УДК 519.7

Татаринов Є.О.[1], інженер першої категорії

Спектр алгоритмів відновлення графів за допомогою блукаючого ним агентом

Досліджується процес відновлення графів. Мета дослідження – виявлення основних етапів відновлення графу за допомогою агента та оптимізація цього процесу. Пропонується спектр алгоритмів відновлення графів, які уточнюють метод відновлення графу за допомогою побудови на його вершинах неявної нумерації. Наводяться переваги та особливості цих алгоритмів.

Ключові слова: відновлення графа агентом, нумерація вершин графа, оптимізація процесів відновлення графа.

[1] Інститут прикладної математики та механіки НАНУ, 83114, м. Донецьк, вул. Рози Люксембург, 74,
e-mail: MDgerelo@yandex.ru

E.A. Tatarinov[1], engineer first category

Spectrum of graph recovery algorithms with moving agent

The graphs recovery are investigated. The main goal of the study to identify the main stages of graph recovery with the agent and optimization of this process. Offers a spectrum of graphs recovery algorithms that define recovery method by building on the graph vertices an implicit enumeration. Given the benefits and features of these algorithms.

Key Words: graph recovery with moving agent, the numbering of graph vertices, optimization of graph recovery

[1] Institute of Applied Mathematics and Mechanics NASU, 83114, Donetsk, R. Luxemburg st., 74
e-mail: MDgerelo@yandex.ru

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Введение

В настоящее время интенсивно развивается одно из направлений математической кибернетики - теория дискретных динамических систем [1 - 2]. В общей схеме Глушкова - Летичевского дискретная система представляется как модель взаимодействия управляющей и управляемой систем (управляющего автомата и операционной среды) [3]. Взаимодействие этих систем зачастую представляется как процесс перемещения управляющего автомата по графу (лабиринту) управляемой системы [1], что привело к появлению обширной и интенсивно развивающейся области исследования поведения автоматов в лабиринтах [1 - 2].

Одной из наиболее важных таких задач является задача полного восстановления графа. Решение этой задачи позволяет получить решение двух других задач: самолокализации и контроля карты [4]. Поэтому задача полного восстановления графа является важной в теоретическом и прикладном плане.

Целью данной работы является планомерное исследование процессов восстановления графов с целью их оптимизации и выявления основных этапов и параметров восстановления. Предлагается спектр алгоритмов, полученных в результате такого исследования, приводятся их преимущества и особенности.

Основные определения

Пусть K – класс простых, неориентированных, связных графов, без петель кратных ребер, и граф $G(V,E)\in K$, где V – множество вершин, мощности n, а $E\subseteq V\times V$ – множество ребер, мощности m. Обхват графа – это длина кратчайшего простого цикла. Цикломатическое число графа – это минимальное число ребер, которые надо удалить, чтобы граф стал ациклическим. Тройку $((u,v),v)$ будем называть инцидентором (точкой прикосновения) ребра (u,v) и вершины v. Множество инциденторов графа обозначим I. Пусть $L=V\cup E\cup I$ множество элементов графа G. Будем обозначать $\mu(e)$ метку элемента e. Окрестностью $O(v)$ вершины v будем называть множество элементов графа, состоящее из вершины v, всех вершин u смежных с v, всех ребер (v,u) и всех инциденторов $((v,u),v),((u,v),u)$.

Нумерация $F:V\to N$ вершин графа G – это инъективное отображение множества его вершин во множество N натуральных чисел. Прямой нумерацией называется нумерация вершин графа, соответствующая порядку их обхода при поиске в глубину. Древесными ребрами называются ребра, порождающие дерево поиска при обходе графа в глубину, остальные ребра –

обратными. Будем обозначать $T^*(n)$ и $T_*(n)$, соответственно, верхнюю и нижнюю оценку временной сложности выполнения алгоритма.

