УДК 004.8+519.5

ОПТИМИЗАЦИЯ ПРОЦЕССОВ ВОССТАНОВЛЕНИЯ ГРАФА

Татаринов Е.А., Грунский И.С.
Институт прикладной математики и механики НАНУ, г. Донецк

1 Введение

В настоящее время интенсивно развивается одно из направлений математической кибернетики – теория дискретных динамических систем. В общей схеме Глушкова – Летичевского дискретная система представляется как модель взаимодействия управляющей и управляемой системы (управляющего автомата и операционной среды) [1]. Взаимодействие этих систем зачастую представляется как процесс перемещения управляющего автомата (агента) по графу (лабиринту) операционной среды [2].

5

Общая проблема движения в графе операционной среды содержит ряд задач, среди которых важное место занимает проблема восстановления ранее неизвестного графа операционной среды. Аналогичные проблемы возникают в робототехнике как проблемы планирования движения робота в своей операционной среде и при анализе и верификации программ. Данная работа посвящена построению базового метода восстановления графа и анализу его модификаций.

2 Основные определения

В статье под графом G понимается конечный, неориентированный, связный граф без петель и кратных ребер. Пусть $G(V,E)$ - граф, у которого V – множество вершин, мощность $n \geq 1$ и $E \subseteq V^2$ – множество ребер. Здесь V^2 - множество всех двух элементных подмножеств множества V. Мощности соответствующих множеств будем обозначать так $n = |V|$, а $m = |E|$. Если $e = (u,v)$ – ребро графа, то пары $\{u,e\}$, $\{v,e\}$ - называются инциденторами ребра e. При этом говорят, что инцидентор $\{u,e\}$ примыкает к вершине u. Через I_u обозначается множество всех инциденторов примыкающих к u. Будем называть 1 – окрестностью вершины u инциденторы I_u и множество всех вершин смежных с u. Через $\mu(v)$ и $\mu(v,e)$ обозначим цвет вершины v и цвет инцидентора (v,e) соответственно. S будем называть вершину, в которой в данный момент находится агент, а $\delta(S,e)$– вершину смежную с S через ребро e.

Будем говорить что граф G изоморфен графу G' тогда и только тогда, когда существует сюрьективное отображение $\varphi : V \to V'$, для которого $(v_1, v_2) \in E$ тогда и только тогда когда $(\varphi(v_1), \varphi(v_2)) \in E'$. Нумерацией F вершин графа $G(V,E)$ называется [3] инъективное отображение множества вершин графа V во множество N натуральных чисел, $F : V \to N$. Номером вершины p в нумерации F обозначается $F(p)$. М-нумерацией вершин графа называется [4] нумерация вершин графа, соответствующая порядку их обхода при поиске в глубину. N-нумерацией вершин графа называется [3] нумерация вершин графа, соответствующая порядку их обхода при поиске в ширину. Элементом графа будем назвать элементарную часть графа, которую агент может выделить (различить) находясь в вершине графа. Примером элементов графа могут быть вершины и инциденторы графа. Метка – существующий в данный момент способ раскраски либо элемента графа, либо совокупности элементов графа. В каждой конкретной прикладной задаче способ реализации меток обуславливается сенсорными возможностями агента и/или свойствами элементов графа.

Все неопределяемые понятия общеизвестны и их можно найти в [3].

3 Постановка задачи

Задан граф G неориентированный, конечный, связный, без петель и кратных ребер, вершины и инциденторы которого можно метить специальными красками. Задан агент, который может передвигаться из вершины в вершину неизвестного ему графа G по ребру, соединяющему их, воспринимать и анализировать некоторую локальную информацию об 1 – окрестности вершины, в которой он находится. Агент обладает ресурсом для изменения цветов элементов графа, для реализации меток вершин, конечной, но бесконечно наращиваемой памятью, в которую он будет записывать список ребер графа H. Предполагается, что метки на элементах графа могут изменяться только агентом, при этом старая метка элемента графа уничтожается, после чего на него наносится новая. Если элемент графа метится камнем, то краска на элементе графа не уничтожается, но становится недоступной для восприятия агента, до тех пор, пока агент не снимет камень с элемента графа. Изначально предполагается, что все элементы графа G не помечены. Агент помещается в произвольную вершину графа G.

