УДК 519.7

Н.К. Шатохина (канд. техн. наук, доц.), П.А. Шатохин (канд. техн. наук, доц.)
ГВУЗ «Донецкий национальный технический университет», г. Донецк
кафедра автоматизированных систем управления
E-mail: nshatokh@rambler.ru, pshatokh@mail.ru

РАСПОЗНАВАНИЕ ГРАФА МОЗАИЧНОЙ СТРУКТУРЫ, СОСТОЯЩЕГО ИЗ СВЯЗНЫХ КОМПОНЕНТОВ И МОСТОВ, КОЛЛЕКТИВОМ АГЕНТОВ

Рассмотрена задача описания структуры графа специального вида без дыр, состоящего из нескольких связных компонентов и мостов, на основе информации, полученной при обходе его по границе. Описан алгоритм решения задачи, приведены оценки его временной и емкостной сложности.
Ключевые слова: *мозаика, лабиринт, коммутативная группа, агент, алгоритм.*

Общая постановка проблемы

Задачи восстановления структуры графа по частичной о нем информации, например по информации, полученной при обходе графа по его границе, возникают в робототехнике, молекулярной физике, в игровых приложениях: построение модели карты среды, модели объекта [1].

В [2] приведен обзор решения подобных задач с использованием одного автомата (далее агента). В работах [3, 4] предлагаются алгоритмы, с помощью которых производится распознавание структуры графа, состоящего из одной связной компоненты, и описание ее в символьном виде с использованием двух агентов. Данная работа является продолжением этих работ, и рассматривает случай графа без дыр, состоящего из нескольких связных компонент.

Первый агент-исследователь (АИ) выполняет перемещения по исходному графу и передачу информации второму агенту-экспериментатору (АЭ). Второй агент по полученной информации описывает структуру графа.

Основные определения и обозначения

В данной работе рассматривается неориентированный граф G без дыр, петель и кратных ребер. Для графа G(V,E), где V – множество вершин, $E=\{(v_i,v_j)|\ (v_i,v_j) \in V\}$ – множество ребер, заданы такие ограничения, описывающие его мозаичную структуру:

1) для всех $v \in V$ выполняется соотношение $1 \le deg(v) \le 6$;
2) для всех внутренних вершин выполняется deg(v)=6 и существуют шесть маршрутов длины 3 из v в v, включающие всевозможные пары соседних ребер;
3) длина всех ребер $e=(v_1,v_2) \in E$ постоянна, $|e|$ = const.

Граф называется связным, если для любых двух вершин v, w его вершин существует маршрут $(v,v_1),(v_1,v_2),\ldots,(v_{n-1},w)$, связывающий эти вершины.

Лучом будем называть ребро, одна из вершин которого имеет единственное инцидентное ребро.

Обозначим множество инцидентных ребер вершины v через $E(v)=\{e_1, \ldots\}$, $|E(v)| \le 6$. На E(v) определим функцию q: $E(v) \rightarrow H=\{x, x^{+1}, x^{+2}, x^{+3}, x^{+4}, x^{+5}\}$.

Множество H описывает метки инцидентных ребер вершины v, расставляя их смежным ребрам в порядке поворота на угол 60° против часовой стрелки относительно некоторого случайно взятого ребра x=(w,v). Соответственно, если процесс расстановки меток осуществлять в порядке поворота по часовой стрелки, то множество $H=\{x, x^{-1}, x^{-2}, x^{-3}, x^{-4}, x^{-5}\}$. Здесь степень букв обозначает количество выполненных поворотов вокруг вершины v против часовой стрелки на фиксированный угол 60° относительно ребра x. Заметим, что $x^{+3}= x^{-3} = \bar{x}$,здесь $\bar{x}$ является ребром диаметрально противоположным ребру x.

Далее будем рассматривать маршруты по границам графа, в которых порядок просмотра ребер будет производиться против часовой стрелки.

Заметим, если рассматривать расположенные на контуре мозаики вершины v, то множество меток H инцидентных ребер таких вершин будет состоять из меньшего количества элементов, т.к. $deg(v)\leq 6$. Если же рассматривать последовательности меток на ребрах, порождаемые при перемещении агента по контуру мозаики, то на основе множеств меток H(x) будут создаваться строки вида $x\, x^{l1}\, x^{l2} \ldots x^{ln+1}\, x^{ln}$. Каждый элемент этой строки описывает направление l_i очередного ребра i относительно первого выбранного ребра x.

