

ТЕХНОЛОГИИ КОГНИТИВНОЙ ВИЗУАЛИЗАЦИИ ТЕМПОРАЛЬНЫХ КОМПЛЕКСНЫХ СЕТЕЙ

К. Д. МУХИНА, К. О. БОЧЕНИНА, А. С. КАРСАКОВ, А. В. БУХАНОВСКИЙ

*Университет ИТМО, 197101, Санкт-Петербург, Россия
E-mail: mukhinaks@gmail.com*

Представлено описание методов визуализации темпоральных комплексных сетей. Рассмотрены основные методические и алгоритмические особенности процесса визуализации темпоральных сетей и динамических процессов. Приведены примеры использования описанных технологий.

Ключевые слова: *темпоральная комплексная сеть, динамический процесс, визуализация графа, когнитивная визуализация*

Введение. Комплексные сети (КС) являются математическим формализмом, используемым для описания различных сложных систем реального мира, таких как Интернет [1], системы цитирования [2], сложные биохимические соединения [3, 4]. При этом по своей природе КС могут быть как статическими, так и динамическими, когда новые объекты и связи в сети создаются, изменяются и исчезают с течением времени вследствие различных протекающих в них процессов, к которым относятся, например, распространение информации в социальных сетях [5], распространение компьютерных вирусов [6], распространение сигналов в нейронных сетях [7] и пр. Такие сети называются *темпоральными* (temporal).

Информационная модель КС основывается на многомерной и многосвязной структуре данных, для анализа и интерпретации которой целесообразно, наравне с формальными методами, использовать технологии когнитивной визуализации. За последние два десятилетия было предпринято несколько попыток создать универсальное приложение, позволяющее визуализировать сети различных размеров и типов. Удачными примерами являются программы Pajek [8], Tulip [9], GeoMI [10], GraphViz [11, 12] и Gephi [13]. Несмотря на то что эти приложения в том или ином виде могут быть использованы для визуализации темпоральных КС, ни одно из них не предназначено для интерактивной визуализации динамических процессов в режиме реального времени. Это связано с тем, что для когнитивной визуализации сетей нельзя использовать экстенсивные расширения известных статических решений — для этого необходимы динамические подходы, позволяющие учитывать особенности структуры самой сети и специфику протекающих в ней процессов.

Методические особенности визуализации темпоральных сетей. Общая модель КС представляется в виде графа $G = G(V, E)$, определяемого как совокупность множества вершин $V, V \neq \emptyset$, и множества ребер $E = (u, v), u, v \in V$. Темпоральная КС, в свою очередь, может быть описана как набор состояний — графов $\mathcal{G} = \{G_0 = (V_0, E_0), G_1 = (V_1, E_1), \dots, G_n = (V_n, E_n)\}$, где G_n — граф, соответствующий моменту времени t [14]. Сам процесс эволюции комплексной сети из одного состояния в другое может быть описан с помощью оператора эволюции Γ :

$$(V, E, f)_{t+1} = \Gamma(V, E, f)_t, (V, E, f)_{t=0} = (V_0, E_0, f_0),$$

где $f: V \rightarrow \mathbb{N}$ — отображение, которое ставит в соответствие каждой вершине ее вес.

Важной характеристикой КС в динамике является темпоральный путь длиной l из вершины u в вершину v ; он определяется как последовательность вершин $\{u = v_0, v_1, \dots, v_l = v\}$, в которой $(v_{i-1}, v_i) \in E$ и $a_{ij}(t) = 1$, где $t_1 \leq t_2 \leq \dots \leq t_l$ для всех $i = 1, \dots, l$, здесь i, j — индексы вершин. Коэффициент $a_{ij}(t)$ — индекс смежности (функция присутствия), который определяется следующим образом:

$$a_{ij}(t) = \begin{cases} 1, & \text{если } (v_i, v_j) \in E_t, \\ 0 & \text{— иначе.} \end{cases}$$

При анализе процессов в темпоральных сетях важным является выделение структурных особенностей, к которым относятся динамические циклы, волны, каскады, параллельные процессы и кластеры.

