метавершинам, пересечения фигур метавершин, которые возникают при решении задачи визуализации метаграфа. Данные проблемы рассмотрены на примере. Определены критерии, по которым полученное изображение считается наглядным для пользователя и однозначно соответствует заданному метаграфу. Поставлена задача визуализации метаграфа. Для ее решения предложен метод визуализации, основанный на принципах силовых алгоритмов. В рамках данного метода определены правила действия сил между узлами. Эти силы зависят от типа узлов, между которыми действуют, и наличия ребер между ними. Применение предложенного метода позволяет строить графическое представление метаграфа, который соответствует заданной предметной области. Метод не предполагает участия человека при формировании изображения, что позволяет экономить время при частых изменениях, которые требуют перестройки изображения. Полученное изображение позволяет визуализировать сущности предметной области и связи между ними, применить методы графического анализа. Определены дальнейшие направления работы – минимизация количества пересечений ребер и площади, занимаемой результирующим изображением. Модификация метода визуализации метаграфа с учетом данных направлений позволит улучшить визуальное представление метаграфа.

Ключевые слова: метаграф, визуализация, граф, метавершина, графическое представление.

METAGRAPH VISUALIZATION METHOD

O.S. Shtogrina^а, O.S. Kryvenko^а

^а National Technical University of Ukraine «Kyiv Polytechnic Institute», Institute of Telecommunication Systems, Kyiv, Ukraine, L_Shtogrina@mail.ru

Analysis of visualization techniques for various types of graphs is done. Conclusion about the relevance of method development for metagraph visualizing is made. The paper deals with issues that arise when solving the problem of metagraph visualization, such as: the placement of nodes that belong to metaverices and do not belong to them, figures intersection of metaverices. These problems are exemplified. Criteria are proposed for the final image to be considered coherent, understandable for users and corresponding to a predetermined metagraph. The problem of metagraph visualization is posed. The method based on the principles of force algorithms is proposed for the problem solving. The rules for forces between metagraph nodes are defined in the framework of the method. These forces depend on nodes type between which they are performed and the presence of edges between them. This method does not involve human intervention during the formation of the image, thus saving time at frequent changes that require rebuilding of the image. The resulting image gives the possibility to portray clearly the entities of the subject area and the links between them. Graphical analysis methods can be applied to the image. Areas for the future researches are identified. They are: minimization of the number of edges intersections and the area occupied by the resulting image. Results of the modified visualization method can improve the visual metagraph representation.

Keywords: metagraph, visualization, graph, metavertex, graphical representation.

Введение

Графы являются естественным и наглядным способом представления сложных структур. Это позволяет широко использовать их в задачах коммуникации и транспортировки, в задачах поиска маршрутов и создания карт, в системах обработки информации для параллельных и распределенных вычислений, в системах поддержки принятия решений, в системах обработки транзакций и др. Теория графов применяется при разработке эффективных алгоритмов оптимизации и наглядном представлении предметных областей для их дальнейшего анализа. Представление в виде простого графа позволяет рассматривать отдельные элементы информации и их связи между собой. Однако во многих случаях недостаточно представления и изучения взаимодействия отдельных элементов. Рассмотрение взаимосвязанных множеств элементов и взаимодействия между группами элементов может предоставить качественно новый взгляд на задачу и ее решение. Например, может возникнуть необходимость вместе рассматривать несколько переменных в моделях систем поддержки принятия решений, несколько логических переменных в правилах баз знаний, несколько документов в системах документооборота. Существуют структуры, которые позволяют частично решить эту проблему – это гиперграфы и hi-графы [1], но они имеют свои недостатки. В работе [2] для решения описанной выше проблемы предложена такая разновидность графов, как метаграфы, и рассмотрено применение метаграфов для формализации логического вывода, моделирова-

ния функционирования устройств, моделирования бизнес-процессов с использованием диаграмм выполнения работ (workflow).

