УДК 519.7

©2016. **А. В. Стёпкин**

ИССЛЕДОВАНИЕ СТРУКТУРЫ ГРАФА ПРИ ПОМОЩИ ДВУХ АГЕНТОВ

В работе рассматривается решение задачи распознавания конечных неориентированных графов двумя агентами. Один агент-исследователь передвигается по графу, считывает и изменяет метки элементов графа и передает информацию о своих действиях агенту-экспериментатору, который строит представление исследуемого графа. Предложен алгоритм квадратической (от числа вершин графа) временной, емкостной и коммуникационной сложностей, который распознает любой конечный неориентированный граф. Для распознавания графа требуется 2 различные краски. Метод основан на методе обхода графа в глубину.

Ключевые слова: *распознавание графа, обход графа, обход в глубину, агенты.*

Введение.

Актуальной проблемой математической кибернетики является проблема взаимодействия управляющей и управляемой систем [1]. Ранее подобное взаимодействие было рассмотрено в [2–4], в предположении, что оно представлено передвижением агентов-исследователей (АИ) по неизвестному графу и обменом данными с агентом-экспериментатором (АЭ), который и производил распознавание графа по данным, полученным от АИ.

Перемещение агента в операционной среде невозможно без построения полной модели выбранной среды. В вопросах такого моделирования определен ряд подходов, одним из которых является топологический [5]. При котором блуждающему агенту доступна информация о связях между различными областями среды и недоступна метрическая и алгоритмическая информация о среде. Зачастую подобная ситуация возникает в роботике [6]. Топологическая модель представляет собой граф, оснащенный дополнительной информацией на ребрах, в вершинах и инциденторах.

Данная работа посвящена решению задачи, в предположении, что взаимодействие управляющей и управляемой систем представляется процессом перемещения одного АИ по конечному неориентированному графу [7]. А суть взаимодействия заключается в обмене данными АИ с АЭ, на основе которого возможно распознавание графа.

1. Основные определения и обозначения.

Для распознавания произвольного конечного неориентированного графа требуется разработать такой алгоритм функционирования агентов, что АИ, помещенный в произвольную вершину неизвестного агентам графа G, все элементы которого окрашены белым цветом, через конечное число шагов обойдет его, передавая АЭ информацию. АЭ, в свою очередь, используя эту информацию, через конечное число шагов построит представление графа H, изоморфного G, то есть распознает граф G.

Рассматриваются конечные, неориентированные графы без петель и кратных ребер. Все неопределяемые понятия общеизвестны, и с ними можно ознакомиться, например, в [8, 9]. Пусть $G = (V, E)$ — граф, у которого V — множество вершин, E — множество ребер, мощности которых равны n и m соответственно. Для уточнения принадлежности множеств V и E некоторому графу G приняты обозначения V_G и E_G соответственно. Ребром будем называть двухэлементное подмножество $\{u, v\}$ множества V, вершины u, v — смежными, а ребро $\{u, v\}$ — инцидентным вершинам u, v. Такое ребро будем обозначать (u, v) или (v, u). Тройку $((v, u), u)$ будем называть инцидентором (точкой соединения) ребра (v, u) и вершины u. Под дальним инцидентором вершины v будем понимать инцидентор $((v, u), u)$, а под ближним — $((v, u), v)$. Множество таких троек обозначим I. Множество $L = V \cup E \cup I$ называется множеством элементов графа G. Пусть M некоторое конечное множество. Функцией раскраски графа G назовем сюръективное отображение $\mu : L \to M$. Пара (G, μ) называется раскрашенным графом.

Окрестностью $Q(v)$ вершины v назовем множество элементов графа, состоящее из вершины v, всех вершин u смежных с v, всех ребер (v, u) и всех инциденторов $((v, u), v), ((v, u), u)$.

Последовательность $u_1, u_2, ..., u_k$ попарно смежных вершин называется путем в графе G, а k — длиной этого пути. Простым путем будем называть путь, в котором все вершины $u_1, u_2, ..., u_k$ попарно различны. В работе рассматриваются только связные графы, то есть графы, в которых между двумя произвольными вершинами существует хотя бы один путь.

