АЛГОРИТМЫ СТРУКТУРНОГО МОДЕЛИРОВАНИЯ ТРУБОПРОВОДНЫХ СИСТЕМ

Крыгин А.А., Лубков Н.В.
Институт проблем управления им. В.А. Трапезникова РАН
andreyakr@yandex.ru, lbknv@mail.ru

Аннотация: В работе рассмотрено решение с помощью методов теории графов задач по структурному моделированию трубопроводных систем, которые возникли в процессе создания автоматизированной информационно-поисковой системы. Описываются эффективные способы хранения данных и основные алгоритмы, в частности алгоритм нахождения всех путей между двумя вершинами.

Ключевые слова: трубопроводные системы, алгоритм нахождения путей в графах, реализация структуры хранения данных о графе.

Введение

Трубопроводные системы (ТПС) занимают важное место в реализации производственно-технической деятельности и в сфере коммунального хозяйства. Типичными примерами таких систем являются системы водоснабжения, водоотведения, газоснабжения комплексов городского хозяйства. Возникновение нештатных - аварийных ситуаций в процессе функционирования систем по причине отказов (утрате работоспособности) узловых элементов, нарушения целостности участков трубопроводов или их засоров может иметь серьезные последствия. Нельзя исключать и внешние неблагоприятные воздействия природного, техногенного характера, а также злонамеренные действия.

Для решения задачи анализа последствий аварии (и ряда других задач) ТПС представляется в виде направленного графа. Это позволяет задействовать аппарат теории графов и разработанные в этой области алгоритмы. Одной из трудностей, возникающих при программной реализации, является большое количество узловых элементов (порядка десятков тысяч) и связей между ними.

Работа состоит из двух частей. Первая часть посвящена программной реализации алгоритмов и способов хранения данных в рамках теории графов. Вторая часть посвящена практическому использованию полученных алгоритмов при создании информационно-поисковой системы автоматизации деятельности диспетчера теплосети

1 Реализация графовых алгоритмов

1.1 Формализованное описание графа

Для целей математической обработки информации, представленной в виде графа, наиболее часто используется задание графа матрицей смежности, которая представляет собой квадратную матрицу размерность, которой n равна числу вершин графа:

$C_{ij}=1$, если имеется дуга (s_i,s_j), и $C_{ij}=0$ в противном случае. Т.е. для описания требуется n^2 элементов.

Основной недостаток такого задания – существенные ограничения на размерность матрицы, которые возникают на практике. Кроме того, для слабосвязанных графов (к которым относятся графы ТПС) матрица связей будет разреженной. Вопросам хранения и обработки разреженных матриц посвящено много работ, например [1-2], в которых показана избыточность такого способа хранения. Поэтому матрицу связей C из соображений компактности записи целесообразно видоизменить. Новую матрицу определим как таблицу, количество строк которой, как и у прежней, будет равно количеству вершин графа, а длина строк будет переменной. Длина строки выбирается равной количеству дуг, выходящих из вершины, соответствующей данной строке. Если все вершины графа перенумерованы числами от 1 до n, то в i–ой строке должны записываться номера вершин, в которые идут дуги из вершины s_i. Для описания требуется n элементов для перекодировки вершин и s элементов для задания дуг.

1.2 Алгоритм нахождения путей в графе.

Все графовые задачи, которые возникли в процессе создания информационно-поисковой системы автоматизации деятельности диспетчера теплосети, базировались на задаче нахождения путей в графе с теми или иными начальными условиями. Поэтому проектирование подсистемы работы с графами строилось по принципу наибольшего быстродействия модуля нахождения путей. Это существенно повлияло на структуры данных по графу и структуры данных в указанном модуле.

Наиболее удобным вследствие регулярности выполняемых элементарных действий является алгоритм поиска путей, основанный на возведении в степень матрицы связей. При этом r-я степень матрицы содержит пути длины r. Максимальная длина пути (без циклов) r_{max} в графе не превышает размерности графа n. Поэтому теоретически может потребоваться последовательно перемножать матрицы n раз. Если же максимальная длина путей $r_{max} < n$, то процесс перемножения прекращается, поскольку все последующие степени матрицы связей будут содержать только нулевые элементы.

