

УДК 004.8; 519.688; 681.3

## ИЕРАРХИЧЕСКИЙ ПОДХОД В ЗАДАЧАХ ПЛАНИРОВАНИЯ ТРАЕКТОРИИ НА ПЛОСКОСТИ

К.С. Яковлев (*yakovlev@isa.ru*)

Институт Системного Анализа РАН, Москва

В работе рассматривается задача планирования траектории на плоскости, как задача отыскания пути на графе специальной структуры. Описывается иерархический алгоритм решения указанной задачи. Приводятся результаты экспериментов, подтверждающие эффективность применения предлагаемого алгоритма на практике.

### Введение

Одной из ключевых проблем, возникающих при разработке систем управления (СУ) беспилотными транспортными средствами (БТС), является проблема создания эффективного модуля планирования траектории. Если речь идет о малых, полностью автономных БТС, то вопросы снижения вычислительной нагрузки выходят на первый план. Причиной этому – совокупность факторов. Во-первых, малые автономные БТС, зачастую, не могут быть оснащены мощными вычислителями в силу своих размеров, малой грузоподъемности и особенностей конструкции. Во-вторых, модуль планирования траектории – лишь один из немногих программных компонент СУ БТС, поэтому ресурсы бортового вычислителя БТС не могут использоваться им в исключительной манере. Таким образом, встает вопрос о разработке и практической реализации вычислительно-эффективных методов и алгоритмов планирования траектории.

Задача планирования траектории на плоскости может быть рассмотрена как задача поиска пути на взвешенном графе, вершинам которого соответствуют координаты точек двумерного пространства, а весам ребер – соответствующие расстояния. Так как в качестве исходных данных для построения графа используются цифровые карты и модели местности (ЦКМ), которые хранятся в памяти бортового вычислителя БТС, целесообразно (с точки зрения экономии вычислительных ресурсов) использовать такие графовые модели, которые могут быть извлечены напрямую из цифровой карты местности. Поскольку в двумерном случае (при навигации БТС на/в плоскости) ЦКМ фактически представляют из

себя матрицу, элементам которой соответствуют проходимые либо непроходимые области пространства, в качестве подобной модели может выступать метрический топологический граф (МТ-граф) [Яковлев, 2009b].

Для осуществления поиска пути на МТ-графе могут быть использованы любые известные алгоритмы поиска пути на графе, наиболее распространенными из которых являются эвристические алгоритмы семейства  $A^*$  [Hart *et al*, 1968; Korf, 1996]. Однако, даже используя при поиске в качестве эвристик метрик МТ-графа (то есть, оперируя «лучшими» из возможных эвристик),  $A^*$ -поиск производит слишком большой объем вычислений, требующих сохранения промежуточных результатов. Именно факт неэкономного использования оперативной памяти алгоритмами семейства  $A^*$  препятствует их применению в ряде задач, когда размер (число вершин) вовлеченных графов (в том числе и МТ-графов) достаточно велик [Edelkamp *et al*, 2008].

Традиционный путь к снижению вычислительной нагрузки в рамках  $A^*$ -поиска заключается в применении взвешенных эвристик, т.е. эвристик вида  $w \cdot h^*$ ,  $w \geq 1$ , при поиске. Алгоритмы, использующие взвешенные эвристики принято обозначать как  $WA^*$ . В качестве примеров подобных алгоритмов можно привести  $ARA^*$  [Gordon *et al*, 2004],  $\text{Alpha-}A^*$  [Reese, 1999]. Однако даже применение взвешенных эвристик зачастую не приводит к существенному сокращению пространства поиска из-за проблемы локального минимума [Яковлев, 2009a; Likhachev *et al*, 2008].

Для решения проблемы локального минимума предлагается использовать принципиально иной алгоритм построения пути на МТ-графе, основанный на иерархическом подходе. Основная идея, лежащая в основе предлагаемого алгоритма, заключается в разбиении исходной задачи планирования на совокупность таких подзадач, каждая из которых может быть легко разрешена известными методами. В работе описывается подобный алгоритм –  $HGA^*$  (Hierarchical  $A^*$  for МТ-Graphs). Даются оценки емкостной сложности алгоритма при определенных ограничениях. Приводятся результаты практических экспериментов, доказывающих перспективность применения  $HGA^*$  для ряда задач.

## 1. Постановка задачи

Будем рассматривать задачу планирования траектории на плоскости, как задачу поиска пути на графе особой структуры – метрическом топологическом графе (МТ-графе) [Яковлев, 2009b].

