УДК 519.7

Стёпкин А.В.

заведующий лабораториями ФЭУ, СГПУ

e-mail: stepkin.andrey@rambler.ru

РАСПОЗНАВАНИЕ КОНЕЧНЫХ ГРАФОВ КОЛЛЕКТИВОМ АГЕНТОВ.

В работе рассматривается задача распознавания конечных графов тремя агентами. Предложен алгоритм квадратических (от числа вершин графа) временной и емкостной сложностей, который распознает любой конечный неориентированный граф, без петель и кратных ребер. Для распознавания графа каждому агенту требуется по 2 краски (всего 3 краски). Метод основан на методе обхода графа в глубину.

Ключевые слова: *граф, распознавание, агенты.*

Введение

В настоящее время одной из центральных проблем компьютерной науки является проблема взаимодействия управляющей и управляемой систем. Подобное взаимодействие зачастую представляется как процесс перемещения агента по помеченному графу. К примеру, в [1] оно представлено передвижением одного агента-исследователя (АИ) по неизвестному графу и обменом данными с агентом-экспериментатором (АЭ), который и производит восстановление графа по данным, полученным от АИ.

Данная работа посвящена исследованию проблемы, в предположении, что взаимодействие управляющей и управляемой систем представляется процессом перемещения двух АИ A и B по неизвестному конечному графу (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 рассмотрен алгоритм построения маршрутов АИ, позволяющих АЭ точно восстановить граф. Для этого у каждого АИ есть две краски: у A это r и b, у B – y и b. В отличие от [2] оптимизированы процедуры распознавания обратных ребер и перешейков, что позволило улучшить временную сложность алгоритма до $O\left(n^2\right)$, где n – число вершин графа. Полученный алгоритм использует результаты и обозначения из [2, 3].

Основные определения и обозначения

Рассматриваются конечные, неориентированные графы без петель и кратных ребер. Все неопределяемые понятия общеизвестны, их можно найти, например, в [4, 5]. 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 $\mu : L \to \{w, r, y, b\}$, где w интерпретируется как белый цвет (краска), r – красный, y – желтый, b – черный. Пара (G, μ) называется раскрашенным графом. Последовательность $u_1, u_2, ...u_k$ попарно смежных вершин называется путем в графе G, а k - длиной 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 $\varphi : V_G \to V_H$, 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.

Мобильные агенты A и B характеризуются следующими свойствами. Они 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 (v, u) они будут перемещаться и как будут перекрашивать элементы v, u, (v, u), $((v, u), v)$, $((v, u), u)$. 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.

Основная часть

Рассмотрим подробнее алгоритмы обхода и восстановления.

Алгоритм работы агента A:

1. АИ красит вершину v, в которой находится $\mu(v) := r$;
2. запрос AN;

3. *if* $AN \neq 1$ *then do* запрос *BN*;
4. *if* $BN = 0$ *then* $МЕТИМ\_ПЕР\_A(v)$;
5. *else* $ВЫБОР\_ХОДА\_A(v)$; *end do*;
6. *else* $РАСП\_ПЕР\_A(v)$;

$РАСП\_ПЕР\_A(v)$:

1. *if* в $O(v)$ не обнаружено ребра, у которого $\mu(v,u) = y$ *then do*
2. *if* в $O(v)$ есть ребро, у которого $((\mu(v,u) = r)\, and\, (\mu((v,u),u) = r))\, or$ $or\, (\mu(v,u) = w)\, or\, ((\mu((v,u),u) = r)\, and\, ((\mu(u) = y)))$ *then do*
3. агент A выполняет процедуру $ОТСТУП\_A(v)$;
4. *go to 1* данной процедуры; *end do*; *else do*
5. агент A выполняет процедуру $НАЗАД\_A(v)$;
6. *go to 1* данной процедуры; *end do*; *end do*; *else do*
7. запрос UDP_B;
8. *if* $UDP\_B = TRUE$ *then* $РАСП\_ABB(v)$; *else* $РАСП\_ABBb(v)$;
9. запрос K;
10. *if* $K \neq 0$ *then go to* 1 данной процедуры; *else do*
11. агент A выполняет процедуру $ОБН\_A(v)$;
12. *if* в $O(v)$ есть ребро, у которого $(\mu\ ((v,u),v) = r)\, and$ $and\, (\mu\ (v,u) = b)\, and\, (\mu\ ((v,u),u) = r)$ *then do*
13. агент A выполняет процедуру $ВПЕРЕД\_AR\_N(v)$;
14. *go to* 12 данной процедуры; *end do*;
15. *if* в $O(v)$ есть ребро, у которого $(\mu(v,u) = r)\, and$ $(\mu((v,u),u) = r)\, and\, (\mu(u) = r)$ *then do*
16. агент A выполняет процедуру $ВПЕРЕД\_\ AR(v)$;
17. *go to* 15 данной процедуры; *end do*;
18. *else go to* 2 алгоритма обхода; *end do*; *end do*.

