

Сотрудник факультета экономики и управления СГПУ

Возможность и сложность распознавания конечного графа коллективом агентов

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

1. Введение

Основной проблемой компьютерной науки является проблема взаимодействия управляющей и управляемой систем (управляющего автомата, агента и его операционной среды). В рассмотренном ниже случае взаимодействие этих систем представляется как процесс перемещения агентов по графу среды. Целенаправленное перемещение агента в его операционной среде невозможно без формирования достаточно полной модели среды. В вопросе моделирования операционных сред определен ряд подходов, одним из которых является топологический. В этом случае агенту доступна информация только о связях между различными областями среды и недоступна метрическая и алгоритмическая информация о среде. Зачастую подобная ситуация возникает в роботике [1]. Топологические модели представляют собой графы с помеченными различными способами вершинами, дугами, инциденторами.

В настоящее время выделились три основные задачи исследования среды агентом: 1) самолокализация агента; 2) контроль соответствия модели среды и самой среды (проверка карты); 3) построение агентом модели карты среды [1]. Задача распознавания среды (построения карты) с помощью тех или иных средств агента находится в центре внимания и ей посвящено значительное количество работ. Однако на наш взгляд, мало исследована возможность и сложность распознавания графа более чем одним агентом [2].

В настоящей работе предложен новый алгоритм распознавания графа в котором используется три агента: два агента-исследователя (АИ), которые перемещаются по графу, и агент - экспериментатор (АЭ), который принимает сообщения АИ и по ним строит представление исследуемого графа, причем, основное внимание уделено анализу того, что и как распознается на каждом шаге алгоритма.

2. Основные определения и обозначения

2.1 Раскрашенные графы

Рассматриваются конечные, неориентированные графы без петель и кратных ребер. Все неопределяемые понятия общеизвестны. Пусть $G=(V,E)$ - граф, у которого V - множество вершин, E - множество ребер, т.е. двухэлементных подмножеств (u,v) , где $u,v \in V$. Тройку $((u,v),v)$ будем называть инцидентором (“точкой прикосновения”) ребра (u,v) и вершины v . Множество таких троек обозначим I . Множество $L=V \cup E \cup I$ назовем множеством элементов графа G . Функцией раскраски графа G назовем отображение $\mu:L \rightarrow \{w,r,y,b\}$, где w интерпретируется как белый цвет (краска), r - красный, y - желтый, b - черный. Пара (G,μ) называется раскрашенным графом.

Последовательность u_1, u_2, \dots, u_k попарно смежных вершин называется путем в графе G , а k - длина пути. При $u_1 = u_k$ этот путь называется циклом. 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 $((v,u),v), ((v,u),u)$. Мощность множеств вершин V и ребер E обозначим через n и m соответственно. Ясно что $m \leq \frac{n(n-1)}{2}$. Изоморфизмом графа G и графа H назовем такую биекцию $\varphi:V_G \rightarrow V_H$, что $(v,u) \in E_G$ точно тогда, когда $(\varphi(v),\varphi(u)) \in E_H$. Таким образом, изоморфные графы равны с точностью до обозначения вершин и раскраски их элементов.

2.2 Мобильные агенты

Мобильные агенты A и B передвигаются по графу из вершины v в вершину u по ребру (v,u) . При этом агенты могут изменять окраску вершин v,u , ребра (v,u) , инциденторов $((v,u),v), ((v,u),u)$. Находясь в вершине v , агенты A и B воспринимают (“видят”) метки всех элементов окрестности $O(v)$ и на этом основании определяют по какому ребру они будут перемещаться и как будут перекрашивать элементы графа. АЭ передает, принимает и идентифицирует сообщения АИ, обладает конечной, неограниченно растущей внутренней памятью, в которой фиксируется результат функционирования АИ на каждом шаге, и, кроме того, постепенно выстраивается представление графа G вначале неизвестного агентам, списками ребер и вершин.

2.3 Постановка задачи

Требуется разработать такой алгоритм функционирования АИ (т.е. передвижения по графу, раскраски его элементов и передачи информации АЭ) и АЭ (восстановления графа по данным полученным от АИ), что АИ будучи помещены в произвольные вершины произвольного неизвестного АИ и АЭ графа G , все элементы которого окрашены цветом w , через конечное число шагов обойдут его, пошагово передавая АЭ информацию. АЭ в свою очередь, используя эту информацию, через конечное число шагов восстановит граф H , изоморфный G , т.е. распознает G .