МТ-граф – это неупорядоченная пара:

$$\text{МТ-Gr} = \langle A, d \rangle, \quad (1.1)$$

где:

$A$  – множество клеток, представляющее собой матрицу  $A_{m \times n} = \{a_{ij}\}$ :  
 $a_{ij} = 0 \vee 1, \forall i, j: 0 \leq i < m, 0 \leq j < n, m, n \in \mathbb{N} \setminus \{0\}$ .

$d$  – метрика на множестве  $A$ .

Клетку МТ-графа будем называть проходимой, если  $a_{ij}=0$ ; непроходимой, если  $a_{ij}=1$ . Множество клеток, смежных с  $a_{ij}$  будем обозначать как  $adj(a_{ij})$ . Множество смежных непроходимых клеток МТ-графа будем называть препятствием:  $Obs=\{a_{i0j0}, a_{i1j1}, a_{i2j2}, \dots, a_{isjs} | a_{ijk}=1, a_{ijk} \in adj(a_{i(k-1)j(k-1)}) \forall k=0,1,2, \dots, s, s \in \mathbf{N}\}$ .

В качестве метрики  $d$  будем использовать функцию:

$$H(a_{ij}, a_{lk}) = \begin{cases} c_d \Delta_j + c_{hv} (\Delta_i - \Delta_j), & \text{если } \Delta_i(a_{ij}, a_{lk}) \geq \Delta_j(a_{ij}, a_{lk}) \\ c_d \Delta_i + c_{hv} (\Delta_j - \Delta_i), & \text{если } \Delta_i(a_{ij}, a_{lk}) < \Delta_j(a_{ij}, a_{lk}) \end{cases} \quad (1.2)$$

где  $\Delta_i = \Delta_i(a_{ij}, a_{lk}) = |i-l|$ ;  $\Delta_j = \Delta_j(a_{ij}, a_{lk}) = |j-k|$ ;  $c_d = \sqrt{2} \cdot c_{hv}$  (вес перехода между диагонально смежными клетками);  $c_{hv} \in R^+ = \{r | r \in \mathcal{R}, r > 0\}$  (вес перехода между горизонтально и вертикально смежными клетками).

Задача планирования траектории формально представляется в виде тройки:

$$PTask = \langle \text{MT-Gr}, a_{startI, startJ}, a_{goalI, goalJ} \rangle, \quad (1.3)$$

и формулируется следующим образом. Пусть на МТ-графе зафиксированы начальная  $a_{startI, startJ}$  и целевая  $a_{goalI, goalJ}$  клетки, такие что  $a_{startI, startJ} \neq a_{goalI, goalJ}$ ,  $a_{startI, startJ} = 0$ ,  $a_{goalI, goalJ} = 0$ . Необходимо найти путь  $\pi(a_{startI, startJ}, a_{goalI, goalJ})$ , то есть последовательность клеток МТ-графа  $\pi = \{a_{i0j0}, a_{i1j1}, a_{i2j2}, \dots, a_{isjs}\}$  такую, что  $a_{i0j0} = a_{startI, startJ}$ ,  $a_{isjs} = a_{goalI, goalJ}$ ,  $\forall v: 1 \leq v < s \ a_{iv, jv} \in adj(a_{i(v-1)j(v-1)})$ .

Весом пути  $\pi$  будет называть величину  $c(\pi)$ , равную сумме весов переходов по всем смежным клеткам, входящим в путь. Величину  $r = \max\{|startI - goalI|, |startJ - goalJ|\}$  будем называть глубиной решения задачи (1.3).

Нуль-траекторией между двумя различными клетками  $a_{ij}$  и  $a_{lk}$  будем называть последовательность смежных клеток МТ-графа  $tr(a_{ij}, a_{lk}) = \{a_{i0j0}, a_{i1j1}, a_{i2j2}, \dots, a_{irjs}\}$ , представляющую собой отрезок дискретной прямой проходящей через клетки  $a_{ij} = a_{i0j0}$  и  $a_{lk} = a_{irjs}$  (см. рис. 2б). Формальное определение нуль-траектории (как совокупности клеток МТ-графа, удовлетворяющих определенному набору ограничений) приведено в [Яковлев, 2009b]. Весом нуль-траектории будем называть величину  $c(tr(a_{ij}, a_{lk}))$ , определяемую аналогично весу пути  $\pi(a_{ij}, a_{lk})$ . Примем без доказательства следующее утверждение:  $c(tr(a_{ij}, a_{lk})) = H(a_{ij}, a_{lk})$ .