Изоморфизмом графов G и 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.

Нумерацией $F : V \to N$ вершин графа G называется инъективное отображение множества его вершин во множество натуральных чисел. Номер вершины v в нумерации F обозначается $F(v)$. Прямой нумерацией вершин графа называется нумерация, соответствующая порядку их обхода при поиске в глубину [8]. Далее будет рассматриваться прямая нумерация. Древесными ребрами [9] называются ребра, порождающие дерево поиска при обходе графа в глубину. Обратные ребра — это ребра соединяющие две несмежные вершины одного дерева поиска в глубину.

По аналогии с [10] агентом будем называть нечто воспринимающее свою среду с помощью датчиков и воздействующее на неё с помощью определенных механизмов. В данной работе рассматривается коллектив, состоящий из двух агентов: агента-исследователя и агента-экспериментатора. Агент-исследователь A обладает следующими свойствами: он передвигаются по графу из вершины v в вершину u по ребру (v, u), может изменять окраску вершин v, u, ребра (v, u) и инциденторов $((v, u), v), ((v, u), u)$ метками из множества M, где $M = \{w, r, b\}$, где w интерпретируется как белый цвет (краска), r — красный, b — черный. Краской будем называть неограниченный ресурс агента. При пометке элемента графа, предыдущая краска стирается и вместо неё, если это необходимо, наносится новая. Помеченным путем будем называть простой путь, образованный последовательностью вершин, помеченных красками в порядке их посещения агентом. Вершины помеченного пути соединяются ребрами, окрашенными красной краской r, один из инциденторов такого ребра помечен красной краской. Если все вершины помеченного пути окрашены красной краской, то будем называть его красным путем. Таким длина красного пути не превосходит число n. Отметим, что в начале работы рассмотренных ниже алгоритмов все элементы графа окрашены в белый цвет.

Находясь в вершине v, агент-исследователь воспринимает метки всех элементов окрестности $Q(v)$ и на основании этой информации определяет по какому ребру (v, u) будет дальше перемещаться и как будет окрашивать элементы из $Q(v)$. Вершину, в которой агент-исследователь находится в данный момент будем называть текущей для данного агента-исследователя вершиной.

Агент-экспериментатор обладает следующими свойствами: это неподвижный агент, который может передавать необходимую информацию агенту-исследователю, а также принимать и идентифицировать сообщения от агента-исследователя; обладает конечной, неограниченно растущей внутренней памятью, в которой фиксируется результат функционирования агента-исследователя на каждом шаге; строит представление графа G, вначале неизвестного агентам, списками ребер и вершин.

Под экспериментом понимается движение агента-исследователя по графу, раскраска элементов графа, сбор локальной информации об окрестности вершины и дальнейший обмен информацией с агентом-экспериментатором. А также построение агентом-экспериментатором графа, изоморфного исследуемому, с точностью до отметок на графе. Для возможности проведения требуемых экспериментов, по априорной информации о графе и о ресурсах агентов, необходимо создать алгоритм работы агентов, осуществляющих эти эксперименты.

Обозначим через $T(n)$ временную сложность выполнения алгоритма распознавания графа, $S(n)$ — емкостную сложность, а через $K(n)$ — коммуникационную сложность. Для оценки сложности вводится асимптотическое обозначение $O(n)$ [11]. В работе асимптотические оценки сложности алгоритмов считаются, как обычно, в равномерной шкале [12].

Отметим, что работа алгоритма осуществляется следующим образом: АИ помещается в произвольную вершину графа G, эта вершина сразу окрашивается в красный цвет. АИ передвигается, пошагово передавая АЭ информацию, АЭ в свою очередь обрабатывает полученные от АИ команды.

2. Стратегия решения задачи.

Рассматриваемый алгоритм основан на стратегии поиска в глубину. Эта стратегия такова: агент идет «в глубину», пока это возможно, возвращается назад, ищет другой путь с еще не посещенными вершинами и не пройденными ребрами. После того, как агент обойдет все вершины и ребра он возвращается в исходную вершину и завершает обход.