Постановка задачи

Пусть задан класс K графов все элементы которых окрашены белой краской w. Задан мобильный агент A. Он может передвигаться по графу из вершины v в вершину u по ребру (v,u). При этом он может изменить окраску вершин v,u, ребра (v,u), инциденторов $((v,u),v),((v,u),u)$ метками. Находясь в вершине v, агент A воспринимает («видит») метки всех элементов окрестности $O(v)$ и на этом основании определяет по какому ребру (v,u) он будет перемещаться и как будет перекрашивать элементы из $O(v)$. Агент A обладает конечной, неограниченно растущей внутренней памятью, в которой фиксируется результат функционирования агента на каждом шаге, и, кроме того, постепенно выстраивается представление графа G, вначале неизвестного агенту, списками ребер и вершин.

Требуется разработать такой алгоритм функционирования агента A (т.е. передвижения по графу и раскраски его элементов), что он, будучи помещен в произвольную вершину произвольного неизвестного агенту графа $G \in K$, через конечное число шагов построит граф H, изоморфный G, т.е. восстановит G.

Базовый алгоритм

Очевидно, для восстановления графа агенту необходимо обойти все вершины, ребра и установить вершины, инцидентные каждому ребру. Поэтому алгоритм восстановления графа является специальным обходом этого графа.

В качестве основы этого обхода выбран обход в глубину, так как он содержит меньшее число возвратов по графу, сравнительно с обходом графа в ширину, и разбивает множество ребер графа всего на два множества: древесные и обратные, поскольку исследуемый граф G неориентированный [5]. При обходе в глубину агент создает на вершинах исследуемого графа с помощью красок неявную прямую нумерацию [5] и на ее основе определяет инцидентность ребер вершинам. Известен ряд алгоритмов поиска в глубину для известного графа [5]. Предлагаемая ниже стратегия обладает рядом особенностей. Во - первых, граф G агенту не известен и агент на каждом шаге обладает информацией только о метках элементов из окрестности $O(v)$ рабочей вершины v. Во - вторых, при прохождении вершин графа G агент создает неявную нумерацию пройденных вершин: при первом посещении вершины u она окрашивается красным цветом и ей ставится в соответствие номер, равный по значению переменной Сч. На основе этой нумерации и происходит восстановление (распознавание) графа путем построения графа H изоморфного G. В процессе поиска агент строит неявное дерево поиска в глубину и относительно этого дерева все ребра разделяются на древесные, т.е. принадлежащие этому дереву и окрашиваемые при первом прохождении по ним в красный цвет r, и обратные – не принадлежащие дереву и окрашиваемые при первом прохождении в черный цвет b. Древесные ребра проходятся по крайне мере 2 раза и при последнем проходе окрашиваются агентом в черный цвет. Обратные – проходятся один раз.

Древесные ребра распознаются агентом при первом проходе агента по ним. Красные вершины графа G, на каждом шаге алгоритма, образуют красный путь. С помощью этого пути распознаются обратные ребра, при проходе в новую вершину красный путь удлиняется, при проходе назад – укорачивается, при распознавании обратного ребра – не изменяется. Вершина, у которой все инцидентные ребра распознаны, красится черной краской. Алгоритм оканчивает работу, когда красный путь становиться пустым, а все вершины черными. Таким образом, агент использует две краски (красную и черную) и не использует камней. Данный алгоритм подробно описан в [6] далее приведем краткое его описание.

Базовый алгоритм (БА).

Вход: граф $G \in K$ неизвестный агенту, все элементы G окрашены в w, агент помещен в произвольную вершину v.

Выход: все элементы графа G окрашены в b, агент находится в вершине v, список вершин V_H и ребер E_H графа H, изоморфного G.

Данные: v – рабочая вершина графа G, в которой находится агент, V_H, E_H – списки вершин и ребер графа H, изоморфного G. Сч – счетчик числа посещенных вершин G. $k(1)k(2)...k(t)$ – список номеров вершин красного пути, t – длина этого списка, j – счетчик, используемый для восстановления обратных ребер.

Алгоритм

1. Сч:=1; $t:=1$; $k(t):=$ Сч; $V_H:=\{1\}$; $E_H:=\varnothing$;

2. Агент красит: $\mu(v):=r$;

3. if $O(v)$ есть ребро (v,u), у которого $\mu(u,v)=w$ then goto 4 else goto 5

4. if $O(v)$ есть ребро (v,u), у которого $\mu(u,v)=w$ и $\mu(u)=\mu(v)=r$ then ВОСТ(v) else ВПЕРЕД(v)

5. if в $O(v)$ есть ребро (v,u), у которого $\mu((v,u),v)=r$ then do НАЗАД(v) goto 3 end do

6. Агент красит: $\mu(v):=b$;

ВПЕРЕД(v) – достраивание дерева и восстановление древесного ребра. НАЗАД(v) – возврат по дереву, когда все ребра, инцидентные вершине, восстановлены. ВОСТ(v) – восстановление обратного ребра.