Важно отметить, что нуль-траектория может быть построена автоматически с помощью одного из множества распространенных и эффективных алгоритмах машинной графики (например, с помощью известного алгоритма Брезенхема [Bresenham, 1965]). Таким образом, задачу автоматического построения нуль-траектории будем считать тривиальной в рассматриваемом домене.

Секцией будем называть упорядоченную пару клеток МТ-графа  $\langle a_{ij}, a_{lk} \rangle$ . Секция  $\langle a_{ij}, a_{lk} \rangle$  проходима, тогда и только тогда когда нуль-траектория  $tr(a_{ij}, a_{lk})$  содержит лишь проходимые клетки МТ-графа. Весом секции назовем величину  $c(\langle a_{ij}, a_{lk} \rangle) = c(tr(a_{ij}, a_{lk})) = H(a_{ij}, a_{lk})$ .

Пусть  $a_{ij}, a_{lk} \in A_{m \times n}$ , при этом  $a_{ij} \neq a_{lk}$ ,  $a_{ij} \neq 1$ ,  $a_{lk} \neq 1$ . Частичным путем из  $a_{ij}$  в  $a_{lk}$  будем называть последовательность клеток МТ-графа  $PP = \{a_{i_0 j_0}, a_{i_1 j_1}, a_{i_2 j_2}, \dots, a_{i_s j_s}\}$ , такую что  $a_{i_0 j_0} = a_{ij}$ ,  $a_{i_s j_s} = a_{lk}$ . Если при этом каждая из секций  $\langle a_{i_0 j_0}, a_{i_1 j_1} \rangle, \langle a_{i_1 j_1}, a_{i_2 j_2} \rangle, \dots, \langle a_{i_{s-1} j_{s-1}}, a_{i_s j_s} \rangle$  является проходимой, то частичный путь  $PP$  будем называть допустимым, в противном случае – недопустимым. Клетки, входящие в частичный путь будем называть опорными. Вес частичного пути  $C(PP)$  определим как сумму весов всех секций, образуемых смежным опорным клеткам.

Используя приведенную выше терминологию, задача (1.3) может рассматриваться, как задача отыскания на МТ-графе допустимого частичного пути из  $a_{start, startJ}$  в  $a_{goal, goalJ}$ . Перед тем как описать иерархический алгоритм решения задачи планирования траектории в такой формулировке дадим еще несколько определений.

Будем говорить, что клетка  $a_{ij}$  расположена левее (правее) клетки  $a_{lk}$ , если  $j < k$  ( $j > k$ ). Будем говорить, что клетка  $a_{ij}$  расположена ниже (выше) клетки  $a_{lk}$ , если  $i < l$  ( $i > l$ ). Будем говорить, что препятствие  $Obs$  лежит между клетками  $a_{ij}$  и  $a_{lk}$ , если  $tr(a_{ij}, a_{lk}) \cap Obs \neq \emptyset$ .

## 2. Иерархический подход в задачах планирования траектории

### 2.1. Простейшие иерархические алгоритмы планирования траектории.

Задача планирования траектории может рассматриваться как задача построения множества клеток (называемых опорными), некоторое подмножество которого образует допустимый частичный путь. Таким образом, задача планирования может быть разбита на три подзадачи:

1. Выделение опорных клеток.
2. Упорядочивание опорных клеток (формирование частичных планов).
3. Выбор опорных клеток, для формирования решения задачи (выбор частичного плана из множества кандидатов).

Опишем простейший недетерминированный алгоритм построения пути на МТ-графе, осуществляющий решение указанных выше подзадач в иерархической манере.

Шаг 1. Добавить в искомый частичный путь начальную и целевую клетки:  $PP = \{a_{start, startJ}, a_{goal, goalJ}\}$

Шаг 2. Если текущий частичный путь  $PP$  является допустимым, то вернуть  $PP$ .

Шаг 3. Выбрать две опорные клетки,  $a_{ij}$  и  $a_{lk}$ , входящие в  $PP^1$ .

Шаг 4. Построить нуль-траекторию  $tr(a_{ij}, a_{lk})$  с помощью алгоритма Брезенхема.

Шаг 5. Если нуль-траектория  $tr(a_{ij}, a_{lk})$  проходима, то перейти к шагу 2.

Шаг 6. Выбрать клетку  $C \in A$

Шаг 7. Добавить клетку  $C$  в частичный путь  $PP$  (а именно – заменить последовательность  $a_{ij}$  и  $a_{lk}$  в  $PP$  на  $a_{ij}, C, a_{lk}$ ).