Предлагаемая стратегия обладает рядом особенностей: 1) Граф G агентам не известен; 2) При обходе графа G, агенты создают неявную нумерацию пройденных вершин: при первом посещении вершины она окрашивается агентом в красный цвет и ей фактически ставится в соответствие номер, равный значению переменной $Сч\_\ A$. На основе построенной нумерации и происходит распознавание графа G путем построения графа H изоморфного G. В процессе обхода агент строит неявное дерево поиска в глубину. Относительно этого дерева все ребра разделяются на древесные (окрашиваются при первом прохождении по ним красным цветом) и обратные (не принадлежат дереву и окрашиваются при первом прохождении в черный цвет). Древесные ребра проходятся как минимум 2 раза и при последнем проходе окрашиваются агентами в черный цвет. Обратные ребра проходятся от одного до двух раз.

Красные вершины графа G, на каждом шаге алгоритма, образуют красный путь. При проходе в новую вершину красный путь удлиняется, при проходе назад – укорачивается, при распознавании обратного ребра – не изменяется. Вершина, у которой все инцидентные ребра распознаны, окрашивается в черный цвет. Алгоритм заканчивает работу, когда красный путь становятся пустым, а все вершины черными.

3. Обход графа.

В работе АИ можно выделить 2 режима:

1) *Обычный режим.* АИ движется вперед по белым вершинам, окрашивая вершины, соединяющие их ребра и дальние инциденторы в красный цвет. Если нет возможных путей перемещения, то АИ возвращается назад, окрашивая пройденные вершины, ребра и ближние инциденторы в черный цвет. Вернувшись в начальную вершину, АИ завершает работу. На каждом шаге АИ обменивается данными с АЭ.

2) *Распознавание обратных ребер.* Если при движении вперед в вершине v было обнаружено обратное ребро, то АИ прекращает работу в обычном режиме и переключается в режим распознавания обратных ребер. АИ красит в красный цвет ближние инциденторы всех обратных ребер инцидентных вершине v. Завершив покраску инциденторов, АИ передвигается назад по своему пути, до обнаружения вершины инцидентной помеченному обратному ребру (под помеченным обратным ребром понимается белое ребро, у которого дальний инцидентор и дальняя вершина окрашены в красный цвет), переходит по этому ребру, окрашивая его в черный цвет. На этом этапе возможны два случая:

2.1) Распознаны не все, помеченные АИ, обратные ребра. В этом случае АИ возвращается назад по пройденному на предыдущем шаге ребру, окрашивая в черный цвет ближний инцидентор, и продолжает движение назад по своему пути, до обнаружения следующей вершины, инцидентной помеченному обратному ребру.

2.2) Распознаны все, помеченные АИ, обратные ребра. В этом случае АИ окрашивает ближний инцидентор ребра, по которому он перешел на предыдущем шаге, в черный цвет и завершает работу в режиме распознавания обратных ребер.

4. Алгоритмы обхода и восстановления.

Опишем подробно алгоритмы, реализующие описанную выше стратегию. Процесс распознавания состоит из двух принципиально разных типов алгоритмов: «Обход» и «Восстановление». Первый тип алгоритма описывает обход неизвестного графа G агентом-исследователем, с целью проведения серии элементарных экспериментов и передачи информации АЭ. Второй тип алгоритма представляет собой анализ результатов элементарных экспериментов и их объединение, в результате которого будет построен граф H, изоморфный распознаваемому графу G с точностью до отметок на графе.

Рассмотрим непосредственно сами алгоритмы.

Алгоритм работы АИ:

Вход: граф G неизвестный АИ и АЭ, все элементы графа G окрашены краской w, АИ помещен в произвольную вершину v.

Выход: все элементы графа G, которые попадут в область работы АИ, окрашены краской b; АИ находится в исходной вершине v; пошагово выданы команды АЭ.

Данные: v – рабочая вершина графа G, в которой находится агент.