Однако очевиден и недостаток данного алгоритма – трудоемкость алгоритма имеет порядок n^4 ниже рассматривается модификация стандартного алгоритма, учитывающая специфику поставленной задачи.

По условию задачи анализа нештатной ситуации на ТПС, требуется определить пути, ведущие из множества начальных вершин, соответствующих источнику ресурса, в некоторое подмножество конечных вершин, отождествляемых с потребителями ресурса. Отсюда следует, что первоначально описанная задача поиска путей в графе может быть сужена, и благодаря этому может быть получен существенный выигрыш в трудоемкости алгоритма. Предлагается следующая схема: выделяется подграф G_n графа G; G_n будет содержать все вершины графа G, через которые проходят пути, соединяющие подмножество начальных вершин с подмножеством конечных вершин;

В результате выполнения первой процедуры размерность $n_п$подграфа G_n может оказаться существенно ниже размерности n графа G.

Выделение подграфа G_n

Пусть заданы два подмножества вершин графа: M1 – подмножество начальных вершин и M2 - подмножество конечных вершин. Требуется выделить подграф G_n графа G, в котором вершины из подмножества M1 будут корневыми (т.е. в них не будет входить ни одна дуга), а вершины из подмножества M2 будут концевыми (т.е. из них не будет выходить ни одна дуга). Добавим две фиктивные вершины V1 и V2 так, что из вершины V1 дуги будут входить в каждую вершину подмножества M1, а из каждой вершины подмножества M2 будет идти дуга в вершину V2. После этого, выполняя последовательно операцию нахождения всех путей, выходящих из V1 и путей, входящих в V2, получим искомый подграф G_n. В схематичном виде это показано на рис. 1.