Шаг 8. Перейти к шагу 2.

Из-за недетерминированного выбора клетки на 6-м шаге приведенный алгоритм не может быть реализован на имеющихся вычислительных машинах<sup>2</sup>. Простейшим способом избавиться от недетерминированного выбора является случайный выбор определенного числа опорных клеток. Соответствующий алгоритм представлен ниже.

Шаг 1. Сформировать множество частичных путей кандидатов  $PPC$ , первоначально содержащее лишь частичный путь  $PP = \{a_{startI, startJ}, a_{goalI, goalJ}\}$ .

Шаг 2. Выбрать из  $PPC$  кратчайший частичный путь  $PP^3$ .

Шаг 3. Если текущий частичный путь  $PP$  является допустимым, то вернуть  $PP$ .

Шаг 4. Выбрать две опорные клетки,  $a_{ij}$  и  $a_{lk}$ , входящие в  $PP$ .

Шаг 5. Построить нуль-траекторию  $tr(a_{ij}, a_{lk})$ .

Шаг 6. Если нуль-траектория  $tr(a_{ij}, a_{lk})$  проходима, то перейти к шагу 2.

Шаг 7. Случайным образом выбрать  $N$  клеток  $C_1, C_2, \dots, C_n \in A$

Шаг 8. Разбить  $PP$  на  $N$  вариантов, а именно – для каждого  $PP \in PPC$ , содержащего последовательность  $a_{ij}, a_{lk}$  выполнить следующие шаги:

а) скопировать  $PP$   $N$  раз в  $PPC$ ;

б) для каждого из  $N$  скопированных частичных путей заменить последовательность  $a_{ij}, a_{lk}$  на  $a_{ij}, C_1, a_{lk}; a_{ij}, C_2, a_{lk}; \dots; a_{ij}, C_n, a_{lk}$  соответственно.

Шаг 9. Перейти к шагу 2.

Приведенный алгоритм может быть реализован на любом современном вычислителе, однако, в общем случае он не является корректным. Можно лишь утверждать, что при  $N \rightarrow \infty$  вероятность нахождения решения (если таковое существует) стремится к 1.

Для удобства дальнейшего изложения будем называть разбиением секции  $\langle a_{ij}, a_{lk} \rangle$  процесс, выполняемый на 7-8 шагах алгоритма. Клетки  $a_{ij},$

---

<sup>1</sup> Здесь и далее будем предполагать, что выбираются клетки, образующие первую непроходимую секцию в  $PP$ .

<sup>2</sup> Машинах Тьюринга

<sup>3</sup> Кратчайший частичный путь – путь с минимальным значением  $C(PP)$

$a_{ik}$  выбираемые на шаге 2 будем называть локальными начальной и целевой клетками соответственно.

## 2.2. HGA\*

Алгоритм HGA\* - это алгоритм поиска пути на МТ-графе, реализующий иерархический подход к решению задачи планирования, главное отличие которого от алгоритма, описанного в предыдущем разделе – детерминированный выбор опорных клеток на этапе разбиения секции. Перед тем как описать алгоритм более подробно, предположим, что начальная клетка расположена не правее целевой<sup>4</sup>. Данное предположение необходимо лишь для упрощения дальнейшего изложения и не может рассматриваться как ограничение, т.к. указанного расположения клеток всегда можно добиться с помощью поворота МТ-графа (формальное определение этой операции приведено в [Яковлев 2009], а неформальное – очевидно). Сформулируем теперь очевидное утверждение, лежащее в основе алгоритма HGA\*.

Если некоторое препятствие лежит между начальной и целевой клетками МТ-графа, то допустимый частичный путь, являющийся решением задачи планирования, необходимо содержит клетки, расположенные выше (либо ниже) данного препятствия.

Перед тем как описать, каким образом алгоритм HGA\* использует это утверждение для выбора опорных клеток при разбиении секции, предположим, что МТ-граф содержит лишь препятствия прямоугольной формы, не соприкасающиеся друг с другом. Это достаточно сильное утверждение, однако оно необходимо для того, чтобы представить первую упрощенную версию HGA\*.