$РАСП\_A(v)$:

1. *while* в $O(v)$ есть ребро, т.ч. $(\mu\ (v,u) = w)\, and\, (\mu\ (v) = \mu(u) = r)$ *do*
2. агент A красит $(\mu((v,u),v) := r)$;
3. агент A записывает в M: МЕТКА_ОР_А; *end do*;
4. агент A выбирает из $O(v)$ ребро, у которого $(\mu(v,u) = r)\, and$ $and\, (\mu((v,u),v) = r)\, and\, (\mu(u) = r)$ и переходит по нему в вершину u;
5. $v := u$;
6. агент A записывает в M: ОТСТУПИЛ_А;
7. *if* в $O(v)$ нет ребра, у которого $(\mu(v,u) = w)\, and\, (\mu((v,u),u) = r)\, and$ $and\, (\mu(v) = \mu(u) = r)$ *then go to* 4 данной процедуры; *else do*
8. агент A переходит по ребру (v,u), красит $(\mu(v,u) := b)$;
9. $v := u$;

10. агент A записывает в список M: ВПЕРЕД_ОР_А; *end do*;
11. запрос $UDOBR\_A$;
12. *if* $UDOBR\_A = TRUE$ *then do*
13. агент A выбирает из $O(v)$ ребро (v, u), у которого $(\mu(v,u) = b)\, and\, (\mu((v,u),v) = r)\, and\, (\mu(v) = \mu(u) = r)$;
14. агент A красит $\mu((v,u),v) := b$;
15. агент A записывает в M: РЕБРА_РАСПОЗНАНЫ_А;*end do*;
16. *else do*
17. агент A выбирает из $O(v)$ ребро, у которого $(\mu(v,u) = b)\, and$ $and\, (\mu((v,u),v) = r)\, and\, (\mu(v) = \mu(u) = r)$, переходит по нему в u;
18. агент A красит $\mu((v,u),v) := b$;
19. $v := u$;
20. агент A записывает в список M: НАЗАД_ОР_А;
21. агент A выполняет *go to* 4 данной процедуры; *end do.*

Неописанные процедуры агента A, представлены в [3].

Алгоритм работы агента B:

1. АИ красит вершину s, в которой находится $(\mu(s) := y)$;
2. запрос BN;
3. *if* $BN \neq 1$ *then do* запрос AN;
4. *if* $\mu(s) = ry$ *then do*
5. агент B выполняет процедуру ВОЗВРАТ_$B(s)$;
6. агент B выполняет процедуру МЕТИМ_*ПЕР*_$B(s)$; *end do*;
7. *else if* AN=0 *then* *МЕТИМ_ПЕР*_$B(s)$;
8. *else* *ВЫБОР_ХОДА*_$B(s)$; *end do*;
9. *else* *РАСП_ПЕР*_$B(s)$.

Выполняя процедуру *ВОЗВРАТ*_$B(s)$, агент B выбирает из окрестности $O(s)$ ребро, у которого $(\mu(s,z) = y)\, and\, (\mu((s,z),s) = y)$ и переходит по нему в вершину z, выполняет присвоение $s := z$ и записывает в список N сообщение: ВОЗВРАТ_В. Процедуры *РАСП*_$B(s)$ и *РАСП_ПЕР*_$B(s)$ агента B аналогичны процедурам *РАСП*_$A(v)$ и *РАСП_ПЕР*_$A(v)$ (с учетом того, что “чужой” вершиной перешейка может быть красно-желтая вершина) агента A. Неописанные процедуры агента B, представлены в [3].

Алгоритм “Восстановление”, и процедуры, не рассмотренные ниже, приведены в [3].

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*="ОТСТУПИЛ_А" *then ОТСТУПИЛ_А*();
10. *if Mes*="РЕБРА_РАСПОЗНАНЫ_А" *then РЕБРА_РАСПОЗНАНЫ_А*();
11. *if Mes*="ОТСТУП_А" *then ОТСТУП_А*();
12. *if Mes*="МЕТКА_ОР_А" *then МЕТКА_ОР_А*();
13. *if Mes*="ВПЕРЕД_ОР_А" *then ВПЕРЕД_ОР_А*().

