

Содержание

1	Введение	3
1.1	Актуальность работы	4
1.2	Определения и обозначения	5
1.3	Графические обозначения	6
2	Метод факторизации	6
3	Методы редукции размерности графа	8
3.1	Удаление лишних компонент связности	8
3.2	Стягивание цепей	9
3.3	Удаление висячих вершин	10
3.3.1	Висячая вершина — невыделенная	10
3.3.2	Висячая вершина — выделенная	10
3.4	Стягивание надёжных вершин	12
4	Графы, допускающие непосредственный расчёт	13
4.1	Графы с одной выделенной вершиной	13
4.2	Расчёт деревьев	13
4.3	Формулы для графов малой размерности	14
5	Простая факторизация	14
6	Реализация алгоритмов и численные эксперименты	15
7	Пример расчётов	18
8	Заключение	20

1 Введение

Беспроводные сенсорные сети представляют собой сложные иерархические системы, анализ и оптимизация которых требует решения нетривиальных математических задач, в частности исследования их структурной надёжности. При этом данные сети характеризуются специфическими показателями надёжности, такими как доля надёжно мониторируемой области, математическое ожидание числа связных пар и т.д. Под основным показателем надёжности сети в этой работе понимается вероятность связанности заданного подмножества элементов сети. Точный расчёт этого показателя сопряжён с большими временными и вычислительными трудностями и требует разработки новых алгоритмов.

При рассмотрении задач, связанных с надёжностью сетей, сеть обычно описывается случайным графом, где рёбра отображают сетевые каналы, а в качестве узлов выступают рабочие станции, серверы, переключатели и другие устройства. В нашем случае, в качестве узлов будут выступать сенсоры, а в качестве рёбер - всевозможные виды связей между сенсорами. Поскольку в сенсорных сетях именно сенсоры подвержены отказам, то мы будем моделировать сеть случайным графом, в котором все рёбра надёжны (имеют вероятность присутствия равную единице), а вершины могут отказывать (каждая вершина имеет вероятность отказа). Отказ вершины сети будет заключаться в её удалении из графа. Некоторые подходы для расчёта и оценивания различных показателей надёжности на таких графах описаны в статьях [3], [4], [5]. Целевые вершины мы будем считать надёжными (присутствующими с вероятностью равной единице).

В общем случае требуется рассмотреть все возможные разрушения графа, что является NP-полной задачей [2]. Следовательно, сложность задачи растёт экспоненциально (от числа невыделенных вершин) и поэтому нужны методы, ускоряющие расчёт.

В данной работе рассматривается метод факторизации [1] для направленного разрушения графа с последующими редукциями размерности.

1.1 Актуальность работы

Сенсорные сети активно применяются для военного наблюдения [8], изучения вулканов [6], среды обитания [7]. Исследуемый показатель надёжности напрямую связан с возможностью сообщения между критически важными узлами в сенсорной сети. Помимо сенсорных сетей, этот показатель может быть использован в других областях, где необходимо обеспечить связь между важными элементами, которая реализуется путём передачи сообщений между ненадёжными элементами, например:

1. Связь командных пунктов во время боевых действий или чрезвычайных ситуаций.
2. Связность выделенной подсети оптической сети.

В данной работе объектом исследования является система сетевой структуры, в которой в качестве модели используется случайный граф с абсолютно надёжными рёбрами, моделирующими связи в сети, и ненадёжными вершинами, моделирующими узлы.

В качестве предмета исследования выбран такой показатель надёжности сети как вероятность связности выделенного множества узлов в сети.

Целью дипломной работы является разработка и реализация алгоритма точного расчёта описанного показателя. Алгоритм должен абсолютно точно вычислять искомый показатель и завершаться за разумное время.

Для достижения указанной цели были поставлены следующие задачи:

1. Разработать точный алгоритм расчёта искомого показателя надёжности, завершающийся за разумное время.
2. Разработать различные методы ускорения.
3. Реализовать алгоритм с использованием многопоточности с помощью средств языка C#.

Методы исследования базируются на теории вероятностей и теории графов. Все предложенные алгоритмы реализованы на языке C#. Практическая значимость заключается в возможности точного расчёта указанного показателя надёжности, что в свою очередь позволяет ускорить и улучшить процесс проектирования надёжных сетей.

