DOI: 10.46548/CIT-2021-0034-0004

О.В. Прокофьев, И.Ю. Сёмочкина

МОДЕЛИРОВАНИЕ БОЛЬШИХ СЕТЕВЫХ СТРУКТУР С ПОМОЩЬЮ ПРОГРАММЫ PAJEK

Россия, г. Пенза, Пензенский государственный технологический университет

Large networks with hundreds or thousands of vertices and lines of communication can be found in many different fields, for example, in computer networks, social networks, transportation networks, in genealogy, in program flow graphs, in molecules, in intra- and inter-organizational networks. Many standard network algorithms are time consuming and memory intensive, and therefore are not suitable for analyzing such networks. The article presents the main approaches to the analysis and visualization of large networks, implemented in the Pajek program. Typical examples of using the program are also given.

Pajek - свободно распространяемый программный продукт с бесплатной лицензией (https://ru.freedownloadmanager.org/Windows-PC/Pajek-FREE.html), который вместе с документацией и дополнительными материалами можно скачать и использовать в некоммерческих целях http://mrvar.fdv.uni-lj.

Pajek (словенское слово "паук") - программный пакет для Windows для анализа и визуализации сетей, содержащих до десяти миллиардов вершин (нет ограничений - кроме размера памяти - на количество строк). Это программа для анализа больших сетей (http://mrvar.fdv.uni-lj.si/pajek/) [1]. Большие сети можно найти в самых разных областях. Обычно они производятся автоматически, с использованием компьютеров, из разных источников данных, которые уже доступны в машиночитаемой форме, например:

- большие генеалогии (родословные с несколькими людьми);
- сети, полученные из словарей и других текстов (сети мутации / вставки / удаления символов в английских словах);
- транспортные сети (авиалинии с аэропортами);
- большая молекула (молекула, состоящая из тысяч атомов, например ДНК);
- сети связи: ссылки между страницами или серверами в Интернете, использование Usenet, телефонные звонки;
- потоковые графы программ;
- библиографии, сети цитирования, графы Эрдёша.

Такие сети нельзя эффективно исследовать с помощью стандартных инструментов сетевого анализа, которые в основном основаны на матричном представлении и поэтому ограничены сетями умеренного размера (несколько десятков или сотен вершин).

Основными целями при разработке Pajek являются (рисунок 1): поддержка абстракции путем (рекурсивной) факторизации большой сети на несколько более мелких сетей, которые можно обрабатывать в дальнейшем с использованием более сложных методов; предоставить пользователю несколько мощных инструментов визуализации; реализовать набор эффективных алгоритмов для анализа больших сетей. Один из подходов к поддержке абстракции: найти кластеры (компоненты, окрестности «центральных» вершин, ядра и т. д.); в сети извлечь и показать вершины, которые принадлежат одним и тем же кластерам по отдельности, возможно, с частями контекста (подробный локальный вид); сжимать вершины в кластерах и показывать отношения между кластерами (глобальный вид). Пример выделенного кластера в области патентов

изображён на рисунке 2.