Динамические циклы. Цикл (рис. 1, а) представляет собой замкнутую фигуру последовательно связанных между собой узлов (обозначенных на рисунке цифрами), соответственно цикл может быть представлен как путь из вершины u в вершину v , где $u = v$. В сети может быть предусмотрена возможность циклической передачи сообщения (сигнала) между узлами, тем не менее в конкретный момент времени сам цикл может отсутствовать, что усложняет его обнаружение при отдельных статических состояниях сети.

Волна. Волновой паттерн представляет собой последовательное во времени изменение некоторой характеристики элементов сети (рис. 1, б). Волновой паттерн может быть описан следующим образом: пусть $V_i \subset V_t$ — набор вершин, смежных v_i и существующих в момент времени t , тогда $\Gamma: f_t(v) = f_{t-1}(v_i)$ для любой вершины $v \in V_i$. Подобного вида паттерн сложно обнаружить с помощью формальных методов, тогда как визуально они легко отслеживаются. Присутствие волнового паттерна в сети говорит о наличии зависимости некоторой характеристики узла от соседних узлов, причем изменение в одном узле приводит к такому же изменению соседних некоторое время спустя.

Каскад. Массовое одинаковое изменение характеристик узлов (рис. 1, в) может быть признаком некоторого критического события. В данном случае паттерн может быть представлен в следующем виде: $V_{\text{cascade}} \subset V_t$ — некоторый набор вершин, существующих в момент времени t , тогда для любой вершины $v \in V_{\text{cascade}}$: $\Gamma: f_t(v) = m$. Присутствие каскада в сети может означать, что рассматриваемая сеть является синтетической. Подобное явление может возникать на начальном этапе массового распространения информации, например, в социальных сетях [15].

Не менее важным аспектом визуализации сетевых структур является непосредственная визуализация процессов, происходящих в сетях. Визуальный анализ динамических сетевых структур позволяет не только подтвердить или опровергнуть статистические характеристики, полученные с помощью математических вычислений, но и сфокусироваться на конкретных узлах или связях в сети, выявить их свойства и особенности, которые могут оказать существенное влияние на дальнейшие исследования.

Параллельные процессы. Процесс в сети может разветвляться (передача информации) или даже проходить параллельно (рис. 1, г). Формально процесс образования сети в операторном виде может быть представлен как набор вершин $V_i \subset V_t$, смежных v_i и существующих в момент времени t , тогда $\Gamma: f_t(v) = f_t(v_i)$ для любой вершины $v \in V_i$. В случае парал-

тельного процесса можно выделить подграфы G_t , для каждого из которых будет действовать свой оператор Γ .

Динамические кластеры. Динамические кластеры в сети (рис. 1, д) отражают во времени скопления узлов сети с идентичными свойствами и могут определяться через характеристику модулярности КС [16], расширенную на временную область:

$$Q = \frac{1}{2\mu} \sum_{ijsr} \left[\left(a_{ij}(s) - \gamma_s \frac{k_i(s)k_j(s)}{2M_s} \right) \delta_{sr} + C_{jsr} \delta_{ij} \right] \delta(g_{is}, g_{jr}),$$

где $2\mu = M_s + \sum_{ijsr} C_{jsr}$, $M_s = \frac{1}{2} \sum_i k_i(s)$ — общее число ребер в фиксированный момент времени s ; $C_{jsr} = 1$, если существует ребро, соединяющее узел с индексом j в момент s с таким же узлом в момент r ; $a_{ij}(s)$ — индекс смежности для момента s ; $k_i(s)$, $k_j(s)$ — степени узлов с индексами i и j ; γ_s — параметр разрешения; δ_{sr} и δ_{ij} — дельта-функции Кронекера; $\delta(g_{is}, g_{jr}) = 1$, если i -й узел в момент s и j -й узел в момент r принадлежат к одному сообществу.

Характеристика Q может быть использована далее для кластеризации стандартными методами [17].