Оценки сложности сформулируем виде теоремы.

Теорема 1. Нижняя оценка временной сложности БА $T_*(n)=\Omega(n+qg)$, а верхняя равна $T^*(n)=O(n+qt)$. Верхняя оценка емкостной сложности– $S(n)=O(n^2)$.

Здесь g – обхват графа, q – цикломатическое число, t – длина наибольшего простого цикла в графе. Агент использует две различные краски (красную и черную). Таким образом $T^*(n)=O(n^3)$.

Модификации Базового Алгоритма (БА)

Анализ выполнения этапов предложенного метода позволил найти модификации алгоритма, которые будут иметь лучшую оценку сложности. Существенным достоинством полученных модификаций является то, что они позволяют установить взаимоотношение между числом ресурсов, используемых агентами и оценками сложности алгоритма.

Модификация 1 (М1), декомпозиция агента

Первой естественной модификацией БА является разделение процессов непосредственного взаимодействия агента с графом и оперирования с внутренней памятью агента при восстановлении графа. Таким образом, проводится декомпозиция агента на два агента: агент - исследователь (АИ) и агент - экспериментатор (АЭ). Первый – передвигается по исследуемому графу, считывает и изменяет метки на элементах графа из окрестности текущей вершины, передает сообщения агенту экспериментатору и на основе разметки элементов графа выбирает дальнейшее направление движения. Второй, по сообщениям от агента АИ, строит представление исследуемого графа в виде списка смежности. Агент АЭ не передает никаких сообщений АИ. Подробно модификация описана в [7].

Достоинством этой модификации является то, что агент АИ не требует для своего функционирования хранения таблицы всего исследуемого графа или хранения информации о пройденном пути, поэтому он может быть достаточно мал, а для некоторых частных случае графов, которые будут рассмотрены далее, он может быть конечным автоматом. При применении этой модификации агенту АИ не требуется выполнять сложные вычисления, дополнительные трудоемкие действий и / или дополнительные ресурсы для пометки элементов графа. Таким образом, декомпозиция агента не ухудшает временную, емкостную сложности и не увеличивает число различных красок и камней.

К особенностям данной модификации можно отнести следующее. Для передачи сообщения от АИ к АЭ потребуется канал односторонней связи, очередь сообщений и протокол передачи данных.

Модификация 2 (М2), одновременное восстановление обратных ребер инцидентных одной вершине красного пути

В БА кубическая временная сложность возникает из - за того, что каждое обратное ребро агент восстанавливал отдельно и при этом мог совершить число шагов соизмеримое с n. Поэтому следующей модификацией стало восстановление всех обратных ребер инцидентных рабочей вершине красного пути за один проход по вершинам красного пути. В этом случае агенту потребуется пометить все такие обратные ребра. Например, если агент воспринимает 1- окрестность $O(v)$, можно поставить в вершине камень и затем двигаться в обратном направлении до начальной вершины с целью поиска ребер, которые смежны с вершиной помеченной камнем. Таким образом, считая число переходов по ребрам красного пути агент может однозначно вычислить прямой номер вершины, из которой он увидит смежную вершину помеченную камнем, а значит сможет узнать какой вершине в восстанавливаемом графе она соответствует, так же как и при восстановлении обратного ребра в БА. Подробно данная модификация описана в [8].

Теорема 2. Нижняя оценка временной сложности алгоритма равна $T_*(n)=\Omega(n)$, а верхняя – $T^*(n)=O(n^2)$, и емкостная сложность – не ухудшается. При этом используется две краски и один камень l.

Таким образом, предложенная модификация имеет верхнюю оценку временной сложности на порядок лучше, чем БА, однако требует дополнительный камень для пометки вершины, инцидентной обратным ребрам, которую агент восстанавливает.