Визуализация метаграфа, сформированного для решения одной из вышеперечисленных задач, предоставит возможность графического анализа задачи. Это, в свою очередь, позволит находить противоречия и неточности в модели предметной области, представленной метаграфом, которые трудно выявить, используя текстовое или формальное представление. Большое количество элементов предметной области, а также ее частое изменение приводят к необходимости частого обновления изображения метаграфа, что затруднительно реализовывать вручную. Вследствие этого задача автоматического построения визуального представления метаграфа является актуальной.

Проблема визуализации метаграфа

На сегодняшний день развитие получили методы генерации изображений графов и графовых структур как малых (до 200 вершин), так и больших размеров (тысячи вершин). Классические силовые алгоритмы основаны на физических аналогиях и могут быть использованы для построения графов произвольного вида. Изображения, созданные с их помощью, содержат мало пересечений ребер и симметрию [3]. К этой группе относятся «пружинные алгоритмы» [4, 5], алгоритмы, имитирующие действие сил гравитации [6] и магнитных сил, а также алгоритмы, основанные на минимизации энергии [7]. Для визуализации графов больших размеров существуют многоуровневые алгоритмы [8, 9]. Они требуют решения NP-полных задач, задач кластеризации графа, GC-фильтрации графа [10]. В работах [11, 12] рассматриваются методы визуализации гиперграфов для отображения данных больших объемов. Данные подходы эффективны для графов, но не подходят для метаграфа, так как метаграф имеет отличную от графа структуру.

Наиболее полное и подробное описание метаграфа дано в работе [2], где метаграф был определен как двойка: порождающее множество и множество ребер. В работе [13] в силу специфики поставленной задачи, по аналогии к теории гиперграфов, к определению метаграфа были добавлены еще две составляющие: множество метавершин и множество метаребер. Так как для решения задачи визуализации метаграфа важны типы узлов и их соотношения, целесообразно расширить определение, данное в [2], и выделить отдельно множество вершин и множество метавершин метаграфа. Выделять и отдельно рассматривать метаребра нет необходимости, так как при визуализации метаграфа важным является наличие ребер, но не их тип. В связи с этим множество ребер будет содержать все ребра метаграфа, независимо от того, какие типы узлов метаграфа они соединяют.

Приведем определение метаграфа, которое учитывает описанные выше особенности и будет использоваться при изложении последующего материала. Метаграф – это тройка множеств $S = \langle V, M, E \rangle$,

где $V = \{v_r | r = \overline{1, N_V}\}$ – порождающее множество (множество вершин метаграфа), N_V – количество вершин метаграфа; $M = \{m_q | q = \overline{1, N_M}\}$ – множество метавершин, N_M – количество метавершин;

$E = \{e_h | h = \overline{1, N_E}\}$ – множество ребер, N_E – общее количество ребер в метаграфе.

Метавершина метаграфа – это вершина, которая включает множество вершин $m_q = \{v_r | v_r \in V, r = \overline{1, N_{m_q}}\}$, где N_{m_q} – количество вершин, входящих в метавершину m_q .

Для упрощения дальнейшей подачи материала введем понятие узла метаграфа $mv \in (V \cup M)$, таким образом, это вершина или метавершина. Тогда ребро метаграфа определяется как $e_h = (mv_{out}, mv_{in})$, где mv_{out} – узел, из которого выходит ребро, mv_{in} – узел, в который входит ребро.

Так как алгоритмы визуализации графов не учитывают структуру метаграфа, то в них нет механизмов учета вложенности вершин в метавершины.

Рассмотрим подробнее проблемы, которые могут возникнуть, если для нахождения позиций вершин метаграфа использовать алгоритм визуализации графов. В этом случае можно получить неправильное графическое представление метаграфа, которое не соответствует заданному метаграфу.

Пусть дан метаграф (рис. 1)

$$S_1 = \left\langle \begin{array}{l} \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}, \\ \{m_1, m_2, m_3, m_4\}, \\ \{e_1, e_2, e_3, e_4, e_5\} \end{array} \right\rangle,$$

где $m_1 = \{v_6, v_7\}$, $m_2 = \{v_3, v_4\}$, $m_3 = \{v_3, v_5\}$, $m_4 = \{v_1, v_2, v_3\}$, $e_1 = (m_1, v_9)$, $e_2 = (m_2, v_9)$, $e_3 = (v_4, v_7)$, $e_4 = (v_8, m_4)$, $e_5 = (v_5, v_{10})$.