1.2 Определения и обозначения

Используется модель случайного графа с абсолютно надёжными ребрами и ненадёжными вершинами. Считаем, что вероятность присутствия для каждой вершины одинакова. Под случайным графом понимаем случайную реализацию заданного графа. В дальнейшем рассматриваем только случайные графы, поэтому слово “случайный” опускаем.

Введем обозначения:

$G(V, E)$ – неориентированный граф с множеством вершин V и множеством ребер E .

$K \subseteq V$ – множество выделенных вершин.

\mathcal{G} – множество всех подграфов G , включающих множество K

$P(H)$ – вероятность реализации $H \in \mathcal{G}$

p_i – вероятность присутствия невыделенной вершины с номером i . Вершины из K считаем присутствующими с вероятностью 1.

$R(G, K)$ – вероятность связности множества вершин K в графе G .

В дальнейшем надёжными вершинами будем называть вершины, имеющие вероятность присутствия равную единице.

1.3 Графические обозначения

На всех рисунках приняты следующие графические обозначения, примеры которых приведены на рисунке 1: вершина с номером 0 является целевой, вершина с номером 1 является ненадёжной, вершина с номером 2 является надёжной, но не целевой.

2 Метод факторизации

Основой всех предложенных методов будет являться метод факторизации или метод Мура-Шеннона [1]. Его суть состоит в следующем: необходимо выбрать вершину графа, имеющую ненулевую вероятность отказа, и перейти к рассмотрению двух подграфов, в одном из которых выбранная вершина имеет



Рис. 1: Изображение вершин в графе на рисунках

вероятность присутствия 1, а в другом выбранная вершина будет отсутствовать. При этом вероятность первого случая будет равна вероятности присутствия соответствующей вершины, а второго – вероятности её отсутствия. Будем называть вершину онадѐженной, если она уже рассмотрена и её вероятность принята равной единице. К полученным подграфам применять эту процедуру до тех пор, пока не останется незатронутых вершин. При этом формула полной вероятности будет давать требуемое выражение исследуемого показателя надёжности:

$$R(G, K) = p_v R(G/v, K) + (1 - p_v) R(G \setminus v, K), \quad (1)$$

где G/v – граф с онадѐженной вершиной v , $G \setminus v$ – граф с удалѐнной вершиной v . Рекурсии длятся до тех пор, пока не будут получены графы, допускающие непосредственный расчёт. Очевидно, сложность вычислений расчёт экспоненциально от числа невыделенных вершин и поэтому нужны методы, позволяющие редуцировать размерность графа.

3 Методы редукции размерности графа

Были разработаны или адаптированы несколько приёмов редукции. Такие приёмы как удаление лишних компонент связности и обработка висячих вершин — стандартные приёмы при вычислении различных показателей надёжности на случайных графах, которые, тем не менее, требуют индивидуального подхода и могут сильно отличаться в зависимости исследуемого показателя. Возможность стягивания надёжных вершин является особенностью исследуемого показателя.

3.1 Удаление лишних компонент связности

В процессе удаления вершин граф может быть разбит на несколько компонент связности. Очевидно, что компоненты связности, не включающие целевые вершины, не влияют на связность целевых и поэтому могут быть отброшены.

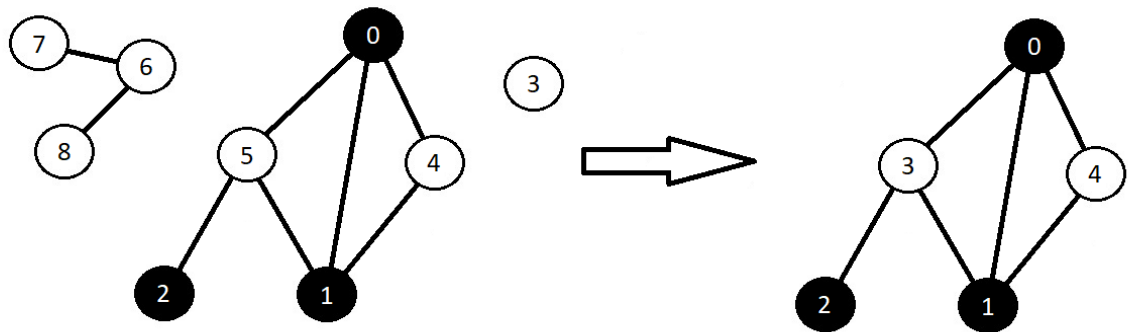


Рис. 2: Удаление лишних компонент связности

Например, для графов, представленных на рисунке 2, показатели надёжности равны ($R(G_1, K) = R(G_2, K)$), т.к. удаленные из первого графа вершины не влияют на связность выделенных вершин.