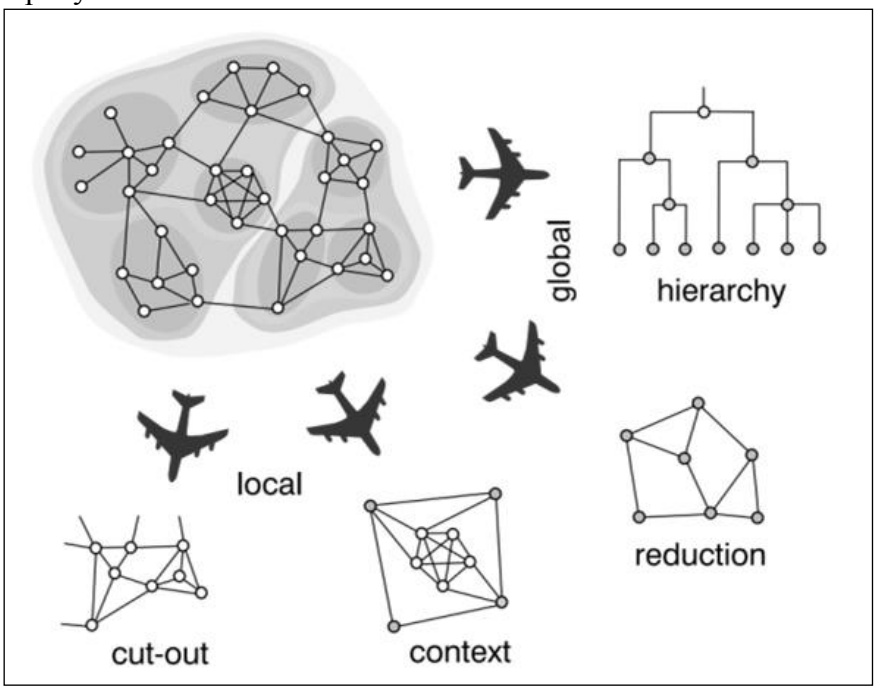

Рисунок 1: Цели разработки программы

Эффективные алгоритмы для больших сетей должны выполняться за ограниченное время и при ограниченном объёме памяти. Сложность алгоритма - это оценки времени T*(n)* и объема памяти S*(n)*, необходимых для его запуска на экземплярах объектов большого размера (в нашем случае размер - количество вершин или связей *n*).

В большинстве крупных сетей количество линий связи того же порядка, что и количество вершин, или большего порядка (такие сети считаются разреженными). Зачастую нужно анализировать большие, но разреженные сети.

Судя по возможностям современных компьютеров, сложность организации пространства для хранения разреженных сетей больше не имеет решающего значения. Проблема может быть решена с использованием соответствующих структур данных для внутреннего представления данных (представление сетей двусвязными списками используется в Pajek). Наличие гораздо более быстрых компьютеров не очень помогает в случае больших временных сложностей. В теории алгоритмов относительно легкими считаются задачи, решаемые с помощью алгоритмов полиномиальной сложности.

Ниже, в Таблице 1, показана временная эффективность алгоритмов для различных значений размера задачи сортировки. Хотя затраты времени приведены для "условного" компьютера с невысокими характеристиками (Pentium/64M/90MHz), производительность конкретного компьютера здесь не так важна, как факт нелинейного нарастания сложности задачи.

