

# ОБ ОДНОЙ ПРОБЛЕМЕ, ВОЗНИКАЮЩЕЙ ПРИ ПЛАНИРОВАНИИ ТРАЕКТОРИИ НА ПЛОСКОСТИ.

Яковлев К.С.<sup>а</sup> (*yakovlev@isa.ru*), Сарафанов В.Ю., Храмоин И.В.<sup>б</sup>

а) Институт системного анализа РАН; б) Российский университет Дружбы народов

В работе рассматривается важный частный случай задачи планирования траектории, а именно, задача планирования траектории на плоскости, когда в окружающем пространстве присутствует лишь одно препятствие. Описывается декомпозиционный подход к решению поставленной задачи и алгоритм его реализующий. Приводятся результаты вычислительного эксперимента.

## 1. ВВЕДЕНИЕ

Проектирование и разработка беспилотных транспортных средств (БТС) различного типа и назначения является одной из наиболее активно развивающихся областей в современной науке и технике. Основной целью, преследуемой исследователями в этой области, является повышение степени автономности разрабатываемых транспортных средств [Осипов и др., 2009]. Очевидно, что для достижения указанной цели необходимо создание эффективных алгоритмов планирования траектории, т.е. перемещения транспортного средства в пространстве (окружающей среде).

Задача планирования траектории на плоскости традиционно рассматривается как задача поиска пути на взвешенном графе, вершинам которого соответствуют координаты точек двумерного пространства, а весам ребер – соответствующие расстояния. Таким образом, для планирования траектории необходимо, во-первых, построить граф, моделирующий окружающую среду БТС, во-вторых, осуществить поиск пути на построенном графе.

Так как обычно в качестве исходных данных для построения графа используются цифровые карты и модели местности (ЦКМ), хранящиеся в памяти бортового вычислителя БТС, целесообразно 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 поиска пути на МТ-графе могут использоваться любые известные алгоритмы поиска пути на взвешенном графе. Самым известным и распространенным алгоритмом поиска кратчайшего пути является алгоритм Дейкстры [Dijkstra, 1959]. На практике же, с целью экономии вычислительных ресурсов, обычно используются эвристические модификация алгоритма Дейкстры – алгоритмы семейства  $A^* - A^*$ ,  $WA^*$ , BeamSearch и другие [Hart *et al*, 1968; Korf, 1996]. Указанные алгоритмы основаны на принципе итерационного обхода клеток графа до выполнения некоторого критерия. В процессе обхода,

для каждой клетки рассчитывается и сохраняется в оперативной памяти ряд числовых значений, которые впоследствии используются для формирования решения. Именно числом рассмотренных клеток и определяется вычислительная нагрузка этих алгоритмов. И, как показывает теоретический и практический анализ, даже эвристические алгоритмы поиска, использующие «лучшую» эвристику (а именно – метрику МТ-графа), не в состоянии избежать рассмотрения чрезмерного количества клеток при решении большинства практических задач [Яковлев, 2009а; Edelkamp *et al*, 2008].

Существует и другой подход к решению задачи поиска пути на МТ-графе, заключающийся в разбиении исходной задачи на ряд легко-разрешимых подзадач, именуемых локальными. Итоговое решение получается за счет объединения решений локальных подзадач. Такой подход будем называть декомпозиционным. Ключевой компонентой декомпозиционного подхода является стратегия разбиения исходной задачи на локальные. В настоящее время большинство методов, реализующих декомпозиционный подход, опирается на случайный выбор при разбиении на подзадачи [Morgan *et al*, 2004]. В то же время перспективным представляется подход, реализующий детерминированный выбор. Описанию алгоритма, реализующего такой подход, посвящена данная работа.

В работе будет рассмотрен важный частный случай задачи планирования траектории на плоскости – а именно случай, когда в окружающей среде присутствует лишь одно препятствие произвольной формы. Будут описана общая схема алгоритма декомпозиции, а также ее частные реализации – реализация, опирающаяся на случайный выбор и реализация, реализующая выбор по детерминированному критерию (алгоритм HGA\*). Будут приведены результаты вычислительных экспериментов, подтверждающие эффективность алгоритма HGA\* по сравнению с имеющимися аналогами.