5

Необходимо разработать такой метод обхода графа G, раскраски его элементов, сбора информации о нем, чтобы можно было построить граф H изоморфный графу G с точностью до меток на вершинах графов.

4. Базовый метод восстановления графа.

Базовый метод восстановления графа подробно описан в работе [4]. В нем выделены два основных этапа распознавание графа «Обход графа» и «Восстановление графа», которые могут выполняться последовательно или параллельно. Напомним основные сведения о нем.

Идея: Обойти граф «в глубину», строя при этом неявную *M*-нумерацию, на вершинах графа *G*, и неявное дерево поиска в глубину, при помощи неявной *M*-нумерации восстанавливать обратные ребра при первом их появлении в вершине, в которой находится агент, а древесные при первом прохождении по ним. Неявность нумерации заключается в том, что явно номер не ставится на вершину или другой элемент графа, а вычисляется на основе меток на элементах графов.

Метод: Предлагаемый метод основан на стратегии поиска в глубину. Эта стратегия такова: идти «вглубь графа», пока это возможно, возвращаясь назад искать другой путь с еще не посещенными ребрами и вершинами. Предлагаемая ниже стратегия обладает рядом особенностей.

1. Граф *G* агенту неизвестен и агент на каждом шаге обладает информацией о красках на элементах из 1- окрестности вершины, в которой он находится.
2. При прохождении графа агент создает неявную *M*-нумерацию пройденных вершин графа *G*: при первом посещении некоторой вершины она окрашивается красным цветом и ей фактически ставится в соответствие следующий номер по порядку, который будет соответствовать номеру вершине в графе *H*, которая будет соответствовать только что посещенной вершине в графе *G*.
3. «Красные вершины графа» *G*, на каждом шаге образуют простой путь «красный путь», который используется для вычисления *M*-номера вершины, которой инцидентно восстанавливаемое обратное ребро. Вершина, у которой все обратные ребра восстановлены, красится черным и удаляется из красного пути.
4. В процессе поиска в глубину агент строит неявное дерево поиска в глубину и относительно этого дерева все ребра разделяются на два множества: древесные и обратные,- т.е. ребра, не принадлежащие этому неявному дереву поиска в глубину и соединяющие две вершины красного пути.
5. Древесные ребра проходятся по крайней мере два раз, в прямом и обратном направлении, при первом прохождении инцидентор древесного ребра метится красной краской, а при последнем посещении черной краской.
6. Обратные ребра проходятся один раз и при первом посещении оба их инцидентора метятся черной краской. При восстановлении обратных ребер агент совершает вычисление неявного номер вершин, которым оно инцидентно. Для этого агент совершает переход по

5

обратному ребру и идет по вершинам красного пути в его конец, подчитывая количество переходов. Номер второй вершины соответствует *M*-номеру конца красного пути.

7. Алгоритм заканчивает работу, когда все вершины помечены черной краской и красный путь пуст.

Для данного метода случае приоритет операций «Исследовать» - «Вперед» - «Назад».

Терема 1. Временная сложность алгоритма, реализующего базовый метод восстановления графа, является кубической функцией от числа вершин в исследуемом графе. При этом агент использует две краски и квадратичное от числа вершин в исследуемом графе количество ячеек памяти.

Утверждение 1. Нижней оценкой временной сложности является линейная функция от числа вершин в исследуемом графе.

5

5 Модификации базового метода восстановления графа

Базовый метод модифицируется путем увеличения приоритета операции «Вперед». В этом случае приоритет операций «Вперед» - «Исследовать» - «Назад» и агент не начинает исследовать обратные ребра вершины до тех пор, пока не попадет в вершину, из которой он может попасть только в ранее посещенные. Таким образом становится возможным восстановление всех обратных ребер за один проход по красному пути, однако агент предварительно должен их пометить специальной краской, а после двигаться по «красному пути» до стартовой вершины, а затем - возвращается в исследуемую.

Терема 2. Временная сложность алгоритма, реализующего базовый метод восстановления графа, является квадратичной функцией от числа вершин в исследуемом графе. При этом агент использует три краски и квадратичное от числа вершин в исследуемом графе количество ячеек памяти.

Рассмотренная выше модификация имеет существенный недостаток, поскольку при исследовании обратных ребер агент проходит весь «красный путь» в прямом и обратном направлении.