Итак, пусть некоторая локальная начальная клетка  $s$  расположена не правее локальной целевой  $g$ , а нуль-траектория  $tr(s, g)$  пересекает препятствие в некоторой клетке  $X$  (см. рис. ТАКОЙ-ТО Б). Рассмотрим следующую процедуру, выделяющую 4 опорные клетки для последующего разбиения секции.

```
GetBaseCellsForExtension(cell s, cell g, cell X)
int i_up, i_down, j_right, j_left=X.j-1; cell tmp=X;

while (tmp==1)
    tmp.i--;
i_up=tmp.i; tmp.i++;

while (tmp==1)
    tmp.j++;
j.right=tmp.j; tmp=X;
```

---

<sup>4</sup> Это же предположение будем считать выполненным для любых локальных начальных и целевых клеток, о которых далее пойдет речь.

```

while (tmp==1)
    tmp.i++;
i_down=tmp.i;

if (i_up>=0){
    A.i=B.i=i_up; A.j=j_left; B.j=j_right
}else
    A=B=null;

if (i_down<m){
    C.i=D.i=i_down; C.j=j_left; D.j=j_right
} else
    C=D=null;

return {A, B, C, D}

```

Процедура *GetBaseCellsForExtensions* в точности возвращает клетки *A*, *B*, *C*, *D* «обрамляющие» препятствие по углам (см. рис. такой-то). С помощью этих клеток не составляет труда осуществить разбиение секции  $\langle s, g \rangle$  по следующему алгоритму: для каждого  $PP \in PPC$ , содержащего последовательность *s*, *g* выполнить:

Шаг 1. Скопировать *PP* в *PP1* и *PP2*

Шаг 2. В *PP1* заменить последовательность *s*, *g* на *s*, *A*, *B*, *g*

Шаг 3. В *PP2* заменить последовательность *s*, *g* на *s*, *D*, *C*, *g*

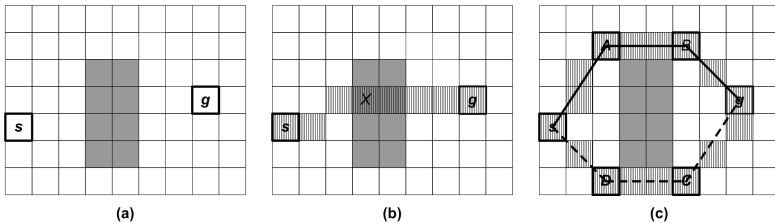


Рис.2. Разбиение секции алгоритмом HGA\*. а) начальная и целевая клетки; б) нуль-траектория; в) сформированные частичные пути

Заметим, что если процедура *GetBaseCellsForExtensions* возвращает только клетки *A* и *B* (либо *C* и *D*), то нет необходимости в копировании частичного пути *PP*, требуется лишь замена *s*, *g* на *s*, *A*, *B*, *g* (*s*, *D*, *C*, *g*). Если же  $A=B=C=D=null$ , то не существует пути из *s* в *g*, что в свою очередь означает, что не существует решения рассматриваемой задачи планирования. Таким образом, детерминированный, корректный алгоритм построения частичного пути – HGA\* – может быть представлен в следующем виде.

Шаг 1. Сформировать множество частичных путей кандидатов *PPC*, первоначально содержащее лишь частичный путь  $PP = \{a_{start, start}, a_{goal, goal}\}$ .

Шаг 2. Выбрать из *PPC* частичный путь *PP*

Шаг 3. Если текущий частичный путь  $PP$  является допустимым, то вернуть  $PP$ .

Шаг 4. Выбрать две опорные клетки,  $a_{ij}$  и  $a_{lk}$ , входящие в  $PP$ .

Шаг 5. Построить нуль-траекторию  $tr(a_{ij}, a_{lk})$ .

Шаг 6. Если нуль-траектория  $tr(a_{ij}, a_{lk})$  проходима, то перейти к шагу 2.

Шаг 7. Выполнить процедуру GET

Шаг 8. Если все клетки, возвращенные Get, равны  $null$ , то осуществить выход (пути из  $a_{startI, startJ}$  в  $a_{goalI, goalJ}$  в не существует)

Шаг 9. Осуществить разбиение секции  $\langle a_{ij}, a_{lk} \rangle$  с помощью найденных клеток

Шаг 10. Перейти к шагу 2.

