

АЛГОРИТМ РАСПОЗНАВАНИЯ КОНЕЧНЫХ ГРАФОВ КОЛЛЕКТИВОМ АГЕНТОВ

А.В. Стёпкин.

Институт прикладной математики и механики НАН Украины

E-mail: stepkin.andrey@rambler.ru

Аннотация

Стёпкин А.В. Алгоритм распознавания конечных графов коллективом агентов.

Рассматривается задача распознавания неизвестного графа коллективом агентов. Два агента-исследователя одновременно передвигаются по графу, считывают и изменяют метки на элементах графа, передают информацию агенту-экспериментатору, который и выполняет восстановление графа. Предложен алгоритм, который распознает любой конечный неориентированный граф. Для распознавания графа каждому агенту требуется 2 различные краски (всего 3 краски), квадратическое (от числа вершин графа) число шагов и квадратичная память. Метод основан на методе обхода графа в глубину.

Введение. В настоящее время интенсивно развивается такое направление математической кибернетики, как теория дискретных динамических систем [1]. В общей схеме Глушкова – Летичевского такая система представляется в виде модели взаимодействия управляющей и управляемой систем [2]. Подобное взаимодействие рассматривалось в [3], в предположении, что оно представлено передвижением одного агента-исследователя (АИ) по неизвестному графу и обменом данными с агентом-экспериментатором (АЭ), который и производил восстановление графа по данным, полученным от АИ. На мой взгляд, мало исследована возможность и сложность распознавания графов коллективом агентов.

Постановка задачи. Необходимо разработать такой алгоритм функционирования АИ и АЭ, что АИ, будучи помещены в произвольные, но не совпадающие вершины произвольного неизвестного агентам-исследователям и агенту-экспериментатору графа G , все элементы которого окрашены цветом w , через конечное число шагов обойдут его, передавая АЭ информацию. АЭ, в свою очередь, используя эту информацию, через конечное число шагов распознает граф G .

Решение задачи. Данная работа посвящена исследованию нашей проблемы, в предположении, что интересующее нас взаимодействие представляется процессом перемещения двух АИ A и B по неизвестной среде, заданной конечным графом [2] (АИ могут окрашивать вершины, ребра и инциденторы графа, воспринимать эти отметки и на их основании осуществлять перемещение), и обменом данными с АЭ (восстанавливает граф, по данным полученным от АИ, а также передает АИ информацию, необходимую для их дальнейшего функционирования).

В работе предложен алгоритм построения маршрутов АИ по графу, позволяющих АЭ точно восстановить граф среды. Для этого у каждого АИ есть две краски: у A это r и b , у B – u и v . При описании алгоритма используются результаты и обозначения из [4,5].

Рассматриваемый алгоритм основан на стратегии поиска в глубину. Эта стратегия такова: агенты идут «в глубину», пока это возможно, возвращаются назад, ищут другой путь с еще не посещенными вершинами и не пройденными ребрами. В случае обнаружения смежной вершины, окрашенной в «чужой» цвет, агент метит все перешейки из текущей вершины в чужую область и сообщает второму АИ, через агента-экспериментатора, о необходимости распознавания помеченных перешейков. Пока второй АИ распознает перешейки, первый АИ не может метить новые перешейки. В случае отсутствия других возможных вариантов перемещения, кроме как пометить новый перешеек, первый АИ

останавливается до того момента, пока второй АИ не распознает все помеченные перешейки.

Предлагаемая стратегия обладает рядом особенностей: 1) Граф G агентам не известен. 2) При обходе графа G , агенты создают неявную нумерацию пройденных вершин: при первом посещении вершины она окрашивается агентом A в красный цвет (в желтый цвет в случае агента B) и ей фактически ставится в соответствие номер, равный значению переменной $Sч_A$ для агента A ($Sч_B$ для агента B). Отметим, что $Sч_A$ и $Sч_B$ принимают соответственно нечетные и четные значения. На основе нумерации и происходит восстановление графа путем построения графа H изоморфного графу G . В процессе обхода агенты строят неявные деревья поиска в глубину. Относительно этих деревьев все ребра разделяются на древесные (окрашиваются при первом прохождении по ним красным либо желтым цветом), обратные (не принадлежат дереву и окрашиваются при первом прохождении в черный цвет) и ребра перешейки (соединяют деревья, построенные разными агентами). Древесные ребра проходятся как минимум 2 раза и при последнем проходе окрашиваются агентами в черный цвет. Обратные ребра проходятся не более двух раз. А вот ребра перешейки проходятся каждым агентом по два раза: первый прошедший по нему агент метит перешеек, окрашивая его в красный цвет в случае агента A (в желтый цвет в случае агента B), второй прошедший по нему агент распознает перешеек и красит его в черный цвет.

