

УДК 004.421+519.178

ПРИМЕНЕНИЕ АЛГОРИТМА СВЕРТКИ ДЛЯ РАЗДЕЛЕНИЯ ГРАФА НА ПРОПОРЦИОНАЛЬНЫЕ ПОДГРАФЫ

Н. Е. СЕРГЕЕВ¹, Е. Р. МУНТЯН²

¹nesergeev@sfedu.ru, ²ermuntyan@sfedu.ru

ФГАОУ ВО «Южный Федеральный университет» (ЮФУ)

Поступила в редакцию 23.10.2017

Аннотация. В работе поставлена одна из задач, возникающих при моделировании процессов управления сложными комплексами охраны объектов. Предложено использование графовой парадигмы для учета и прогнозирования ситуаций. Для наглядности рассмотрен пример связного неориентированного графа ситуаций, описаны его метрические характеристики с целью учета информации о его структуре. В статье выполнен обзор известных методов разделения графа на равные части, продемонстрированы результаты их выполнения. Предложено использование алгоритма свертки для разделения графа на пропорциональные подмножества для решения частных задач, возникающих при моделировании графов ситуаций.

Ключевые слова: граф; модель; ситуация; список; алгоритм; представление; поглощение; разделение; метрические характеристики.

ВВЕДЕНИЕ

При проектировании систем охраны сложных протяженных или площадных объектов возникает задача управления группами разнотипных компонентов таких систем. Управление выполняется на основании мониторинга действий и предполагаемых «враждебных намерений» некоторой неконтролируемой группы объектов. Предполагается наличие устойчивых связей между группами с возможно различной смысловой нагрузкой. В группы могут объединяться объекты системы или ситуации, складывающиеся на охраняемом объекте. Граф ситуаций рассматривается на уровне планирования действий для обеспечения охраны, граф охраняемого объекта лежит в основе представлений его компонентов.

В качестве объектов в нашем случае могут рассматриваться сложные технические

объекты, входящие в состав охранной системы. В случае ситуационного представления все ситуации рассматриваются как штатные для охранной системы, на которые она должна реагировать соответствующим образом. Примером ситуаций могут служить: проникновение в охраняемую зону, угроза проникновения в охраняемую зону, сокращение функциональных возможностей собственных сил и средств охранной системы и т.д.

В качестве способа представления групп объектов и прогнозирования ситуаций могут использоваться различные подходы, в том числе графовое [1], когнитивное [2–4] и нейросетевое [5] моделирование.

Сложные объекты, как в нашем случае, не могут быть представлены нейросетевыми и когнитивными структурами. Для прогнозирования развития ситуаций с помощью нейросетевых технологий необходимо достаточно большое количество обучающих образцов, которыми в настоящее время авторы не располагают. Для решения нашей

задачи требуется пошаговый анализ процесса прогнозирования. Когнитивное моделирование скорее применимо для процессов, в которых основную роль играет свойства личностей субъектов и их групп. В нашем случае в качестве объекта моделирования рассматривается сложная техническая система, в которой лица, принимающие решения, должны действовать по некоторым заранее известным правилам или стратегиям. В связи с этим в данной работе сделан акцент на графовых моделях.

ОПИСАНИЕ ГРАФА СИТУАЦИЙ

Имеется связный граф ситуаций $G = (G_V, G_E)$, где $G_V = \{gv_1, gv_2, \dots, gv_n\}$ – множество вершин графа (gv – graphvertex), $G_E = \{\langle ge_1, \mu_1 \rangle, \langle ge_2, \mu_2 \rangle, \dots, \langle ge_m, \mu_m \rangle\}$ – множество ребер графа (ge – graphedge).

Рассмотрим неориентированный граф $G_1 = (G_{1V}, G_{1E})$, представляющий ситуации и возможные переходы между ними (рис. 1).

Граф G_1 задан множествами вершин $G_{1V} = \{gv_1, gv_2, \dots, gv_{18}\}$ и ребер $G_{1E} = \{\langle ge_1, \mu_1 \rangle, \langle ge_2, \mu_2 \rangle, \dots, \langle ge_{39}, \mu_{39} \rangle\}$.

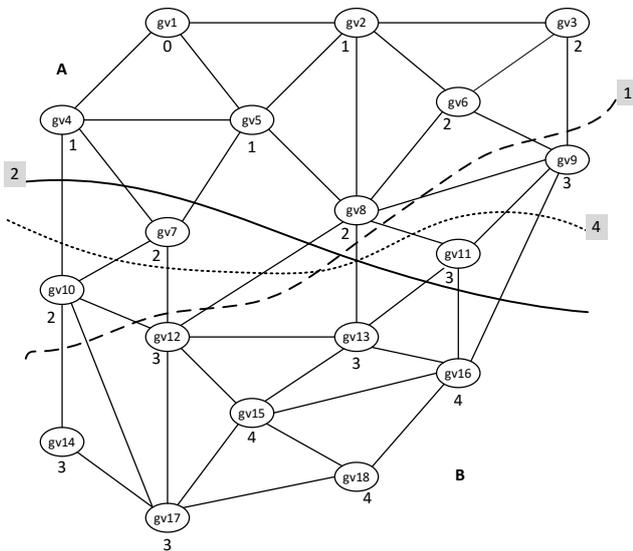


Рис. 1. Пример графа ситуаций G_1 и результаты разбиения графа на две части различными методами

В исследуемых задачах зачастую нередко возникает необходимость учета информации о структуре графа. Структура графа

может быть дополнительно описана его метрическими параметрами, такими как центр, радиус и диаметр графа. Опишем метрические характеристики графа ситуаций G_1 в соответствии с [6]. Для определения метрических характеристик графа авторами была составлена таблица минимальных расстояний между всеми вершинами при условии равенства расстояний между любой парой соседних вершин значению единицы. В этом случае все веса ребер графа из множества G_{1E} будут одинаковыми ($\mu_i = 1$ для $i = 1 \dots 39$).