Так же можно отбрасывать из рассмотрения графы, в которых целевые вершины находятся в разных компонентах связности, как, например, на рисунке 3.

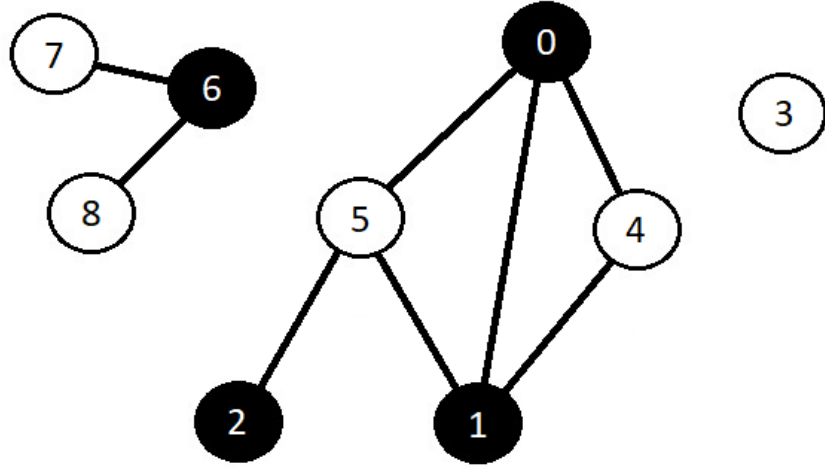


Рис. 3: Граф с заведомо несвязными целевыми вершинами

В любых реализациях графа вершины 1 и 6 не могут быть связны, и поэтому $R(G, K) = 0$.

3.2 Стягивание цепей

При обнаружении цепи, состоящей из нецелевых вершин, мы можем стянуть её, заменив на одну вершину, имеющую вероятность присутствия равной произведению стянутых. Такое преобразование возможно, т.к. возможность пройти по этой цепи появляется тогда и только тогда, когда присутствуют все элементы цепи. Пример стягивания изображён на рисунке 4. При этом

$$p_{1'} = \prod_{i=1}^n p_i \quad (2)$$

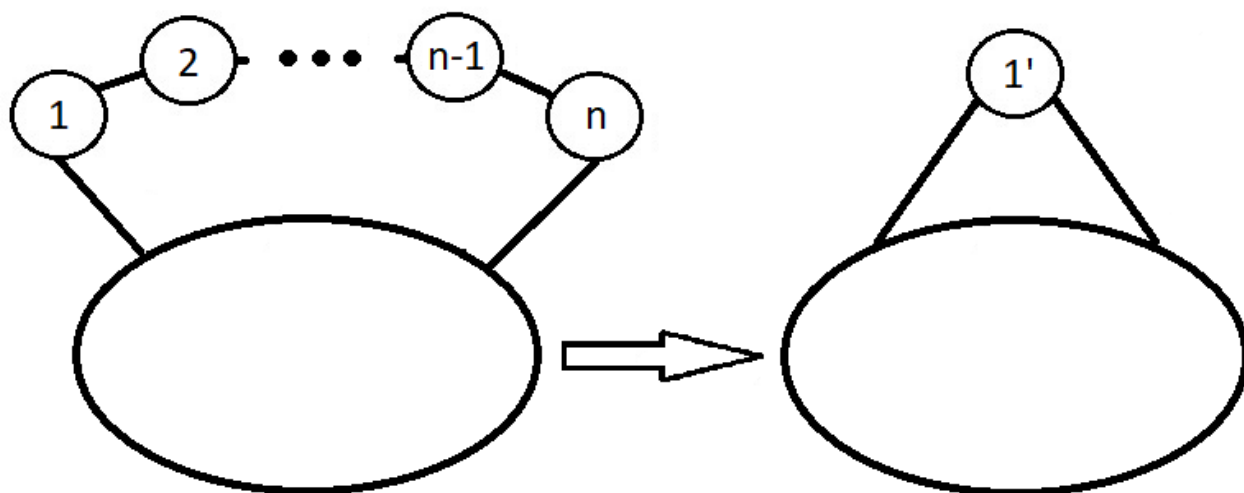


Рис. 4: Стягивание цепи

3.3 Удаление висячих вершин

При обнаружении висячей вершины возможно два варианта:

1. Висячая вершина является целевой.
2. Висячая вершина не является целевой.