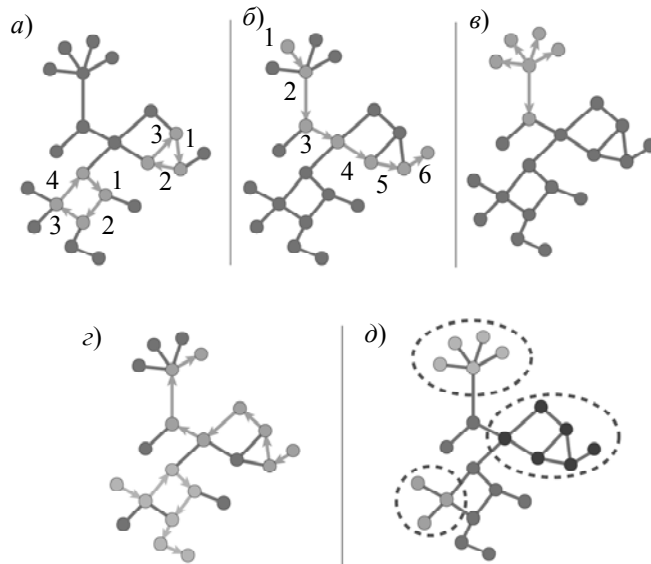


Рис. 1

При визуальном отображении темпоральной КС, в которой присутствуют отдельные кластеры, важно избежать перенасыщения изображения информацией [18]. Для этого используются механизмы агрегации, предусматривающие объединение узлов, принадлежащих к одному кластеру, в общий метаузел для улучшения восприятия. Пусть $B_{\text{cluster}} \subset V_t$ — набор вершин, принадлежащих к одному кластеру, v_{cluster} — вершина, представляющая кластер, тогда агрегация осуществляется следующим образом:

$$\Gamma: G_t = (V_t, E_t) \rightarrow (V_t \setminus B_{\text{cluster}} \cup v_{\text{cluster}}, E_t).$$

При детализации оператор имеет вид

$$\Gamma: G_t = (V_t, E_t) \rightarrow (V_t \setminus v_{\text{cluster}} \cup B_{\text{cluster}}, E_t).$$

При этом v_{cluster} может быть как фиктивным узлом, так и одним из элементов множества B_{cluster} , например узлом с максимальной степенью.

Исходя из вышеперечисленных особенностей темпоральных сетей для их качественной визуализации должен учитываться тот факт, что при эволюции КС ее конечное и промежуточные состояния могут существенно отличаться от начальных данных. Динамические процессы в КС могут быть отображены посредством изменения ее топологии (появление и исчезновение вершин и связей со временем), управления визуальными характеристиками ее составляющих, а также с помощью изменения взаимодействия между узлами.

Алгоритмические особенности визуализации темпоральных сетей. В соответствии с вышеизложенным основной целью визуализации темпоральной КС является раскладка (размещение вершин и связей в двумерном пространстве) графа таким образом, чтобы его особенности были хорошо различимы. При этом в алгоритме раскладки должны учитываться изменения свойств связей, а также изменения атрибутов узлов во времени. В общем случае разработанные для визуализации статических сетей алгоритмы без выполнения ряда модификаций не позволят получить в результате информативное изображение. В частности, реализация алгоритма должна обеспечивать такие характерные для темпоральных сетей приемы, как фиксация узлов, динамический пересчет раскладки, детализация и динамическая агрегация.

Фиксация узлов. В процессе отображения эволюции сети позиции „интересных“ объектов должны быть зафиксированы: изменение положения узла запрещается даже при изменении параметров КС. На рис. 2 представлен процесс образования сети репостов из социальной сети ВКонтакте (группа „Красивый Санкт-Петербург“) в течение года, в центре изображений зафиксирован узел, представляющий группу, откуда совершаются репосты первого уровня.