## 2. МТ-ГРАФЫ В ЗАДАЧЕ ПЛАНИРОВАНИЯ ТРАЕКТОРИИ НА ПЛОСКОСТИ

В задачах планирования траектории на плоскости входными данными обычно являются дискретные цифровые карты местности, которые могут быть представлены в виде графов специального вида – метрических топологических графов [Яковлев, 2009b] – см. Рис. 1.

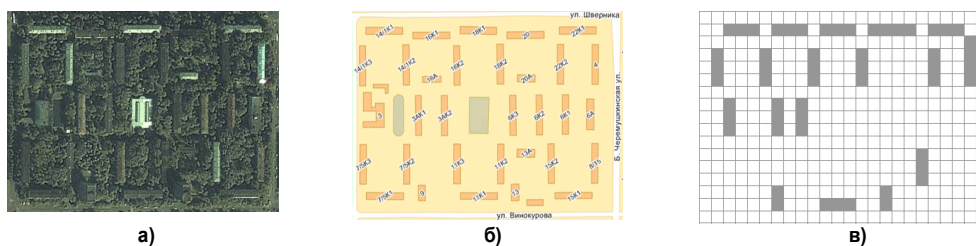


Рис. 1. Представление карты местности в виде МТ-графа: а) спутниковый снимок местности; б) карта местности; в) МТ-граф.

Под МТ-графом будем понимать неупорядоченную тройку:

$$\text{MT-Gr} = \langle A, d, c \rangle,$$

где

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

$d$  – метрика на множестве  $A^+ = \{a_{ij} | a_{ij} \in A, a_{ij} = 0\}$ ;

$c: E \rightarrow (0, +\infty)$  – коммутативная функция, определяющая веса переходов между клетками МТ-графа (здесь,  $E \subset A \times A$ ).

Для упрощения записи будем обозначать МТ-граф как  $A_{m \times n}$ . Величины  $i, j$  будем называть координатами либо индексами клеток МТ-графа. Клетку МТ-графа будем называть проходимой, если  $a_{ij} = 0$ , и – 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:  $Obs = \{a_{i_0 j_0}, a_{i_1 j_1}, a_{i_2 j_2}, \dots, a_{i_s j_s} | a_{i_k j_k} = 1, a_{i_k j_k} \in adj(a_{i_{k-1} j_{k-1}}) \forall k = 0, 1, 2, \dots, s, s \in \mathbb{N}\}$ . Графически МТ-граф можно представить в виде таблицы (сетки), содержащей  $m$  строк и  $n$  столбцов – см. Рис. 2. Ячейки таблицы соответствуют клеткам  $a_{ij}$ , при этом проходимым клеткам соответствуют светлоокрашенные ячейки, а непроходимым – темноокрашенные.

Пусть  $ADJ \subset A \times A$  – множество всех пар смежных клеток,  $c_{hv} \in (0, +\infty)$ ,  $c_d = k \cdot c_{hv}$ ,  $1 < k < 2$ .

Определим функцию  $c: E \rightarrow (0, +\infty)$  следующим образом:  $c: ADJ \rightarrow \{c_{hv}, c_d, +\infty\}$ .

Вес перехода из клетки  $a_{ij}$  в клетку  $a_{lk}$  будем обозначать  $c(a_{ij}, a_{lk})$ . При этом:

$c(a_{ij}, a_{lk}) = c_{hv}$ , если  $a_{ij} = 0 \wedge a_{lk} = 0$  и клетки  $a_{ij}, a_{lk}$  являются горизонтально или вертикально смежными;

$c(a_{ij}, a_{lk}) = c_d$ , если  $a_{ij} = 0 \wedge a_{lk} = 0$  и клетки  $a_{ij}, a_{lk}$  являются диагонально смежными;

$c(a_{ij}, a_{lk}) = +\infty$ , если  $a_{ij} = 1 \vee a_{lk} = 1$ .

Будем считать, что в качестве метрики на МТ-графе используется функция

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

обычно именуемая в литературе по искусственному интеллекту диагональной метрикой/эвристикой. Здесь  $\Delta_i = \Delta_i(a_{ij}, a_{lk}) = |i-l|$ ;  $\Delta_j = \Delta_j(a_{ij}, a_{lk}) = |j-k|$ .