Маршрут называется выпуклым, если для любых двух последовательных элементов $s^{li}\, s^{li+1}$ выполняется условие $0\leq l_{i+1}-l_i\leq 2$, т.е. между двумя последовательными направлениями агент выполнил $l_{i+1}-l_i$ поворотов. Нетрудно сообразить, что разность $l_{i+1}-l_i$ оценивает, какое по счету из H взято направление относительно x в качестве следующего.

Фрагмент маршрута называется вогнутым, если для любой пары подряд следующих направлений $x^{li}\, x^{li+1}$, выполняется условие $l_{i+1}-l_i\geq 4$. Первая вершина вогнутого фрагмента называется вершиной перегиба.

Показано, что единичному перемещению агента по ребру (v_i,v_{i+1}) в некотором направлении l_i соответствует свободный вектор, с началом в вершине v_i и концом в вершине v_{i+1}. В дальнейшем будем считать понятия направления перемещения и свободного вектора, расположенного вдоль его, эквивалентными. Процессу перемещения агента по контуру мозаики можно поставить в соответствие последовательность векторов (строку направлений), для которой выполняется: начало каждого следующего вектора располагается в конце предыдущего. Показано, что процесс перемещения описывается сложением соответствующих свободных векторов. Рассмотрены свойства операции сложения векторов, приведена таблица Кэли, описывающая эти свойства:

Таблица 1

Таблица Кэли

	x	$\mathbf{x^{+1}}$	$\mathbf{x^{+2}}$	$\mathbf{x^{+3}}$	$\mathbf{x^{+4}}$	$\mathbf{x^{+5}}$
x	2 x	$x^{+1}x$	x^{+1}	0	x^{+5}	$x^{+5}x$
$\mathbf{x^{+1}}$	xx^{+1}	$2\, x^{+1}$	$x^{+2}x^{+1}$	x^{+2}	0	x
$\mathbf{x^{+2}}$	x^{+1}	$x^{+1}x^{+2}$	$2\, x^{+2}$	$x^{+3}x^{+2}$	x^{+3}	0
$\mathbf{x^{-1}}$	0	x^{+2}	$x^{+2}\, x^{+3}$	$2\, x^{+3}$	$x^{+4}\, x^{+3}$	x^{+4}
$\mathbf{x^{-2}}$	x^{+5}	0	x^{+3}	$x^{+3}x^{+4}$	$2\, x^{+4}$	$x^{+5}x^{+4}$
$\mathbf{x^{-3}}$	xx^{+5}	x	0	x^{+4}	$x^{+4}x^{+5}$	$2\, x^{+5}$

Показано, что данная операция позволяет задать функцию $f: (E(v), q(v,w))\rightarrow V$ так, что $f(v, q(v,w))=w$, где (v,w) ребро с меткой q(v,w), определяя на G лабиринт.

Лабиринтом L=(G, t, q, f) называется граф G, на котором определены t – разметка вершин, q – разметка ребер и f – функция перемещения таким образом, что метки на ребрах определяются углом поворота агента относительно некоторого первого ребра.

На основе этой таблицы показано, что данная операция определяет коммутативную группу. Комбинации векторов, именующие строки и столбцы, на пересечении которых находится 0, являются обратным векторам.

Нетрудно построить правило для проверки являются ли вектора обратными, а направления инверсными:

$$|i-j|=3 \tag{1}$$

Комбинации векторов, именующие строки и столбцы x_i, x_j, на пересечении которых находится один вектор, равный их сумме x_p, описывает основные элементы рассматриваемой мозаики – базовый и обратный треугольники [3]. По смыслу эта ситуация соответствует тому факту, что агент из одной вершины в другую может пройти либо по двум ребрам, которые соответствуют заголовкам строк и столбцов, либо по одному – указанному в клетке таблицы. Такие комбинации будем называть сокращаемыми.

Анализируя таблицу Кэли, нетрудно построить правила сокращения векторов:

$$\begin{array}{l} \text{Если } j-i=2 \text{ или } i-j=4, \text{ тогда } \quad p=i+1 \text{ (для выпуклого треугольника);} \\ \text{Если } j-i=4 \text{ или } i-j=2, \text{ тогда } \quad p=j+1 \text{ (для вогнутого треугольника).} \end{array} \tag{2}$$

Важным свойством коммутативной группы является тот факт, что сумма элементов в выражении не зависит от их порядка следования, поэтому приведенные правила могут быть применены к произвольно расположенным векторам в строке, а не только к соседним.