ОТСТУП_А(): $i := i + 1$.
ВПЕРЕД_АВВ(): $E_H := E_H \cup \{(N\_B,\, r\,(t - i))\}$.
ОТСТУПИЛ_А(): $i := i + 1$.
РЕБРА_РАСПОЗНАНЫ_А(): $i := 0$.
МЕТКА_ОР_А(): $KOBR\_A := KOBR\_A + 1$.
ВПЕРЕД_ОР_А(): $KOBR\_A := KOBR\_A - 1$;
$UDOBR\_A := (KOBR\_A = 0)$; $E_H := E_H \cup \{(r\,(t)\,,\, r\,(t - i))\}$.
ФИКС_А(): $N\_A := Сч\_A$; $BN := 1$; $E := 0$; $Q := F$;
$UDP\_A := (((F = Q)\, or\, (F = 1))\, and\, (Q \neq 1))$.
НАЗАД_АВВ(): $K := K - 1$; $UDP\_B := (((K = Z)\, or\, (K = 1))\, and\, (Z \neq 1))$.

Процедура *ОБР_СП_В*() аналогична процедуре *ОБР_СП_А*(), только добавлено условие: *if Mes*="ВОЗВРАТ_В" *then* ВОЗВРАТ_В().

ВОЗВРАТ_В(): $E_H := E_H \backslash \{(y(p-1), y(p))\}$; $V_H := V_H \backslash \{Сч\_B\}$; *Сч_В:=Сч_В-2*; $p := p - 1$; *y(p):=Сч_В*; $L := 1$; $K := K + 1$.

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

Выводы

В работе предложен алгоритм распознавания графа среды временной и емкостной сложностей $O\left(n^2\right)$. Выполняя этот алгоритм, агенты распознают любой конечный граф G с точностью до изоморфизма. АИ имеют конечную память, независимую от n, где n – число вершин графа, и используют по две краски каждый (всего три краски).

Литература

[1] *Грунский И. С., Татаринов Е. А.* Распознавание конечного графа блуждающим по нему агентом. Вестник Донецкого университета. Серия А. Естественные науки — 2009, вып. 1. — с. 492 — 497.

[2] *Грунский И.С., Стёпкин А.В.* Распознавание конечного графа коллективом агентов. – Труды ИПММ НАН Украины. – 2009. – Т.19. – С. 43-52.

[3] *Стёпкин А.В.* Распознавание графов коллективом агентов. // Мат. IV міжнар. наук.-практ. конф. «Сучасна інформаційна Україна: інформатика, економіка, філософія» 2010, Донецьк, — т.1 — с. 139-143.

[4] *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. — М.: МЦНМО, 2001 — 960 с.

[5] Графы в программировании: обработка, визуализация и применение. — СПБ.: БХВ — Петербург, 2003. — 1104 с.

УДК 378.147

Овчарова О.І.

старший викладач кафедри алгебри, СДПУ

e-mail: znpfizmatsdpu@ukr.net

ІНДИВІДУАЛЬНИЙ ПІДХІД У НАВЧАННІ НА ЗАНЯТТЯХ З «ІНФОРМАТИКИ І ТЗН»

Дана стаття присвячена деяким питанням особистісно-орієнтованого навчання на заняттях з «Інформатики і ТЗН» при підготовці майбутнього вчителя.

Ключові слова: *особистісно-орієнтоване навчання, індивідуальне навчання.*

Вступ

Індивідуальний підхід у навчанні є одним з важливих принципів дидактики. Він сприяє кращому засвоєнню навчального матеріалу, формуванню творчої особистості, позитивної мотивації навчання. Індивідуалізація навчання є особливо актуальною в умовах кредитно-модульної системи навчання у вищих навчальних закладах. При особистісно-орієнтованому навчанні акцент переноситься з навчальної діяльності викладача на пізнавальну діяльність студента, що передбачає підвищення рівня його особистісної активності.

Якість підготовки майбутнього вчителя пов'язана з ефективністю й успішністю навчальної діяльності студента. Водночас успішність будь-якої діяльності, як відомо, визначається активністю особистості та її оптимальним психічним станом. Отже, щоб підвищити якість підготовки майбутніх вчителів, слід активізувати пізнавальну діяльність студентів, забезпечити її ефективність і такі умови навчання, за яких психічні функції були в оптималь-