Задача планирования траектории формально представляется в виде тройки:

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

и формулируется следующим образом. Пусть на МТ-графе зафиксированы различные проходимые клетки: начальная  $a_{startI\ startJ}$  и целевая  $a_{goalI\ goalJ}$ . Необходимо найти путь  $\pi(a_{startI\ startJ}, a_{goalI\ goalJ})$ , то есть последовательность клеток МТ-графа  $\pi = \{a_{i_0 j_0}, a_{i_1 j_1}, a_{i_2 j_2}, \dots, a_{i_s j_s}\}$  такую, что  $a_{i_0 j_0} = a_{startI\ startJ}$ ,  $a_{i_s j_s} = a_{goalI\ goalJ}$ ,  $\forall v: 1 \leq v < s \ a_{i_v j_v} \in adj(a_{i_{v-1} j_{v-1}})$  – см. Рис. 2.

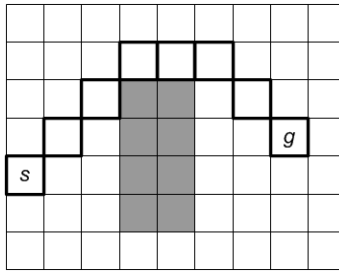


Рис. 2. МТ-граф и путь на нем (начальная и целевая вершины обозначены как  $s$  и  $g$ ).

Весом пути  $\pi$  будем называть величину  $c(\pi)$ , равную сумме весов переходов по всем смежным клеткам, входящим в путь.

### 3. ДЕКОМПОЗИЦИОННЫЙ ПОДХОД К ПОИСКУ ПУТИ НА МТ-ГРАФЕ

#### 3.1. Основные определения

Секцией будем называть упорядоченную пару клеток МТ-графа  $\langle a_{ij}, a_{lk} \rangle$ . Будем считать секцию  $\langle a_{ij}, a_{lk} \rangle$  проходимой, если известен хотя бы один путь из  $a_{ij}$  в  $a_{lk}$ . В этом случае под весом секции будем подразумевать вес соответствующего пути  $c(\langle a_{ij}, a_{lk} \rangle) = c(\pi(a_{ij}, a_{lk}))$ . Если секция непроходима, то есть не известен ни один путь из  $a_{ij}$  в  $a_{lk}$ , то под весом такой секции будем понимать эвристическую оценку веса пути  $c(\langle a_{ij}, a_{lk} \rangle) = d(a_{ij}, a_{lk})$ .

Пусть  $a_{ij}, a_{lk} \in A_{m \times n}$ , при этом  $a_{ij} \neq a_{lk}$ ,  $a_{ij} = 0$ ,  $a_{lk} = 0$ . 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 планирования траектории  $PTask$  может рассматриваться, как задача отыскания на МТ-графе допустимого частичного пути из  $a_{startI} startJ$  в  $a_{goalI} goalJ$ . Эта задача, в свою очередь, заключается в выделении на МТ-графе множества клеток (называемых опорными), некоторое подмножество которого и образует допустимый частичный путь. То есть, фактически, речь идет о декомпозиции исходной задачи на ряд легко-разрешимых подзадач.

#### 3.2. Алгоритм $R^*$

Опишем простейший алгоритм построения частичного пути на МТ-графе, реализующий декомпозиционный принцип.

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  $PP$  согласно некоторому критерию.

Шаг 3. Если текущий частичный путь  $PP$  является допустимым, то вернуть  $PP$  в качестве решения задачи планирования.

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

Шаг 5. Попытаться построить путь  $\pi(a_{ij}, a_{lk})$ .

Шаг 6. Если путь  $\pi(a_{ij}, a_{lk})$  удастся построить «легко», 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.