Красные (желтые) вершины графа G , на каждом шаге алгоритма, образуют красный (желтый) путь. При проходе в новую вершину красный (желтый) путь удлиняется, при проходе назад – укорачивается, при распознавании обратных ребер и перешейков – не изменяется. Вершина, у которой все инцидентные ребра распознаны, окрашивается в черный цвет. Алгоритм заканчивает работу, когда красный и желтый пути становятся пустыми, а все вершины черными.

Рассмотрим подробнее алгоритм обхода графа.

В работе АИ можно выделить 5 режимов:

1. *Обычный режим.* АИ движется вперед по белым вершинам, окрашивая вершины, соединяющие их ребра и дальние инциденторы в «свой» цвет. Если нет возможных путей перемещения, то АИ возвращается назад, окрашивая пройденные вершины, ребра и ближние инциденторы в черный цвет. Вернувшись в начальную вершину, АИ завершает работу. На каждом шаге АИ обменивается данными с АЭ.

2. *Распознавание обратных ребер.* Если при движении вперед было обнаружено обратное ребро, то агент прекращает работу в обычном режиме и переключается в режим распознавания обратных ребер. АИ метит все обратные ребра из текущей вершины (окрашивая ближние инциденторы обратных ребер в «свой» цвет), возвращается назад по своему пути, до обнаружения ближайшей вершины, инцидентной помеченному обратному ребру. Далее АИ переходит по этому ребру, окрашивая его в черный цвет, проверяет, нет ли ещё нераспознанных обратных ребер, помеченных рассматриваемым агентом. В результате поиска возможны два случая:

2.1 *Остались нераспознанные обратные ребра.* В этом случае АИ возвращается по обратному ребру (по которому на предыдущем шаге он перешел в текущую вершину), окрашивая ближний инцидентор в черный цвет, и продолжает движение назад для распознавания оставшихся обратных ребер.

2.2 *Не осталось нераспознанных обратных ребер.* В этом случае АИ окрашивает ближний инцидентор обратного ребра (по которому на предыдущем шаге был совершен переход в текущую вершину) в черный цвет. Далее АИ переключается в обычный режим работы.

На каждом шаге АИ обменивается данными с АЭ.

3. *Пометка перешейков.* Если в процессе обхода графа был обнаружен перешеек, то при условии, что все ранее помеченные данным АИ перешейки были распознаны, агент

переключается в режим пометки перешейков. В этом режиме АИ переходит по перешейку в чужую область, окрашивая ребро и дальний инцидентор в «свой» цвет. На следующем шаге возвращается по этому перешейку, ничего не окрашивая, и ищет другие перешейки из этой вершины. Пометив все перешейки, АИ сообщает об этом АЭ, который в свою очередь, дает команду второму АИ о необходимости распознавания помеченных перешейков. По завершению режима пометки перешейков АЭ содержит информацию о количестве помеченных перешейков.

4. *Распознавание перешейков.* Получив от АЭ команду о необходимости распознавания перешейков, АИ переключается в режим распознавания перешейков. Если в этот момент агент работает в режиме распознавания обратных ребер, то АИ переключится в режим распознавания перешейков, лишь по завершению работы над распознаванием обратных ребер. В этом режиме АИ возвращается назад по своему пути до обнаружения ближайшей вершины, инцидентной помеченному перешейку. Под помеченным перешейком понимается ребро, у которого ближний инцидентор, ребро и дальняя вершина этого ребра окрашены в «чужой» цвет. Далее возможны два случая:

4.1 *Помечено один перешеек.* АИ переходит по нему в чужую область, окрашивая перешеек в «свой» цвет, а дальний инцидентор в черный. На следующем шаге возвращается по этому перешейку в свою область, окрашивая дальний инцидентор и перешеек в черный цвет. Далее движется вперед по своему пути, пока не вернется в вершину, в которой было произведено переключение в режим распознавания перешейков.

4.2 *Помечено не менее двух перешейков.* АИ переходит по первому найденному помеченному перешейку в чужую область, окрашивая его в «свой» цвет, а дальний инцидентор в черный. На следующем шаге АИ возвращается по этому перешейку в свою область, окрашивая его в черный цвет, а оба инцидентора в красный. Далее АИ движется назад по своему пути, пока не будет найден следующий помеченный перешеек. В данной ситуации возможны два случая: а) *Следующий помеченный перешеек не последний.* АИ переходит по найденному перешейку, окрашивая его в «свой» цвет, а дальний инцидентор в черный. На следующем шаге АИ возвращается по этому перешейку в свою область, окрашивая его и оба инцидентора в черный цвет. И снова возвращается назад по своему пути до следующего помеченного перешейка. б) *Следующий помеченный перешеек последний.* АИ переходит по найденному перешейку, окрашивая его в «свой» цвет, а дальний инцидентор в черный. На следующем шаге АИ возвращается по этому перешейку в свою область, окрашивая его в черный цвет, а оба инцидентора в красный.