1. Агент A красит $(\mu(v) := r)$;
2. $if\ \exists(v,u) \in Q(v) : (\mu(v,u) = w)\, and\, (\mu(u) = \mu(v) = r)\ then\ РАСП\_A(v)$;
3. $else\ if\ \exists(v,u) \in Q(v) : (\mu(v,u) = w)\, and\, (\mu(u) = w)\ then\ do$
4. агент A выполняет процедуру $ВПЕРЕД\_A(v)$;
5. $go\ to$ 2;
6. $end\ do$;
7. $else\ if\ \exists(v,u) \in Q(v) : (\mu(v)=r) and (\mu((v,u),v)=r) and (\mu(v,u)=r)\ then\ do$
8. агент A выполняет процедуру $НАЗАД\_A(v)$;
9. $go\ to$ 2;
10. $end\ do$;
11. $else$ агент A выполняет процедуру $СТОП\_A(v)$;

$РАСП\_A(v)$

1. $while\ \exists(v,u) \in Q(v) : ((\mu(v) = \mu(u) = r)\, and\, (\mu(v,u) = w))\ do$
2. агент A красит $\mu\left((v,u), v\right) := r$;
3. агент A записывает в M: МЕТКА_ОР_А;
4. $end\ do$;

5. агент A выбирает $(v,u) \in Q(v) : (\mu\left((v,u)\,,v\right) = r)\, and\, (\mu\,(v,u) = r)\, and$ $and\,(\mu\,(u) = r)$ и переходит по нему в вершину u;
6. $v := u$;
7. агент A записывает в M: ОТСТУП_А;
8. *if* $\nexists(v,u) \in Q(v) : (\mu(v,u) = w) and (\mu((v,u),u) = r) and (\mu(v) = \mu(u) = r)$ *then go to* 5 данной процедуры;
9. *else do*
10. агент A переходит по ребру (v,u), красит $\mu(v,u) := b$;
11. $v := u$;
12. агент A записывает в список M: ВПЕРЕД_ОР_А;
13. *end do*;
14. запрос $UDOBR\_A$;
15. *if* $UDOBR\_A = TRUE$ *then do*
16. агент A выбирает $(v,u) \in Q(v) : (\mu(v,u) = b)\, and\, (\mu\left((v,u),v\right) = r)\, and$ $and\,(\mu(v) = \mu(u) = r)$ и красит $\mu\left((v,u),v\right) := b$;
17. агент A записывает в список M: РЕБРА_РАСПОЗНАНЫ_А;
18. *end do*;
19. *else do*
20. агент A выбирает $(v,u) \in Q(v) : (\mu(v,u) = b)\, and\, (\mu\left((v,u),v\right) = r)\, and$ $and\,(\mu\,(v) = \mu\,(u) = r)$ и переходит по нему в вершину u;
21. агент A красит $\mu\left((v,u),v\right) := b$;
22. $v := u$;
23. *go to* 5 данной процедуры;
24. *end do*;

При выполнении процедуры *ВПЕРЕД_А*(v), агент A выбирает из окрестности $Q(v)$ произвольное ребро (v,u), у которого $\mu(v,u) = w, \mu(u) = w$ и переходит по нему в вершину u. При этом окрашивает $\mu(v,u) := r, \mu\left((v,u),u\right) := r, \mu(u) := r$, выполняет присвоение $v := u$ и записывает в список M сообщение: ВПЕРЕД_А.

В процессе выполнения процедуры $НАЗАД\_A(v)$, агент A выбирает из окрестности $Q(v)$ ребро, для которого выполняется условие $(\mu(v) = r)\, and\, (\mu\,((v,u),v) = r)\, and\, (\mu(v,u) = r)$, и переходит по нему в вершину u. При этом, производит окрашивание $\mu(v) := b, \mu\,((v,u),v) := b, \mu(v,u) := b$, выполняет присвоение $v := u$ и записывает в список M сообщение: $НАЗАД\_A$.

При выполнении процедуры $СТОП\_A(v)$, агент A окрашивает $\mu(v) := b$, записывает в список M сообщение: $СТОП\_A$ и завершает работу.