3.3.1 Висячая вершина — невыделенная

Если она не является целевой, то независимо от того, с чем она была связана, мы можем удалить её, т.к. она никак не влияет на связность целевых вершин.

На рисунке 5 изображено удаление невыделенной висячей вершины.

При этом

$$R(G_1, K) = R(G_2, K) \quad (3)$$

3.3.2 Висячая вершина — выделенная

Если же висячая вершина является целевой, тогда нам нужно объявить связанную с ней вершину целевой, учитывая при этом изменения вероятности,

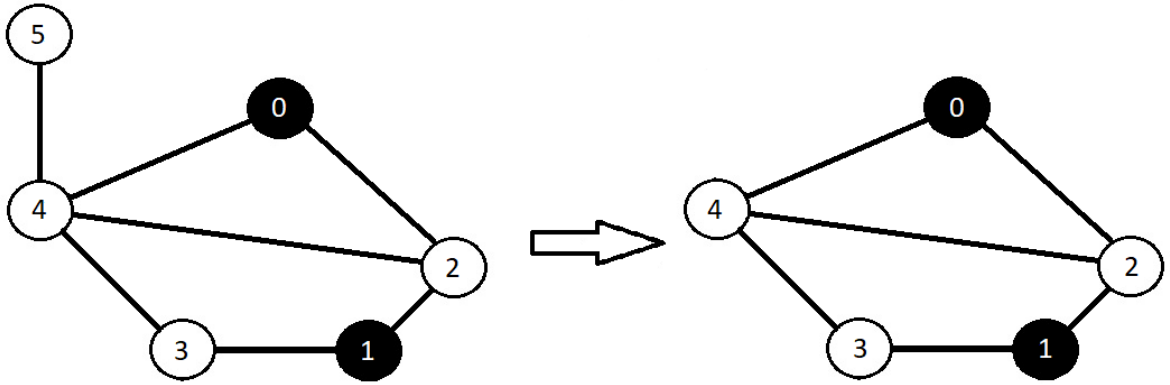


Рис. 5: Удаление висячей невыделенной вершины

затем удалить её. На рисунке 6 изображено удаление висячей выделенной вершины.

