

РАСПОЗНАВАНИЕ КОНЕЧНОГО ГРАФА КОЛЛЕКТИВОМ АГЕНТОВ*Стёпкин А.В.**Славянский государственный педагогический университет***Введение**

Проблемой компьютерной науки, решению которой уделяется основное внимание, является проблема взаимодействия управляющей и управляемой систем. В рассмотренном ниже случае взаимодействие этих систем представляется как процесс перемещения агентов по графу среды. Пока не сформирована достаточно полная модель операционной среды, целенаправленное перемещение агента по ней невозможно. 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]. Выделяют три основные задачи исследования среды агентом:

1. самолокализация агента;
2. контроль соответствия модели среды и самой среды;
3. построение агентом модели карты среды [1].

Много работ посвящено задаче распознавания среды с помощью тех или иных средств агента. 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.

1 Основные определения и обозначения

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 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 Распознавание графа

Предлагаемый алгоритм основан на стратегии поиска в глубину. Известен ряд алгоритмов поиска в глубину для известного графа. Предлагаемая ниже стратегия обладает рядом особенностей:

1. граф G -й агентам не известен;
2. 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 $Sч_A$ для агента A или $Sч_B$ для агента B . Агент A ведет нумерацию нечетными числами, а агент B четными. На основе этой нумерации и происходит восстановление графа путем построения графа H изоморфного G . Процесс распознавания состоит из двух

принципиально разных алгоритмов: «Обход» и «Восстановление». Отметим, что совместная работа алгоритмов осуществляется следующим образом: сначала ходит агент А, АЭ обрабатывает полученные от него данные, далее ходит агент В, АЭ обрабатывает полученные данные и т.д. АИ помещаются в различные вершины графа G, эти вершины сразу окрашиваются в красный и желтый цвета для А и В соответственно.

Рассмотрим алгоритм работы агента А:

1. Агент А красит ($\mu(v) = r$);
2. Запрос AN;
3. if AN=0 then do
4. Запрос F;
5. if F=0 then
6. МЕТИМ_ПЕР_А(v);
7. else do
8. ВЫБОР_ХОДА_А(v);
9. end do;
10. end do;
11. else do
12. РАСП_ПЕР_А(v);
13. end do.

Рассмотрим процедуры, используемые в данном алгоритме.

МЕТИМ_ПЕР_А(v):

1. Если в $O(v)$ обнаружено ребро, у которого $(\mu(v, u) = w) \text{ and } (\mu(u) = y)$ then do
2. Агент А выполняет процедуру МЕТИМ_АВ(v);
3. Перемещаемся в строку 1 данной процедуры;
4. end do;
5. Иначе do
6. Агент А запрашивает у АЭ значение переменной E;
7. Если E=0 тогда перемещаемся в строку 8 алгоритма обхода;
8. end do;
9. Иначе do
10. Агент А выполняет процедуру ФИКС_А(v);
11. Перемещаемся в строку 2 алгоритма обхода;
12. end do;

ВЫБОР_ХОДА_А(v):

1. Если в $O(v)$ агент А обнаружил ребро, у которого $(\mu(v, u) = w) \text{ and } (\mu(u) = \mu(v) = r)$ then do
2. Агент А выполняет процедуру РАСП_А(v);
3. Перемещаемся в строку 2 алгоритма обхода;
4. end do;
5. Иначе, если в $O(v)$ обнаружено ребро, у которого $(\mu(v, u) = w) \text{ and } (\mu(u) = w)$ then do
6. Агент А выполняет процедуру ВПЕРЕД_А(v);
7. Перемещаемся в строку 2 алгоритма обхода;
8. end do;
9. Иначе, если в $O(v)$ есть ребро, у которого $(\mu(v, u) = w) \text{ and } (\mu(u) = y)$ then do
10. Агент А выполняет процедуру СТОИТ_А(v);
11. Перемещаемся в строку 2 алгоритма обхода;
12. end do;
13. Иначе, если в $O(v)$ есть ребро, у которого $(\mu(v, u) = y)$ then do
14. Агент А выполняет процедуру СТОИТ_А(v);

15. Перемещаемся в строку 2 алгоритма обхода;
16. end do;
17. Иначе, если в $O(v)$ есть ребро, у которого $(\mu(v,u)=r) \text{ and } (\mu(u)=y)$ then do
18. Агент А выполняет процедуру СТОИТ_А(v);
19. Перемещаемся в строку 4 алгоритма обхода;
20. end do;
21. иначе, если 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 А выполняет процедуру НАЗАД_А(v);
23. Перемещаемся в строку 2 алгоритма обхода;
24. end do;
25. Иначе агент А выполняет процедуру СТОП_А;
РАСП_ПЕР_А(v):

1. Если в $O(v)$ обнаружено ребро, у которого $(\mu(v,u)=y)$ then do
2. Агент А выполняет процедуру РАСП_АВВ(v);
3. Агент А запрашивает у АЭ значение переменной К;
4. Если $K \neq 0$ тогда переходим в строку 1 данной процедуры;
5. Иначе do
6. Агент А выполняет процедуру ОБН_А(v);
7. Если в $O(v)$ есть ребро, у которого $(\mu(v,u)=r) \text{ and } (\mu(u)=r) \text{ and } (\mu((v,u),u)=r)$ then do
8. Агент А выполняет процедуру ВПЕРЕД_АР(v);
9. Перемещаемся в строку 7 данной процедуры;
10. end do;
11. Иначе перемещаемся в строку 2 алгоритма обхода;
12. end do;
13. end do;
14. Иначе do
15. Агент А выполняет процедуру ОТСТУП_А(v);
16. Перемещаемся в строку 1 данной процедуры;
17. end do;