Одной из реализаций описанного выше подхода, опирающегося на случайный выбор опорных клеток, является алгоритм  $R^*$ . В качестве критерия выбора на шаге 2 используется следующее правило – выбирается частичный путь с наименьшим весом. При этом для расчета весов непроходимых секций (т.е. секций проходимость, которых неизвестна), входящих в частичный путь, используется взвешенная диагональная эвристика. Эта же эвристика используется при построении «локальных» путей, т.е. путей между выбранными на шаге 4 опорными клетками, с помощью алгоритма  $WA^*$  на шаге 5. Если таковой путь не удастся построить меньше, чем за определенное число шагов (задается с помощью настроечного параметра), то соответствующая секция помечается, как «нежелательная». Эта метка влияет на выбор секции на шаге 4 – алгоритм выбирает первую непроходимую секцию частичного пути, не являющуюся «нежелательной». При выборе опорных клеток на шаге 7, алгоритм  $R^*$  использует настроечный параметр  $\Delta$  – расстояние (вычисляется с помощью метрики МТ-графа), на котором расположены опорные клетки от клетки  $a_{ij}$  (см. Шаг 8). То есть алгоритм  $R^*$  случайным образом выбирает  $N$  клеток, удаленных на  $\Delta$  от клетки, являющейся начальной в текущей секции. Более подробное описание алгоритма приведено в [Likhachev *et al*, 2008].

### 3.3. Алгоритм $HGA^*$

$HGA^*$  – алгоритм поиска пути на МТ-графе, реализующий декомпозиционный подход и опирающийся на детерминированную стратегию выбора опорных клеток.

Здесь и далее будем рассматривать частный случай, когда МТ-граф содержит лишь одно препятствие  $Obs$ . Также будем полагать, что начальная ( $a_{startI\ startJ}$ ) и целевая ( $a_{goalI\ goalJ}$ ) клетки расположены таким образом, что  $|startJ - goalJ| \geq |startI - goalI|$  и  $goalJ > startJ^1$ . Будем говорить, что начальная клетка лежит левее целевой, а целевая – соответственно правее начальной. Последнее условие не является ограничением, т.к. можно показать, что решение задачи поиска пути при любом расположении начальной и целевой клеток, может быть сведено к решению задачи с указанным расположением клеток [Яковлев, 2011].

Нуль-траекторией между двумя различными клетками  $a_{ij}$  и  $a_{lk}$  будем называть последовательность смежных клеток МТ-графа  $tr(a_{ij}, a_{lk}) = \{a_{i_0j_0}, a_{i_1j_1}, a_{i_2j_2}, \dots, a_{i_rj_r}\}$ , представляющую собой отрезок дискретной прямой проходящей через клетки  $a_{ij} = a_{i_0j_0}$  и  $a_{lk} = a_{i_rj_r}$  – см. Рис. 3а). Формальное определение нуль-траектории (как совокупности клеток, удовлетворяющих определенному набору ограничений) приведено в [Яковлев, 2011].

Отметим, что нуль-траектория может быть построена автоматически с помощью одного из распространенных и эффективных алгоритмах машинной графики (например, с помощью алгоритма Брезенхема [Bresenham, 1965]). Таким образом, задачу автоматического построения нуль-траектории будем считать легко-разрешимой.

Будем говорить, что препятствие  $Obs$  лежит между клетками  $a_{ij}$  и  $a_{lk}$ , если  $tr(a_{ij}, a_{lk}) \cap Obs \neq \emptyset$ . Здесь и далее будем считать, что  $Obs$  лежит между начальной и целевой клетками (т.к. в противном случае задача поиска пути разрешается в один шаг с помощью построения нуль-траектории по алгоритму Брезенхема). Очевидно, что в этом случае искомый путь содержит клетки, лежащие выше (либо ниже) препятствия  $Obs$ , таким образом, имеет смысл идентифицировать эти клетки и использовать их в качестве опорных клеток для декомпозиции задачи.

Для выделения опорных клеток, алгоритм НГА\* использует процедуру обхода контура препятствия, т.е. последовательного перебора проходимых клеток МТ-графа, смежных между собой и в то же время смежных с клетками, образующими препятствие.

Будем считать, что нуль-траектория пересекает препятствие в клетке  $x$ . Перебор клеток МТ-графа, начиная с клетки, смежной с  $x$  и расположенной левее  $x$ , в порядке, показанном на рис. 3б, будем называть обходом контура препятствия по часовой стрелке. Аналогично определяется процедура обхода против часов стрелки.

Упорядоченные тройки клеток, изображенные на рис. 3в, будем называть поворотами (по часовой стрелке) слева направо снизу вверх (ЛП-НВ), слева направо сверху вниз (ЛП-

<sup>1</sup> На рис.2 и последующих рисунках изображены МТ-графы именно с таким расположением начальной и целевой клеток.