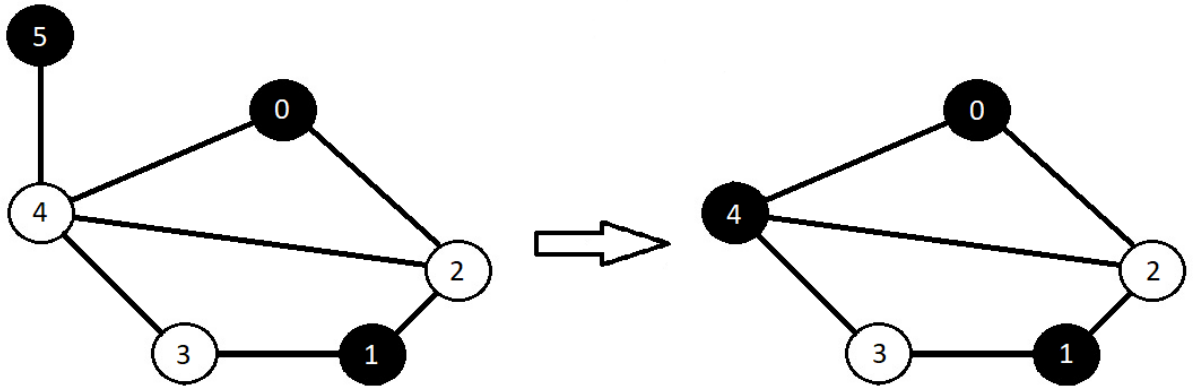


Рис. 6: Удаление висячей выделенной вершины

При этом

$$R(G_1, K) = p_4 R(G_2, K') \quad (4)$$

где $K' = K \cup \{4\} \setminus \{5\}$

3.4 Стягивание надёжных вершин

В процессе факторизации может сложиться такая ситуация, что надёжные вершины оказываются смежными. В таком случае мы можем стянуть их, заменив на одну, т.к. возможность дойти до любой из этих вершин равносильна возможности добраться до всех остальных. При этом возможно две ситуации: среди стягиваемых вершин есть выделенные или их нет. Если есть хотя бы одна выделенная вершина, то итоговая вершина также будет выделенной. Если среди стягиваемых выделенной не было, то итоговая вершина будет просто онадёженной. Значение показателя надёжности при этом не изменится.

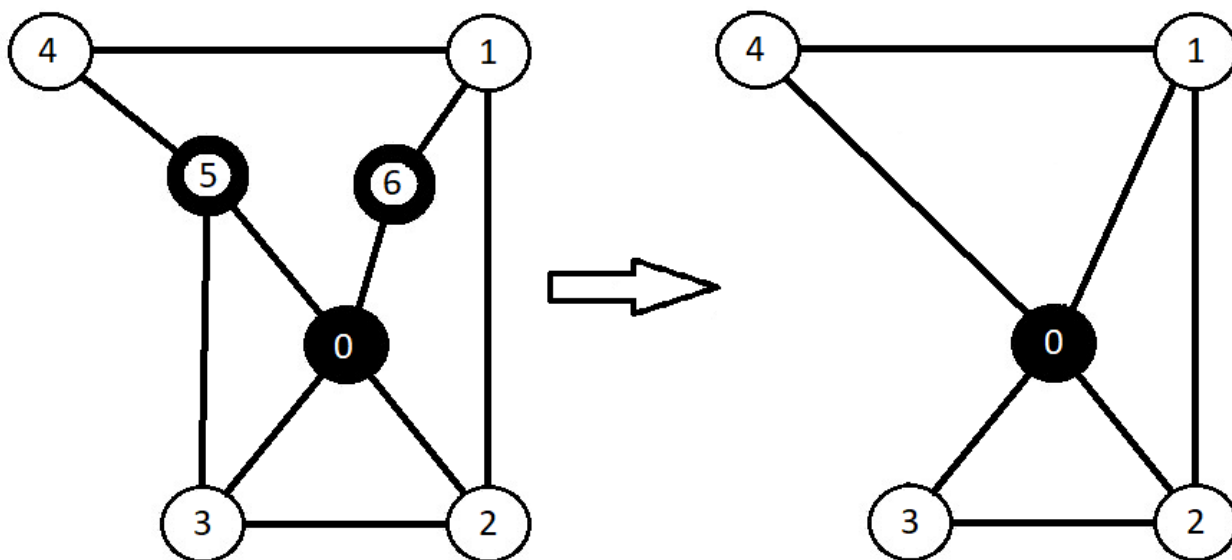


Рис. 7: Стягивание надёжных вершин

На рисунке 7 вершины 0, 5 и 6 могут быть объединены в одну и так как среди них есть выделенные, то итоговая вершина будет также выделенной.

4 Графы, допускающие непосредственный расчёт

Для некоторых видов графов можно вывести формулы прямого расчёта искомого показателя надёжности. К таким видам графов можно отнести:

1. Графы имеющие одну выделенную вершину.
2. Деревья.
3. Графы малой размерности (до 4-х).

Проверка на возможность непосредственного расчёта будет проводиться каждый раз при возможном изменении структуры графа. Проверка будет проводиться только для связных графов, поэтому во всех пунктах ниже графы считаются связными.

4.1 Графы с одной выделенной вершиной

В процессе объединения вершин, описанному в пункте 3.4, все выделенные вершины могут объединиться в одну. Это будет означать что целевые вершины связаны через надёжные вершины или напрямую и поэтому значение искомого показателя будет равно единице.