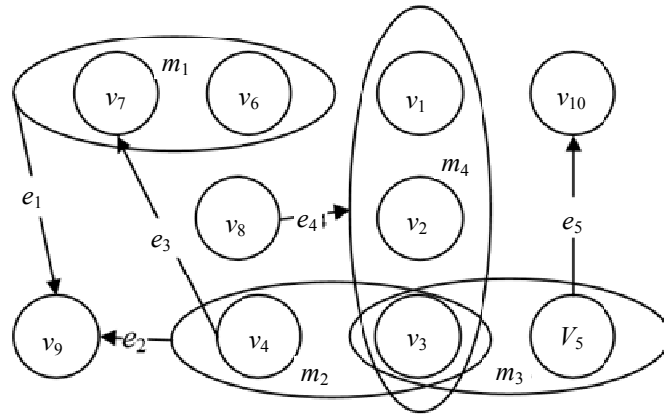


Рис. 1. Пример правильного графического представления метаграфа \$S_1\$

Если алгоритм визуализации графов применить только к множеству вершин, то не будут рассчитываться собственно положения метавершин, а также учитываться ребра, инцидентные метавершинам. Метавершины в таком случае можно сформировать за счет обводки принадлежащих им вершин. После такой визуализации метавершин может возникнуть проблема размещения в метавершине узлов, не принадлежащих ей. На рис. 2 показан пример такого размещения. Узлы, расположенные неправильно, выделены темным цветом, \$v_1 \notin m_1\$, \$v_5 \notin m_4\$, \$m_3 \notin m_4\$, но на рисунке видно, что это не так.

В случае применения алгоритма визуализации графов к множеству вершин и метавершин вершины, принадлежащие метавершине, будут расположены вне метавершины за счет того, что их положение будет вычисляться независимо друг от друга.

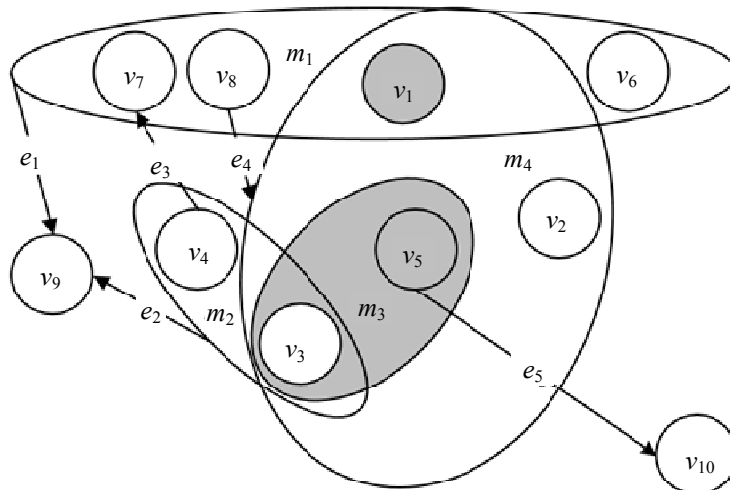


Рис. 2. Пример неправильного графического представления метаграфа \$S_1\$

Критерии правильной визуализации метаграфа

Каждая вершина \$v_r\$ и метавершина \$m_q\$ имеет позицию – координаты на плоскости \$\mathbf{p}_{v_r} = (x_{v_r}, y_{v_r})\$, \$\mathbf{p}_{m_q} = (x_{m_q}, y_{m_q})\$. Вектора \$\mathbf{P}_V = (\mathbf{p}_{v_1}, \mathbf{p}_{v_2}, \dots, \mathbf{p}_{v_{N_V}})\$ – вектор позиций вершин, \$\mathbf{P}_M = (\mathbf{p}_{m_1}, \mathbf{p}_{m_2}, \dots, \mathbf{p}_{m_{N_M}})\$ – вектор позиций метавершин, полностью определяют расположение узлов для визуализации. Введем \$\mathbf{p}_{mv_i}\$ – обозначение для координаты узла, который может быть вершиной, если \$mv_i \in V\$, или метавершиной, если \$mv_i \in M\$. Расстояние между узлами рассчитывается как евклидово расстояние между точками: \$l_{e_h} = \|\mathbf{p}_{mv_i} - \mathbf{p}_{mv_j}\|\$.