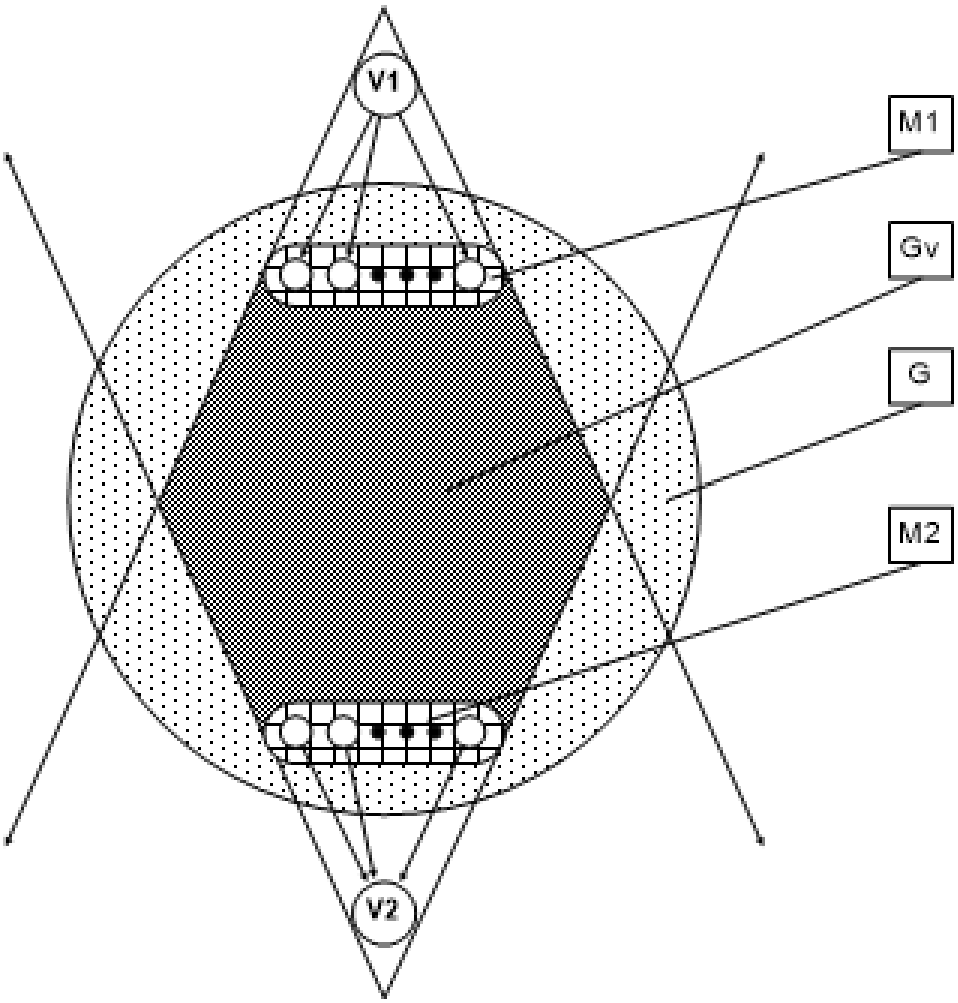

Рис.1 Выделение подграфа $G_П$

Определение путей в графе G_n.

Для поиска путей в графе G_n используется алгоритм последовательного умножения строки матрицы, соответствующей начальной вершине (для определенности установим номер начальной вершины равным 1) на матрицу связей по следующему правилу.

Str(k) = Str1(k-1)×MS – k-я степень первой строки, определяет пути длины k, выходящие из начальной вершины, определяется следующим образом:

Str(1) [j] = MS[1,j];

$Str(k)[j] = \bigcup_i Str(k-1)[i] \& MS[i, j] = \bigcup_i Str(k-1)[i] \& MS`[j, i]$, где & - операция логического «и», $\bigcup_i$ - операция логического «или», MS – матрица связей, MS`– транспонированная матрица связей.

1.3 Программная реализация

Эффективность алгоритма при программной реализации в значительной мере зависит от выбора структуры представления данных и наличия дополнительной информации, обеспечивающей навигацию по массивам данных при выполнении отдельных шагов алгоритма. По этой причине в качестве базового представления данных были выбраны многомерные динамические массивы, что позволяет использовать лишь минимально необходимые объемы памяти на каждом этапе выполнения алгоритма и повысить быстродействие их выполнения.

Приведем структуру основных массивов в реализации алгоритма нахождения путей; в этих массивах хранятся уже перекодированные номера вершин.

Двумерный массив для хранения таблицы смежных вершин графа (матрицы связей) *mMS* (рис. 2). По вертикали размер массива равен *n+1*, нулевой столбец массива – служебный, он используется для хранения текущих размерностей строк с целью повышения быстродействия алгоритмов. Номер каждой строки (кроме нулевой) соответствует номеру вершины, размер строки на единицу больше количества дуг, выходящих из соответствующей вершины (из-за служебного столбца). В служебных полях массива содержаться следующие данные:

mMs[*0,0*] – содержит значение *n* размерности матрицы (количества вершин графа), размер нулевой строки равен 1;

mMs[*i,0*] – содержит значение k_i количества дуг, выходящих из *i*-ой вершины графа.

Остальные элементы массива *mMs*[*i,j*] – содержат номер *j*-ой по счету вершины, в которую идет дуга из *i*-ой вершины.