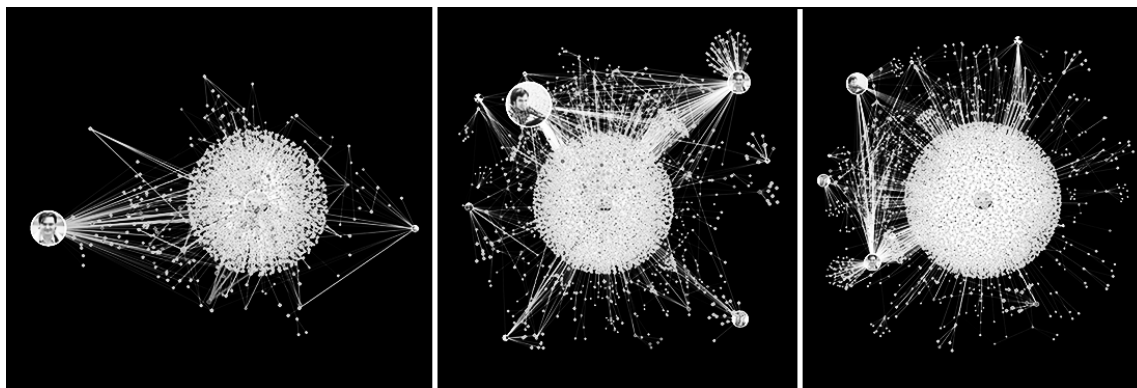


Рис. 2

Пересчет раскладки. Раскладка графа должна пересчитываться в течение времени при каждом изменении в структуре сети (рис. 3). При этом раскладка должна зависеть от параметров сети, а время, затрачиваемое на вычисления, не должно превышать времени внутренних изменений характеристик сети. Варьируя параметры раскладки, можно выделять конкретные узлы, хабы, кластеры и темпоральные пути.

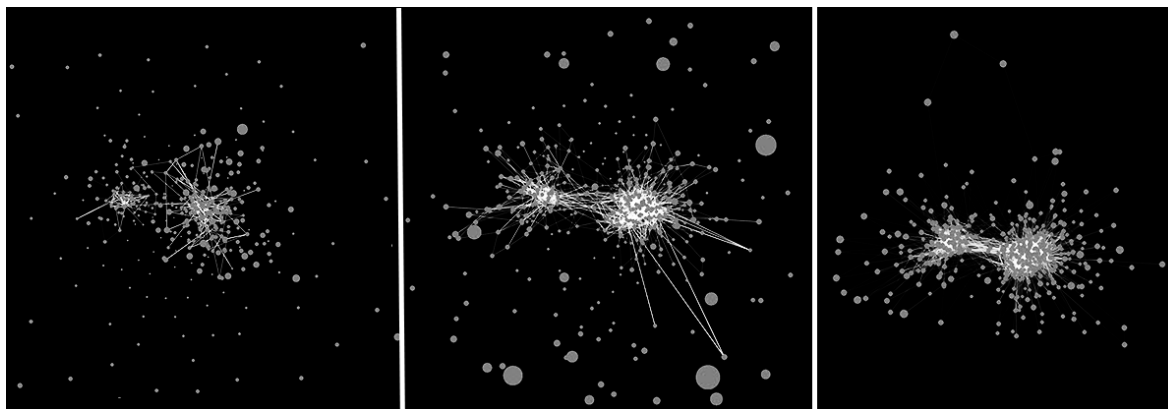


Рис. 3

При этом можно эффективно сочетать приемы фиксации узла и динамической раскладки. Так, объект (или объекты) наблюдения, выбранный оператором, может быть зафиксирован в определенном месте, а раскладка остальных элементов графа осуществляется относительно него или по собственному алгоритму. Таким образом, происходит сочетание нескольких алгоритмов раскладки в рамках одной визуализации. В листинге 1 приведен псевдокод алгоритма фиксации узла и динамической раскладки одновременно.

Листинг 1

```
foreach (node in graph.nodes){
    node.property.Update(time);
    if (node.isNotFixed){
        node.force = ForceRepulsion(node.property) +
            ForceAttraction(node.property);
        node.acceleration = node.force / node.mass;
        node.position += node.velocity * time +
            node.acceleration * time^2 / 2;
    };
};
```

Детализация и агрегация. Для улучшения визуализации после выделения кластеров осуществляется агрегация группы узлов, находящихся в текущий момент на значительном расстоянии от видеокамеры, в один узел. Детализация сети должна производиться либо при масштабировании изображения, либо по требованию оператора. Псевдокод алгоритма приведен в листинге 2.