Определим «графическое представление метавершины» на плоскости как геометрическую фигуру \$F_{m_q}\$, которая имеет позицию \$\mathbf{p}_{m_q}\$ и форму. Позиция метавершины на конечном изображении метаграфа

– центр ее фигуры. «Графическое представление ребра» определим как кривую F_{e_h} , которая задана между координатами начала и конца ребра. Тогда «графическое представление метаграфа» будет определяться тройкой $W_S = \langle P_V, F_M, F_E \rangle$, где $F_M = \{F_{m_q}\}$ – множество фигур метавершин, $F_E = \{F_{e_h}\}$ – множество кривых, определяющих ребра.

Графическое представление будет правильным, т.е. будет соответствовать заданному метаграфу, если отображение $S \rightarrow W_S$ изоморфно. Определим критерии, при выполнении которых графическое представление метаграфа будет правильным:

1. Координаты вершин не равны: $\forall i, \forall j: i \neq j \Rightarrow p_{v_i} \neq p_{v_j}$, причем координаты метавершин могут быть равны при наличии общих внутренних вершин.
2. Только координаты вершин, принадлежащих метавершине m_q , находятся в F_{m_q} :

$$\forall v_r: v_r \in m_q \Rightarrow p_{v_r} \in F_{m_q}, \forall v_r: v_r \notin m_q \Rightarrow p_{v_r} \notin F_{m_q}.$$

3. Фигуры метавершин, не имеющих общих внутренних вершин, не пересекаются: $\forall j, \forall k: m_j \cap m_k = \emptyset \Rightarrow F_{m_j} \cap F_{m_k} = \emptyset$.

Основные требования к эстетичности графического представления графа – минимальное количество пересечений ребер, приблизительно равная длина ребер и др. – описаны в работах [3–5]. Эти требования также справедливы и для метаграфа. Но для метаграфа более важно правильное расположение метавершин и их форма, чем длина ребер.

Метод визуализации метаграфа

Опишем постановку задачи нахождения правильного графического представления метаграфа.

Дано:

1. Метаграф $S = \langle V, M, E \rangle$.
2. P_V, P_M – начальное расположение узлов.
3. Критерии правильной визуализации метаграфа.
4. Прямоугольная область U , в которой должно размещаться изображение метаграфа.

Найти: Графическое представление метаграфа.

Для решения поставленной задачи необходимо определить:

1. расположение вершин, не входящих в метавершины;
2. расположение метавершин с учетом взаимного расположения метавершин, которые имеют общие внутренние вершины;
3. взаимное расположение вершин в метавершине;
4. форму фигур метавершин;
5. кривые F_{e_h} , определяющие ребра.

Предложенный метод визуализации метаграфа основан на алгоритме Фрюхтермана–Рейнгольда для визуализации ненаправленных графов [5]. Граф рассматривается как система объектов, соединенных пружинами по определенным правилам. Каждая из пружин действует на соединенные ею вершины с силой притяжения или отталкивания. Под действием пружин система движется до тех пор, пока не установится равновесие – сумма сил, действующих на узел, равна нулю. Гарантируется, что узлы размещаются внутри U – прямоугольной области экрана и (или) бумаги. С каждой итерацией алгоритма узлы двигаются на расстояние $\delta(t)$ в направлении вектора суммарной силы, действующей на узел. Функция $\delta(t)$ зависит от времени и убывает. Если вектор смещения выталкивает узел за границу области U , соответствующая координата вектора смещения обрывается, а сила продолжает действовать по касательной к границе области U .