4.2 Расчёт деревьев

Если мы получили дерево, то необходимо начать удалять все невыделенные листья до тех пор, пока таковых не останется. Как только все листья окажутся выделенными, то возможность сообщения между ними появится тогда

и только тогда, когда присутствуют все элементы графа и поэтому искомая формула (применимая только для деревьев с выделенными вершинами в листьях) будет иметь вид

$$R(G, K) = \prod_{i=1}^n p_i \quad (5)$$

4.3 Формулы для графов малой размерности

Получены формулы для искомого показателя надёжности для графов, размерности меньше 4. Граф размерности 1 заведомо связан, следовательно, значение искомого показателя равно единице. В графе размерности 2 целевые вершины также связаны в любой реализации. В графе размерности 3 возможны два варианта - две или три целевые вершины. Если их три, то граф заведомо связан. Если две, то они связаны с помощью нецелевой (иначе они уже были бы стянуты), а значит показатель надёжности равен вероятности присутствия единственной нецелевой вершины.

5 Простая факторизация

Предложенные улучшения факторизации будут сравниваться с простой факторизацией. Её суть заключается в следующем - проводя факторизацию по всем ненадёжным вершинам, рассмотреть все возможные реализации графа и проверить связность выделенных вершин в них. Ответ будет получен по формуле

$$R(G, K) = \sum_{H \in \mathcal{G}} P(H) \chi(H, K) \quad (6)$$

Где $\chi(H, K)$ – характеристическая функция, показывающая связаны выделенные вершины в подграфе H или нет.

$$\chi(H, K) = \begin{cases} 1, & \text{множество вершин } K \text{ связано в графе } H; \\ 0, & \text{иначе.} \end{cases} \quad (7)$$

6 Реализация алгоритмов и численные эксперименты

Было реализовано 4 версии алгоритма:

1. Последовательная простая факторизация.
2. Параллельная простая факторизация.
3. Последовательная факторизация с использованием предложенных ускорений.
4. Параллельная факторизация с использованием предложенных ускорений.

Все алгоритмы были реализованы на языке C#. Параллелизм реализован следующим образом: при выборе разрешающей вершины выделялась отдельная задача (Task из библиотеки TPL в языке C#) для ветки разрушения, ветка онадёживания продолжала обрабатываться в этом же потоке. В начале вычислений в параллельных версиях алгоритмов создавался объект-сумматор, к которому приходили сообщения вида (вероятность, соединено выделенное вершин множество или нет) от разных потоков программы, который суммировал полную вероятность и вероятность того, что выделенное множество

вершин соединенно. Когда полная вероятность достигала единицы, объект менял свойство готовности и был готов отдать ответ.

При выделении ветки разрушения потоку отдавались граф, разрешающая вершина и вероятность реализации. В простой факторизации удалялась вершина, после чего происходила попытка выбора разрешающей вершины. Если она не удавалась - все изначально ненадёжные вершины были удалены или онадѣжены, то запускался поиск в глубину из первой целевой вершины и проверка вхождения остальных целевых вершин в её компоненту связности, после чего соответствующее сообщение отправлялось сумматору. Если находилась вершина с ненулевой вероятностью отказа, то производилась дальнейшая факторизация.

В модифицированном алгоритме в выделенном потоке производились удаление вершины и проверка на возможность проведения описанных редукций. Если в ходе проведения редукций был получен граф, допускающий непосредственный расчёт, то происходил расчёт, в противном случае происходил новый выбор разрешающей вершины и новая факторизация.

Используемый процессор - Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz
количество оперативной памяти - 8GB. Были случайным образом были сгенерированы 20 графов для каждой пары (количество вершин, количество рѣбер), для каждого рассчитан исследуемый показатель надёжности и в таблицу 1 занесено среднее арифметическое времени расчёта в секундах. На малых размерностях разрыв между всеми версиями очень мал, так как накладные расходы на параллельность и редукции не окупаются, но на больших размерностях выигрыш начинает становиться заметным. Несмотря на то, что задача NP-трудна, виден очень большой разрыв при использовании предложенных

Таблица 1: Результаты экспериментов

Версия алгоритма	1	2	3	4
(20,35)	0,06	0,03	0,02	0,026
(25,45)	1,7	1,28	0,03	0,034
(30,60)	70,48	58,40	0,45	0,35
(32,70)	373,35	279,15	3,56	2,66

ускорений, который также будет увеличиваться при увеличении размерности. Для каждой пары (количество вершин, количество рёбер) была получена статистика по числу различных применённых методов редукции (среднее арифметическое по всем графам заданной размерности), которая занесена в таблицу 2.