Центром данного графа является вершина O_{G_1} , максимальное расстояние от которой до всех остальных вершин является минимальным. Анализ составленной таблицы позволяет определить центр графа G_1 ($O_{G_1} = gv_8$).

Диаметр графа d_{G_1} определяется как максимальное расстояние из всех кратчайших расстояний между парами вершин графа. В нашем случае $d_{G_1} = 5$.

Полезным может быть формирование множества периферийных вершин графа G_1 (G_{1PV}). Периферийными являются вершины, эксцентриситет которых равен диаметру графа. Т.к. диаметр d_{G_1} равен значению 5, то множество периферийных вершин исходного графа определяется в соответствии с таблицей минимальных расстояний, как $G_{1PV} = \{gv_1, gv_3, gv_{14}, gv_{18}\}$.

При этом диаметрально расположены две пары вершин: $\langle gv_1, gv_{18} \rangle$ и $\langle gv_3, gv_{14} \rangle$.

Радиус графа определяется как максимальное расстояние от центра O_{G_1} до остальных вершин графа ($r_{G_1} = 3$).

При моделировании взаимодействия групп может возникнуть необходимость распределения различных зон, что влечет за собой потребность в разбиения графа на части с последующим вычислением центров этих частей.

ЗАДАЧА РАЗБИЕНИЯ ГРАФА СИТУАЦИЙ

Для решения данной задачи разработано множество методов и алгоритмов, среди которых можно выделить основные группы [7, 8]: геометрические, спектральные, комбинаторные, диффузионные, генетические

алгоритмы и алгоритмы, используемые в рамках иерархического подхода для многоуровневой оптимизации.

Спектральные методы сводят задачу дискретной оптимизации к задаче непрерывной оптимизации. Среди спектральных методов разбиения графов можно выделить алгоритм спектральной бисекции. Суть спектральных методов заключается в получении дискретного оператора Лапласа графа, при помощи которого вычисляются меры расстояния, потом они сортируются, и полученный список делится на части, обеспечивая тем самым разбиение [7]. Это объясняет временные издержки при использовании спектральных методов. Время выполнения спектральных методов может быть уменьшено на порядок, если использовать многоуровневые алгоритмы для вычислений. Однако даже применение таких алгоритмов не помогает существенно снизить временные затраты [8].

Геометрические методы основаны на использовании координат узлов сети без учета их связности и являются достаточно быстрыми, но результаты разделения у таких алгоритмов обычно хуже, чем у других методов [8]. Среди них известны: рекурсивное покоординатное разбиение; рекурсивный инерционный метод разбиения и метод разбиения с помощью кривых, заполняющих пространство (например, кривых Пеано или кривых Гильберта). Для решения наших задач геометрические методы не представляют интереса, т.к. они используются для разделения сети, а не графа, который ее представляет.

Комбинаторные методы нацелены на распределение вершины по группам с учетом их связности. Среди таких методов можно выделить алгоритмы:

- алгоритмы вложенных сечений;
- метод деления с учетом связности (*Levelized Nested Dissection, LND*);
- алгоритм Кернигана–Лина (*KL*) и его модификации;
- алгоритм *Balanced Label Propagation (BLP)*.

Алгоритмы вложенных сечений относятся к методам рекурсивного деления по-

полам и используются для упорядочивания с целью сокращения заполнения при разложении разреженной матрицы. Суть алгоритма можно сформулировать следующим образом. В графе выбирается одна вершина gv_i в качестве корневой, этой вершине присваивается номер 1 и считается, что он относится к нулевому уровню смежности (L_0). Такая вершина называется псевдопериферийной.

Условие псевдопериферийности для вершины gv_i заключается в равенстве расстояния между вершинами gv_i и gv_j и эксцентриситету gv_i . Указанное определение гарантирует, что эксцентриситет псевдопериферийной вершины будет «близким» к диаметру графа.

Затем постепенно все оставшиеся вершины графа разбиваются на иерархические уровни смежности ($L_1 \dots L_{k-1}$), внутри которых осуществляется сквозная нумерация вершин по порядку.

Метод деления с учетом связности *LND* заключается в следующем: произвольным образом выбранной вершине присваивается номер $i=0$, вершинам, смежным с i -ой вершиной, присваиваются номера $i=i+1$ и т.д. в порядке увеличения номеров. Когда все вершины оказываются пронумерованы, граф делится на n частей в порядке нумерации. Очевидно, что для улучшения качества разбиения графа на части рекомендуется в качестве начальной вершины выбрать периферийную вершину [6]. Однако и это не гарантирует оптимальное разбиение графа на n подграфов. Под оптимальным разделением графа [9] обычно понимают разбиение его вершин на непересекающиеся подмножества с максимально близкими суммарными весами вершин и минимальным суммарным весом ребер, проходящих между полученными подмножествами вершин.

На рис. 1 пунктирной линией №1 отмечены результаты разделения графа G_1 рассмотренным выше методом. В качестве периферийной вершины выбрана gv_1 , которой присвоена метка 0. Далее метки распределены между вершинами с учетом инцидентности ребер. Множество вершин графа G_1 можно представить как $G_1 = G_1A \cup G_1B$. При этом подграф A содержит подмноже-

ство вершин $G1A_V = \{gv_1, gv_2, gv_3, gv_4, gv_5, gv_6, gv_7, gv_8, gv_{10}\}$, а подграф $B - G1B_V = \{gv_9, gv_{11}, gv_{12}, gv_{13}, gv_{14}, gv_{15}, gv_{16}, gv_{17}, gv_{18}\}$.