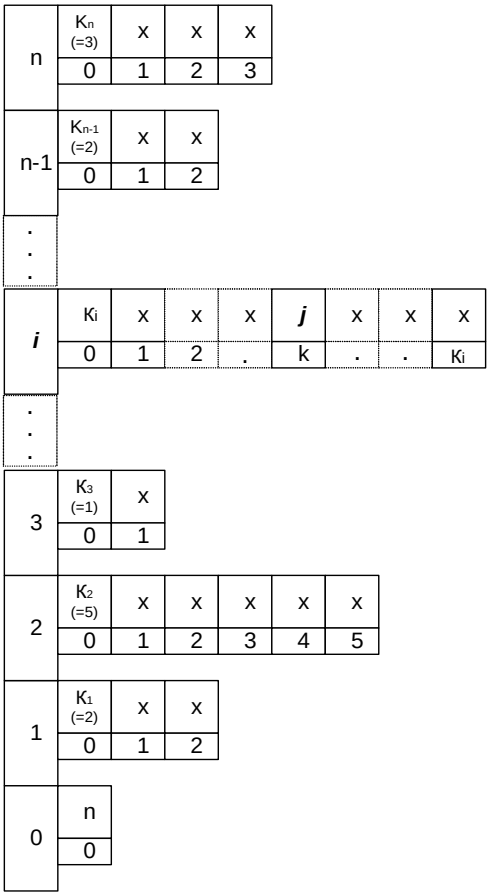

Рис.2 Структура массива таблицы смежных вершин

1. Для формирования и хранения результатов перемножения строки *Str(k)* на матрицу *MS* предлагается специальная структура данных – четырехмерный динамический массив *Str_n* , структура которого показана на рис. 3.