Листинг 2

```
List<Node> aggregationGroup = CalculateGroup();
Node aggregator = new Node();
foreach (node in aggregationGroup){
    aggregator.Recalculate(node);
    graph.nodes.Delete(node);
};
graph.nodes.Add(aggregator);
```

Для визуализации больших сетей кластеризацию можно осуществлять исходя из положения узлов в пространстве визуализации, рассчитанного с помощью алгоритма раскладки. Таким образом, узлы, находящиеся рядом, будут объединены для уменьшения количества вычислений. На рис. 4 показаны исходная сеть, выделение кластеров и агрегация.

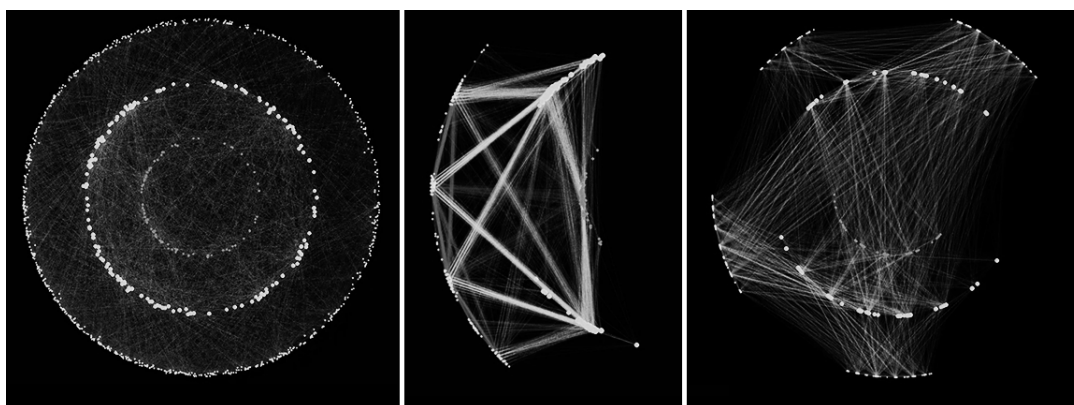


Рис. 4

Программная реализация комплекса визуализации КС. Рассмотренные выше методы и алгоритмы реализованы программно на основе платформы Fusion [19] с использованием языка программирования C# в качестве основного. Полученный программный комплекс обладает следующими функциональными возможностями.

Визуализация процессов, для чего используется комбинация следующих методов:

— управление визуальными характеристиками — изменение цвета, размера узлов или ребер для отображения процесса; для обозначения появления или удаления элемента изображения изменяется прозрачность объекта; для визуализации дополнительного процесса, например распространения информации в сети, возможно переключение режимов;

— управление физическими процессами: событие, произошедшее в сети, влечет за собой изменение взаимодействия между объектами, что реализуется с помощью изменения параметров раскладки графа: например, при удалении ребра узлы перестают притягиваться друг к другу.

Детализация и агрегация узлов. Возможно выделение кластеров в любой момент динамического процесса и наблюдение за ними на всем временном интервале. При формировании кластеров используется один из двух способов:

— алгоритмический, при котором узлы группируются в соответствии с определенными характеристиками, например типами или информированностью;

— пользовательский, при котором пользователь может выбрать определенные узлы и выделить их в отдельный кластер.

Раскладка графа осуществляется с применением силовых моделей графа (force-directed layout) и его модификаций [20], что для визуализации сетей в настоящий момент является наиболее эффективным методом как с позиций простоты реализации и гибкой настройки параметров, так и с позиции, характеризующей эстетическую оценку полученного результата [21]. Суть таких моделей заключается в поиске оптимального местоположения вершин (узлов) графа в 2D- или 3D- пространстве путем расчета сил взаимодействия (сил взаимного притяжения и отталкивания) между узлами или минимизации энергии системы. Как результат, все ребра графа имеют приблизительно равную длину и пересечения между ними минимальны.

