УДК 519.1

РАСПОЗНАВАНИЕ КОНЕЧНЫХ НЕОРИЕНТИРОВАННЫХ ГРАФОВ ТРЕМЯ АГЕНТАМИ

Стёпкин А.В.
Институт прикладной математики и механики НАН Украины.
E-mail: stepkin.andrey@rambler.ru

Аннотация

Стёпкин А. В. Распознавание конечных неориентированных графов тремя агентами. *В работе рассматривается проблема распознавания конечного неориентированного графа коллективом агентов. Построен алгоритм распознавания неизвестных графов временная сложность которого равна $O(n)$, а емкостная – $O(n^2 \log n)$, где n - число вершин графа. Для распознавания графа каждому агенту, передвигающемуся по графу, необходимо 2 различные краски (всего 3 краски). Метод основан на методе обхода графа в глубину.*

Общая постановка проблемы

В настоящее время актуальным направлением математической кибернетики является распознавание неизвестной среды [1]. Распознавание среды, заданной конечным неориентированным графом, ранее рассматривалось в [2,3]. В [3] был рассмотрен алгоритм распознавания в котором два агента-исследователя (АИ) одновременно передвигаются по неизвестному графу, обмениваются данными с агентом-экспериментатором (АЭ), который и восстанавливает исследуемый граф по данным, полученным от АИ. Алгоритм работы затрачивал $O(n^2)$ шагов и требовал $O(n^2)$ памяти, где n – количество вершин исследуемого графа. В данной работе рассматривается модификация алгоритма рассмотренного в [3], позволяющая на порядок улучшить временную сложность за счет добавления памяти агентам и в вершины, то есть увеличения емкостной сложности.

Алгоритм работы

При описании алгоритма используются результаты и обозначения из [3]. Распознавание графа осуществляется двумя типами агентов: 1. *Агент-исследователь*. АИ перемещается по графу, окрашивает (у каждого АИ есть две краски: у A это r и b, у B – y и b) вершины, ребра и инциденторы графа, а также записывает в каждую посещенную им вершину номер (в двоичном коде), позволяющий однозначно идентифицировать любую, ранее посещенную (в том числе и вторым АИ) вершину. Так же АИ воспринимает отметки на элементах графа, на основании которых и осуществляет перемещение. 2. *Агент-экспериментатор*. АЭ восстанавливает граф, по данным полученным от АИ, а также пошагово передает АИ информацию, необходимую для их дальнейшего функционирования.

Рассмотрим *алгоритм обхода графа*. При описании режимов работы АИ, в скобках указываются сообщения, отправляемые АИ, в рассматриваемой ситуации («СООБЩЕНИЕ_АГЕНТА_А»; «СООБЩЕНИЕ_АГЕНТА_В»). 1. *Обычный режим.* АИ движется вперед по белым вершинам, окрашивая вершины, соединяющие их ребра и дальние инциденторы в «свой» цвет, а так же записывая в вершины номера (номера, записываемые АИ в вершины, агенты получают на каждом шаге от АЭ) («ВПЕРЕД_А»; «ВПЕРЕД_В»). Если нет возможных путей перемещения, то АИ возвращается назад, окрашивая пройденные вершины, ребра и ближние инциденторы в черный цвет («НАЗАД_А»; «НАЗАД_В»). Вернувшись в начальную вершину, АИ завершает работу («СТОП_А»; «СТОП_В»). На каждом шаге АИ обменивается данными с АЭ. 2. *Распознавание обратных ребер* (белое ребро, дальняя вершина которого окрашена в «свой» цвет). Если при движении вперед было обнаружено обратное ребро, то АИ сканирует окрестность вершины, в которой он находится,

считывает номера дальних вершин инцидентных обратным ребрам и отправляет их АЭ («ОБР_А($x_1, x_2, ..., x_l$)»; «ОБР_В($k_1, k_2, ..., k_l$)», где x_i, k_i – номера, записанные агентами A и B соответственно, в вершинах своего пути). 3. *Распознавание перешейков* (ребро, соединяющее вершины, принадлежащие областям работы разных агентов). Этот режим немного отличается для каждого АИ. Если агент A обнаружил в вершине перешейки, и ни в одной из дальних вершин, этих перешейков нет агента B (или же агент B находится в одной из этих вершин, но уже распознал все, ранее обнаруженные ним перешейки в эту вершину), то агент A отправляет все номера дальних вершин перешейков АЭ («ПЕР_А($x_1, x_2, ..., x_l$)», где x_i – номера, записанные агентом B, в вершинах своего пути). Если же агент B находится в одной из дальних вершин перешейков, и ещё не распознавал инцидентные ей перешейки, то агент A отправляет АЭ номера всех дальних вершин обнаруженных перешейков, кроме номера вершины, в которой находится агент B. Если перешейки обнаружил агент B и в одной из дальних вершин этих перешейков находится агент A, то B не выполняет никаких действий до ухода A из окрестности вершины, в которой находится B. Далее агент B передает АЭ номера всех дальних вершин обнаруженных перешейков (*ПЕР_В*($k_1, k_2, ..., k_l$), где k_i – номера, записанные агентом A, в вершинах своего пути). 4. *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 B на следующем шаге отступает назад по своему пути и переключается в обычный режим работы. Агент A видит разноцветную вершину как свою, но при распознавании окрашивает в черный цвет обе половинки.

Рассмотрим *алгоритм работы АЭ*.