Утверждение 1 Множество отметок ребер контура представляет циклическую строку, содержащую все возможные пути обхода графа.

Действительно, поскольку маршрут по граничным вершинам является замкнутым, то соответствующая строка поворотов является циклической, т.е. $v_n= v_1$ и $x_n = x$.

Далее с помощью этой строки однозначно описываются все возможные пути обхода графа. Путь по граничным вершинам из вершины v_i описывается строкой $x_i\, x_1\, x_{i+1} \ldots x_n\, x_2 \ldots x_{i-1}$.

Утверждение 2 Каждой связной компоненте мозаики соответствует линейно зависимая комбинация свободных векторов.

Известно, что сумма свободных векторов равна нулю для любого замкнутого контура, вдоль которого они последовательно размещаются. Следовательно, если граф состоит из нескольких связных компонентов, то для каждой связной его компоненты выполняется это утверждение.

Поэтому решение задачи выделения из графа связных компонентов сводится к выделению из строки векторов (направлений) подстрок, которым соответствует линейно зависимая комбинация свободных векторов, т.е. равная 0.

Базовые мозаики

В [3] рассмотрены основные базовые фигуры, которые используются при описании структуры исследуемой мозаики.

Для **треугольника** маршрут по граничным ребрам против часовой стрелки представим строкой поворотов вида $S=s_1^{+2}\, s_2^{+2}\, s_3^{+2}$, а по часовой стрелке - строкой $S= s_1^{-2}\, s_2^{-2}\, s_3^{-2}$. Здесь параметр s_1, s_2, s_3 – натуральные числа, равные количеству ребер, расположенных вдоль сторон треугольника, и $s_1=s_2=s_3$.

Для **параллелограммов**, имеют место две строки, описывающие маршруты обхода: $S=s_1 s_2^{+1} s_3^{+3} s_4^{+4}$ ($S= s_1 s_2^{+4} s_3^{+3} s_4^{+1}$ - по часовой стрелке) и $S= s_1 s_2^{+2} s_3^{+3} s_4^{+5}$ ($S= s_1 s_2^{+5} s_3^{+3} s_4^{+2}$ - по часовой стрелке).

Для **трапеций** (их может быть 4 вида в зависимости от начальной стороны), маршруты описываются следующим образом:

–если обход начинается с большего основания трапеции, то строка обхода будет $S = s_1^{+2}s_2^{+1}s_3^{+1}s_4^{+2}$ и по часовой стрелке - $S = s_1^{-2}s_2^{-1}s_1^{-1}s_2^{-2}$, причем $s_1 = s_2 + s_3$, $s_4 = s_2$;

–в случае если обход начинается с правой боковой стороны трапеции, то строка обхода будет $S = s_1^{+1}s_2^{+1}s_3^{+2}s_4^{+2}$ и по часовой стрелке - $S = s_1^{-1}s_2^{-1}s_1^{-2}s_2^{-2}$, причем $s_4 = s_2 + s_1$, $s_1 = s_3$;

–если обход начинается с меньшего основания трапеции, то строка обхода будет $S = s_1^{+1}s_2^{+2}s_3^{+2}s_4^{+1}$, по часовой стрелке - $S = s_1^{-1}s_2^{-2}s_1^{-2}s_2^{-1}$, причем $s_4 = s_2 + s_1$, $s_1 = s_3$;

–если обход начинается левой боковой стороны трапеции, то строка обхода будет $S = s_1^{+2}s_2^{+2}s_3^{+1}s_4^{+1}$, по часовой стрелке - $S = s_1^{-2}s_2^{-2}s_1^{-1}s_2^{-1}$, причем $s_2 = s_4 + s_3$, $s_1 = s_3$;

Шестиугольник. Маршруту по границе мозаики против часовой стрелки соответствует строка $S = s_1^{+1}s_2^{+1}s_3^{+1}s_4^{+1}s_5^{+1}s_6^{+1}$, по часовой стрелке - $S = s_1^{-1}s_2^{-1}s_3^{-1}s_4^{-1}s_5^{-1}s_6^{-1}$, а также $s_1 = s_2 = s_3 = s_4 = s_5 = s_6$.