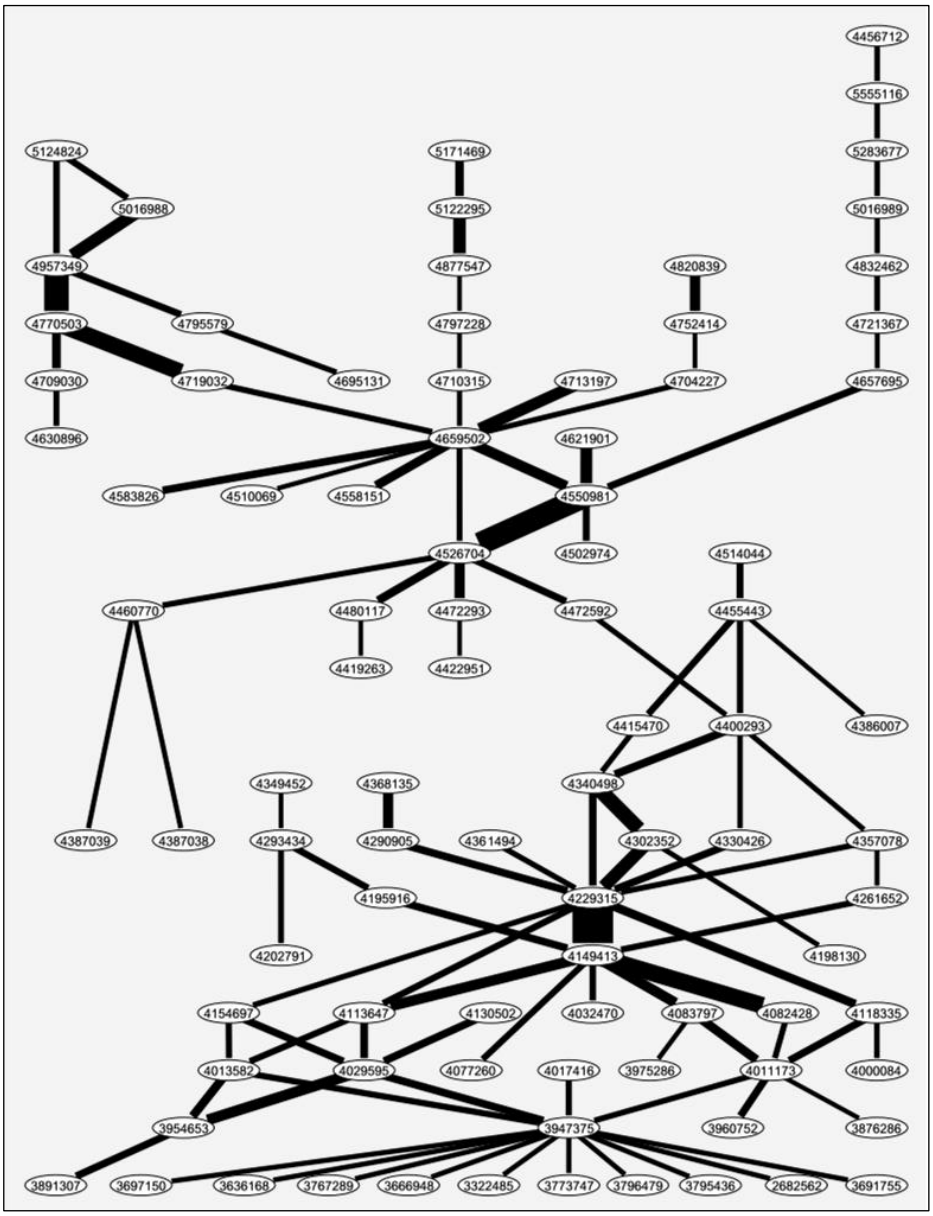

Рисунок 2. Патенты США "Жидкокристаллический дисплей" [1] - *Main island*

Таблица 1: Временные сложности алгоритмов

Метод	T*(n)*	1000	10000	100000	1000000
Перемешивание	*O(n)*	0,00 сек	0,015 сек	0,17 сек	2,22 сек
Быстрая сортировка	*O(n log n)*	0,00 сек	0,00 сек	0,40 сек	5,14 сек
Сортировка	*O(n log n)*	0,00 сек	0,06 сек	0,98 сек	14,35 сек

кучи					
Сортировка вставками	$O(n^2)$	0,07 сек	7,50 сек	12,5 мин	20,83 час
XY-сортировка	$O(n^3)$	0,10 сек	1, 67 мин	1,16 дня	3,17 года

На рисунке 3 изображена упорядоченная матрица значений торгового оборота между всеми странами мира [1] как пример задачи сортировки и визуализации большого объёма данных. В случае ещё большего объёма, уже на практике алгоритмы могут быть слишком медленными для интерактивного использования, что можно увидеть в Таблице 1.