Выполняя процедуру $ОТСТУП\_A(v)$, агент A выбирает из окрестности $Q(v)$ ребро (v,u), у которого $(\mu\,((v,u),v) = r)\, and\, (\mu(v,u) = r)$ и переходит по нему в вершину u, выполняет присвоение $v := u$ и записывает в список M сообщение: ОТСТУП_А.

Алгоритм работы АЭ:

Вход: список сообщений M от агента-исследователя.

Выход: список вершин V_H и ребер E_H графа H, изоморфного графу G.

Данные: V_H, E_H списки вершин и ребер графа H, изоморфного графу G. $Сч\_A$ — счетчик числа посещенных вершин графа G агентом A. M — список сообщений от агента-исследователя. i — счетчик используемый агентом A при подсчете номеров вторых вершин помеченных обратных ребер. $STOP\_A$ — переменная, используемая агентом A, для сигнализации агенту-экспериментатору, о завершении обхода графа. $UDOBR\_A$ — логическая переменная, используемая агентом A для определения является ли рассматриваемое обратное ребро последним из помеченных. $KOBR\_A$ — переменная, в которой агент A записывает количество помеченных обратных ребер. $r(1), r(2), ..., r(t)$ — список номеров вершин красного пути, где t — длина этого списка. Mes — текущее сообщение.

1. $Сч\_A := 1, M := \varnothing, i := 0, E_H := \varnothing, STOP\_A := 0, UDOBR\_A := FALSE$, $KOBR\_A := 0$, $t := 1$, $r(t) :=Сч\_A$, $V_H := \{1\}$;
2. *while* $STOP\_A = 0$ *do*
3. *if* $M \neq \varnothing$ *then do*
4. считать в Mes сообщение и удалить его из очереди M;
5. $ОБР\_СП\_A()$;
6. *end do*;
7. *end do*;
8. печать V_H, E_H.

$ОБР\_СП\_A()$:

1. *if* $Mes =$ "$ВПЕРЕД\_A$" *then* $ВПЕРЕД\_A()$;
2. *if* $Mes =$ "$НАЗАД\_A$" *then* $НАЗАД\_A()$;

3. $if\ Mes =$"$РЕБРА\_\ РАСПОЗНАНЫ\_\ A$"$then$ $РЕБРА\_\ РАСПОЗНАНЫ\_\ A()$;
4. $if\ Mes =$"$ОТСТУП\_\ A$"$then$ $ОТСТУП\_\ A()$;
5. $if\ Mes =$"$МЕТКА\_\ ОР\_\ A$"$then$ $МЕТКА\_\ ОР\_\ A()$;
6. $if\ Mes =$"$ВПЕРЕД\_\ ОР\_\ A$"$then$ $ВПЕРЕД\_\ ОР\_\ A()$;
7. $if\ Mes =$"$СТОП\_\ A$"$then$ $СТОП\_\ A()$.

$ВПЕРЕД\_\ A()$: выполняются операции: $Сч\_\ A{:=}Сч\_\ A{+}1$; $t := t+1$; $r\,(t) := Сч\_\ A$; $V_H := V_H \bigcup \{Сч\_\ A\}$; $E_H := E_H \bigcup \{(r(t-1), r(t))\}$;
$НАЗАД\_\ A()$: из списка $r(1)...r(t)$ удаляется элемент $r(t)$; $t := t-1$;
$РЕБРА\_\ РАСПОЗНАНЫ\_\ A()$: $i := 0$;
$ОТСТУП\_\ A()$: $i := i+1$;
$МЕТКА\_\ ОР\_\ A()$:$KOBR\_A := KOBR\_A + 1$;
$ВПЕРЕД\_\ ОР\_\ A()$:$KOBR\_A := KOBR\_A - 1$; $UDOBR\_A := (KOBR\_A = 0)$; $E_H := E_H \bigcup \{(r\,(t)\,,\, r\,(t-i))\}$;
$СТОП\_\ A()$: $STOP\_A := 1$;

5. Свойства алгоритма распознавания.