Поскольку рассматриваются графы произвольной структуры, описание которых будет производиться в виде комбинации базовых, то для идентификации базовых структур из вышеприведенных строк удаляются последние стороны [3], т.к. они будут представлены в обеих базовых фигурах. Например, для треугольника используется правило: $S = s_1^{+2}s_2^{+2}$ и $s_1 = s_2$.

Для идентификации в произвольном графе основных базовых фигур по указанным правилам разработаны отдельные алгоритмы.

Свойства агентов

Установленный в некоторую вершину графа, АИ может пользоваться локальной информацией о вершине – ее степенью. Он может выполнять повороты на 60° по часовой и против часовой стрелки, а также может перемещаться из вершины в вершину по ребрам и определять направление своего перемещения, и передавать информацию АЭ, отмечать некоторые ребра.

Агент АЭ принимает сообщения от АИ, выделяет фрагменты строки, соответствующие обходу отдельным связным компонентам, описывает представление их в символьном виде – в виде формулы, отражающей состав компоненты как комбинацию базовых структур.

Постановка задачи

Требуется разработать такие алгоритмы передвижения агента АИ по исходному графу G с передачей информации об этом, и восстановления графа агентом АЭ по полученным данным в виде формулы, что АИ за конечное число шагов независимо от начального размещения обойдет часть внутренних вершин G и все граничные вершины, пошагово передавая информацию, по которой АЭ через конечное число шагов опишет граф P, изоморфный исходному графу G.

Процесс распознавания состоит из двух алгоритмов как и в [3,4]. Первый алгоритм «Обход» описывает перемещение АИ по неизвестному графу G и генерацию информации для АЭ. Второй алгоритм «Восстановление», представляет собой анализ полученной информации, на основании которой создается символьное описание графа P, изоморфного исходному G.

Алгоритм «Обход» работы АИ

Предполагается, что агент случайным образом может быть помещен в любую вершину графа, поэтому алгоритм состоит из двух этапов. На первом этапе АИ выходит на границу, а на втором этапе АИ обходит граф против часовой стрелки по граничным ребрам до тех пор, пока не возвратится в вершину выхода на границу. Первый этап может отсутствовать.

Приведем подробный алгоритм ввиду его отличия от ранее приведенного.

Алгоритм АИ выхода на границу. Агент, помещенный в произвольную вершину v, определяет ее степень, если это внутренняя вершина, т.е. deg(v)=6, то перемещается по диаметрально противоположному ребру относительно первого встреченного ребра до следующей вершины. Эту процедуру АИ повторяет до тех пор, пока не найдет граничную вершину. Выбирает первое граничное ребро путем поворота против часовой стрелки, которое помечает маркером черного цвета.

```
Вход: v-некоторая вершина графа.
Выход: S – строка направлений.
{  i=0;
   определить deg(v)
  If deg(v)=6   then                          // выход на границу
   { выбрать  x=(v,w);                        // случайно выбранное ребро
    x=x^-3 ;                                  // диаметрально противоположное ребро
    v= (v,x );                                // переход к следующей вершине
    while deg(v)=6
     {       x=x^-3; v= (v,x );               // движение внутри графа
      }
  }
else
{ while (v,x^-1) ≠∅                   // поиск отсутствующего ребра
   { x= x^-1; v= (v,x);}
     x= x^+1;    S_1 ={x};  i=i+1;    // ребро найдено, сформировать строку направлений
     V_1 =black;                              //пометить  черным цветом
}}
```

Алгоритм обхода графа по граничным вершинам. Дальнейшее перемещение агента происходит по границе графа в направлении против часовой стрелки. Агент перемещается по выбранному ребру, фиксируя направление своего перемещения. Далее АИ проверяет метку ребра. Если ребро не помечено, тогда он пересылает букву АЭ и продолжает движение. Если ребро помечено черным цветом, тогда он помечает его красным цветом, пересылает букву АЭ и продолжает движение. Ребро помечено красным цветом – завершает свою работу.

```
{  i=2;
   While V_i ≠ red
      {  Найти граничное ребро x_i^li;
         S_i = x_i^li ;                  //  формируется строка направлений
         If   V_i = black then   V_i = red;                // формирование метки  ребра
              i=i+1;
}   }
```

Временная сложность оценивается [3] как O(n) шагов, емкостная сложность E(n) отсутствует, т.к. агенту ничего не требуется сохранять.