ВПЕРЕД_А(v): агент А 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 М: ВПЕРЕД_А;

НАЗАД_А(v): агент А 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 М: НАЗАД_А;

СТОИТ_А(v): агент А не выполняет никаких действий, в М записывает: СТОИТ_А;

МЕТИМ_АВ(v): агент А 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 М: ВПЕРЕД_АВ, на следующем шаге агент А 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 М: НАЗАД_АВ;

РАСП_АВВ(v): агент А 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 М: ВПЕРЕД_АВВ. На следующем шаге агент А 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 $\mu(v,u):=b$, выполняет $v:=u$ и записывает в список М: НАЗАД_АВВ;

ВПЕРЕД_AR(v) (в конец красного пути после распознавания меченых перешейков): агент А 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 М: ВПЕРЕД_AR;

ОТСТУП_A(v): агент А 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 М: ОТСТУП_A;

ФИКС_A(v): агент А записывает в М: ФИКС_A;

ОБН_A(v): агент А записывает в М: ОБН_A;

РАСП_A(v) (распознавание обратного ребра):

1. Агент А 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 А красит $\mu(v,u)=b$;
3. Агент А 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 А записывает в М: РАСП_A
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 А переходит по ребру (u,l) в вершину l ;
8. $u:=l$;
9. Агент А записывает в М: ОТСТУПИЛ_A;
10. end do;
11. Агент А записывает в М: РЕБРО_РАСПОЗНАНО_A;

Алгоритм работы агента В аналогичен алгоритму работы агента А.

Алгоритм «Восстановление»:

1. $Сч_A:=1$; (счетчик числа посещенных вершин графа G агента А)
2. $Сч_B:=2$; (счетчик числа посещенных вершин графа G агента А)
3. $AN:=0$;
4. $BN:=0$;
5. $N_A:=0$; (номер вершины из которой мы помечали агентом А перешейки)
6. $N_B:=0$; (номер вершины из которой мы помечали агентом В 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 А пометил перешеек)
12. $L:=0$; (если $L=1$, то на предыдущем шаге агент В пометил перешеек)
13. $i:=0$; (подсчет номера второй вершины перешейка и обратного ребра для А)
14. $j:=0$; (подсчет номера второй вершины перешейка и обратного ребра для В)
15. $t:=1$; (длина списка номеров вершин красного пути)
16. $p:=1$; (длина списка номеров вершин желтого пути)
17. $r(t):=Сч_A$; (номера вершин красного пути)
18. $y(p):=Сч_B$; (номера вершин желтого пути)
19. $VH:=\{1,2\}$;
20. $EH:=\emptyset$;
21. While $(M \neq \emptyset) \text{ and } (N \neq \emptyset)$ do
22. Прочитать в Mes сообщение и удалить его из очереди M;
23. if $M \neq \emptyset$ then do

24. ОБР_СП_А()
25. end do;
26. if $N \neq \emptyset$ then do
27. ОБР_СП_В()
28. end do;
29. end do;
30. Печать VH, EH.

Рассмотрим используемые в алгоритме «восстановление» процедуры.

ОБР_СП_А():

1. if Mes= «ВПЕРЕД_А» then ВПЕРЕД_А();
 2. if Mes= «ВПЕРЕД_АВ» then ВПЕРЕД_АВ();
 3. if Mes= «ВПЕРЕД_АВВ» then ВПЕРЕД_АВВ();
 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: $Cч_A := Cч_A + 2$; $t := t + 1$; $r(t) := Cч_A$;
 $VH := VH \cup \{Cч_A\}$; $EH := EH \cup \{r(t-1), r(t)\}$;
 НАЗАД_А(): из списка $r(1) \dots r(t)$ удаляется элемент $r(t)$, значение t уменьшается на 1;
 ВПЕРЕД_АВ(): F увеличивается на 1;
 НАЗАД_АВ(): E присваивается значение 1;
 ВПЕРЕД_АВВ(): $EH := EH \cup \{N_A, r(t)-i\}$;
 НАЗАД_АВВ(): K присваивается значение $K-1$;
 ОТСТУП_А(): i присваивается значение $i+2$;
 ФИКС_А(): выполняется следующая цепочка операций: $N_A := Cч_A$, $VN := 1$ и $E := 0$;
 ОБН_А(): отменяется команда «назад» для A ($AN := 0$), и обнуляется параметр i ;
 РАСП_А():
1. Прочитать в Mes сообщение из M и удалить его;
 2. While Mes=«ОТСТУПИЛ_А» do
 3. $i := i + 2$;
 4. end do;
 5. $EH := EH \cup \{r(t), r(t)-i\}$;
 6. $i := 0$;

Процедуры обработки списка сообщений от агента В аналогичны.

3 Свойства алгоритма распознавания

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

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

4 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 $O(n^2)$.

Автор выражает глубокую признательность своему научному руководителю И.С. Грунскому за оказанную помощь в работе.

Литература

- [1] Dudek G, Jenkin M. Computational principles of mobile robotics – Cambridge Univ. press, Cambridge, 2000. – 280 p.
- [2] Татаринов Е.А. Построение эксперимента для восстановления графа // Компьют. науки и информац. техноглг.: Материалы междунар. науч. конф.-Саратов: Изд-во Саратов. ун-та, 2009. - С. 185-188.