Как видно на рис. 1, подграфы $G1A$ и $G1B$ соединяются 10 ребрами. Очевидно, что такое разделение не является оптимальным с точки зрения [9].

Алгоритм Кернигана–Лина [10] позволяет решить проблему улучшения уже существующего разбиения, поочередно меняя местами вершины из разных подмножеств. Суть алгоритма может быть сформулирована таким образом: формируются всевозможные пары из вершин разных подмножеств, вершины из каждой пары переносятся в другие подмножества, образуя тем самым новый вариант разбиения. Эти варианты разбиений фиксируются. Из множества новых разбиений выбирается наилучшее.

Существуют различные модификации алгоритма Кернигана–Лина. Например, Fiduccia и Mattheyses [11] предложили алгоритм (FM), в котором за один шаг перемещается только одна вершина между всеми подмножествами. Алгоритм KL/FM позволяет избежать локальных минимумов за счет возможности временного увеличения числа связей между подграфами, однако качество его работы напрямую зависит от первоначального разбиения графа.

Если использовать предложенные алгоритмы для улучшения уже выполненного нами разбиения графа $G1$, результаты которых отмечены сплошной линией № 2 на рис. 1, то можно увидеть, что подграф A содержит подмножество вершин $G1A_V = \{gv_1, gv_2, gv_3, gv_4, gv_5, gv_6, gv_8, gv_9, gv_{11}\}$, а подграф $B - G1B_V = \{gv_7, gv_{10}, gv_{12}, gv_{13}, gv_{14}, gv_{15}, gv_{16}, gv_{17}, gv_{18}\}$. При этом следует отметить, что полученные подграфы связывает 8 ребер.

Подходы, основанные на применении алгоритма Кернигана–Лина и многоуровневой оптимизации, достаточно эффективны, но не подходят для разбиения больших графов.

Для разделения больших графов можно предложить алгоритм Balanced Label Propagation (BLP) [12], основанный на распро-

странении меток. Согласно алгоритму BLP каждой вершине присваивается уникальная метка, затем последовательно выполняются итерации замены метки каждой вершины на наиболее часто встречающуюся метку среди ее соседей.

При этом для распределенных вычислений эффективно использование принципа Map-Reduce. Принцип параллельного программирования Map-Reduce нацелен на обработку больших данных (петабайтов) и состоит из двух шагов: Map- и Reduce-. На Map-шаге выполняется предварительная обработка данных, а именно главный компьютер (*master node*) разбивает входные данные на части и передает их на рабочие компьютеры (*worker node*). На Reduce-шаге происходит сбор данных от *worker node* и выдача результата.

Авторами конкретизированы правила распространения меток для рассмотренного алгоритма, с учетом которых алгоритм BLP можно сформулировать следующим образом:

- 1) задание входных параметров:
 - номер шага ($t = 1$);
 - множество вершин $G_V = \{gv_i\}$, где $i=1, 2, \dots, n$;
 - множество меток вершин $M = \{M_j\}$, где $j = 1, 2, \dots, k$;
- 2) одновременное присвоение меток из множества M вершинам множества G_V . Одновременно каждой i -ой вершине из множества вершин G_V присваивается уникальная метка $M_j = 1, 2, \dots, k$, например совпадающая с номером вершины (в этом случае $k = n$);
- 3) переход к шагу $t = t + 1$;
- 4) последовательное присвоение меток вершинам множества G_V :
 - 4.1) определение исходной вершины gvi , возможно с младшим номером. Тогда $i = 1$;
 - 4.2) анализ меток вершины gvi и соседних с ней вершин;
 - 4.3) присвоение вершине gvi метки наиболее часто встречающейся среди ее соседей. Если таких меток несколько, то одна из них выбирается случайным образом или по определенному критерию, например метка вершины с наименьшим количеством

ребер. В случае нескольких меток вершин с наименьшим количеством ребер выбирается метка из присвоенных позже по времени;

5) если $i < n$, то $i = i + 1$;

6) переход в п. 4;

7) если $i = n$, то всем вершинам графа присвоены метки на шаге t . Переход в п. 3, если требуется новое расставление меток.

Рассмотрим работу данного алгоритма на примере графа $G1$.

Шаг 1 ($t = 1$).

П. 2. Вершинам $gv_1, gv_2, \dots, gv_{18}$ присвоить соответственно метки $M_j = \{1, 2, \dots, 18\}$ (как показано на рис. 2);

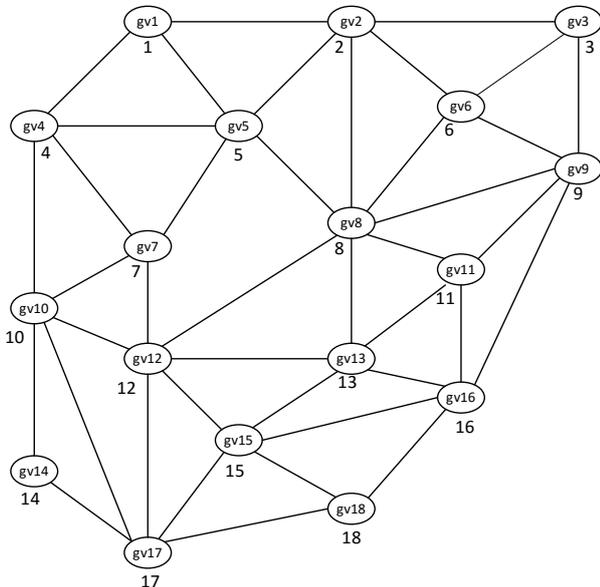


Рис. 2. Расстановка меток по алгоритму BLP в графе ситуаций $G1$ (шаг 1)

Шаг 2 ($t = 2$).