Отметим, что практическая реализация HGA\* возможна и без явного хранения частичных планов. Достаточно для каждой опорной клетки хранить указатель на предыдущую опорную клетку. Более того можно показать, что при определенных ограничениях требуется хранение не более 2 указателей для каждой опорной клетки. Таким образом, емкостная сложность HGA\* может быть оценена следующим образом. Пусть объем памяти, необходимый для того, чтобы «хранить» опорную клетку оценивается в  $O(1)$ . Если ни одно препятствие не лежит между  $a_{startI, startJ}$  и  $a_{goalI, goalJ}$ , то требуется хранение 2 опорных клеток. Если между  $a_{startI, startJ}$  и  $a_{goalI, goalJ}$  лежит 1 препятствие, то требуется хранение не более 6 клеток; если – 2 препятствия, то – не более 10; и так далее. По принципу математической индукции получаем, что всего требуется хранить на более  $2+4 \cdot NOBS$  опорных клеток, где  $NOBS$  – число препятствий, лежащих между  $a_{startI, startJ}$  и  $a_{goalI, goalJ}$ . Поскольку  $NOBS$  очевидно не может превышать  $r$  (где  $r$  – глубина решения задачи (1.3)), то емкостная сложность HGA\* может быть оценена как  $O(r)$ . В то время, как известно, что емкостная сложность эвристических алгоритмов семейства  $A^*$  –  $O(r^2)$  [Korf, 1996].

## 2.2. HGA\* в сложных доменах

Ослабим предположение о том, что все препятствия на МТ-графе имеют прямоугольную форму. Опишем неформально процедуру выделения опорных клеток в этом случае.

Как и раньше будем полагать, что клетка  $s$  лежит на правее клетки  $g$ , а нуль-траектория  $tr(s, g)$  пересекает препятствие  $Obs$  в клетке  $X$ . Будем двигаться вдоль границы препятствия по часовой стрелке до тех пор, пока не совершим движение в горизонтальном направлении слева направо. Таким образом, мы выделим опорную клетку  $A$  (см. рис. 2-а), первую из серии опорных клеток, расположенных выше препятствия, которые войдут в искомый частичный путь. Для отыскания первой клетки, расположенной ниже, выполним аналогичные действия – будем двигаться



вдоль границы препятствия, но против часовой стрелки, пока не совершим движение в горизонтальном направлении слева направо.

После выделения опорных клеток  $A$  и  $B$ , разбиение секции  $\langle s, g \rangle$  может быть осуществлено стандартным образом на  $\{s, A, g\}$  и  $\{s, B, g\}$ . Однако при таком подходе, возникает следующая проблема. Если, после нескольких итераций алгоритма,  $A$  и  $g$  становятся локальными начальной и целевой клетками, то при разбиении секции  $\langle A, g \rangle$  клетка  $B$  будет обнаружена снова. И после разбиения  $\langle A, g \rangle$  множество  $PPC$  будет содержать взаимно противоречивые элементы:  $PP_1 = \{\dots, s, A, C, g, \dots\}$ ,  $PP_2 = \{\dots, s, A, B, g, \dots\}$ ,  $PP_3 = \{\dots, s, B, g, \dots\}$  (см. рис. 2).

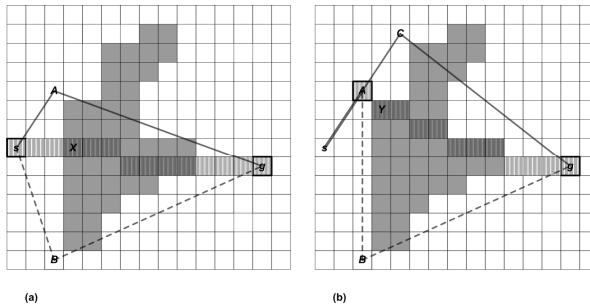


Рис. 2. Взаимное противоречие частичных путей. а) Разбиение секции  $\langle s, g \rangle$ ; б) Последующее разбиение секции  $\langle A, g \rangle$

Для поддержания целостности множества  $PPC$  необходимо следовать следующему простому правилу: если некоторая секция  $\langle s, g \rangle$  разбивается с помощью некоторой клетки  $A$ , лежащей выше препятствия  $Obs$ , секция  $\langle A, g \rangle$  не должна разбиваться с помощью клетки, лежащей ниже  $Obs$ . Соблюдение данного правила может быть реализовано с помощью приписывания клетке  $A$  не только указателя на клетку  $s$ , но и на клетку  $X$  (клетку, в которой  $tr(s, g)$  пересекает  $Obs$ ). Впоследствии, если клетка  $X$  будет обнаружена при «скольжении» вдоль контура препятствия по (против) часовой стрелки, во время разбиения  $\langle A, g \rangle$ , необходимо остановить процедуру поиска опорной клетки и продолжить скольжение лишь в противоположную сторону.