ВН), справа налево сверху вниз (ПЛ-ВН), справа налево снизу вверх (ПЛ-НВ). «Среднюю» клетку поворота будем называть опорной клеткой поворота – см. Рис. 3б.

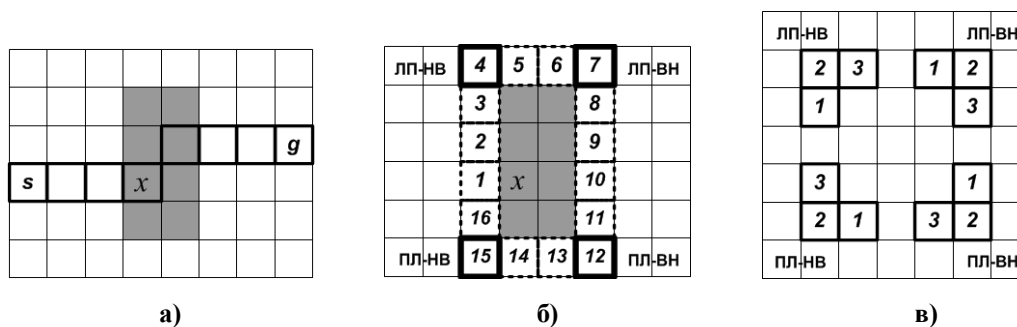


Рис. 3. а) Нуль-траектория; б) Обход контура препятствия по часовой стрелке; в) Повороты по часовой стрелке.

Рассмотрим подробнее процедуру обхода препятствия по часовой стрелке. Результатом работы процедуры является упорядоченная последовательность (список) проходимых клеток МТ-графа, которые в дальнейшем будут использованы как опорные.

Пусть текущая клетка – клетка, смежная с  $x$  и расположенная левее  $x$ . Будем двигаться (перебирать проходимые клетки) вверх по контуру препятствия до того момента пока, либо не будет зафиксирован поворот ЛП-НВ, либо не будет зафиксирована непроходимая клетка (препятствие «изгибается» и необходимо «идти в обратную сторону»). Пока будем считать, что зафиксирован поворот ЛП-НВ, в этом случае опорная клетка поворота помещается в список опорных клеток возвращаемых процедурой и производится проверка – если  $i$ -я координата опорной клетки больше либо равна  $i$ -й координате целевой клетки, то процедура прекращает свою работу, в противном случае – обход контура продолжается до того момента, как будет зафиксирован поворот ЛП-ВН (то есть мы «упадем» за препятствие/часть препятствия), при этом повороты ЛП-НВ (мы «взбираемся» на препятствие), если такие будут зафиксированы, также помещаются в список – см. Рис. 4.

Рассмотрим отдельно случай, когда «препятствие изгибается в обратную сторону». При этом будем считать, что препятствие изгибается «справа налево» при изначальном обходе контура «снизу вверх» (остальные случаи рассматриваются аналогично) – см. Рис. 4. Итак, при переборе клеток смежных с препятствием (движении «вдоль препятствия справа налево, в сторону начальной клетки»), мы сравниваем  $j$ -ые координаты текущей и начальной клеток. Как только (и если) они окажутся равными, в стек требуемых поворотов (содержащий изначально лишь поворот ЛП-НВ) добавляем поворот ПЛ-НВ. Продолжаем обход, и как только поворот ПЛ-НВ будет зафиксирован, добавляем клетку поворота в список опорных клеток, продолжаем обход (т.к. поворот ЛП-НВ еще не зафиксирован) – см. Рис. 4.