П. 4. В качестве исходной вершины выбираем вершину gv_1 с младшим номером ($i=1$). $M_1 = 1$. Для gv_1 соседними являются вершины gv_2, gv_4 и gv_5 с метками 2, 4 и 5 соответственно. В данном случае каждая метка уникальна, поэтому анализируем количество инцидентных ребер каждой из интересующих нас вершин. Для gv_2 –

5 ребер, gv_4 – 4 ребра и gv_5 – 5 ребер. Исходя из п. 4.3 алгоритма, вершине gv_1 присваиваем метку 4.

Переходим к вершине gv_2 . Соседние с ней вершины: gv_1, gv_3, gv_5 и gv_6 с метками 4, 3, 5 и 6 соответственно. Каждая метка уни-

кальна, поэтому анализируем количество инцидентных ребер каждой из интересующих нас вершин. Для gv_1 – 3 ребра, gv_3 – 3 ребра, gv_5 – 5 ребер и gv_6 – 4 ребра. Исходя из п. 4.3 алгоритма, выбираем случайным образом между метками вершин gv_1 и gv_3 , например, вершине gv_2 присваиваем метку 4.

Аналогично выполняем полный перебор вершин последовательно в сторону увеличения номеров. Результаты шага 2 изображены на рис. 3 пунктирной линией № 3.

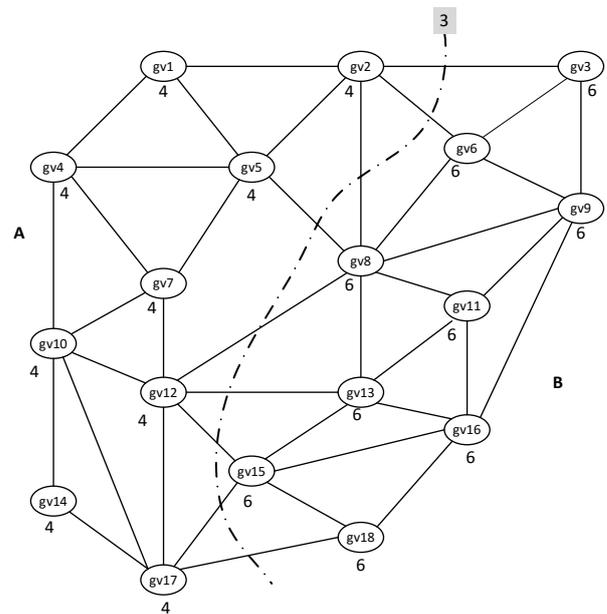


Рис. 3. Расстановка меток по алгоритму BLP в графе ситуаций $G1$ (шаг 2)

Следует отметить, что для разбиения графа ситуаций $G1$ на два подмножества A и B с использованием алгоритма BLP оказалось достаточно 2 шагов. При этом $G1A_V = \{gv_1, gv_2, gv_4, gv_5, gv_7, gv_{10}, gv_{12}, gv_{14}, gv_{17}\}$, а $G1B_V = \{gv_3, gv_6, gv_8, gv_9, gv_{11}, gv_{13}, gv_{15}, gv_{16}, gv_{18}\}$. Количество связей между подграфами A и B составляет 9 ребер.

Некоторые из рассмотренных выше алгоритмов являются средствами локальной оптимизации графов, т.к. способствуют улучшению уже существующих разбиений графа на подграфы. С целью решения данной проблемы предлагается подход многоуровневой оптимизации, который состоит из нескольких этапов: огрублению графа, разбиению огрубленного графа и уточнении

графа. Огрубление графа можно выполнить за счет стягивания пар смежных вершин в графе, как в [13].

В то же время в [14] авторами описана возможность обобщения графов ситуаций на основе алгоритма свертки. В предложенном алгоритме для исходной вершины, выбранной определенным образом, выполняется поиск инцидентных ребер и смежных вершин, которые все вместе образуют кластер первого уровня. Затем все вершины образованного кластера сворачиваются в одну вершину с предварительным представлением в виде списков смежности или ребер. Такая свертка выполняется до тех пор, пока исходный граф не будет представлен одной вершиной или до необходимого уровня детализации кластеров.

В случае использования спискового алгоритма свертки для решения нашей задачи предлагается следующий подход. Если заранее известно количество подграфов k при отсутствии заданных заранее критериев разбиения графа на части, то произвольным образом выбирается $k/2$ пары диаметрально противоположных периферийных вершин. Для каждой из выбранных периферийных вершин применяется алгоритм свертки до необходимого уровня кластеризации либо до полной свертки подграфов. Таким образом, возможны два варианта: либо получается естественное разбиение графа на k подграфов, либо свернутый граф следующим этапом разделяется k на подграфов одним из описанных выше способов.

Рассмотрим пример. Предположим, необходимо разбить граф $G1$ на два подграфа.

На рис. 4 представлен граф $G1$ с выделенной парой периферийных вершин $\langle gv_1, gv_{18} \rangle$ и образованными подмножествами A и B соседних вершин для gv_1 и gv_{18} . Условимся ребра, инцидентные выбранной вершине, обозначать пунктирной линией.