Таблица 2: Статистика применения редукций

	(20,35)	(25,45)	(30,60)	(32,70)
Висячих вершин	108	327	5731	73409
Удалено вершин не связанных с основной компонентой	0	0	0	102
Цепей	1	9	599	7925
Факторизаций	19	284	5048	60744
Непосредственно рассчитанных графов	8	82	1440	21939
Из них деревьев	3	17	678	6397

Из таблицы видно, как быстро растёт число факторизаций, а следовательно и сложность задачи, в зависимости от размерности. Удаление лишних компонент почти не происходило, что говорит о высокой плотности сгенерированных графов. Остальные методы редукции применяются достаточно часто,

что в сочетании с показанным выше ускорением говорит о высокой степени применимости и эффективности предложенных методов.

7 Пример расчётов

В качестве примера рассмотрим несколько вариантов размещения стоков в сенсорной сети, представленной сеткой 5 на 5 сенсоров, каждый из которых имеет вероятность отказа 0,5. Будем производить сравнения только параллельных версий алгоритмов.

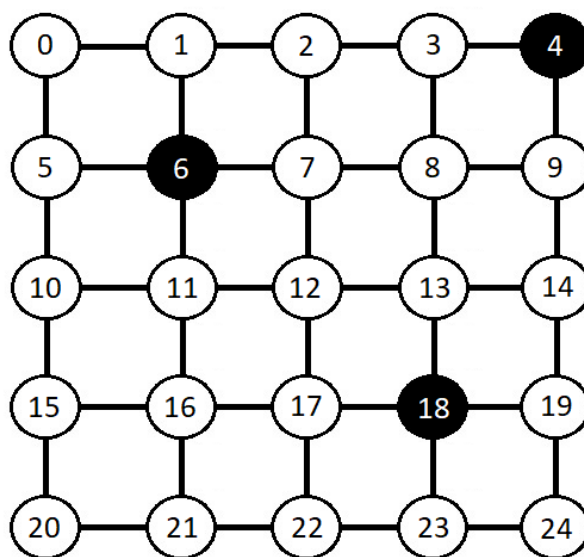


Рис. 8: Первый вариант размещения

Для сети, изображённой на рисунке 8 показатель надёжности равен 0,264. Для сети, изображённой на рисунке 9 показатель надёжности равен 0,186. Для сети, изображённой на рисунке 10 показатель надёжности равен 0,187. Время вычисления в секундах сведено в таблицу 3. Примерно одинаковый результат метода простой факторизации определяется практически равным количеством операций, в то время как при использовании редукций существенно