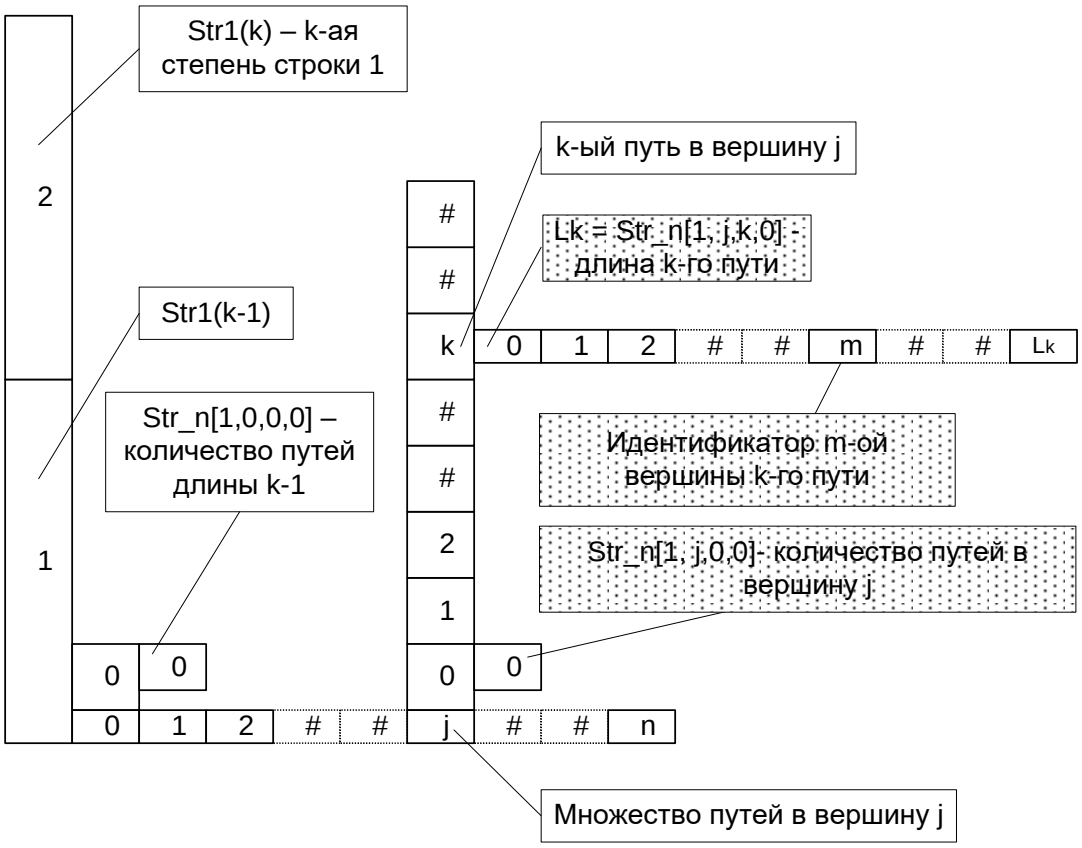

Рис.3. Структура массива Str_n

Размерность 1-го измерения равна двум, фактически, это два трехмерных массива: в одном хранятся результаты предыдущего умножения строки на матрицу, в другой записываются результаты текущего умножения. За исключением служебных полей размерность 2-го измерения равна *n*, в *j*-ом двухмерном массиве хранится множество путей. Размерность третьего измерения *№j* равна количеству путей в *j*-ую вершину, а размерность четвертого измерения *№k* равна длине *k*-го пути. Как видно из рисунка, этот массив является «плотно упакованным» и требует для хранения минимально возможных объемов памяти. Кроме того, он содержит служебную информацию (ячейки с индексами «0»), позволяющую сократить время работы алгоритма при многократном выполнении часто повторяющихся процедур.

2 Практическое применение разработанных методов (на примере сети теплоснабжения)

Инженерные распределительные сети предназначены для переноса ресурса от источников снабжения к потребителям. В упрощенном виде объекты теплоснабжения можно разделить на источники снабжения (ТЭЦ, РТС, КТС), объекты теплосети (камеры, участки теплопровода, коммутирующие устройства), тепловые пункты (ТП, ЦТП) и потребителей ресурса (строения, абоненты). Для обеспечения надежности объекты теплосети резервируются, соответственно, имеется множество конфигураций переноса ресурса. Текущая конфигурация сети определяется состоянием коммутирующих устройств (открыто/закрыто). Для краткости будем называть все источники снабжения ТЭЦ, коммутирующие устройства – коммутаторами, тепловые пункты – ТП.

При создании информационно-поисковой системы автоматизации деятельности диспетчера теплосети был выявлен ряд задач, которые удобно решать с помощью методов теории графов. Для решения этих задач была создана библиотека, практически не привязанная к предметной области, которая содержала следующие основные функции по работе с графами.

1. Инициализация

Для дальнейшей работы задается исходный граф и отдельно все ТЭЦ и ТП.

2. Построение покрывающего дерева

По заданному объекту передачи ресурса, в зависимости от заданного направления функция строит покрывающее дерево от объекта ко всем возможным ТП или от всех возможных ТЭЦ к объекту.

3. Поиск всех путей между двумя вершинами

Эта функция строит все пути между двумя заданными вершинами.

4. Поиск путей из заданной вершины во множество вершин

По заданным двум подмножествам вершин *V1* и *V2* и заданной вершине функция строит все пути из заданной вершины в вершины подмножества *V1* и из всех вершин подмножества *V2* в заданную вершину.

5. Поиск связей между подмножествами

Для двух произвольных подмножеств вершин *V1* и *V2* (подмножества могут пересекаться или совпадать). Необходимо построить граф, у которого каждая дуга *(a,b)* обладает следующими свойствами:

- $a \in V_1; b \in V_2$;

- В исходном графе существует путь $a \to b$, вершины которого (кроме *a* и *b*) не принадлежат *V1* и *V2*.

С помощью этих функций было выполнено решение следующих задач.

2. Визуализация схемы инженерной сети (на примере сети теплоснабжения)

Эта задача заключается в графическом отображении схемы теплоснабжения в удобном для пользователя виде и навигации по ней. В связи с тем, что количество объектов теплосети крупного города составляет сотни тысяч, для удобного представления схемы была выбрана четырехуровневая модель: схема соединения магистралей, укрупненная схема выбранной магистрали, детализация связей между объектами на выбранном фрагменте магистрали и схема подключения ТП к выбранной камере ввода. При создании программного модуля визуализации рассматриваемые методы использовались для следующих подзадач:

- расстановка объектов на схеме, автоматическое определение их координат;
- построение графа укрупненной схемы магистрали.

Расстановка объектов на схеме

Графовые модели необходимых схем (кроме схемы соединения магистралей) представляют собой лес с небольшим количеством дополнительных дуг. Поэтому для таких схем стандартный способ отображения деревьев будет удобным с точки зрения визуального восприятия (рис. 4).