В соответствии с алгоритмом свертки [14]:

1) задаем подмножества $A = \{gv_1\}$ и $B = \{gv_{18}\}$, состоящие из найденных периферийных вершин графа $G1$;

2) определяем соседние вершины для gv_1 : gv_2, gv_4, gv_5 ; для gv_{18} : $gv_{15}, gv_{16}, gv_{17}$;

3) формируем новые подмножества вершин $A = \{gv_1, gv_2, gv_4, gv_5\}$ и $B = \{gv_{18}, gv_{15}, gv_{16}, gv_{17}\}$ (рис. 4);

4) представляем подмножества вершин A и B в виде списков ребер: $XA_{p1} = (gv_1: gv_2), (gv_1: gv_4), (gv_1: gv_5), (gv_2: gv_5), (gv_4: gv_5)$;

$XB_{p1} = (gv_{18}: gv_{15}), (gv_{18}: gv_{16}), (gv_{18}: gv_{17}), (gv_{15}: gv_{16}), (gv_{15}: gv_{17})$;

5) вершины подмножеств A и B сворачиваем в вершины gv_{19} и gv_{20} , т.е. $A = \{gv_{19}\}$ и $B = \{gv_{20}\}$;

6) определяем соседние вершины для gv_{19} : $gv_3, gv_6, gv_7, gv_8, gv_{10}$; для gv_{20} : $gv_9, gv_{10}, gv_{11}, gv_{12}, gv_{13}, gv_{14}$.

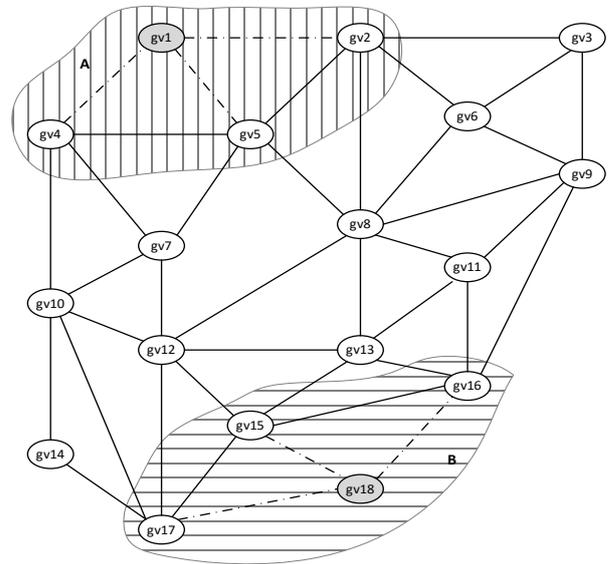


Рис. 4. Первый шаг свертки графа ситуаций $G1$

Как видно, вершина gv_{10} входит в оба подмножества A и B , так как является соседней для двух вершин gv_{19} и gv_{20} . В этом случае возможно два решения: либо изъятие такой вершины из одного подмножества, либо мы получаем пересекающиеся подмножества A и B . Выберем первый вариант для наглядности.

7) формируем новые подмножества вершин A и B (рис. 5):

$A = \{gv_{19}, gv_3, gv_6, gv_7, gv_8, gv_{10}\}$ и $B = \{gv_{20}, gv_9, gv_{11}, gv_{12}, gv_{13}, gv_{14}\}$;

8) представляем подмножества вершин A и B в виде списков ребер: XA_{p2} и XB_{p2} .

$XA_{p2} = (gv_{19}: gv_3), (gv_{19}: gv_6), (gv_{19}: gv_7), (gv_{19}: gv_8), (gv_{19}: gv_{10}), (gv_3: gv_6), (gv_6: gv_8), (gv_7: gv_{10}).$

$XB_{p2} = (gv_{20}: gv_9), (gv_{20}: gv_{11}), (gv_{20}: gv_{12}), (gv_{20}: gv_{13}), (gv_{20}: gv_{14}), (gv_9: gv_{11}), (gv_{11}: gv_{12}), (gv_{12}: gv_{13}).$

9) вершины подмножеств A и B сворачиваем в вершины gv_{21} и gv_{22} , т.е. $A = \{gv_{21}\}$ и $B = \{gv_{22}\}$.

При этом разделение графа на подграфы $G1A_V$ и $G1B_V$ совпадает с вариантом на рис. 1 (пунктирная линия № 1).

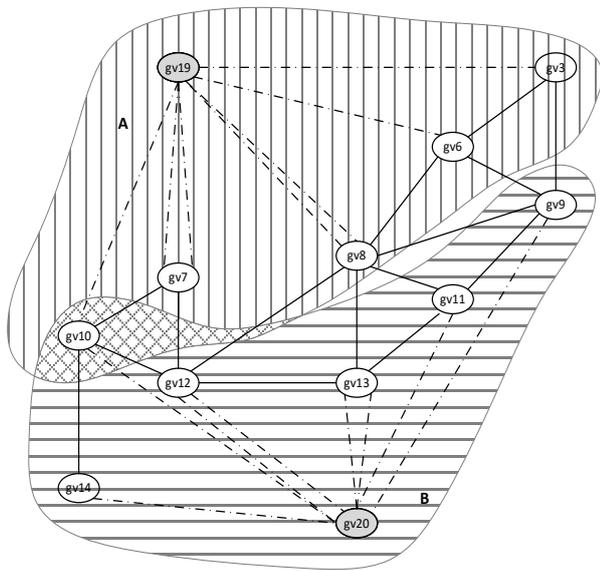


Рис. 5. Второй шаг свертки графа ситуаций $G1$

Если задан критерий формирования подграфов по принадлежности вершин к определенным группам объектов и известны центры этих групп (центры подграфов), то описанный подход может быть незначительно изменен на этапе выбора исходных вершин. В качестве исходных вершин предлагается использовать вершины – «центры» групп, затем этапы свертки и разбиения графа выполняются аналогично рассмотренному выше алгоритму.

Проанализировав литературные источники, можно отметить интерес к решаемой задаче в настоящее время. Например, в [9] рассматривается задача разделения графа на подграфы с минимизацией средних перемещений по подграфам и предлагается многоуровневый алгоритм разделений, в [8, 13] также предлагаются алгоритмы для реше-

ния задачи оптимального разделения графа на непересекающиеся подмножества.