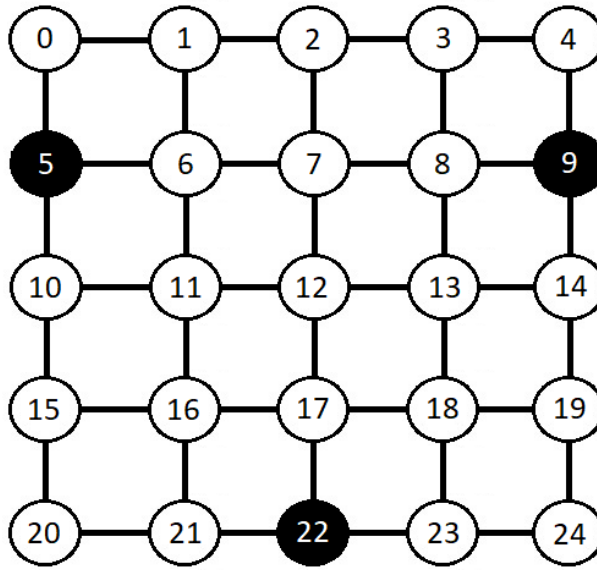


Рис. 9: Второй вариант размещения

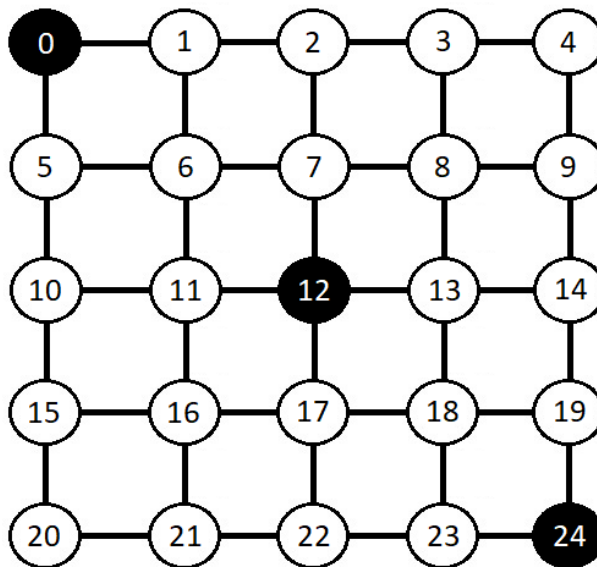


Рис. 10: Третий вариант размещения

зависит от размещения выделенных вершин, чем и определяется более чем двукратная разность времени расчёта. Статистика применённых редукций показана в таблице 4.

Таблица 3: Время вычисления

	Метод факторизации с редукциями	Метод простой факторизации
Первый вариант	1,94	38,1
Второй вариант	3,8	38,6
Третий вариант	4,2	37,7

Таблица 4: Статистика применения редукций

	1	2	3
Висячих вершин	18937	59476	58715
Удалено вершин не связанных с основной компонентой	0	89	0
Цепей	1890	5566	2850
Факторизаций	13153	41932	41844
Непосредственно рассчитанных графов	1660	10464	10329
Из них деревьев	1660	5381	6324

8 Заключение

В данной работе в качестве показателя надёжности была выбрана вероятность связности выделенного множества вершин. Этот показатель непосредственно связан с надёжностью и устойчивостью сенсорных сетей. Для расчёта этого показателя был реализован алгоритм полного перебора и его модификации, которые ускорили алгоритм. Таким образом, были решены следующие задачи:

1. Разработан алгоритм точного расчёта.
2. Предложены различные методы ускорения вычислений.

3. Реализован как последовательный вариант, так и реализация с учётом многопоточности.
4. Проведены численные эксперименты для подтверждения эффективности алгоритма по сравнению с простой факторизацией.

В результате численных экспериментов были получены следующие результаты:

1. Многопоточная версия работает быстрее, чем последовательная, что говорит об эффективности распараллеливания алгоритма.
2. Получен существенный выигрыш по сравнению с простой факторизацией, который увеличивается с увеличением размерности.

Была заложена программная основа для дальнейших исследований в области исследуемого показателя надёжности, которые будут проводиться по следующим направлениям:

1. Разработка и исследование методов выбора разрешающей вершины.
2. Поиск и анализ новых методов редукции.
3. Поиск других видов графов, допускающих непосредственный расчёт.
4. Исследование различных вариаций алгоритма.
5. Исследование применения метода Чена-Ли к данному показателю надёжности.

Список литературы

- [1] Moore E.F., Shannon C.E. Reliable Circuits Using Less Reliable Relays // J. Franklin Inst., Vol. 262, n. 4b, 1956. P. 191–208.
- [2] Leslie G. Valiant The Complexity of enumeration and reliability problems / Valiant Leslie G. // Society for Industrial and Applied Mathematics. -1979. -№3. - P.410-421.
- [3] Shooman A. M. Algorithms for network reliability and connection availability analysis // Electro/95 Intern. Prof. Program Proc. Hynes Convention Center, Boston, MA, June 21–23, 1995. S. l.: IEEE, 1995. P. 309–333
- [4] Мигов Д.А. Об одном показателе надёжности для сетей с отказами узлов / Д.А. Мигов // Средства и системы защиты информации и сетевых ресурсов. -2013. -№2. -С.43-48.
- [5] Liu Sh., Cheng K., Liu X. Network reliability with node failures // Networks. 2000. V. 35. P. 109–117
- [6] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. IEEE Internet Computing, 10:18–25, 2006.
- [7] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In First European Workshop on Wireless Sensor Networks, pages 307–322, 2004.
- [8] P. Kikiras and J. Avaritsiotis. Unattended ground sensor network for force protection. Journal of Battlefield Technology, 7(3):29–34, 2004.

- [9] Мистюрин В.В. Исследование алгоритмов расчёта и оценивания показателей структурной надёжности сенсорных сетей // Информационные технологии: Научноёмкое программное обеспечение: Материалы междунар. научной студ. конф. — М., 2017. С. 167.