Программный комплекс апробирован на различных комплексных сетях, отражающих явления реального мира разной природы.

Примеры. Одним из примеров визуализации процесса в сети является *граф звонков* — моделирование процесса распространения информации в мобильной телефонной сети. Исходная сеть состояла из 1000 абонентов, принадлежащих к одному из трех типов: деловые люди (совершают очень короткие звонки, имеют большой список контактов), организаторы (совершают много коротких звонков, средний список контактов) и обычные люди (длинные звонки, небольшой список контактов). Задача визуализации — показать процесс распространения информации в данной сети с учетом дополнительных данных. Для улучшения восприятия абонентам разных групп соответствуют узлы, расположенные на окружностях, имеющих разные радиусы. Оптимальные радиусы рассчитываются исходя из максимально возможного размера узла и максимального количества абонентов одного типа. На рис. 5 продемонстрирован процесс распространения информации в телефонной сети в начальный момент времени (а), спустя 20 ч (б) и по окончании моделирования (в).

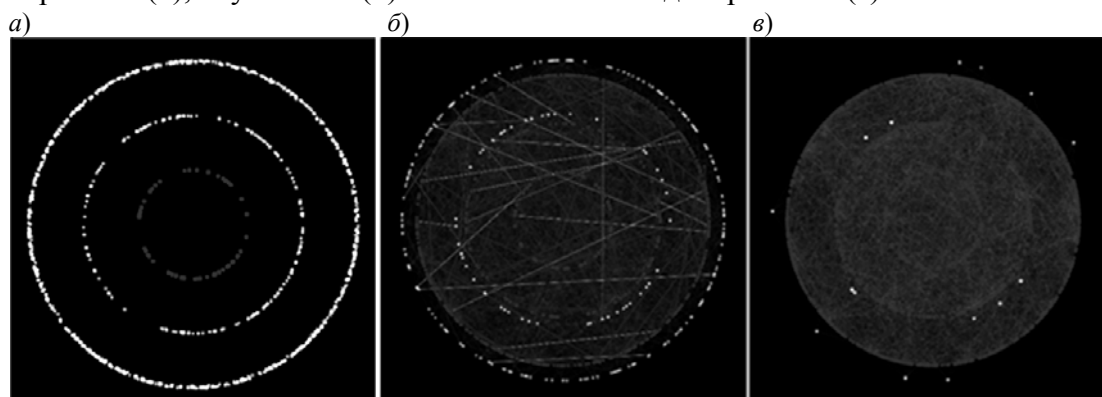


Рис. 5

Другой пример — визуальное представление динамического *графа косвенных контактов врачей и пациентов* (рис. 6), построенного на основе данных Медицинского центра им. В. А. Алмазова (Санкт-Петербург). Исходная информация (на 2014 г.) — это выборка данных по 216 пациентам. Каждый узел графа ассоциируется с врачом или сотрудником Центра, к которому было зафиксировано обращение пациента; ребро между двумя узлами — переход пациента от одного врача к другому, соответственно один и тот же пациент может совершать такой переход несколько раз.