Например, в [9] предлагается следующий алгоритм разделения графа:

10) формируется матрица смежности графа;

11) полученная матрица смежности произвольным образом разделяется на k равных частей, образуя тем самым условное разделение графа на подграфы $G = \{G_A, G_B, \dots\}$;

12) определяются суммарные весовые коэффициенты подграфов $\mu = \{\mu_A, \mu_B, \dots\}$;

13) определяется средний весовой коэффициент подграфа как среднее значение $\mu_k = (\mu_A + \mu_B + \dots)/k$;

14) проверяется выполнение условия оптимизации на соответствие суммарных весовых коэффициентов подграфов μ_A, μ_B и т.д. среднему значению μ_k ;

15) если необходимо, выполняется перенос вершин из одного подграфа в другой для выравнивания суммарных весовых коэффициентов подграфов.

К недостаткам такого алгоритма следует отнести необходимость составления матрицы смежности, что затруднительно выполнить для больших графов [1, 14].

Рассмотрим работу данного алгоритма на примере графа $G1$.

16) матрица смежности графа $G1$ составляется стандартно;

17) полученную матрицу смежности делим пополам и получаем $G1 = \{G1A, G1B\}$, при этом $G1A = \{gv_1, gv_2, gv_3, gv_4, gv_5, gv_6, gv_7, gv_8, gv_9\}$ и $G1B = \{gv_{10}, gv_{11}, gv_{12}, gv_{13}, gv_{14}, gv_{15}, gv_{16}, gv_{17}, gv_{18}\}$;

18) суммарные весовые коэффициенты подграфов $\mu = \{\mu_A, \mu_B\} = \{9, 9\}$ с учетом, что $\mu_{gvi} = 1, i = 1 \dots 18$;

19) средний весовой коэффициент подграфа $\mu_k = (\mu_A + \mu_B)/k = 9$ при количестве подграфов $k = 2$;

20) условие оптимизации на соответствие суммарных весовых коэффициентов подграфов μ_A, μ_B среднему значению μ_k выполнено.

Разделение, полученное в результате алгоритма [9], отмечено на рис. 1 пунктирной линией № 4. Полученные подграфы связы-

вает 8 ребер, как и в случае применения алгоритма улучшения разделения, отмеченного линией № 2.

Также следует отметить, что может иметь место задача разбиения графа на частично пересекающиеся подмножества вершин. Частичность пересечения в данном случае может быть обусловлена несколькими причинами, в том числе:

- сложившейся ситуацией, как в примере применения алгоритма свертки для графа ситуаций $G1$ (рис. 4);
- необходимостью для подграфа иметь вершину «своего агента» в соседнем подграфе.

В последнем случае предлагается для каждого подграфа определить свой список периферийных вершин и одну или несколько таких вершин из одного подмножества добавить в соседнее подмножество. При этом может возникнуть ситуация, когда нарушится баланс по количеству вершин в подграфах. Если такое нарушение критично, предлагается обмениваться между подграфами одинаковым количеством периферийных вершин.

С другой стороны, ситуация пересечения подграфов A и B по вершине gv_{10} , сложившаяся в случае применения алгоритма свертки [14] для графа ситуаций $G1$ (рис. 5), показывает целесообразность использования такой вершины в качестве «агента» другого подмножества. Например, по вершине gv_{10} , принадлежащей подграфам A и B на рис. 5, раньше было принято решение передачи ее подмножеству A с целью равномерного распределения вершин по подграфам. В результате получены два подмножества вершин:

$$A = \{gv_1, gv_2, gv_3, gv_4, gv_5, gv_6, gv_7, gv_8, gv_{10}\};$$

$$B = \{gv_9, gv_{11}, gv_{12}, gv_{13}, gv_{14}, gv_{15}, gv_{16}, gv_{17}, gv_{18}\}.$$

В нашем случае возможен обмен вершинами $gv_{10} \in A$ и $gv_{12} \in B$. Для этого сформируем пересечение подграфов A и B по вершинам gv_{10} и gv_{12} , как показано на рис. 6.

Теперь подмножества A и B примут вид:

$$A = \{gv_1, gv_2, gv_3, gv_4, gv_5, gv_6, gv_7, gv_8, gv_{10}, gv_{12}\};$$

$$B = \{gv_9, gv_{11}, gv_{12}, gv_{13}, gv_{14}, gv_{15}, gv_{16}, gv_{17}, gv_{18}, gv_{10}\}.$$

А вершины являются gv_{10} и gv_{12} вершинами-«агентами» для своих подмножеств.

Кроме того, рассмотренные методы и алгоритмы предполагают возможность разбиения графа на равные части, в то время как не решена задача разделения графа на пропорциональные подмножества вершин. Это может потребоваться при решении задачи разбиения больших графов, например, в случае движения по графу в различных направлениях с разной скоростью.

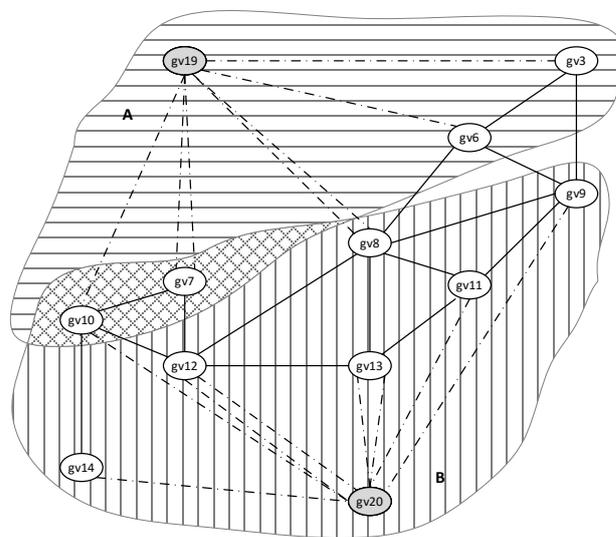


Рис. 6. Пример обмена периферийными вершинами между подграфами A и B

Таким образом, возникает необходимость формулировки новых критериев разделения графа на части, отличных от общепринятых [9].