3. Распознавание графа

Предлагаемый алгоритм основан на стратегии поиска в глубину. Эта стратегия такова: агенты идут «вглубь», 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 метим все перешейки из текущей вершины в чужую область и сообщаем второму АИ, через АЭ о необходимости вернуться и распознать помеченные перешейки. Пока второй агент не распознает все перешейки, выбор дальнейшего пути перемещения первым агентом изменяется. В нем поиск перешейков выполняется непосредственно перед проверкой выполнения условий, связанных с распознаванием дерева, построенного данным АИ. Если отсутствуют все возможные варианты перемещения, наличие которых проверяется до проверки на наличие перешейка, то при обнаружении его, текущий агент останавливается до того момента, пока все помеченные перешейки не будут распознаны вторым агентом.

Стратегия поиска в глубину хорошо известна. Известен ряд алгоритмов поиска в глубину для известного графа. Предлагаемая ниже стратегия обладает рядом особенностей. Во - первых, граф G агентам не известен. Во - вторых, при прохождении вершин графа G агенты создают неявную нумерацию пройденных вершин: при первом посещении вершины она окрашивается красным цветом, если это агент A и желтым цветом в случае агента B и ей фактически ставится в соответствие номер, равный значению переменной $Cч_A$ для агента A или $Cч_B$ для агента B . Агент A ведет нумерацию нечетными числами, а агент B четными, другими словами переменные $Cч_A$ и $Cч_B$ принимают соответственно нечетные и четные значения. На основе этой нумерации и происходит восстановление графа путем построения графа H изоморфного G . В процессе поиска агенты строят неявные деревья поиска в глубину и относительно этих деревьев все ребра разделяются на древесные, т.е. принадлежащие деревьям и окрашиваемые при первом прохождении по ним в красный (желтый) цвет, обратные – не принадлежащие дереву и окрашиваемые при первом прохождении в черный цвет и ребра перешейки, которые соединяют деревья. Древесные ребра проходятся как минимум два раза и при последнем проходе окрашиваются агентом в черный цвет, обратные проходятся 1 раз, а ребра перешейки проходятся каждым агентом по крайней мере 2 раза.

Древесные ребра распознаются агентами при первом проходе по ним. Красные (желтые) вершины графа G , на каждом шаге алгоритма, образуют красный (желтый) путь. С помощью этого пути распознаются обратные ребра, при проходе в новую вершину красный (желтый) путь удлиняется, при проходе назад (в случае если это не возврат для распознавания перешейков) – укорачивается, при распознавании обратного ребра – не изменяется. Вершина, у которой все инцидентные ребра распознаны красится черным. Алгоритм оканчивает работу, когда красный и желтый пути становятся пустыми, а все вершины черными.

Рассмотрим алгоритм работы агента A :

1. Агент A красит $(\mu(v) := r)$;
2. Запрос AN ;
3. *if* $AN=0$ *then do*
4. Запрос F ;
5. *if* $F=0$ *then* $МЕТИМ_ПЕР_A(v)$;
6. *else* $ВЫБОР_ХОДА_A(v)$;
7. *end do*;
8. *else* $РАСП_ПЕР_A(v)$;

Рассмотрим процедуры, используемые в данном алгоритме.

$МЕТИМ_ПЕР_A(v)$:

1. *if* в $O(v)$ обнаружено ребро, у которого $(\mu(v,u) = w) \text{ and } (\mu(u) = y)$ *then do*
2. Агент A выполняет процедуру $МЕТИМ_AB(v)$;
3. *go to* 1 данной процедуры;
4. *end do*;
5. *else do*
6. Агент A запрашивает у $AЭ$ значение переменной E ;
7. *if* $E=0$ *then* $ВЫБОР_ХОДА_A(v)$;
8. *else do*
9. Агент A выполняет процедуру $ФИКС_A(v)$;
10. *go to* 2 алгоритма обхода;
11. *end do*;
12. *end do*;

$ВЫБОР_ХОДА_A(v)$:

1. *if* в $O(v)$ обнаружено ребро, у которого $(\mu(v,u) = w) \text{ and } (\mu(u) = \mu(v) = r)$ *then do*
2. Агент A выполняет процедуру $РАСП_A(v)$;
3. *go to* 2 алгоритма обхода;
4. *end do*;
5. *else if* в $O(v)$ обнаружено ребро, у которого $(\mu(v,u) = w) \text{ and } (\mu(u) = w)$ *then do*
6. Агент A выполняет процедуру $ВПЕРЕД_A(v)$;
7. *go to* 2 алгоритма обхода;
8. *end do*;
9. *else if* в $O(v)$ есть ребро, у которого $(\mu(v,u) = w) \text{ and } (\mu(u) = y)$ *then do*
10. Агент A выполняет процедуру $СТОИТ_A(v)$;
11. *go to* 2 алгоритма обхода;
12. *end do*;
13. *else if* в $O(v)$ есть ребро, у которого $(\mu(v,u) = y)$ *then do*
14. Агент A выполняет процедуру $СТОИТ_A(v)$;
15. *go to* 2 алгоритма обхода;
16. *end do*;
17. *else if* в $O(v)$ есть ребро, у которого $(\mu(v,u) = r) \text{ and } (\mu(u) = y)$ *then do*
18. Агент A выполняет процедуру $СТОИТ_A(v)$;
19. *go to* 4 алгоритма обхода;

20. *end do*;
 21. *else if* в $O(v)$ есть ребро, у которого $(\mu(v,u)=r) \text{ and } (\mu(v)=r) \text{ and } (\mu((v,u),v)=r)$ *then do*
 22. Агент A выполняет процедуру $НАЗАД_A(v)$;
 23. *go to* 2 алгоритма обхода;
 24. *end do*;
 25. *else* агент A выполняет процедуру $СТОП_A$;
- РАСП_ПЕР_A(v):**
1. $Z:=K$
 2. *if* в $O(v)$ обнаружено ребро, у которого $(\mu(v,u)=y)$ *then do*
 3. *if* $((K=Z) \text{ or } (K=1)) \text{ and } (Z \neq 1)$ *then* Агент A выполняет $РАСП_АВВ(v)$;
 4. *else* Агент A выполняет процедуру $РАСП_АВВb(v)$;
 5. Агент A запрашивает у АЭ значение переменной K ;
 6. *if* $K \neq 0$ *then go to* 2 данной процедуры;
 7. *else* Агент A выполняет процедуру $ОБН_A(v)$;
 8. *if* $Z \neq 1$ *then do*
 9. *if* в $O(v)$ есть ребро, т.ч. $(\mu((v,u),v)=r) \text{ and } (\mu(v,u)=b) \text{ and } (\mu((v,u),u)=r)$ *then do*
 10. Агент A выполняет процедуру $ВПЕРЕД_AR_N(v)$;
 11. *go to* 9 данной процедуры;
 12. *end do*;
 13. *else if* в $O(v)$ есть ребро, т.ч. $(\mu(v,u)=r) \text{ and } (\mu(u)=r) \text{ and } (\mu((v,u),u)=r)$ *then do*
 14. Агент A выполняет процедуру $ВПЕРЕД_AR(v)$;
 15. *go to* 13 данной процедуры;
 16. *end do*;
 17. *else go to* 2 алгоритма обхода;
 18. *end do*;
 19. *else go to* 13 данной процедуры
 20. *end do*;
 21. *else do*
 22. Агент A выполняет процедуру $ОТСТУП_A(v)$;
 23. *go to* 2 данной процедуры;
 24. *end do*;

ВПЕРЕД_A(v): агент A выбирает из $O(v)$ произвольное ребро (v,u) , у которого $\mu(v,u)=w, \mu(u)=w$, переходит по нему в вершину u , окрашивая при этом $\mu(v,u):=r, \mu(u):=r, \mu((v,u),u):=r$, выполняет $v:=u$ и записывает в список M : $ВПЕРЕД_A$;

НАЗАД_A(v): агент A выбирает из $O(v)$ произвольное ребро (v,u) у которого $((\mu(v,u)=r) \text{ and } (\mu((v,u),v)=r)) \text{ and } (\mu(v)=r)$ и переходит по нему в вершину u , окрашивая при этом $\mu(v):=b, \mu(v,u):=b$, выполняет $v:=u$ и записывает в список M : $НАЗАД_A$;

СТОИТ_A(v): агент A не выполняет никаких действий, в M записывает: $СТОИТ_A$;

МЕТИМ_AB(v): агент A выбирает из $O(v)$ ребро (v,u) у которого $(\mu(v,u)=w) \text{ and } (\mu(u)=y)$, переходит по нему в вершину u , окрашивая

$\mu(v,u) := r$, $\mu((v,u),u) := r$, выполняет $v := u$ и записывает в список M : ВПЕРЕД_АВ, на следующем шаге агент A выбирает из $O(v)$ ребро, у которого $((\mu(v,u) = r) \text{ and } (\mu(v) = y))$ и переходит по нему в вершину u , выполняет $v := u$ и записывает в список M : НАЗАД_АВ;