Для визуализации метаграфа предлагается модифицировать алгоритм Фрюхтермана–Рейнгольда следующим образом: рассмотреть все возможные пары узлов и ввести дополнительные коэффициенты балансировки сил. Тогда силы будут вычисляться по следующим формулам:

– сила отталкивания узла u от узла v :

$$f_{\text{rep}_{uv}} = Kr_{uv} \frac{l^2}{\|p_{mv_u} - p_{mv_v}\|} p_{0_{vu}} \quad (1)$$

при $u = v$, $f_{\text{rep}_{uv}} = 0$;

– сила притяжения узла u к узлу v :

$$\mathbf{f}_{attr_{uv}} = Ka_{uv} \frac{\|p_{mv_u} - p_{mv_v}\|^2}{l} \mathbf{p}_{0_{uv}} \quad (2)$$

при $u = v$, $\mathbf{f}_{attr_{uv}} = 0$, где l – идеальная длина ребра для метаграфа – вычисляется как функция от площади размещения и количества узлов:

$$l = \sqrt{\frac{area(U)}{N_V + N_M}}, \quad (3)$$

$\mathbf{p}_{0_{vu}}$ – единичный вектор направления из \mathbf{p}_{mv_v} в \mathbf{p}_{mv_u} .

Коэффициенты Kr_{uv} и Ka_{uv} зависят от типа каждого узла в паре и их соотношения. Силы отталкивания действуют между каждой парой узлов с коэффициентом Kr_{uv} . Между парами «метавершина и входящая в нее вершина» $Kr_{uv} = 0$. Коэффициент при силе отталкивания со стороны метавершины зависит от количества ее внутренних вершин. Тогда метавершина с большим количеством внутренних вершин N_{mq} будет отталкивать другие узлы сильнее, чем метавершина с меньшим количеством N_{mq} . Коэффициент при силе отталкивания со стороны вершины равен 1.

Силы притяжения действуют между смежными узлами с коэффициентом Ka_{uv} , что гарантирует расположение узлов ближе друг к другу при наличии ребра между ними. Сила притяжения со стороны метавершины зависит от количества внутренних вершин, как и при вычислении сил отталкивания. Между метавершинами и вложенными в них вершинами сила притяжения действует с коэффициентом $Ka_{uv} > 1$. Эта сила обеспечивает расположение вложенных вершин внутри фигуры метавершины и их движение за метавершиной при ее перемещении под действием сил со стороны остальных узлов. Сила притяжения между всеми вложенными в метавершину вершинами действует с коэффициентом $Ka_{uv} > 1$, за счет чего обеспечивается меньшая площадь фигуры метавершины. Сила притяжения для пар вершина-вершина действует с коэффициентом 1 для обеих вершин.

Тогда общая сила, действующая на узел u , будет рассчитываться по следующей формуле:

$$\mathbf{F}_{spring_u} = \sum_v \mathbf{f}_{rep_{uv}} + \sum_v \mathbf{f}_{attr_{uv}}. \quad (4)$$

Блок-схема алгоритма решения задачи визуализации метаграфа представлена на рис. 3.

Применение данного метода позволяет найти такие координаты узлов, при которых графическое представление метаграфа будет правильным. Фигуру метавершины можно принять за эллипс. Однако для уменьшения занимаемой площади, избегания пересечений фигур метавершин и повышения читаемости изображения для метавершин с тремя и более внутренними вершинами ее можно определять как минимальную выпуклую оболочку множества позиций внутренних вершин. Для нахождения этой оболочки можно использовать алгоритмы Джарвиса, Грэхема или Чана [14].

В самом простом случае ребра можно задать как прямые, соединяющие фигуры узлов. Однако при таком изображении возникают проблемы множественных пересечений ребер и пересечений фигур метавершин ребром. Исходя из этого, необходимо рассчитывать форму ребра, которое будет огибать узлы, как кривую Безье или В-сплайн [15, 16].

Заключение

В работе затронуты основные проблемы визуализации метаграфа, предложены критерии правильного графического представления метаграфа, а также предложен метод визуализации метаграфа, основанный на принципах силовых алгоритмов. Изображение метаграфа, полученное с помощью предложенного метода, удовлетворяет вышеуказанным критериям, т.е. однозначно соответствует заданному метаграфу.