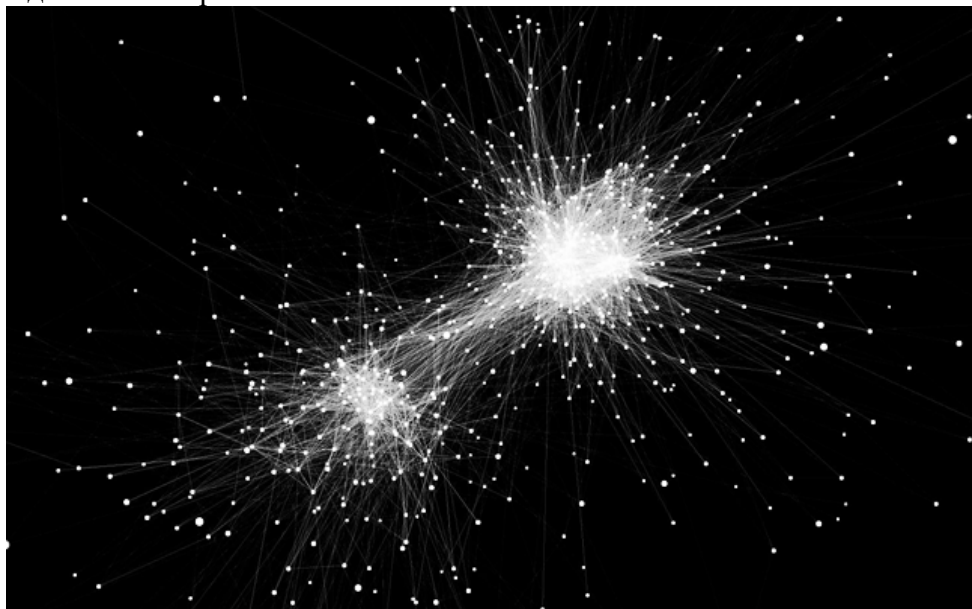


Рис. 6

Заключение. Когнитивная визуализация темпоральных комплексных сетей во многом наследует методы, применяемые для отображения статических сетей, однако в силу зависимости характеристик и структуры КС от времени требует их модификации и адаптации. Выбор алгоритма раскладки графа и даже задание примитивных визуальных характеристик, таких как размер или яркость, должны быть произведены с учетом изменения параметров сети с течением времени. С другой стороны, удовлетворение требований к алгоритмам раскладки позволит существенно улучшить качество итоговой визуализации как в рамках интерактивного приложения, предназначенного для эксперта, так и для получения информативных результирующих изображений.

Статья подготовлена по результатам работы, выполненной при поддержке Российского научного фонда (соглашение № 14-21-00137 от 15.08.2014 г.)

СПИСОК ЛИТЕРАТУРЫ

1. Broder A. et al. Graph structure in the web // Computer Networks. 2000. Vol. 33, N 1. P. 309—320.
2. Palla G., Barabási A. L., Vicsek T. Quantifying social group evolution // Nature. 2007. Vol. 446, N 7136. P. 664—667.
3. Bhalla U. S., Iyengar R. Emergent properties of networks of biological signaling pathways // Science. 1999. Vol. 283, N 5400. P. 381—387.
4. Kohn K. W. Molecular interaction map of the mammalian cell cycle control and DNA repair systems // Molecular Biology of the Cell. 1999. Vol. 10, N 8. P. 2703—2734.
5. Holme P., Saramäki J. Temporal networks // Physics reports. 2012. Vol. 519(3). P. 97—125.
6. Albert R., Barabási A. L. Statistical mechanics of complex networks // Reviews of Modern Physics. 2002. Vol. 74, N 1. P. 47.
7. Jeong H. et al. The large-scale organization of metabolic networks // Nature. 2000. Vol. 407, N 6804. P. 651—654.

8. Batagelj V., Mrvar A. Pajek-program for large network analysis // *Connections*. 1998. Vol. 21, N 2. P. 47—57.
9. Auber D. Tulip — A huge graph visualization framework // *Graph Drawing Software*; Ed.: M. Jünger, P. Mutzel. Springer-Verlag Berlin, Heidelberg, 2004. P. 105—126.
10. Ahmed A. et al. GeoMI: Geometry for maximum insight // *Graph Drawing: Proc. of Intern. Symp.* Springer-Verlag Berlin, Heidelberg, 2005. P. 468—479.
11. Ellson J. et al. Graphviz — open source graph drawing tools // *Graph Drawing: Proc. of Intern. Symp.* Springer-Verlag Berlin, Heidelberg, 2001. P. 483—484.
12. Ellson J. et al. Graphviz and dynagraph — static and dynamic graph drawing tools // *Graph Drawing Software*; Ed.: M. Jünger, P. Mutzel. Springer-Verlag Berlin, Heidelberg, 2004. P. 127—148.
13. Bastian M. et al. Gephi: an open source software for exploring and manipulating networks // *ICWSM*. 2009. Vol. 8. P. 361—362.
14. Zaidi F., Muelder C., Sallaberry A. Analysis and visualization of dynamic networks // *Encyclopedia of Social Network Analysis and Mining*. 2014. P. 37—48.
15. Kempe D., Kleinberg J., Tardos É. Maximizing the spread of influence through a social network // *Proc. of the 9th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*. ACM, 2003. P. 137—146.
16. Blondel V. D. et al. Fast unfolding of communities in large networks // *J. of Statistical Mechanics: Theory and Experiment*. 2008. Vol. 2008, N 10. P. P10008.
17. Nicosia V. et al. Graph metrics for temporal networks // *Temporal Networks*; Ed.: P. Holme, J. Saramäki. Springer-Verlag Berlin, Heidelberg, 2013. P. 15—40.
18. Hu Y., Shi L. Visualizing large graphs // *Wiley Interdisciplinary Reviews: Computational Statistics*. 2015. Vol. 7(2). P. 115—136.
19. Bezgodov A. et al. The framework for rapid graphics application development: the multi-scale problem visualization // *Procedia Computer Science*. 2015. Vol. 51. P. 2729—2733.
20. Jacomy M. et al. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software // *PLoS One*. 2014. Vol. 9(6).
21. Gibson H., Faith J., Vickers P. A survey of two-dimensional graph layout techniques for information visualisation // *Information Visualization*. 2013. Vol. 12(3—4). P. 324—357.