Теорема. *Три агента, выполнив алгоритм распознавания, распознают любой граф G с точностью до изоморфизма.*

Доказательство. В начале работы алгоритма распознавания, при $n \geqslant 3$, как минимум два раза, когда агент-исследователь посещает белые вершины исследуемого графа G, выполняется процедура: $ВПЕРЕД\_\ A(v)$. Процедурой агента-экспериментатора $ВПЕРЕД\_\ A()$ создается новая вершина графа H. Таким образом, процесс выполнения описанного алгоритма индуцирует отображение $\varphi : V_G \to V_H$ вершин графа G в вершины графа H. Причем равенство $\varphi\,(v) = Сч\_\ A$ устанавливается, когда вершина v красится в красный цвет.

Указанное отображение естественным образом устанавливает неявную нумерацию вершин графа G. Более того, отображение φ является биекцией, поскольку рассматриваются связные графы, а в таких графах все вершины достижимы из начальных вершин. Поэтому все вершины посещаются агентом-исследователем (то есть окрашиваются в красный цвет) и при первом посещении вершины агентом ей присваивается единственный номер.

Из описания алгоритма следует, что агент-исследователь проходит все ребра графа G, поскольку по окончании алгоритма все ребра становятся черными. Выполняя процедуру $ВПЕРЕД\_\ A()$ агент-экспериментатор распознает древесное ребро $(v{,}u)$ и так нумерует вершину u, что ребру (v,u) однозначно соответствует ребро $(\varphi(v), \varphi(u))$ графа H. Выполняя процедуру $ВПЕРЕД\_\ ОР\_\ A()$ агент-экспериментатор распознает обратное ребро $(v{,}u)$ и ставит ему в однозначное соответствие ребро $(\varphi(v), \varphi(u))$ графа H. Следовательно, φ является изоморфизмом графа G на граф H. Что и требовалось доказать.

Подсчитаем временную, емкостную и коммуникационную сложности в равномерной шкале [12]. Рассмотрим подробнее свойства красного пути. Из описания алгоритма следует, что на каждом шаге алгоритма красный путь — это простой путь, берущий начало в вершине v с номером $\varphi(v) = 1$ и оканчивающийся в вершине u с номером $\varphi(u) = Сч\_A$. Следовательно, длина красного пути не превосходит n.

Каждый раз при выполнении процедур $ВПЕРЕД\_A(v)$ и $НАЗАД\_A(v)$ агент-исследователь проходит одно ребро. При однократном выполнении процедуры из режима распознавания обратных ребер агент-исследователь метит не более $n-2$ обратных ребер, по одному разу проходит не более $n-1$ ребер красного пути, а также по два раза проходится не более $n-2$ обратных ребер.

При подсчете временной сложности алгоритма, будем считать, что инициализация алгоритма, и выбор одной из возможных процедур занимают некоторое постоянное число единиц времени. Также будем считать, что проход по ребру агента-исследователя и обработка команды агента-экспериментатора полученной на данном этапе от агента-исследователя осуществляется за время равное некоторой константе. Примем, что время пересылки одного сообщения равно константе, которая не превышает времени прохода по ребру агента-исследователя. Причем общее время, затрачиваемое на анализ окрестности $Q(v)$ вершины и выбор необходимых ребер оценивается как $O(n^2)$ [8]. Тогда временная сложность алгоритма определяется следующими соотношениями:

1. Процедуры $ВПЕРЕД\_A(v)$ и $НАЗАД\_A(v)$ выполняются не более чем $2\cdot(n-1)$ раз, общее время их выполнения оценивается как $O\left(n\right)$.
2. Время, затрачиваемое на работу в режиме распознавания обратных ребер оценивается как $n \cdot 4 \cdot O(n)$, то есть как $O\left(n^2\right)$.

Таким образом верхняя оценка числа переходов по ребрам, совершаемых агентом-исследователем равняется $O(n^2)$, а суммарная временная сложность $T(n)$ алгоритма удовлетворяет соотношению: $T\left(n\right) = O\left(n^2\right)$.

Емкостная сложность $S(n)$ алгоритма определяется суммарной сложностью списков $V_H, E_H, r(1), ..., r(t)$, сложность которых соответственно определяется величинами $O\left(n\right), O\left(n^2\right), O\left(n\right)$.