Алгоритм «Восстановление» работы АЭ

В алгоритме анализируется строка направлений, сгенерированная и переданная АИ. Производится поиск подстроки S′, соответствующей очередной связной компоненте. Результатом являются номера начала St_S и конца End_s найденной подстроки.

Алгоритм поиска связных компонент

Основной идеей поиска подстрок является поиск двух позиций в исходной строке, между которыми находится одинаковое количество обратных векторов и нет ни одной пары сокращаемых векторов.

Просмотр исходной строки будет осуществляться в прямом порядке, начиная со второй позиции. При этом для каждой позиции производится просмотр строки в обратном порядке, для поиска обратных и сокращаемых векторов.

Вход: $S=\{ x_1^{l1} x_2^{l2} \dots x_n^{ln} \}$ - строка направлений (свободных векторов);

$V=\{v_1, v_2, \dots , v_n\}$ – множество индексов S, n – количество граничных вершин.

Выход: St_s, End_s – номера начальной и конечной вершин (номера первого и последнего ребра) связной компоненты.

```
{ S′= S;
 { i=2;  j=i-1;
   St_s=Vi ; End_s= Vj ;
   while   n>0                            // просмотр строки в прямом порядке
   {  while   j≥1                         // просмотр строки в обратном порядке
     { j = i-1;
       If  lj-li=2 or  li-lj=4 then       // xi^li ,  xj^lj   организуют выпуклый треугольник
       { Алгоритм Границы;        lj = li +1;
       }
       else
       { if  lj-li=4 or  li-lj=2 then      // xi^li, xj^lj  организуют вогнутый треугольник
         { Алгоритм Границы;       li= lj+1;
         }
         else
         {  if | lj-li=3 |   then          // xi^li ,  xj^lj     удовлетворяют правилу  (1)
             {  if j=i-1 then
               { Алгоритм Границы;
                 j=j-1;    // два последних вектора обратные
               }
               else
                 For p=j+1 to i do  xp-1^lp-1 =xp^lp;  // удалить ,  xj^lj  -  j позицию сдвигом влево
                 j=j-1;
                  i-i-1;   n=n-1;                    //   удалить последнюю позицию  xi^li
             }
             i=i-1; n=n-1;
           }
             else  j=j-1;
       } }
         if i=j then                      // выделена строка связного подграфа
       {  Алгоритм Мост;       //  есть ли мост, соединяющий выделенный подграф?
          If fl=1 then               // перенос невошедшей начальной части строки в конец S
     { For p=1 to i-1  S′n+p= S′p ;
        fl=2;  i=1; j=i+1;
     }     }
     i=i+1;
} }}}
```

Алгоритм Границы;

```
{St_S=Vi ;
If Vj >End_s  then
```

```
If fl=0 then
  { End_s = Vj;      Fl=1;    }
else    End_s = Vj;
}
```

Алгоритм Мост;

{ p=0; v=i;

While $x_i^{li} = x_{i-p+1}^{li-p+1}$ p=p+1;

If p=>0 then // есть мост

{ For k=i+p to n $x_{k-2h}^{k-2p} = x_k^k$;

n=n-2p; F=F+M(v,p);

} }

В S′ производится поиск точки перегиба [4], слева от которой расположен вогнутый фрагмент подстроки, а справа выпуклый. Дальнейшая обработка найденного фрагмента осуществляется стандартными подалгоритмами, перебор которых осуществляется в порядке уменьшения сторон: от шестиугольника к треугольнику. В результате их работы в строку описания графа добавляется символьное обозначение найденной базовой структуры и из строки S′, описывающей связную компоненту, удаляется подстрока, соответствующая найденной базовой структуре.

Процесс описания связной компоненты продолжается до тех пор, пока S′ не станет пустой строкой.

Процесс описания структуры графа продолжается до тех пор, пока исходная строка S не станет пустой.

Теорема 1. Выполняя алгоритмы распознавания, агенты распознают любой граф G с точностью до изоморфизма.

Поскольку для каждой позиции исходной строки производится анализ всех предыдущих позиций, то временная сложность может быть оценена как $O(n^2)$.