Возможно для решения наших задач неважен критерий разбиения графа на равные части (по количеству вершин или по суммарным весам) в условиях минимизации связей между подграфами. Авторами ставится задача разделения графа на пропорциональные подмножества, например на части: $1/3$ и $2/3$. В качестве одного из возможных подходов решения поставленной задачи предлагается использование алгоритма свертки [14] с разной скоростью (для части $2/3$ алгоритм применяется с обычной скоростью v_0 , без задержек, а для части $1/3$ – со скоростью в два раза меньше v_0).

Такой подход позволяет решить задачу пропорционального разбиения графа на части. Возможны модификации такого подхода, когда выполняется разбиение графа на n равных частей, а на последнем шаге происходит объединение некоторых частей для удовлетворения критерию пропорциональности подграфов.

ЗАКЛЮЧЕНИЕ

При проектировании систем охраны сложных объектов возникает задача управления группой разнотипных компонентов таких систем в условиях противодействия некоторой другой стороны.

В качестве средства моделирования возможных ситуаций для охраняемых объектов выбрана графовая модель.

При моделировании взаимодействия групп отмечена необходимость разбиения графа на части с последующим вычислением центров этих частей. Задача разбиения графа возникает при разделении функций между компонентами систем охраны.

Приведена классификация методов разделения графов. Выполнен обзор существующих алгоритмов разделения графа на равные части, продемонстрированы результаты их выполнения на конкретных примерах.

Описан алгоритм свертки графа и предложено его использование для разделения графа на пропорциональные подмножества для решения частных задач, возникающих при моделировании графов ситуаций, что продемонстрировано на конкретном примере.

В отличие от существующих решений, предложенный алгоритм позволяет в реальном времени выполнять коррекцию результатов разбиения графа в зависимости от изменения функциональных возможностей охранной системы и противодействующей стороны. В [14] отмечается увеличение скорости выполнения алгоритма на 20-30% в зависимости от полноты графа по сравнению с алгоритмами, использующими матрицы смежности или инцидентности и последовательный перебор вершин.

Предлагаемый алгоритм решает одну из комплекса задач охранной системы. Полная

структура охранной системы и механизм взаимодействия компонентов будут описаны в последующих публикациях.

СПИСОК ЛИТЕРАТУРЫ

1. **Сергеев Н. Е., Целых А. А.** Нечеткие теоретико-графовые подходы к моделированию и анализу социосемантических сетей знаний для задач принятия решений в научной и научно-технической экспертизе // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. Краснодар: КубГАУ, 2016. №09 (123). С. 390–410. [N. E. Sergeev, A. A. Tselyh, "Fuzzy theoretical-graph approaches to modeling and analysis socio-semantic networks of knowledge for decision problems in the scientific and technical expertise," (in Russian), in *Politematicheskij setevoj ehlektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta*, no. 9 (123), pp. 390-410, 2016.]

2. **Колоденкова А. Е.** Построение системы когнитивных моделей и методов анализа реализуемости проекта по созданию информационно-управляющих систем для атомных станций // Глобальная ядерная безопасность, 2016. № 3 (20). С. 43–50. [A. E. Kolodenkova, "The construction of a system of cognitive models and methods of analyzing the feasibility of a project to create management information systems for nuclear power plants," (in Russian), in *Global'naya yadernaya bezopasnost'*, no. 3(20), pp. 43-50, 2016.]

3. **Таран Т. А., Шемаев В. Н.** Когнитивное моделирование рефлексивных процессов // Искусственный интеллект. 2004. № 2. С. 173–179. [T. A. Taran, V. N. Shemaev "Cognitive modeling of reflexive processes," (in Russian), in *Iskusstvennyj intellekt*, no. 2, pp. 173-179. 2004.]

4. **Колоденкова А. Е.** Оценка реализуемости проекта по созданию информационно-управляющих систем с применением процедуры обучения нечеткой когнитивной модели // Вестник УГАТУ, 2016. Т. 20. № 2 (72). С. 123-133. [A. E. Kolodenkova, "Project feasibility estimation to creation information-control systems with procedure application learning of fuzzy cognitive model," (in Russian), in *Vestnik UGATU*, vol. 20, no. 2 (72), pp. 123-133, 2016.]

5. **Моделирование** нейросетевых взаимодействий с использованием механизма клеточных автоматов / [А. В. Шестаков и др.] // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. Краснодар: КубГАУ, 2016. №10 (124). С. 601-610. [A. V. Shectakov, et al., "Modeling neural interactions with the use of cellular automata," (in Russian), in *Politematicheskij setevoj ehlektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta*. Krasnodar: KubSaU, no. 10 (124), pp. 601-610, 2016.]

6. **Сергеев Н. Е., Мунтян Е. Р.** О решении некоторых задач, возникающих при моделировании взаимодействия социальных групп // VIII Всероссийская научная конференция «Системный синтез и прикладная синергетика»: сборник научных трудов. Ростов-на-Дону; Таганрог: изд-во ЮФУ, 2017. С. 240-249. [N. E. Sergeev, E. R. Muntyan, "About solution of some problems arising in the modeling of the interaction of social groups," (in Russian), in

"Sistemny sintez i prikladnaia sinergetika". Rostov-on-don; Taganrog: izd-vo YuFU, pp. 240-249, 2017.]