РАСП_АВВ(v): агент A выбирает из $O(v)$ ребро (v,u) у которого $\mu(v,u) = y$ и переходит по нему в вершину u , окрашивая $\mu(v,u) := r$, $\mu((v,u),u) := b$, выполняет $v := u$ и записывает в список M : ВПЕРЕД_АВВ. На следующем шаге агент A выбирает из $O(v)$ ребро (v,u) у которого $((\mu(v,u) = r) \text{ and } (\mu((v,u),v) = b))$ и переходит по нему в вершину u окрашивая $\mu((v,u),v) := r$, $\mu(v,u) := b$, $\mu((v,u),u) := r$, выполняет $v := u$ и записывает в список M : НАЗАД_АВВ;

РАСП_АВВb(v): агент A выбирает из $O(v)$ ребро (v, u) у которого $\mu(v,u) = y$ и переходит по нему в вершину u , окрашивая $\mu(v,u) := r$, $\mu((v,u),u) := b$, выполняет $v := u$ и записывает в список M : ВПЕРЕД_АВВ. На следующем шаге агент A выбирает из $O(v)$ ребро (v, u) у которого $((\mu(v,u) = r) \text{ and } (\mu((v,u),v) = b))$ и переходит по нему в вершину u , окрашивая $\mu(v,u) := b$, $\mu((v,u),u) := b$, выполняет $v := u$ и записывает в список M : НАЗАД_АВВ;

ВПЕРЕД_АР_N(v): агент A выбирает из $O(v)$ ребро (v, u) у которого $(\mu((v,u),v) = r) \text{ and } (\mu(v,u) = b) \text{ and } (\mu((v,u),u) = r)$ и переходит по нему в вершину u , окрашивая $\mu((v,u),v) := b$, $\mu((v,u),u) := b$, выполняет $v := u$ и записывает в список M : ВПЕРЕД_АР_N;

ВПЕРЕД_АР(v): агент A выбирает из $O(v)$ ребро (v,u) у которого $(\mu(v,u) = r) \text{ and } (\mu(u) = r) \text{ and } (\mu((v,u),u) = r)$ и переходит по нему в вершину u , выполняет $v := u$ и записывает в M : ВПЕРЕД_АР;

ОТСТУП_А(v): агент A выбирает из $O(v)$ ребро (v,u) у которого $((\mu(v,u) = r) \text{ and } (\mu(v,u),v) = r))$ и переходит по нему в вершину u , выполняет $v := u$ и записывает в M : ОТСТУП_А;

ФИКС_А(v): агент A записывает в M : ФИКС_А;

ОБН_А(v): агент A записывает в M : ОБН_А;

РАСП_А(v) (распознавание обратного ребра):

1. Агент A выбирает из $O(v)$ произвольное ребро (v, u) у которого $((\mu(v) = \mu(u) = r) \text{ and } (\mu(v,u) = w))$ и переходит по нему в вершину u ;
2. Агент A красит $\mu(v,u) = b$;
3. Агент A выбирает из $O(u)$ произвольное ребро (u, l) у которого $(\mu(u,l) = r) \text{ and } (\mu((u,l),l) = r) \text{ and } (\mu(l) = r)$ и переходит по нему в вершину l ;
4. $u := l$;
5. Агент A записывает в M : РАСП_А
6. *while* в окрестности $O(u)$ есть ребро (u,l) , у которого $(\mu(u,l) = r) \text{ and } (\mu((u,l),l) = r) \text{ and } (\mu(l) = r)$ *do*
7. агент A переходит по ребру (u,l) в вершину l ;
8. $u := l$;
9. Агент A записывает в M : ОТСТУПИЛ_А;

10. *end do*;

11. Агент A записывает в M : РЕБРО_РАСПОЗНАНО_А;

СТОП_А(v): агент A красит $\mu(v) := b$, агент A останавливается.

Алгоритм работы агента B аналогичен алгоритму работы агента A .