Емкостная сложность E(n) определяется [3] сложностью списков строки направлений S, множества граничных вершин $V' \subset V$, и строки формул F, сложность которых определяется величинами O(n), O(n), O(n).

Выводы

В работе предложен алгоритм восстановления структуры графа в виде формулы, отражающей состав исходного графа как комбинацию базовых структур. Далее предполагается рассмотреть графы с дырами.

Список использованной литературы

1. Dudek G. Computational Principles of Mobile Robotics / G. Dudek, V. Jenkin. – Cambridge: Cambridge University Press, 2000. – 280 p.
2. Кудрявцев В.Б. О поведении автоматов в лабиринтах / В.Б. Кудрявцев, Ш. Ушчумлич, Г. Килибарда // Дискретная математика. – 1992. – Т.4, вып.3. – С. 3-28.
3. Шатохина Н.К. Распознавание графа мозаичной структуры коллективом агентов. / Н.К. Шатохина, П.А.Шатохин // Наукові праці Донецького національного технічного університету. Серія: Проблеми моделювання та автоматизації проектування (МАП-2011). - 2011. – Випуск 9(179). – С. 111-121.
4. Шатохина Н.К. Об анализе мозаичных лабиринтов с использованием коллектива агентов / Н. К. Шатохина // Математичне та програмне забезпечення інтелектуальних систем (MPZIS

- 2012): X міжнар. наук.-практ. конф., (21-23 листопада 2012 р.): тези допов. - Дніпропетр.: ДНУ ім. Олеся Гончара, 2012. – С. 320 -321.

Надійшла до редакції:
30.04.2013

Рецензент:
д-р техн. наук, проф. Скобцов Ю.О.

Н.К. Шатохіна, П.О. Шатохін
ДВНЗ «Донецький національний технічний університет»

Розпізнавання графа мозаїчної структури, що складається зі зв'язаних компонентів та мостів, колективом агентів. *Розглянута задача опису структури графа спеціального виду без дірок, що складається із декількох зв'язних компонентів і мостів, на основі інформації, отриманої при обході його по кордону. Описано алгоритм розв'язання задачі, наведені оцінки його часової та ємкісної складності.*

Ключові слова: *мозаїка, лабіринт, комутативна група, агент, алгоритм.*

N.K. Shatokhina, P.A. Shatokhin
Donetsk National Technical University

Recognition of a Mosaic-structured Graph, Consisting of the Linked Components and Bridges, by the Collective of Agents. *The paper considers a problem of recovering a graph structure of a special type on the basis of the information obtained by traversing its borders. We study the graphs of mosaic structure consisting of several strongly connected components and bridges. The mosaic structure of the graph is constructed from equilateral triangles, without multiple edges and holes. Restoration of the structure of the graph is produced by two agents. The first agent - researcher (AR) has to perform the movements on the graph, to fix its directions of the motion and to give the extracted information to the second agent - experimenter (AE). The second agent (AE) recreates the structure of the graph in a symbolic form, based on the information received. Each movement of the AR can be characterized by a certain direction of the movement (free vector). The special symbolic notations are introduced for different directions of the motion, so the AR movement on the border of the graph uniquely generates a string of characters. Investigation of the properties of the system of movements along the mosaics of such kind gives a possibility to define a commutative group on a set of free vectors. On this basis, an algorithm for determining such substrings, which describe the strongly connected components of the graph, is built. These substrings are determined from the characters string, which is transmitted from the AR to the AE. Basic structures for the mosaics are defined. Rules are given for the identification of such structures. These rules are used by the AE for the recreation of the original graph. The algorithm of AR may consist of two stages. At the first stage, if the AR has been placed on some internal node of the graph, then it first has to reach the boundary of the graph. The first step may be omitted if the node of its initial placement will be a boundary node. At the second stage the AR passes the graph along its border and delivers the string of the directions to the AE. Algorithm of the AE is composed of a main and a subordinate algorithms. The basic algorithm, by scanning the string of directions, selects such fragments, which correspond to strongly connected subgraphs and bridges. It handles them with the help of the subordinate algorithm. It is shown that the algorithms allow recognizing the graph structure up to isomorphism. Capacitive and time complexities of the algorithms are given.*

Keywords: *puzzle, maze, commutative group, agent, algorithm.*