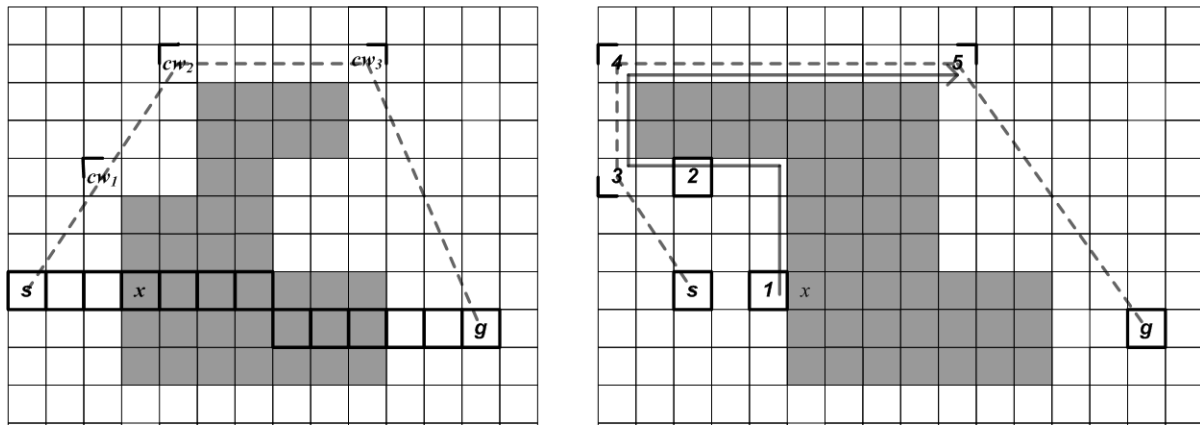


Рис. 4. а) опорные клетки,  $\{cw_1, cw_2, cw_3\}$ , выделенные процедурой обхода препятствия по часовой стрелке и частичный путь (обозначен штрихом), построенный по этим клеткам б) Обход контура «изгибающегося» препятствия (штрихом обозначен построенный путь): 1 - клетка начала обхода, стек поворотов содержит ЛП-НВ; когда клетка 2 становится текущей, в стек поворотов добавляется ПЛ-НВ; 3 - поворот ПЛ-НВ фиксируется и удаляется из стека, стек содержит ЛП-НВ; 4 - поворот ЛП-НВ фиксируется и удаляется из стека, стек пуст, но клетка 4 лежит выше целевой, следовательно, в стек добавляется поворот ЛП-ВН; 5 – поворот ЛП-ВН фиксируется, стек пуст, процедура заканчивает работу.

Отметим, что для успешного функционирования на практике необходимо внести ряд модификаций в описанную схему работы процедуры, предназначенный для обработки частных случаев (спиралевидные препятствия, случай, когда нуль-траектория пересекает препятствие «сверху-вниз», а не «слева-направо» и т.д.), однако подробное описание указанных модификаций представляется нецелесообразным в данной работе. Заметим лишь, что описанная процедура может быть реализована в виде корректного алгоритма.

Опишем теперь алгоритм построения допустимого частичного пути – HGA\*, использующий процедуры обхода препятствия по/против часовой стрелки, для детерминированного выбора опорных клеток.

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

Шаг 2. Выбрать из  $PPC$  частичный путь  $PP$  согласно некоторому критерию.

Шаг 3. Если текущий частичный путь  $PP$  является допустимым, то вернуть  $PP$  в качестве решения задачи планирования.

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. С помощью процедур обхода препятствия по и против часовой стрелки выделить опорные клетки  $\{cw_1, cw_2, \dots, cw_{|cw|}\}$  и  $\{ccw_1, ccw_2, \dots, ccw_{|ccw|}\}$ .

Шаг 8. Осуществить разбиение секции  $\langle a_{ij}, a_{lk} \rangle$  с помощью найденных клеток.

Шаг 9. Перейти к шагу 2.



Заметим, что если обе процедуры обхода по и против часовой стрелке возвращают пустые списки, это означает, что пути из начальной клетки в целевую не существует.

Рассмотрим также важный частный случай, когда препятствие не удается «обойти за один раз» – см. Рис. 5.