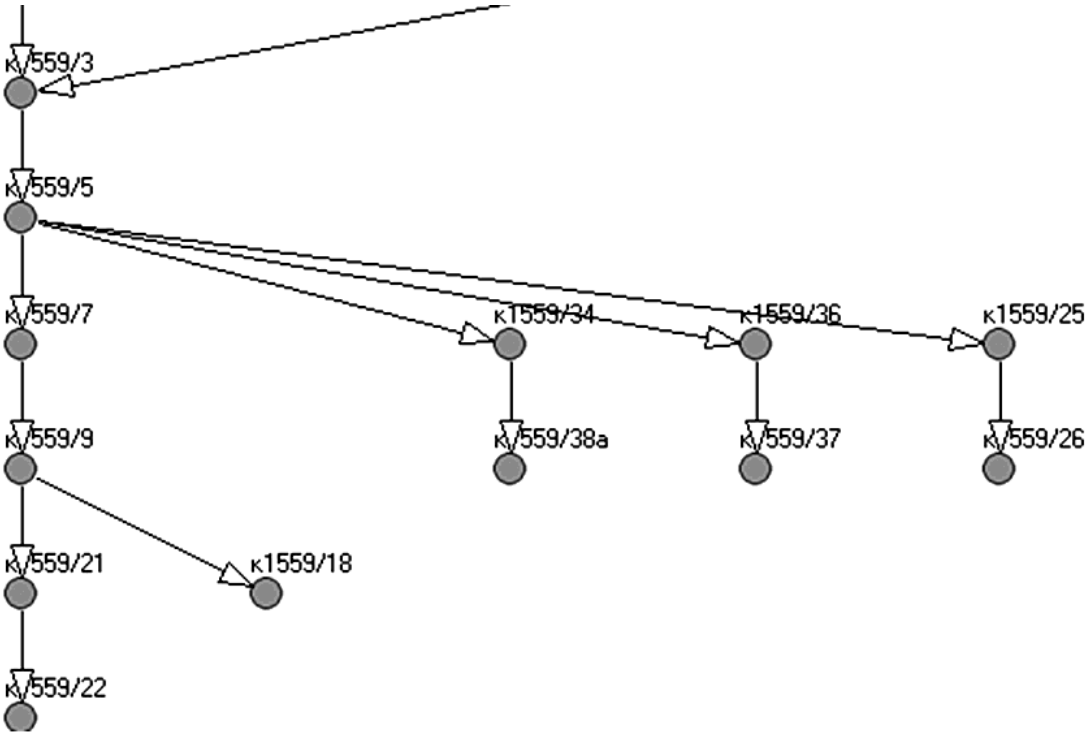

Рис.4. Фрагмент схемы магистрали

Соответственно, при отображении необходимо построить покрывающее дерево для рассматриваемого в данный момент фрагмента сети. Это было сделано с помощью *функции №2*.

Построение графа укрупненной схемы магистрали

В схемах инженерных сетей достаточно часто встречаются продолжительные линейные участки: камера - трубопровод - камера и т.д. Полное отображение таких участков будет существенно загромождать схему магистрали, поэтому при показе укрупненной схемы отображаются только вершины-источники, вершины-потребители и вершины-ветвления. Список таких вершин несложно получить с помощью нескольких SQL запросов к БД, после чего возникает задача определения графа такой укрупненной схемы. Это было сделано с помощью *функции №5*, где в качестве исходного графа задавался граф магистрали, а в качестве *V1* и *V2* - список отображаемых вершин (*V1* и *V2* совпадают).

2. Поиск неисправности в теплосети, отвечающей за нарушение в теплоснабжении строения по конкретному адресу

В процессе работы диспетчер принимает обращения и выдает информацию о ремонтных работах по запросу потребителей. Для оперативного решения этой задачи необходимо по заданному адресу определить ТП, от которого проводится теплоснабжение строения, проследить всю цепочку снабжения ТП от ТЭЦ и найти аварийный или ремонтируемый объект на этой цепочке. Для этой задачи использовалась *функция №4*: в качестве *V1* задавалось пустое множество, в качестве *V2* - все ТЭЦ, в качестве вершины – найденное ТП. После этого определялся путь снабжения ТП - путь, на котором все коммутаторы находятся в состоянии «открыто». Потом выполнялся поиск пересеченья всех объектов найденного пути и списка ремонтируемых объектов.