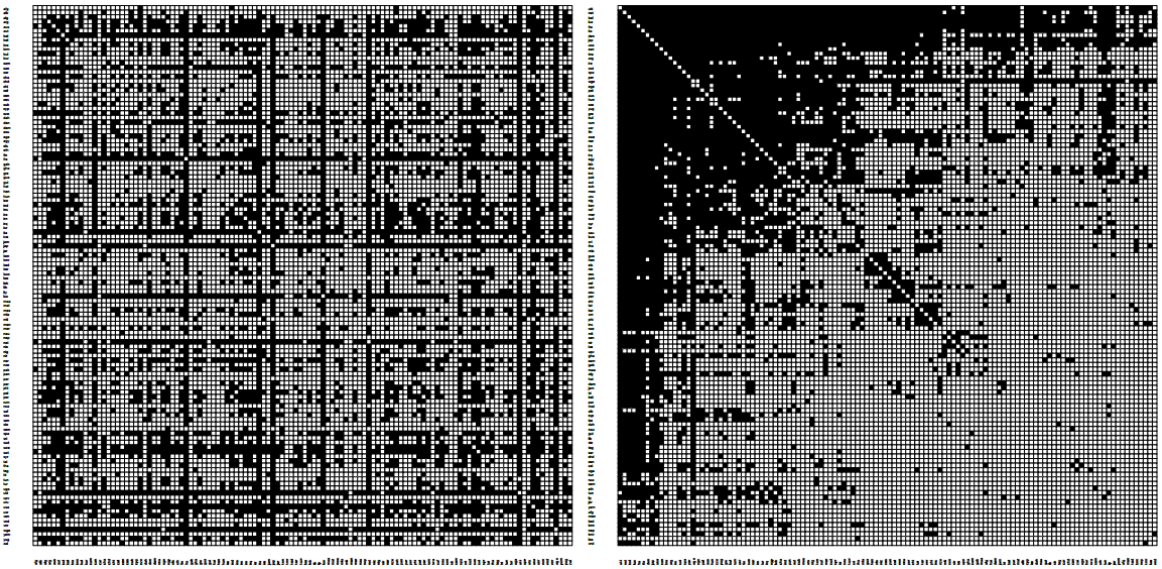

Рисунок 3. Матрица Снайдера и Кика объёма торговли между странами мира [1], слева - упорядоченная по алфавиту, справа - упорядоченная по объёму

Большая часть алгоритмы, реализованные в Pajek, имеют субквадратичную временную сложность: $O(n)$, $O(n \log n)$, $O(n\, n^{1/2})$ или ограничены небольшими наборами выбранных вершин.

В Pajek используются шесть типов объектов (основные показаны в меню программы на рисунке 4).

1. Сети - основные объекты (вершины и линии). Расширение по умолчанию: .net.

Сеть может быть представлена во входном файле по-разному:

с использованием списков дуг и вершин, в формате матрицы, в форматах UCINET, GEDCOM, химических форматах.

Во входной файл также может быть включена дополнительная информация для сетевого чертежа. Это объясняется в разделе «Экспорт в EPS / SVG / VRML».

2. Разбиения - они сообщают для каждой вершины, к какому классу принадлежит вершина.

3. Перестановки - перестановка вершин.

4. Кластеры - подмножество вершин (например, один класс из раздела).

5. Иерархии - иерархически упорядоченные вершины.

6. Векторы - они задают для каждой вершины числовое свойство действительное число).

Процедуры в главном окне Pajek (см. рисунок 4) организованы в соответствии с типами объектов данных, которые они используют в качестве входных.

Перестановки, разбиения и векторы могут использоваться для хранения свойств

вершин и измеряются в разных шкалах: порядковой, номинальной (категориальной) и числовой.

Помимо собственных форматов ввода, Pajek поддерживает несколько других форматов: UCINET DL, GED, генеалогии можно читать как Ope-граф или *p*-граф, некоторые молекулярные форматы: BS (*Ball and Stic*k), MAC (*Mac Molecule*) и MOL (MDL MOLfile).