Следовательно: $S\left(n\right) = O\left(n^2\right)$.

На каждом шаге алгоритма агент-исследователь отправляет агенту-экспериментатору одно сообщение. Поэтому объем передаваемой агентами информации оценивается как $K(n) = O\left(n^2\right)$.

Учитывая описанные выше предположения о способе подсчета временной сложности, имеет место следующая теорема:

Теорема. *Временная, емкостная, коммуникационная сложности алгоритма и число переходов по ребрам, совершаемых агентом-исследователем, равны* $O\left(n^2\right)$, *где* n *— число вершин графа. Для распознавания достаточно двух красок.*

6. Выводы. Предложен алгоритм распознавания графа среды временная, емкостная, коммуникационная сложности и число переходов по ребрам которой равны $O\left(n^2\right)$. Агент-исследователь имеет конечную память, независимую от n, и использует две краски.

1. *Капитонова Ю. В., Летичевский А. А.* Математическая теория проектирования вычислительных систем. – М.: Наука. 1988. – 296 с.
2. *Грунский И.С., Татаринов Е. А.* Распознавание конечного графа блуждающим по нему агентом // Вестник Донецкого университета. Серия А. Естественные науки. – 2009. – Т. 1. – С. 492–497.
3. *Стёпкин А. В.* Использование коллектива агентов для распознавания графов // Компьютерные исследования и моделирование. – 2013. – Т. 5, № 4. – С. 525–532.
4. *Stepkin A.* Using a Collective of Agents for Exploration of Undirected Graphs // Cybernetics and Systems Analysis. – 2015. – V. 51, № 2. – P. 223–233.
5. *Kuipers B.* The spatial semantic hierarchy // Artificial Intelligence. – 2000. – V. 119, № 1–2. – P. 191–233.
6. *Dudek G., Jenkin M.* Computational principles of mobile robotics. – Cambridge Univ. press, Cambridge. 2000. – 280 p.
7. *Кудрявцев В. Б., Алешин С.В., Подкозлин А. С.* Введение в теорию автоматов. – М.: Наука, 1985. – 320 с.
8. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. – М.: МЦНМО, 2001. – 960 с.
9. *Касьянов В. Н., Евстигнеев В. А.* Графы в программировании: обработка, визуализация и применение. – СПБ.: БХВ–Петербург, 2003. – 1104 с.
10. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход. — 2-е издание. — М.: «Вильямс», 2006. — 1408 с.
11. *Килибарда Г., Кудрявцев В. Б., Ушчумлич Щ.* Коллективы автоматов в лабиринтах // Дискретная математика. — 2003. – Т. 15, № 3. — С. 3–40.
12. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979. — 536 с.

A. V. Stepkin

Exploration of the graph structure by two agents.

This paper considers the problem of exploration of finite undirected graphs by two agents. One agent-researcher traverse a graph, read and change labels of graph elements, and send necessary information to the agent-experimenter constructing a representation of the graph being explored. An exploration algorithm is proposed with a quadratic (with respect to the number of nodes) time complexity, space complexity and communication complexity. An algorithm is proposed explored any finite undirected graph. Graph's exploring needs two different colors. The algorithm is based on the depth-first traversal method.

Keywords: *graph exploration, traversal of a graph, depth-first traversal method, agents.*

А. В. Стьопкін

Дослідження структури графа за допомогою двох агентів.

В роботі розглядається розв'язок задачі розпізнавання скінчених неорієнтованих графів двома агентами. Один агент-дослідник рухається графом, зчитує та змінює помітки на елементах графу та передає інформацію про свої дії агенту-експериментатору, який будує уявлення про досліджуваний граф. Пропонується алгоритм квадратичної (від кількості вершин графу) часової, ємнісної та комунікаційної складностей, який розпізнає довільний скінчений неорієнтований граф. Для розпізнавання графу необхідно дві різні фарби. Метод базується на методі обходу графа в глибину.

Ключові слова: *розпізнавання графу, обхід графу, обхід в глибину, агенти.*

Донбасский государственный педагогический университет,
Славянск
stepkin.andrey@rambler.ru

Получено 06.09.16