### 3. Экспериментальный анализ

Был произведен экспериментальный анализ алгоритмов HGA\*, A\*, WA\*-3 и WA\*-5<sup>5</sup>. Для этого были проведены три серии экспериментов по отысканию пути на МТ-графах. Отслеживались следующие индикаторы:

$W_{alg}$  – вес пути, найденного алгоритмом  $alg$ ;

<sup>5</sup> WA\*- $x$  – алгоритм A\*, использующий взвешенную эвристику с весом  $x$

$Q_{alg}$  – число клеток МТ-графа, рассмотренных при нахождении пути, алгоритмом  $alg$ ;

$E_{alg}=(Q_{alg}/W_{alg})/(Q_{A^*}/W_{A^*})$  – интегральный показатель емкостной эффективности  $alg$ .

### 3.1. 1-я серия экспериментов.

Было сгенерировано 150 МТ-графов с различной степенью заполнения препятствиями,  $\lambda$  (0,3; 0,5; 0,8)<sup>6</sup>. Начальная и целевая клетки выбирались таким образом, чтобы глубина решения составляла 50, 100, 250, 500 и 1000. Препятствия являлись прямоугольниками шириной 1 клетку и длиной (в среднем) – 10 клеток. Усредненные (по всем прогонам) результаты экспериментов на МТ-графах с  $\lambda=0,8$  представлены ниже.

**Таблица 1.** Результаты первой серии экспериментов на МТ-графах с  $\lambda=0,8$

Алгоритм	Показатели	Глубина решения				
		50	100	250	500	1000
A*	$W_{A^*}$	628	1235	3024	5938	12377
A*	$Q_{A^*}$	656	1733	10281	32410	158328
A*	$E_{A^*}$	100%	100%	100%	100%	100%
WA*-3	$W_{WA^*-3}$	701	1416	3552	6911	14182
WA*-3	$Q_{WA^*-3}$	206	384	951	1791	3779
WA*-3	$E_{WA^*-3}$	28%	19%	8%	5%	2%
WA*-5	$W_{WA^*-5}$	720	1442	3662	7008	14405
WA*-5	$Q_{WA^*-5}$	203	381	933	1763	3745
WA*-5	$E_{A^*}$	27%	19%	7%	5%	2%
HGA*	$W_{A^*}$	682	1334	3381	6574	13911
HGA*	$Q_{A^*}$	115	265	608	1311	2646
HGA*	$E_{A^*}$	16%	14%	5%	4%	1%

Из полученных данных можно сделать два основных вывода. Во-первых, разница между показателями WA\*-3 и WA\*-5 минимальна. Это значит, в свою очередь что... Во-вторых HGA\* превосходит наиболее «оптимизированный» алгоритм планирования траектории, основанный на A\*-поиске, в среднем в 1,5 раза. При этом на МТ-графах с меньшей плотностью препятствий превосходство HGA\* еще более ярко выражено. Так для МТ-графов с параметром  $\lambda=0,3$  получены следующие результаты:  $E_{WA^*-5}/E_{HGA^*}=4,58$ ;  $W_{WA^*-5}/W_{HGA^*}=4,90$ .

### 3.2. 2-я серия экспериментов.

Было сгенерировано 50 МТ-графов с фиксированной степенью заполнения препятствиями:  $\lambda=0,5$ . Глубина решения была зафиксирована на отметке 100. Ширина препятствий была фиксирована и составляла 1 (клетку). Средняя длина препятствий  $l$  составляла 2, 5, 10, 15, 25 клеток

<sup>6</sup> Использовалась следующая формула  $\lambda=(l \cdot 2 + 4) \cdot N$ , где  $l$  – площадь препятствия,  $N$  – число препятствий.

(по 10 МТ-графов для каждого значения  $l$  было сгенерировано). Усредненные результаты  $A^*$ ,  $WA^*-5$  и  $HGA^*$  представлены в таблице 2.

**Таблица 2.** Результаты второй серии экспериментов

Алгоритм	Показатели	Средняя длина препятствия				
		2	5	10	15	25
$A^*$	$W_{A^*}$	1143	1166	1158	1225	1335
$A^*$	$Q_{A^*}$	499	901	1478	1991	2662
$A^*$	$E_{A^*}$	100%	100%	100%	100%	100%
$WA^*-5$	$W_{WA^*-5}$	1181	1245	1290	1413	1500
$WA^*-5$	$Q_{WA^*-5}$	350	357	380	423	524
$WA^*-5$	$E_{A^*}$	68%	37%	23%	18%	18%
$HGA^*$	$W_{A^*}$	1146	1189	1235	1285	1516
$HGA^*$	$Q_{A^*}$	95	120	187	154	114
$HGA^*$	$E_{A^*}$	19%	13%	12%	7%	4%

Опираясь на полученные результаты можно утверждать, что алгоритм  $HGA^*$  превосходит алгоритмы, основанные на  $A^*$ -поиске, не только на МТ-графах с небольшим числом «больших» препятствий, но и на МТ-графах с большим числом «небольших» препятствий.