Далее АИ переходит по последнему перешейку в чужую область, окрашивая инциденторы в черный цвет. На следующем шаге АИ переходит по первому распознанному перешейку в свою область, окрашивая инциденторы в черный цвет. Далее АИ возвращается в вершину, в которой было произведено переключение в режим распознавания перешейков.

5. *Одновременное попадание двух АИ в одну белую вершину.* При одновременном попадании двух АИ в одну белую вершину, каждый АИ окрашивает вершину наполовину, и она становится красно-желтой. Агент *В* на следующем шаге отступает назад по своему пути и переходит в режим пометки перешейков (при этом ребро, по которому он вернулся уже посчитано как первый перешеек, а длина желтого пути уменьшена на одну вершину). Агент *А* видит разноцветную вершину как свою, но при распознавании окрашивает в черный цвет обе половинки.

Рассмотрим алгоритм восстановления графа.

Восстановление графа производит АЭ, на основе сообщений, которые он на каждом шаге получает от АИ. Агенты *А* и *В* записывают свои сообщения в списки *М* и *Н* соответственно. Далее эти списки обрабатываются АЭ независимо друг от друга. На каждом шаге алгоритма может быть обработано не более чем по одной записи из каждого списка.

Рассмотрим подробно процедуры АЭ, которые в конечном итоге и формируют списки ребер и вершин распознаваемого графа.

ВПЕРЕД_А(): $Cч_A := Cч_A + 2$; $t := t + 1$; $r(t) := Cч_A$; $V_H := V_H \cup \{Cч_A\}$; $E_H := E_H \cup \{(r(t-1), r(t))\}$;

ВПЕРЕД_АВВ(): $E_H := E_H \cup \{(N_B, r(t-i))\}$;

ВПЕРЕД_ОР_А(): $KOBRA := KOBRA - 1$; $UDOBRA := (KOBRA = 0)$; $E_H := E_H \cup \{(r(t), r(t-i))\}$;

ОБР_СП_В() аналогична рассмотренной процедуре *ОБР_СП_А()*, только добавлено условие:

ВОЗВРАТ_В(): $E_H := E_H \setminus \{(y(p-1), y(p))\}$; $V_H := V_H \setminus \{Cч_B\}$; $Cч_B := Cч_B - 2$; $p := p - 1$;

$y(p) := Cч_B$; $L := 1$; $K := K + 1$.

Процедуры работы со списком команд от агента *В*, которые не рассмотрены выше, аналогичны процедурам работы со списком команд от агента *А*.

Доказана корректность рассмотренного алгоритма.

Анализируя построенный алгоритм, получаем следующие теоремы:

Теорема 1. Выполняя алгоритм распознавания, агенты распознают любой граф *G* с точностью до изоморфизма.

Теорема 2. Временная и емкостная сложности алгоритма равны $O(n^2)$, где *n* – число вершин графа, при этом алгоритм использует 3 краски.

Выводы.

В работе предложен новый алгоритм распознавания графа среды временной и емкостной сложности $O(n^2)$. АИ имеют конечную память, независимую от *n*, и используют по две краски каждый (всего три краски). В отличие от [3-6] оптимизированы алгоритмы распознавания обратных ребер и перешейков, что позволило улучшить временную сложность алгоритма до $O(n^2)$, где *n* – число вершин графа.

Литература.

1. Кудрявцев В.Б., Ушчумлич Щ., Килибарда Г. О поведении автоматов в лабиринтах. Дискретная математика, 1992, т. 4, № 3, стр. 3-28.
2. Кудрявцев В.Б., Алешин СВ., Подколзин А.С. Введение теорию автоматов. - М: Наука, 1985,-320 с.
3. Грунский И. С., Татаринев Е. А. Распознавание конечного графа блуждающим по нему агентом. Вестник Донецкого университета. Серия А. Естественные науки -, 2009, вып. 1. – с. 492 – 497.
4. И.С. Грунский, А.В. Стёпкин Распознавание конечного графа коллективом агентов. – Труды ИПММ НАН Украины. – 2009. – Т.19. – С. 43-52.
5. А.В. Стёпкин Распознавание графов коллективом агентов. // Мат. IV міжнар. Наук.-практ. конф. «Сучасна інформаційна Україна: інформатика, економіка, філософія» 2010, Донецьк, – т.1 – с. 139-143.
- 4-6. Стёпкин А.В. Грунский И.С. Распознавание конечного графа коллективом агентов // Мат. V междунар. Науч.-тех. конф. «Информатика и компьютерные технологии» 2009, Донецк, ДонНТУ – т.2 – с. 126-131.

Отформатировано: Отступ:
Выступ: 1,25 см, нумерованный +
Уровень: 1 + Стиль нумерации: 1, 2,
3, ... + Начать с: 1 + Выравнивание:
слева + Выровнять по: 0,63 см +
Табуляция после: 1,27 см + Отступ:
1,27 см