

Литература.

1. Wasserman S. Social Network Analysis: Methods and Applications / Stanley Wasserman, Katherine Faust – New York : Cambridge University Press, 1994. – 857 p.
2. Савельев О.О. Особенности разработки подсистемы хранения информации для системы поддержки принятия решений в области анализа телекоммуникационных данных / О.О. Савельев, А.И. Шевченко // Материалы VI международной научно-технической конференции студентов, аспирантов и молодых ученых «Информатика и компьютерные технологии». – Донецк : ДонНТУ. – 2010. С. 343-349.

Стёпкин А.В.

**Науч. руководитель к.ф.-м.н. проф. каф.
ПОИС Грунский И.С.**

Славянский государственный педагогический университет

**Распознавание конечного неориентированного
графа тремя агентами**

Актуальным вопросом математической кибернетики является распознавание неизвестной среды [1]. В данной работе рассматривается модификация алгоритма рассмотренного в [2], позволяющая улучшить временную

сложность с $O(n^2)$ до $O(n)$

агентам и в вершины, то есть увеличения емкостной сложности с $O(n^2)$ до $O(n^2 \log n)$.

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, записывает в вершины

номера в двоичном коде (получаемые 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»). 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 находится, считывает номера дальних вершин инцидентных обратным ребрам и отправляет их АЭ («ОБР_А(x_1, x_2, \dots, x_l)»);

«ОБР_В(k_1, k_2, \dots, k_l)», где x_i, k_i – номера, записанные агентами А и В 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 А обнаружил в вершине перешейки, и ни в одной из дальних вершин, этих перешейков нет агента В (или же агент В находится в одной из этих вершин, но уже распознал все, ранее обнаруженные ним перешейки в эту вершину), то агент А отправляет все номера дальних вершин перешейков АЭ («ПЕР_А(x_1, x_2, \dots, x_l)», где x_i –

номера, записанные агентом В, в вершинах своего пути). Если же агент В находится в одной из дальних вершин перешейков, и ещё не распознавал инцидентные ей перешейки, то агент А отправляет АЭ номера всех дальних вершин обнаруженных перешейков, кроме номера вершины, в которой находится агент В. Если перешейки обнаружил агент В и в одной из дальних вершин этих перешейков находится агент А, то В не выполняет никаких действий до ухода А из окрестности вершины, в которой находится В. Далее агент В передает АЭ номера всех

дальних вершин обнаруженных перешейков ($ПЕР_B(k_1, k_2, \dots, k_l)$), 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. $Cч_A := 1; Cч_B := 1; M := \emptyset; N := \emptyset; E_H := \emptyset; STOP_A := 0; STOP_B := 0; mr_A = 0; t := 1; p := 1; r(t) := Cч_A; y(p) := Cч_B; V_H := \{A[1], B[1]\};$
2. *while* ($STOP_A = 0$) *or* ($STOP_B = 0$) *do*
3. *if* $M \neq \emptyset$ *then do*
4. прочитать в *Mes* сообщение и удалить его из M ;
5. $ОБР_СП_A()$; *end do*;
6. *if* $N \neq \emptyset$ *then do*
7. прочитать в *Mes* сообщение и удалить его из N ;
8. $ОБР_СП_B()$; *end do*; *end do*;
9. печать V_H, E_H .

Рассмотрим процедуры, используемые в алгоритме.

$ОБР_СП_A()$:

1. *if* $Mes = "ПЕР_A(x_1, x_2, \dots, x_l)"$ *then* $ПЕР_A(x_1, x_2, \dots, x_l)$;
2. *if* $Mes = "ОБР_A(x_1, x_2, \dots, x_l)"$ *then* $ОБР_A(x_1, x_2, \dots, x_l)$;
3. *if* $Mes = "ВПЕРЕД_A"$ *then* $ВПЕРЕД_A()$;
4. *if* $Mes = "НАЗАД_A"$ *then* $НАЗАД_A()$;
5. *if* $Mes = "СТОП_A"$ *then* $СТОП_A()$.

$ПЕР_A(x_1, x_2, \dots, x_l)$:

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

ОБР_А(x_1, x_2, \dots, x_l):

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

ВПЕРЕД_А($Cч_А$): $Cч_А := Cч_А + 1$; $t := t + 1$; $r(t) := Cч_А$;

$V_H := V_H \setminus \{A[Cч_А]\}$; $E_H := E_H \setminus \{(A[r(t-1)], A[r(t)])\}$.

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

СТОП_А(t): $STOP_А := 1$.

Нерассмотренные процедуры работы с командами агента B , аналогичны процедурам с командами агента A .

ОБР_СП_В($Cч_В$):

1. if $Mes = "ВОЗВРАТ_В"$ then *ВОЗВРАТ_В*($Cч_В$)

2. if $Mes = "ПЕР_В(k_1, k_2, \dots, k_l)"$ then *ПЕР_В*(k_1, k_2, \dots, k_l);

3. if $Mes = "ОБР_В(k_1, k_2, \dots, k_l)"$ then *ОБР_В*(k_1, k_2, \dots, k_l);

4. if $Mes = "ВПЕРЕД_В"$ then *ВПЕРЕД_В*($Cч_В$);

5. if $Mes = "НАЗАД_В"$ then *НАЗАД_В*($Cч_В$);

6. if $Mes = "СТОП_В"$ then *СТОП_В*($Cч_В$).

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

$V_H := V_H \setminus \{Cч_В\}$; $Cч_В := Cч_В - 1$; $p := p - 1$; $y(p) := Cч_В$.

ПЕР_В(k_1, k_2, \dots, k_l):

$E_H := E_H \setminus \{(B[y(p)], A[k_1]); (B[y(p)], A[k_2]); \dots; (B[y(p)], A[k_l])\}$; $mr_А := 1$.

ВПЕРЕД_В($Cч_В$): $Cч_В := Cч_В + 1$; $p := p + 1$; $y(p) := Cч_В$;

$V_H := V_H \setminus \{B[Cч_В]\}$; $E_H := E_H \setminus \{(B[y(p-1)], B[y(p)])\}$; $mr_А := 0$.

НАЗАД_В($Cч_В$): из $y(1) \dots y(p)$ удаляется $y(p)$; $p := p - 1$; $mr_А := 1$.

Теорема 1. Выполнив алгоритм распознавания на конечном неориентированном графе, агенты 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. Алгоритм использует 3 краски.

Выводы. Рассмотренная модификация позволила на порядок улучшить временную сложность алгоритма рассмотренного в [2]. За счет увеличения емкостной сложности добавлением памяти агентам и в вершины.

Литература.

1. S. Albers and M. R. Henzinger. Exploring unknown environments. *SIAM Journal on Computing*, 29(4): 1164 – 1188, 2000.

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