3. Задачи, связанные с авариями на теплосети

Действия, которые обязан выполнять диспетчер при возникновении аварии на объекте теплосети, включают в себя следующие задачи, которые были решены с помощью рассматриваемых методов:

- оценка последствий аварии;
- локализация аварии;
- поиск резервных маршрутов.

Оценка последствий аварии

В этой задаче необходимо в наиболее удобном для пользователя виде, отображать данные по строениям, теплоснабжение в которых нарушено по причине возникновения аварии. Для этого нужно найти все ТП, снабжение которых происходит через аварийный объект. Это было сделано с помощью *функции №4*: в качестве *V1* задавались все ТП, в качестве *V2* - пустое множество, в качестве вершины – аварийный объект. После этого определялись все ТП, у которых все коммутаторы находятся в состоянии «открыто» на найденных путях.

Локализация аварии

В этой задаче необходимо отключить аварийный объект от сети теплоснабжения. Необходимо найти все ближайшие коммутаторы к заданному аварийному объекту и перевести их в состояние «закрыто». Для решения использовалась *функция №4*: в качестве *V1* задавались все ТП, в качестве *V2* - все ТЭЦ, в качестве вершины – аварийный объект. После этого для каждого пути по состояниям коммутаторов определялось «открыт» или «закрыт» этот путь и для каждого открытого пути определялись коммутаторы, ближайшие к аварийному объекту, а также, для последующих действий, проводились те же операции для «закрытых» путей.

Поиск резервных маршрутов

После оценки последствий аварии диспетчеру необходимо определить оптимальную схему обеспечения отключенных строений теплом от резервных коммуникаций на время проведения ремонтных работ. Для этого выполняется поиск множества вариантов восстановления теплоснабжения отключенных ТП, среди которых производится выбор наилучшего варианта. Использовавшийся алгоритм поиска всех вариантов восстановления теплоснабжения [5] включает несколько этапов, один из которых – нахождение всех путей снабжения для каждого потребителя. Это было сделано с помощью *функции №3* следующим образом: к исходному графу магистрали было добавлено две фиктивных вершины, от первой вершины добавлены ребра ко всем ТЭЦ и от всех ТП добавлены ребра ко второй вершине. После этого вызывалась *функция №1*, которой передавался модифицированный исходный граф и *функция №3*, которая определяла все пути между двумя фиктивными вершинами.

3 Выводы

В процессе проектирования информационно-поисковой системы автоматизации деятельности диспетчера теплосети был выявлен ряд задач, которые удобно решать с помощью методов теории графов. Для решения этих задач была создана отдельная подсистема работы с графами. При анализе выделенных задач было поставлено два требования к подсистеме: максимальное быстродействие выполнения операции умножения строки на матрицу и экономное расходование памяти, т.к. количество узлов в исследуемом объекте может составлять сотни тысяч. При создании графовой подсистемы были разработаны структуры для хранения данных о графе и выполнении указанной операции умножения, отвечающие поставленным требованиям. В самой информационно-поисковой системе многократно использовались реализованные в модуле функции для визуализации схемы теплоснабжения, поиска неисправностей, локализации и оценке последствий аварий и поиске резервных маршрутов для восстановления снабжения потребителей.

Литература

1. *Тьюарсон Р.* Разреженные матрицы. – М.: Мир, 1977.
2. *Богоявленский А.И.* Использование форматов хранения разреженных матриц при реализации методоа конечных элементов. – М.: Вестник МГТУ им. Н.Э. Баумана. Серия «Естественные науки». 2017 №2 С. 4-11.
3. *Гребенюк Г.Г., Крыгин А.А.* Алгоритмы оптимизации числа переключений при реконфигурации сетей теплоснабжения - Автоматика и телемеханика. 2007. № 12. С. 101-112.