Использование предложенного метода визуализации метаграфа позволяет применять методы графического анализа в задачах, в которых используются метаграфы, без затрат времени на ручное построение изображений. В дальнейшем планируется доработка метода для определения формы ребра, минимизации количества пересечений ребер и занимаемой площади.

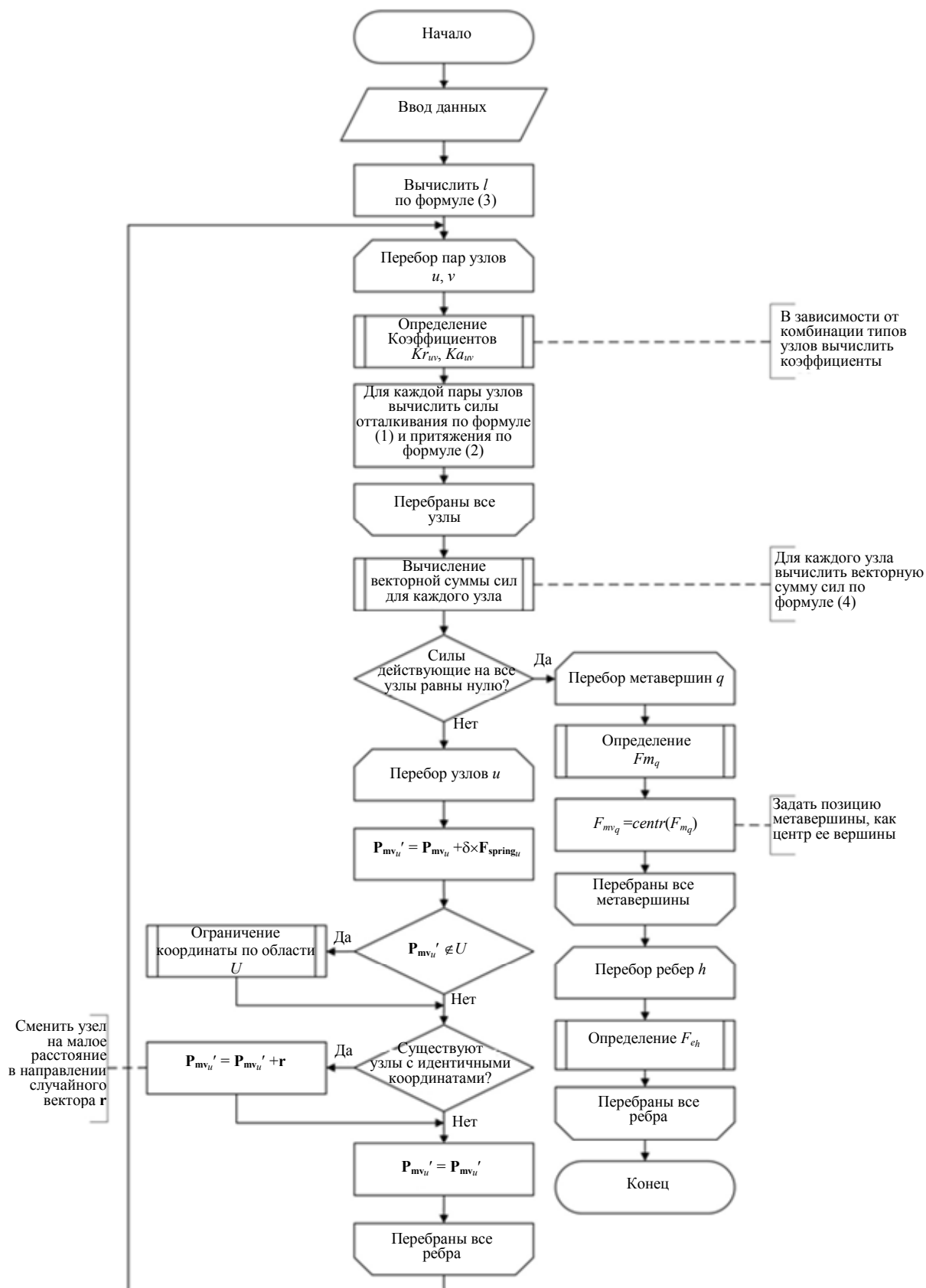


Рис. 3. Блок-схема алгоритма решения задачи визуализации метаграфа

Reference

1. Grossman O., Harel D. *On the algorithmics of higraphs*. Technical Report CS97-15, Rehovot, Israel, 1997, 32 p.
2. Basu A., Blanning R.W. *Metagraphs and their applications*. NY: Springer, 2006, 172 p.
3. Eades P., Lin X. Spring algorithms and symmetry. *Theoretical Computer Science*, 2000, vol. 240, no. 2, pp. 379–405.
4. Eades P. A heuristic for graph drawing. *Congressus Nutnerantiunt*, 1984, no. 42, pp. 149–160.
5. Fruchterman T.M.J., Reingold E.M. Graph drawing by force-directed placement. *Software-Practice and Experience*, 1991, vol. 21, no. 11, pp. 1129–1164.
6. Barnes J., Hut P. A hierarchical O (N log N) force-calculation algorithm. *Nature*, 1986, vol. 324, no. 4, pp. 446–449. doi: 10.1038/324446a0
7. Kamada T., Kawai S. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 1989, vol. 31, no. 1, pp. 7–15.
8. Hachul S., Junger M. Drawing large graphs with a potential-field-based multilevel algorithm. *Lecture Notes in Computer Science*, 2004, vol. 3383, pp. 285–295.
9. Archambault D., Munzner T., Auber D. TopoLayout: Multilevel graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics*, 2007, vol. 13, no. 2, pp. 305–317. doi: 10.1109/TVCG.2007.46
10. Harel D., Koren Y. A fast multi-scale method for drawing large graphs. *Journal of Graph Algorithms and Applications*, 2002, vol. 6, no. 3, pp. 177–202.