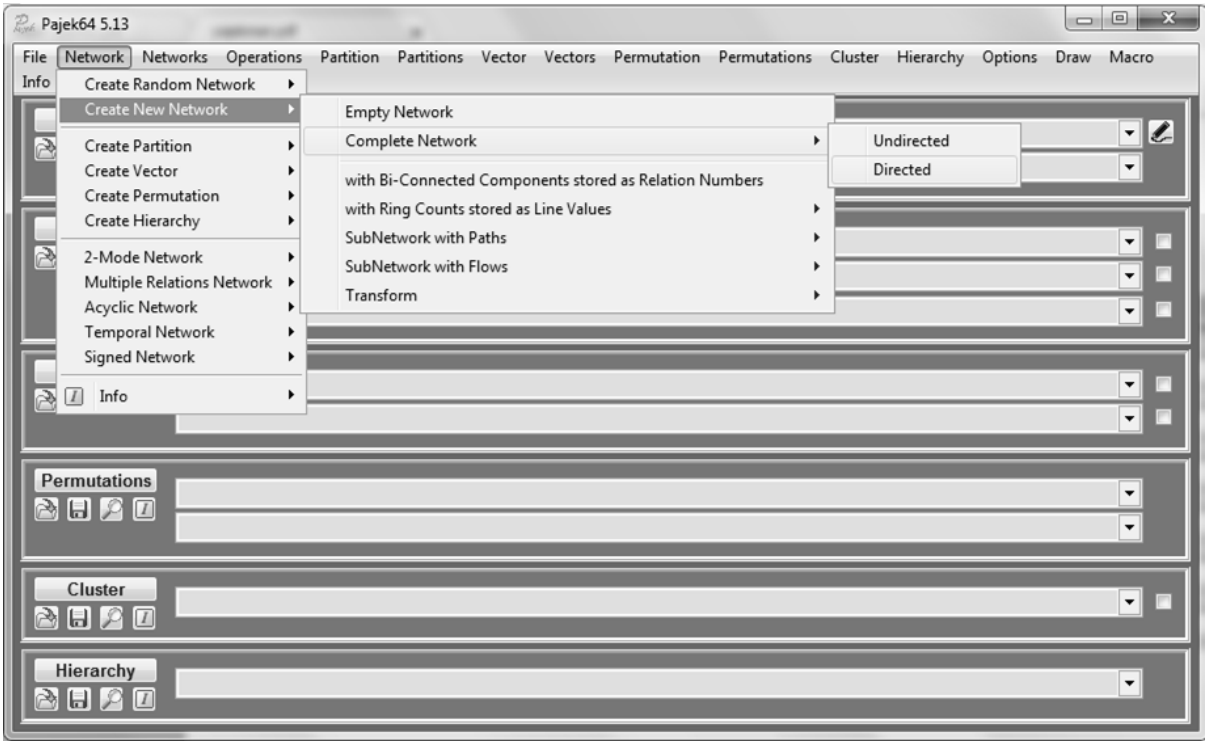

Рисунок 4. Основное диалоговое окно программы

Используя указанные выше структуры данных, был реализован базовый набор эффективных алгоритмов, например:

- подразделения: степень, глубина, ядро, p-клики, центры;
- бинарные операции: объединение, пересечение, разность;
- компоненты: сильная, слабая, двусвязная, симметричная;
- разложения: симметрично-ациклические;
- пути: кратчайший путь (и), все пути между двумя вершинами;
- потоки: максимальный поток между двумя вершинами;
- веса цитирования: Метод подсчета путей и метод SPLC;
- соседство: *k*-соседи;
- *CPM* (метод критического пути);
- выделение подсети;
- сокращение кластеров в сети (обобщенное блочное моделирование);
- переупорядочивание: топологическое упорядочение, нумерация Ричардса, поиск в глубину / ширину;
- редукция: иерархия, подразделение, степень;
- упрощения и преобразования: удаление петель, нескольких линий, преобразование дуг в ребра.

Программа Pajek находится в постоянном развитии. Последняя версия 5.13 доступна на его домашней странице. В ближайшем будущем планируется реализовать следующие дополнительные опции: различные процедуры кластеризации и декомпозиции, дополнительную статистику (подсчет триад), анимацию и представление последовательности сетей, форматирование вывода, управляющие структуры в макросах, тестирование планарности и разметку.

Перспективным направлением развития является возможность применения Pajek для нечётких когнитивных карт [2 - 4] с большим количеством концептов (вершин) и связей между ними. Например, в [5] программа позволила объединить в единую карту представления 29 экспертов о системе экологии Чёрного моря, визуально отобразить коллективное знание опрашиваемых по данной проблеме (рисунок 5). Этот же подход позволяет моделировать последствия нарушения экологического баланса. Pajek как средство моделирования может представлять интерес для специалистов по системному анализу в широком диапазоне прикладных областей: от биологических и экологических систем до сложных технических объектов [6].