Алгоритм «Восстановление»:

1. $Cч_A := 1$; (счетчик числа посещенных вершин графа G агента A)

2. $Cч_B := 2$; (счетчик числа посещенных вершин графа G агента A)

3. $AN := 0$;

4. $BN := 0$;

5. $N_A := 0$; (номер вершины из кот. мы помечали агентом A перешейки)

6. $N_B := 0$; (номер вершины из кот. мы помечали агентом B перешейки)

7. $N := \emptyset$;

8. $M := \emptyset$;

9. $F := 0$; (количество перешейков из вершины N_A)

10. $K := 0$; (количество перешейков из вершины N_B)

11. $E := 0$; (если $E = 1$, то на предыдущем шаге агент A пометил перешеек)

12. $L := 0$; (если $L = 1$, то на предыдущем шаге агент B пометил перешеек)

13. $i := 0$; (подсчет номера второй вершины перешейка и обратного ребра для A)

14. $j := 0$; (подсчет номера второй вершины перешейка и обратного ребра для B)

15. $t := 1$; (длина списка номеров вершин красного пути)

16. $p := 1$; (длина списка номеров вершин желтого пути)

17. $r(t) := Cч_A$; (номера вершин красного пути)

18. $y(p) := Cч_B$; (номера вершин желтого пути)

19. $V_H := \{1, 2\}$;

20. $E_H := \emptyset$;

21. *While* ($M \neq \emptyset$) *or* ($N \neq \emptyset$) *do*

22. *if* $M \neq \emptyset$ *then do*

23. Прочитать в Mes сообщение из M и удалить его из очереди M ;

24. $ОБР_СП_A()$;

25. *end do*;

26. *if* $N \neq \emptyset$ *then do*

27. Прочитать в Mes сообщение из N и удалить его из очереди N ;

28. $ОБР_СП_B()$;

29. *end do*;

30. *end do*;

31. Печать V_H, E_H .

Рассмотрим используемые в алгоритме «восстановление» процедуры.

ОБР_СП_А():

1. *if* $Mes = \langle \text{ВПЕРЕД_A} \rangle$ *then* $ВПЕРЕД_A()$;

2. *if* $Mes = \langle \text{ВПЕРЕД_AB} \rangle$ *then* $ВПЕРЕД_AB()$;

3. *if* $Mes = \langle \text{ВПЕРЕД_ABV} \rangle$ *then* $ВПЕРЕД_ABV()$;

4. *if Mes = «НАЗАД_А» then НАЗАД_А();*
5. *if Mes = «НАЗАД_АВ» then НАЗАД_АВ();*
6. *if Mes = «НАЗАД_АВВ» then НАЗАД_АВВ();*
7. *if Mes = «ФИКС_А» then ФИКС_А();*
8. *if Mes = «ОБН_А» then ОБН_А();*
9. *if Mes = «РАСП_А» then РАСП_А();*
10. *if Mes = «ОТСТУП_А» then ОТСТУП_А();*

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

НАЗАД_А(): из списка $r(1) \dots r(t)$ удаляется элемент $r(t)$, $t := t - 1$;

ВПЕРЕД_АВ(): F присваивается значение $F + 1$;

НАЗАД_АВ(): E присваивается значение 1 ;

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

НАЗАД_АВВ(): K присваивается значение $K - 1$;

ОТСТУП_А(): i присваивается значение $i + 1$;

ФИКС_А(): выполняются операции: $N_A := Cч_А$, $BN := 1$ и $E := 0$;

ОБН_А(): отменяется команда «назад» для A ($AN := 0$), и $i := 0$;

РАСП_А():

1. Прочитать в Mes сообщение из M и удалить его;
2. *While Mes = «ОТСТУПИЛ_А» do*
3. $i := i + 1$;
4. *end do*;
5. $E_H := E_H \cup \{r(t), r(t-i)\}$;
6. $i := 0$;

Процедуры обработки списка сообщений от агента B аналогичны.

4. Свойства алгоритма распознавания

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

Теорема 2 Временная сложность алгоритма распознавания равна $O(n^3)$, а емкостная - $O(n^2)$. При этом алгоритм использует 3 краски.

5. Выводы

Выполняя алгоритм распознавания, АИ и АЭ строят граф H изоморфный исследуемому графу G . При этом каждому АИ требуется 2 краски, одна из которых совпадает по цвету для обоих АИ. Алгоритм останавливается через конечное число шагов, число которых является величиной $O(n^3)$. При этом емкостная сложность есть величина $O(n^2)$.

Литература

1. Dudek G, Jenkin M. Computational principles of mobile robotics – Cambridge Univ. press, Cambridge, 2000. – 280 p.
2. Грунский И.С., Татаринов Е.А. Распознавание конечного графа блуждающим по нему агентом. - Вестник ДонНУ, - естеств. науки. – 2009, вып.1. – С. 492-497.