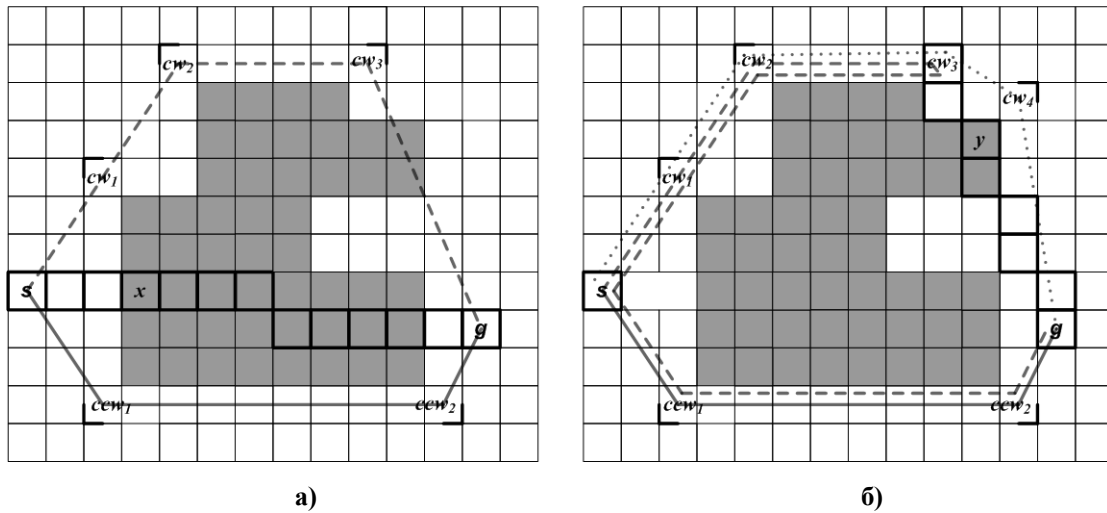


Рис. 5. Обход препятствия за две итерации

Итак, пусть на первой итерации алгоритма HGA\* для разбиения секции  $\langle s, g \rangle$  с помощью процедур обхода контура препятствия по/против часовой стрелки определяются две последовательности клеток –  $\{cw_1, cw_2, cw_3\}$  и  $\{ccw_1, ccw_2\}$ . Секция  $\langle s, g \rangle$  разбивается с помощью найденных клеток – см. Рис. 5а. Далее в качестве текущей секции выбирается секция  $\langle cw_3, g \rangle$ , при этом  $tr(cw_3, g)$  – непроходима и также пересекает препятствие  $Obs$  (в некоторой клетке  $y$ ). Тогда, при осуществлении разбиения секции  $\langle cw_3, g \rangle$ , на этапе выделения опорных клеток, при обходе контура  $Obs$  против часовой стрелки клетки  $cw_3, cw_2, cw_1$  будут обнаружены повторно. И после разбиения  $\langle cw_3, g \rangle$  множество  $PPC$  будет содержать элементы:  $PP_1 = \{s, cw_1, cw_2, cw_3, cw_4, g\}$ ,  $PP_2 = \{s, cw_1, cw_2, cw_3, cw_3, cw_2, cw_1, ccw_1, ccw_2, g\}$ ,  $PP_3 = \{s, ccw_1, ccw_2, g\}$  – см. Рис. 5б. Очевидно, что частичный путь  $PP_2$  должен быть исключен из рассмотрения, т.к. множество  $PPC$  уже содержит частичный путь из  $s$  в  $ccw_2$  (а именно –  $PP_3$ ). Для исключения «дублирующих» путей необходимо следовать следующему правилу: для каждой опорной клетки, найденной при обходе препятствия, запомнить направление движения, при котором она была найдена (по/против часовой стрелки). Если во время обхода препятствия в определенном направлении, координаты текущей клетки  $a$  совпадают с координатами начальной клетки текущей секции  $a'$ , и направление движения не совпадает с сохраненным направлением для  $a'$ , то необходимо прекратить обход препятствия в этом направлении.

#### 4. ЭКСПЕРИМЕНТАЛЬНЫЙ АНАЛИЗ

Рассматривалась задача планирования траектории маловысотного полета вертолета в городских условиях. МТ-графы представляли собой модели карты города Москвы. Препятствиям соответствовали высотные здания, которые должен облетать в горизонтальной плоскости вертолет – см. Рис. 6. Всего было построено 10 МТ-графов размером 500 на 500 клеток, соответствующих 10 разным участкам городской карты.