### 3.3. 3-я серия экспериментов.

Рассматривалась задача автоматического построения траектории маловысотного полета вертолета в городских условиях. МТ-графы представляли собой модели двух фрагментов карты Москвы. Препятствиям соответствовали высотные здания, которые должен был облетать в горизонтальной плоскости воображаемый вертолет. На каждом из двух МТ-графов 10 раз случайным образом выбирались начальная и целевая клетки, так чтобы глубина решения равнялась 110.

**Таблица 3.** Результаты второй серии экспериментов

Алгоритм	Показатели	МТ-граф 1	МТ-граф 2
$A^*$	$W_{A^*}$	1326	1424
$A^*$	$Q_{A^*}$	1344	1695
$A^*$	$E_{A^*}$	100%	100%
$WA^*-5$	$W_{WA^*-5}$	1368	1472
$WA^*-5$	$Q_{WA^*-5}$	372	394
$WA^*-5$	$E_{A^*}$	20%	27%
$HGA^*$	$W_{A^*}$	1404	1532
$HGA^*$	$Q_{A^*}$	229	355
$HGA^*$	$E_{A^*}$	12%	24%

Результаты эксперимента подтверждают, что алгоритм  $HGA^*$  эффективно решает задачу отыскания пути не только на случайно сгенерированных МТ-графах, но и на МТ-графах являющихся моделями городского ландшафта.

## Заключение

В статье была рассмотрена задача планирования траектории на плоскости, как задача поиска пути на графе специальной структуры. Был предложен новый алгоритм решения указанной задачи, основанный на принципах иерархической декомпозиции. Экспериментальный анализ подтвердил перспективность такого подхода и его большую эффективность по сравнению с наиболее распространенными на данный момент подходами (а именно – подходами, основанными на эвристическом поиске) при решении ряда практических задач.

Дальнейшая работа предполагается в направлении изучения теоретических свойств HGA\* и создания модификаций алгоритма для планирования траектории в динамических средах.

**Благодарности.** Работа выполнена при финансовой поддержке РФФИ (проекты №№ 09-07-00043, 09-07-00006), Министерства образования и науки РФ (ФЦП «Научные и научно-педагогические кадры инновационной России»), Президиума РАН, ОНИТ РАН.

## Список литературы

[Яковлев, 2009а] Яковлев К.С. Методы решения проблемы локального минимума при планировании траектории. // Труды IX международной научной конференции им.Таран Т.А. «Интеллектуальный анализ информации ИАИ-2009», Киев: ПРОСВІТА, 2009.

[Яковлев, 2009б] Яковлев К.С. Графы специальной структуры в задачах планирования траектории. // Труды III международной конференции «Системный анализ и информационные технологии САИТ-2009». М: ИСА РАН, 2009.

[Bresenham, 1965] J. E. Bresenham. Algorithm for computer control of a digital plotter. // IBM SystemsJournal, Vol. 4, No.1, 1965.

[Edelkamp *et al*, 2008] S. Edelkamp, E. Hansen, S. Jabbar, R. Zhou. External Memory Graph Search. // 18th International Conference on Automated Planning and Scheduling (ICAPS08), Sydney, Australia, 2008.

[Gordon *et al*, 2004] G. Gordon, M. Likhachev, S. Thrun. ARA\*: Anytime A\* with provable bounds on sub-optimality. // In Advances in Neural Information Processing Systems, MIT Press, 2003.

[Hart *et al*, 1968] P. Hart, N. Nilsson, B. Raphael. A formal basis for the heuristics determination of minimum costs path. // IEEE Transactions on Systems Science and Cybernetics, 2, 1968.

[Korf, 1996] R.E.Korf. Artificial Intelligence Search Algorithms. // In Algorithms and Theory of Computation Handbook, CRC Press, 1996.

[Likhachev *et al*, 2008] M. Likhachev, A. Stentz. R\* Search. // Proceedings of the National Conference on Artificial Intelligence (AAAI), 2008.

[Reese, 1999] B. Reese. Alpha\*: An  $\epsilon$ -admissible Heuristic Search Algorithm. // Institute for Production Technology, University of Southern Denmark 1999 - <http://home1.stofanet.dk/breese/astaralpha-submitted.pdf.gz>