7. **Мунтян Е. Р.** О задаче разделения графа на пропорциональные подмножества вершин // Информационные технологии и информационная безопасность в науке, технике и образовании «ИНФОТЕХ – 2017»: сб. статей всероссийской научно-технической конференции. г. Севастополь: СевГУ, 2017. С. 71-73. [E. R. Muntyan, "About the task of dividing a graph on a proportional subset of vertices," (in Russian), in *Informacionnye tekhnologii i informacionnaya bezopasnost' v nauke, tekhnike i obrazovanii "INFOTECH – 2017": Sevastopol, SSU, 2017, pp. 71-73, 2017. Available: <http://infotech.sevsu.ru/arshive/>]*

8. **Бувайло Д. П., Толлок В. А.** Быстрый высокопроизводительный алгоритм для разделения нерегулярных графов // Вісник Запорізького державного університету, 2002. № 2. С. 47-53. [D. P. Buvajlo, V. A. Tolok, "Fast high-performance separation algorithm for irregular graphs," (in Russian), in *Visnik Zaporiz'kogo derzhavnogo universitetu*, no. 2, pp. 47-53, 2002.]

9. **Шматков С. И., Толстолужская Е. Г., Артюх Ю. А.** Анализ распараллеливания алгоритма задачи оптимального разделения графа на подграфы // сб. науч. тр. Харьковского университета Воздушных Сил, Харьков, 2013. Вып. 2 (35). С. 132-134. [S. I. Shmatkov, E. G. Tolstoluzhskaya, Y. A. Artiuh, "Analysis of the parallelization of the algorithm the problem of optimal division of graph into subgraphs," (in Russian), in *sb. nauch. tr. Harkovskogo universiteta Vozdushnyh Sil, Kharkiv. journal issue 2 (35)*, pp. 132-134, 2013.]

10. **Kernighan B. W., Lin S.** An Efficient Heuristic Procedure for Partitioning Graphs // *Bell System Technical Journal*, 1970. Vol. 49. Pp. 291-307. [B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," in *Bell System Technical Journal*, Vol. 49, pp. 291-307, 1970]

11. **Fiduccia C. M., Mattheyses R. M.** A Linear-time Heuristics for Improving Network Partitions // 19th Design Automation Conference, 1982. Pp. 175-181. [C. M. Fiduccia, R. M. Mattheyses. "A Linear-time Heuristics for Improving Network Partitions," in *19th Design Automation Conference*, pp. 175-181, 1982]

12. **Ugander J., Backstrom L.** Balanced Label Propagation for Partitioning Massive Graphs // *proc. of the Sixth ACM International Conference on Web Search and Data Mining. WSDM'13. Rome, Italy: ACM, 2013. Pp. 507-516.* [J. Ugander and L. Backstrom, "Balanced Label Propagation for Partitioning Massive Graphs," in: *proc. of the Sixth ACM International Conference on Web Search and Data Mining, WSDM'13. ACM, pp. 507-516, 2013.*]

13. **Улучшение** качества разбиения графа с помощью многоуровневой оптимизации / [Р.К. Пастухов и др.] // Труды ИСП РАН, 2014. Т. 26, вып. 4. С. 21-31. [R. K. Pastuhov, et. al., "Improving the quality of graph partitioning using multilevel optimization," (in Russian), in *Trudy ISP RAN*, vol. 26(4), pp. 21-31, 2014.].

14. **Обобщение** графов ситуаций на основе спискового алгоритма свертки для задач ситуационного управления / [Н. Е. Сергеев и др.] // Известия ЮФУ. Технические науки. 2017. № 3. С. 111-121. [N. E. Sergeev, et. al., "Generalization of graph situations on the basis of list-based algorithm for the tasks of situational management," (in Russian), in *Izvestiya YuFU. Tehnicheskie nauki*, no. 3, pp. 111-121, 2017.].

ОБ АВТОРАХ

СЕРГЕЕВ Николай Евгеньевич, проф. каф. ВТ. Дипл. инженер-математик (Таганрогский радиотехнический институт им. В.Д. Калмыкова (ТРТИ), 1979). Д-р техн. наук, диссертация по специальностям «Теоретические основы информатики», «Математическое моделирование, численные методы и комплексы программ» (ТРТИ им. В.Д. Калмыкова, 2004). Иссл. в обл. систем поддержки принятия решения и автоматизированных систем управления.

МУНТЯН Евгения Ростиславна, ст. преподаватель каф. ВТ. Дипл. инженер-системотехник (ТРТИ им. В.Д. Калмыкова, 1992). Готовит дис. о моделировании взаимодействий в группах объектов.

METADATA

Title: Using convolution algorithm to separate a graph on the proportional subgraphs.

Authors: N.E. Sergeev¹, E.R. Muntyan²

Affiliation: Southern Federal University, Russia.

Email: ¹nesergeev@sfedu.ru, ²ermuntyan@sfedu.ru

Language: Russian.

Source: Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 22, no. 1 (79), pp. 121-130, 2018. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

Abstract: In article is one of the tasks arising in the modeling of processes management of complex systems of protection of objects. Proposed using a graph paradigm to consider and predict the possible situations. For visualization, was considered an example connected non-directed graph of situations, described its metric characteristics to take account of information about its structure. In article gives an overview of the well-known methods of separating the graph into equal parts, demonstrated the results of their performance. Suggested usage list-based convolution algorithm to separate a graph into proportional subsets for the solving specific problems arising while modeling of a graph of situations.

Key words: graph; model; situation; list; algorithm; presentation; absorption; division; metric characteristics.

About authors:

SERGEEV, Nikolai Evgenievich, Prof., Dept. of Computer Engineering. Dipl. engineer-mathematician (TSURE, 1979). Dr. of eng. sc. (TSURE, 2004).

MUNTYAN, Evgenia Rostislavna, Senior lecturer Dept. of Computer Engineering. Dipl. systems engineer (TSURE, 1992).