К особенностям метода можно так же отнести проход по ребрам красного пути в прямом и обратном направлении.

Модификация 3 (М3), восстановление обратных ребер при «зацикливании»

Естественной модификацией базового алгоритма и его М2 будет восстановление ребер не при первом их появлении в текущей вершине, а при зацикливании агента. Данная модификация показывает, что порядок восстановления обратных ребер не влияет на верхнюю оценку времени восстановления графа и не требует дополнительных ресурсов по памяти и меткам.

Эта модификация изменяет в основном алгоритме только приоритет выполнения двух процедур $\text{ВОСТ}(v)$, $\text{ВПЕРЕД}(v)$. То есть, раньше, если была возможность восстановить обратное ребро, то агент его восстанавливал и только потом шел вперед, в М3 наоборот, если есть возможность идти вперед, то агент идет вперед не восстанавливая обратных ребер. Подробно модификация описана в [8].

Эта модификация не ухудшает порядка сложности алгоритма, к которому она была применена. Однако она позволяет получить новые алгоритмы восстановления графа и позволяет показать, что порядок восстановления обратных ребер не изменяет сложность алгоритма.

Модификация 4 (М4), ординарные и неординарные вершины

В БА множество ребер разбивается только на два непересекающихся подмножества: древесные и обратные. При этом обратные ребра требуют для восстановления проход по вершинам красного пути. Длина такого прохода может быть соизмерима с n. Поэтому следующей модификацией БА стало выделение во множестве обратных ребер подмножества, которое бы восстанавливалось за константное число шагов. Рассматривается разбиение множества вершин на два подмножества: ординарные и неординарные,– а множество обратных ребер на те, которые не инцидентны неординарным вершинам и те, которые инцидентны хотя бы одной неординарной вершине. Ребра инцидентные неординарным вершинам восстанавливаются за два шага, при проходе по ним агента в прямом и обратном направлении. При этом ординарной вершиной будем назвать вершину, у которой $\deg(v) \le D$, где D (степень ординарности) заранее известная константа и не зависит от n, в противном случае вершина неординарная. Ординарные вершины восстанавливаются согласно стратегии алгоритма, к которому была применена данная модификация (например, к БА или его модификациям М1 - М3). Неординарные вершины метятся камнями. В момент установки камня в неординарной вершине ей неявно присваивается следующий номер по порядку. Этот номер записывается в список соответствия в ячейку, номер которой совпадает с номером установленного камня. Такое соответствие сохраняется на протяжении выполнения всего эксперимента по восстановлению. Это позволяет без выполнения дополнительных шагов определять неявный номер неординарной вершины по номеру камня установленного в ней. Таким образом, при выполнении процедуры $\text{НАЗАД}(v)$ агент при обнаружении неординарной вершины помеченной камнем сможет вычислить ее неявный номер. После чего агент может добавить в E_H ребро, соединяющее текущую вершину с этой неординарной вершиной. Заметим, что выполнять поиск вершин помеченных камнем можно не только при выполнении процедуры $\text{НАЗАД}(v)$, но и при выполнении процедуры $\text{ВПЕРЕД}(v)$. Подробно данная модификация описана в [8]. Оценку сложности выполнения модификации сформулируем виде теоремы.

Теорема 3. Верхняя оценка временной сложности при применении М4 не ухудшается, верхняя оценка емкостной сложности равна $O(pn)$. При этом используется две краски и, дополнительно, не более чем p камней. Здесь p – количество неординарных вершин в графе.

Достоинством этой модификации является то, что агенту необходимо анализировать окрестности только ординарных вершин, что займет меньше времени и конечную память, чем анализ окрестности неординарных вершин. Если же применить М4 вместе с М1, то агент - исследователь может быть конечным автоматом, поскольку ему необходимо будет анализировать окрестность, в которой будет не более чем D ребер, $2D$ инциденторов и $D+1$ вершины.

К особенностям этой модификации можно отнести то, что рассматривается более узкий класс графов, чем в базовом алгоритме и требуются дополнительные камни для установки в неординарных вершинах.