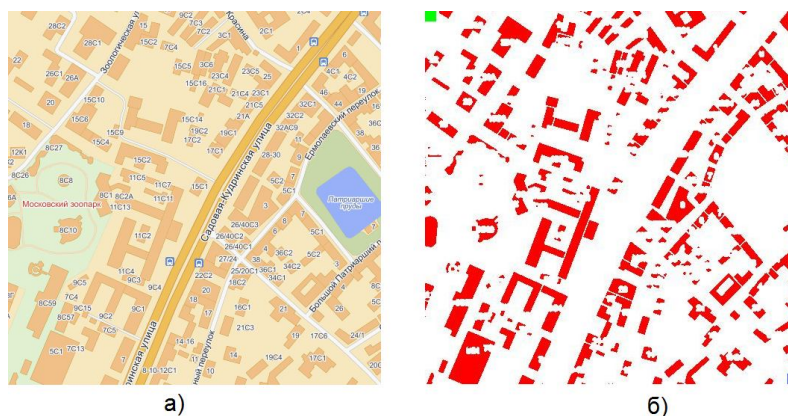


Рис. 6. Карта города и соответствующий ей МТ-граф.

На каждом МТ-графе случайным образом выбирались начальная и целевая клетки так, чтобы величина  $D = \max\{|startI - goalI|, |startJ - goalJ|\}$  – глубина решения – составляла 500 клеток, производился поиск пути. Затем выбиралось новое расположение начальной и целевой клетки, эксперимент повторялся. Всего было произведено 100 экспериментов (по 10 на каждом из 10 МТ-графов).

Между собой сравнивались алгоритмы  $A^*$ ,  $WA^*$ ,  $R^*$ ,  $HGA^*$ . Поскольку алгоритмы  $WA^*$ ,  $R^*$  являются параметризованными, перед экспериментами была произведена настройка алгоритмов и в экспериментах участвовали алгоритмы с «лучшими» значениями параметров, т.е. с такими значениями, которые показали лучший результат.

В каждом эксперименте сравнивались между собой следующие параметры работы алгоритмов: вес найденного пути; число рассмотренных (и сохраненных в оперативной памяти) клеток – емкостная эффективность; затраченное время – временная эффективность. Усредненные значения, полученные в результате экспериментов, приведены в таблице 1.

Таблица 1. Результаты вычислительного эксперимента.

	Фактические параметры			Нормированные параметры		
	1	2	3	4	5	6
	Клетки	Вес пути	Время	Клетки	Вес пути	Время
$A^*$	19049,7	5743,2	17,52562	86,9	1,00	1914,28
$WA^*$	2051,5	6197,6	0,05288	9,4	1,08	5,78
$HGA^*$	219,2	5845,8	0,009155	1,0	1,02	1,00
$R^*$	652,3	6150	0,025537	3,0	1,07	2,79

При нормировке выходных параметров (столбцы 4, 5, 6) в качестве нормирующего выбирался параметр наилучшего алгоритма (в случае веса пути –  $A^*$ , в остальных случаях –  $HGA^*$ ).

Как видно из таблицы, алгоритм  $HGA^*$  существенно превосходит (в 3-9 раз) все рассмотренные аналоги с точки зрения использования вычислительных ресурсов. Заметим при этом, что следом за  $HGA^*$  идет алгоритм  $R^*$ , так же реализующий декомпозиционный подход. Таким образом, можно сделать вывод о целесообразности применения декомпозиционных алгоритмов, в частности –  $HGA^*$ , для эффективного решения практических задач планирования траектории беспилотных транспортных средств.

## 5. 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 принципе декомпозиции –  $HGA^*$ . Экспериментальный анализ подтвердил перспективность такого подхода и его большую эффективность по сравнению с наиболее распространенными аналогами.

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

## СПИСОК ЛИТЕРАТУРЫ

[Осипов и др., 2009] Осипов Г.С., Тихомиров И.А., Хачумов В.М., Яковлев К.С. Интеллектуальное управление транспортными средствами: стандарты, проекты, реализации // Авиакосмическое приборостроение, № 6, 2009. М: НАУЧТЕХЛИТИЗДАТ, 2009. С. 34-43.

[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.

[Яковлев, 2011] Яковлев К.С. Исследование методов и разработка алгоритмов автоматического планирования траектории на плоскости. // Диссертация на соискание ученой степени кандидата наук, ИСА РАН, 2011.

- [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.
- [Dijkstra, 1959]** E. W. Dijkstra. A note on two problems in connexion with graphs. // Numerische Mathematik, 1:269-271, 1959.
- [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.
- [Morgan *et al*, 2004]** S.B. Morgan, M.S. Branicky, Sampling-based planning for discrete spaces}. // In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004