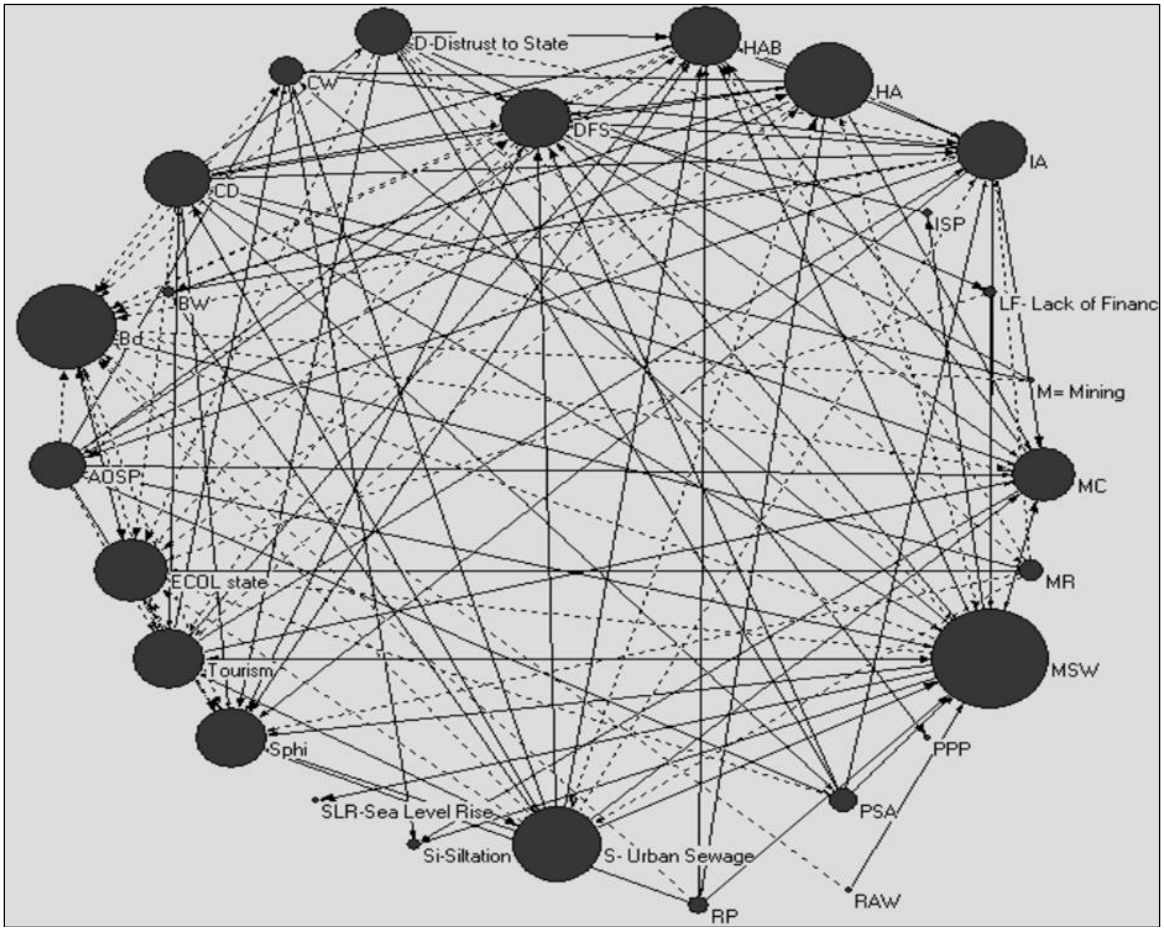

Рис.5. Нечёткая когнитивная карта объединённого знания группы интервьюируемых об экологии Чёрного моря

1. V. Batagelj, A. Mrvar. Pajek. Program for Analysis and Visualization of Large Networks Reference Manual. List of commands with short explanation version 2.05. University of Ljubljana, 2011, pages 96.

2. Голышевский О.А., Прокофьев О.В. Модель системы прогнозирования состояний сложного технического объекта. XXI век: итоги прошлого и проблемы настоящего плюс. 2018. Т. 7. № 4 (44). С. 19 - 23.

3. Прокофьев О.В., Савочкин А.Е. Моделирование системы обнаружения предаварийной ситуации на основе нечеткой когнитивной карты. Надежность и качество сложных систем. 2018. № 2 (22). С. 73 - 79.

4. Прокофьев О.В., Фельдман Ю. Способ построения дескриптивной модели на основе нечеткой когнитивной карты. Современные информационные технологии. 2004. № S2. С. 188 - 189.

5. Kontogianni A, Papageorgiou E, Salomatina L, Skourtos M, Zanou B, Risks for the Black Sea marine environment as perceived by Ukrainian stakeholders: A fuzzy cognitive mapping application, Ocean and Coastal Management(2012), doi: 10.1016/j.ocecoaman.2012.03.006.

6. M.Yu. Micheev, O.V. Prokofiev, A.E. Savochkin. Fuzzy cognitive map of pre-emergency prediction. Fuzzy Technologies in the Industry - FTI 2018, Vol-2258, p.502-509. [Электронный ресурс] http://ceur-ws.org/Vol-2258/paper59.pdf.