Модификация 5 (M5), нумерация вершин набором камней

Пусть у агента имеется p попарно различных камней. Нумерацию вершин агент будет реализовывать неявно, при помощи установки в вершине одного камня. При этом для каждого камня i существует свой счетчик числа непомеченных ребер, которые есть в окрестности текущей вершины v, перед тем как агент установит в ней камень i. Когда агент «видит» в окрестности текущей вершины смежную вершину, помеченную камнем i, его счетчик уменьшается на единицу. Когда счетчик обнулится, камень станет свободным и агент перейдет в вершину, снимет с нее камень i, вершину пометит краской b и вернется обратно. Агент использует для пометки вершины свободный на данный момент камень с наименьшим номером. Камень считается свободным, если его счетчик равен нулю. Перед началом восстановления счетчики всех камней равны нулю. В момент установки камня i вершине неявно присваивается ее номер по порядку $1,2,3$ – ее неявный прямой номер. Камень i ассоциируется с прямым номером, который неявно был присвоен вершине в момент установки в ней камня i. Такая связь сохраняется до тех пор, пока камень не станет свободным. Модификация подробно описана в [9]. Оценку сложности сформулируем в виде теоремы.

Теорема 4. Верхняя оценка временной сложности M5 базового алгоритма $T(n)=O(n)$, а верхняя оценка емкостной сложности не изменяется. При этом используется две различные краски и h_t камней, где h_t - максимальная высота дерева при обходе графа в глубину.

К достоинствам этой модификации можно отнести совпадение верхней и нижней оценки временной сложности. Для любого графа из класса K, существуют классы графов для восстановления, которых агент использует константное число камней.

К особенностям этой модификации следует отнести достаточно большое число камней, которые требуются для выполнения алгоритма, поскольку в общем случае h_t соизмеримо с n. Кроме того, агент проводит дополнительный анализ в вершине при восстановлении обратного ребра и при снятии камня с вершины, в частных случая это может потребовать дополнительного времени.

Выводы

Предлагается Базовый Алгоритм и его модификации, образующие спектр алгоритмов. Модификации позволяют понизить верхнюю оценку временной сложности и выявить основные этапы восстановления графа. Однако, выполнение модификаций требуют от агента использования дополнительных ресурсов. Это позволяет установить определенную зависимость времени выполнения алгоритма от числа используемых агентом ресурсов.

Список використаних джерел

1. *G. Kilibarda, V. B. Kudryavtsev, and S. M. Ushchumlich* Collectives of automata in labyrinths // Discrete mathematics. – 2003, 15, no. 3, C. 3–40. (in Russian).
2. *G. Kilibarda, V. B. Kudryavtsev, and S. M. Ushchumlich* Collectives of automata in labyrinths // Discrete mathematics. – 2003, 15, no. 2, C. 3–39. (in Russian).
3. *Kudryavtsev V.B., Aleshin S.V., Podkolzin A.S.* Intorduction to the Automata Theory. M.: Nauka, 1985. 320 p. (in Russian).
4. *Dudek G., Jenkin M., Milios E., Wilkes D.* Map validation and Robot Self-Location in a Graph-Like World // Robotics and Autonomous Systems. – 1997. – Vol. 22.
5. *Evstigneev V.A.* Application of graph theory to computer programming. M: Nauka, 1985. 352 p. (in Russian).
6. *Grunsky I.S., Tatarinov E.A.* Grpah reconstruction with moving agent // Bulletin of Donetsk National University. Series A. Natural Sciences. – 2009. – no 1. – P. 492–497. (in Russian).
7. *Grunsky I.S., Tatarinov E.A.* Graph reconstructiong algorithm // Proc. of 4th "PACO'2008" Intern. Conf. "Parallel Computations and Control Problems", Moscow, V. A. Trapeznikov Institute of Control Sciences, RAS, October 27-29, 2008. - P. 1483-1498. (in Russian).
8. *Tatarinov E.A.* Basic algorithm for reconstructing graph // Proceedings of the Institute of Applied Mathematics and Mechanics. – 2010. – V. 21. – P. 216-227.
9. *Tatarinov E.A.* Graph reconstruction with stones / / Scientific Papers "Problems of Applied Mathematics, Mathematical Modeling." - Izd. Dnepropetrovsk National. univ., - 2011. - P. 232 - 255.

Надійшла до редколегії 03.12.2012