Сведения об авторах

- | | |
|---|---|
| Ксения Дмитриевна Мухина | — инженер; Университет ИТМО; Институт наукоемких компьютерных технологий; E-mail: mukhinaks@gmail.com |
| Клавдия Олеговна Боченина | — канд. техн. наук; Университет ИТМО; Институт наукоемких компьютерных технологий; ст. науч. сотрудник; E-mail: k.bochenina@gmail.com |
| Андрей Сергеевич Карсаков | — инженер; Университет ИТМО; Институт наукоемких компьютерных технологий; E-mail: karcs3d@gmail.com |
| Александр Валерьевич Бухановский | — д-р техн. наук; Университет ИТМО; Институт наукоемких компьютерных технологий; директор; E-mail: avb_mail@mail.ru |

Рекомендована
Институтом наукоемких компьютерных
технологий Ун-та ИТМО

Поступила в редакцию
12.12.16 г.

Ссылка для цитирования: Мухина К. Д., Боченина К. О., Карсаков А. С., Бухановский А. В. Технологии когнитивной визуализации темпоральных комплексных сетей // *Изв. вузов. Приборостроение*. 2017. Т. 60, № 3. С. 195—203.

TECHNOLOGIES OF COGNITIVE VISUALIZATION OF TEMPORAL COMPLEX NETWORKS

K. D. Mukhina, K. O. Bochenina, A. S. Karsakov, A. V. Boukhanovsky

*ITMO University, 197101, St. Petersburg, Russia
E-mail: mukhinaks@gmail.com*

Methods of temporal network visualization are considered. The basic methodological and algorithmic features of dynamic processes and temporal networks are reviewed. Several examples of programming realization are presented and application of the described technologies is discussed.

Keywords: temporal complex network, dynamic process, graph visualization, cognitive visualization

Data on authors

- | | | |
|----------------------------------|---|---|
| Ksenia D. Mukhina | — | Engineer; ITMO University, eScience Research Institute;
E-mail: mukhinaks@gmail.com |
| Klavdiya O. Bochenina | — | PhD; ITMO University, eScience Research Institute; Senior Scientists; E-mail: k.bochenina@gmail.com |
| Andrey S. Karsakov | — | Engineer; ITMO University, eScience Research Institute;
E-mail: kapc3d@gmail.com |
| Alexander V. Boukhanovsky | — | Dr. Sci.; ITMO University, eScience Research Institute; Director;
E-mail: avb_mail@mail.ru |

For citation: Mukhina K. D., Bochenina K. O., Karsakov A. S., Boukhanovsky A. V. Technologies of cognitive visualization of temporal complex networks // Journal of Instrument Engineering. 2017. Vol. 60, N 3. P. 195—203 (in Russian).

DOI: 10.17586/0021-3454-2017-60-3-195-203