1. $Сч\_A := 1$;
2. $Сч\_B := 1$;
3. обнуляем массивы: $M := \varnothing$, $N := \varnothing$, $E_H := \varnothing$, $STOP\_A := 0$, $STOP\_B := 0$, $mr\_A = 0$;
4. $t := 1$;
5. $p := 1$;
6. $r(t) := Сч\_A$; (*номера вершин красного пути*)
7. $y(p) := Сч\_B$; (*номера вершин желтого пути*)
8. $V_H := \{A[1], B[1]\}$;
9. *while* $(STOP\_A = 0) or (STOP\_B = 0)$ *do*
10. *if* $M \neq \varnothing$ *then do*
11. прочитать в *Mes* сообщение и удалить его из списка M;
12. *ОБР_СП_А*();
13. *end do*;
14. *if* $N \neq \varnothing$ *then do*
15. прочитать в *Mes* сообщение и удалить его из списка N;
16. *ОБР_СП_В*();
17. *end do*;
18. *end do*;
19. печать V_H, E_H.

Рассмотрим процедуры, используемые в алгоритме.

ОБР_СП_А():

1. *if* $Mes = "ПЕР\_А(x_1, x_2, ..., x_l)"$ *then* *ПЕР_А*($x_1, x_2, ..., x_l$);
2. *if* $Mes = "ОБР\_А(x_1, x_2, ..., x_l)"$ *then* *ОБР_А*($x_1, x_2, ..., x_l$);
3. *if* $Mes = "ВПЕРЕД\_А"$ *then* *ВПЕРЕД_А*();

4. *if Mes* = "*НАЗАД_А*" *then НАЗАД_А*();
5. *if Mes* = "*СТОП_А*" *then СТОП_А*().

ПЕР_А($x_1, x_2, ..., x_l$): $E_H := E_H \cup \{(A[r(t)], B[x_1]); (A[r(t)], B[x_2]); ...; (A[r(t)], B[x_l])\}$.

ОБР_А($x_1, x_2, ..., x_l$): $E_H := E_H \cup \{(A[r(t)], A[x_1]); (A[r(t)], A[x_2]); ...; (A[r(t)], A[x_l])\}$.

ВПЕРЕД_А(): $Сч\_A := Сч\_A + 1$; $t := t + 1$; $r(t) := Сч\_A$; $V_H := V_H \cup \{A[Сч\_A]\}$; $E_H := E_H \cup \{(A[r(t-1)], A[r(t)])\}$.

НАЗАД_А(): из списка $r(1)...r(t)$ удаляется элемент $r(t)$; $t := t - 1$.

СТОП_А(): $STOP\_A := 1$.

Процедуры работы со списком команд от агента B, которые не рассмотрены ниже, аналогичны процедурам работы со списком команд от агента A.

ОБР_СП_В():

1. *if Mes* = "*ВОЗВРАТ_В*" *then ВОЗВРАТ_В*()
2. *if Mes* = "*ПЕР_В*($k_1, k_2, ..., k_l$)" *then ПЕР_В*($k_1, k_2, ..., k_l$);
3. *if Mes* = "*ОБР_В*($k_1, k_2, ..., k_l$)" *then ОБР_В*($k_1, k_2, ..., k_l$);
4. *if Mes* = "*ВПЕРЕД_В*" *then ВПЕРЕД_В*();
5. *if Mes* = "*НАЗАД_В*" *then НАЗАД_В*();
6. *if Mes* = "*СТОП_В*" *then СТОП_В*().

ВОЗВРАТ_В(): $E_H := E_H \setminus \{(y(p-1), y(p))\}$; $V_H := V_H \setminus \{Сч\_B\}$; $Сч\_B := Сч\_B - 1$; $p := p - 1$; $y(p) := Сч\_B$.

ПЕР_В($k_1, k_2, ..., k_l$): $E_H := E_H \cup \{(B[y(p)], A[k_1]); (B[y(p)], A[k_2]); ...; (B[y(p)], A[k_l])\}$; $mr\_A := 1$.

ВПЕРЕД_В(): $Сч\_B := Сч\_B + 1$; $p := p + 1$; $y(p) := Сч\_B$; $V_H := V_H \cup \{B[Сч\_B]\}$; $E_H := E_H \cup \{(B[y(p-1)], B[y(p)])\}$; $mr\_A := 0$.

НАЗАД_В(): из списка $y(1)...y(p)$ удаляется элемент $y(p)$; $p := p - 1$; $mr\_A := 1$.

Для предложенного в работе алгоритма доказаны следующие теоремы.

Теорема 1. Три агента, выполнив алгоритм распознавания на конечном неориентированном графе, распознают этот граф с точностью до изоморфизма.

Теорема 2. Временная сложность алгоритма равна $O(n)$, емкостная сложность алгоритма равна $O(n^2 \log n)$, где n – число вершин графа, при этом алгоритм использует 3 краски.

Выводы

Рассмотренная модификация позволила на порядок улучшить временную сложность алгоритма рассмотренного в [3] и получить временную сложность равную $O(n)$ за счет увеличения его емкостной сложности с $O(n^2)$ до $O(n^2 \log n)$**.** Для распознавания АИ используют по две краски каждый (всего три краски).

Список литературы

9. Albers S. Exploring unknown environments / S. Albers, M. R. Henzinger // SIAM Journal on Computing. – 2000. – 29(4). – P. 1164 – 1188.

10. Грунский И. С. Распознавание конечного графа блуждающим по нему агентом. / И. С. Грунский, Е. А. Татаринов // Вестник Донецкого университета. Серия А. Естественные науки. – 2009. – вып. 1. – С. 492 – 497.

11. Стёпкин А. В. Алгоритм распознавания конечных графов коллективом агентов // А. В. Стёпкин / Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС КМ – 2011) : матеріали II Всеукр. наук. – техн. конф. студентів, аспірантів та молодих вчених (Донецьк 11-13 квітня 2011). − Донецьк, ДонНТУ 2011. − т.1. − С. 213 − 216.