Избежать полного прохода по «красному пути» при исследовании можно в том случае, если агент будет знать количество обратных ребер, которое ему надо будет восстановить. В результате чего он сможет завершить проход по «красному пути» не доходя до его начала, а в тот момент, когда будут восстановлены все обратные ребра. Данная модификация не изменяет приоритет выполнения операций, однако накладывает на граф ограничение – все его вершины должны быть ординарными, т.е. степени не более чем D, где D - заранее известная константа. Это ограничение позволяет получить линейность верхней оценки временной сложности, на классах графов и сохранять ее при применении к ним операций описанных ниже. Подробна данная модификация рассмотрена в [5].

6 Классы графов и операции над ними

Класс T - деревья.

Класс T_D - деревья с ограниченной степенью вершин. Будем говорить, что граф $G(V,E) \in T_D$, если граф G является деревом и для $\forall v \in V\left(G\right)$ выполняется соотношение $\deg(v) \leq \mathrm{D}$.

Класс K_0^c графов, у которых цикломатическое число равно χ. Будем говорить, что граф $G \in K_0^{\chi}$, если $G \in \mathrm{K}_0$ и $\chi(G) = \chi$, где χ – цикломатическое число графа.

Класс T^k - класс квази – деревьев. Будем говорить, что граф $G \in T^k$, если $G = \left(G_0 O_5 \left(...G_{k-1} O_5 \left(G_k\right)\right)\right)$, $k \in N$, $u_0,..,u_{k-1}$ – вершины сочленения, если отождествив G_j, $j = 0,..,k$ с вершинами, а вершины $u_0,..,u_{k-1}$ с ребрами мы получим граф вида дерево.

Класс S^k - класс квази – кольцо. Будем говорить, что граф $G \in S^k$, если $G' = \left(G_0 O_5 \left(...G_{k-1} O_5 \left(G_k O_5 G_0\right)\right)\right)$, $k \in N$,

$u_0,..,u_k$ – вершины сочленения, если отождествив G_j, $j=0,..,k$ с вершинами, а вершины $u_0,..,u_k$ с ребрами мы получим граф вида – цепочка вершин соединенных по кругу.

Операция O_1 – операция добавления неординарной вершины к графу.

Операция O_2 – удаление неординарной вершины.

Операция O_3 – добавление висячей вершины без нарушения ординарности

Операция O_4 – введение вершины в ребро.

Операция O_5 – соединение двух графов через вершину сочленения.

Применение к графу G принадлежащему некоторому классу одной или нескольких операций порождает новый класс графов. Это связано с тем, что при выполнении операций в них не используются все вершины графа, а только их часть. Таким образом, варьируя вершины графа, используемые в операциях, мы получим множество различных графов, которые объединим в класс.

5

7 Выводы

Предложен базовый метод восстановления ранее неизвестного графа с точностью до меток на элементах графов, при помощи передвигающемуся по нему агента. Данный метод использует две краски, и имеет нижнюю и верхнюю оценку временной сложности выполнения метода, которая является линейной и кубической функцией от числа вершин в восстанавливаемом графе. Базовый метод допускает модификации, которые позволяют получить лучшую оценку временной сложности в зависимости от приоритета операций, которые выполняет агент, однако, накладывают ограничения на восстанавливаемый граф. Найдены такие операции над графами, что результирующий граф имеет верхнюю оценку временной сложности не хуже, чем исходный граф.

Литература

[1] Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. - М: Наука, 1985, – 320 с.

[2] Килибарда Г., Кудрявцев В.Б., Ушчумлич Щ. Независимые системы автоматов в лабиринтах // Дискретная математика, 2003, Т. 15, вып. 2, С.3-39.

[3] Касьянов В.Н., Евстигнеев В.А. Графы в программировании, визуализация и применение – СПБ.: БХВ - Петербург, 2003. – 1104 с.

[4] Вестник Донецкого Национального Университета. Грунский И.С. , Татаринов Е.А. Распознавание конечного графа блуждающим по нему агентом. Вестник Донецкого университета. Серия А. Естественные науки -, 2009, вып. 1. – С. 492-497.

[5] Грунский И.С., Татаринов Е.А. Алгоритм распознавания графов // Труды 4 междунар. конф. «Параллельные вычисления и задачи управления» PACO’2008, Москва, Ин-т Проблем Управления им. В.А. Трапезникова РАН, 2008 – С. 1483-1498.