11. Jin R., Xiang Y., Fuhry D., Dragan F.F. Overlapping matrix pattern visualization: A hypergraph approach. *Proceedings - IEEE International Conference on Data Mining, ICDM*. Pisa, Italy, 2008, art. no. 4781126, pp. 313–322. doi: 10.1109/ICDM.2008.102
12. Paquette J., Tokuyasu T. Hypergraph visualization and enrichment statistics: How the EGAN paradigm facilitates organic discovery from big data. *Proc. SPIE - The International Society for Optical Engineering*, 2011, vol. 7865, art. no. 78650E. doi: 10.1117/12.890220
13. Astanin S.V., Dragnish N.V., Zhukovsky N.K. Vlozhennye metagrafy kak modeli slozhnykh ob"ektov [Nested metagraphs as models of complex objects]. *Inženernyj vestnik Dona*, 2012, vol. 23, no. 4-2 (23), p. 76.
14. Cormen T.H., Leiserson C.E., Rivest R., Stein C. *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw Hill, 2001, 1202 p.
15. Rogers D.F., Adams J.A. *Mathematical Elements for Computer Graphics*, 2nd ed. NY, McGraw-Hill, 1990, 611 p.
16. Holten D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 2006, vol. 12, no. 5, pp. 741–748. doi: 10.1109/TVCG.2006.147

- Штогриня Елена Сергеевна** – ассистент, Национальный технический университет Украины «Киевский политехнический институт», Институт телекоммуникационных систем, Киев, Украина, L_Shtogrina@mail.ru
- Кривенко Александра Сергеевна** – студент, Национальный технический университет Украины «Киевский политехнический институт», Институт телекоммуникационных систем, Киев, Украина, alexakryvenko@gmail.com
- Olena S. Shtogrina** – assistant, National Technical University of Ukraine «Kyiv Polytechnic Institute», Institute of Telecommunication Systems, Kyiv, Ukraine, L_Shtogrina@mail.ru
- Oleksandra S. Kryvenko** – student, National Technical University of Ukraine «Kyiv Polytechnic Institute», Institute of Telecommunication Systems, Kyiv, Ukraine, alexakryvenko@gmail.com

Принято к печати 12.02.14

Accepted 12.02.14