

МУРАВЬИНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ ПУТИ

Способность избегать препятствий является важным требованием при проектирования любых мобильных роботов.

Определение свободного от столкновений пути между начальной и целевой позициями среди препятствий, загромождающих рабочее пространство, занимает центральное место при планировании пути автономного робота.

Планирование пути исследования охватывает широкую область исследований в робототехнике, поскольку оно улучшает системы навигации в статических и динамических средах.

С хорошей системой планирования пути и навигацией мобильные роботы могут перемещаться самостоятельно без вмешательства человека, чтобы достичь целевого назначения.

Проблемы, возникающие при навигации робота, представлены на рис.

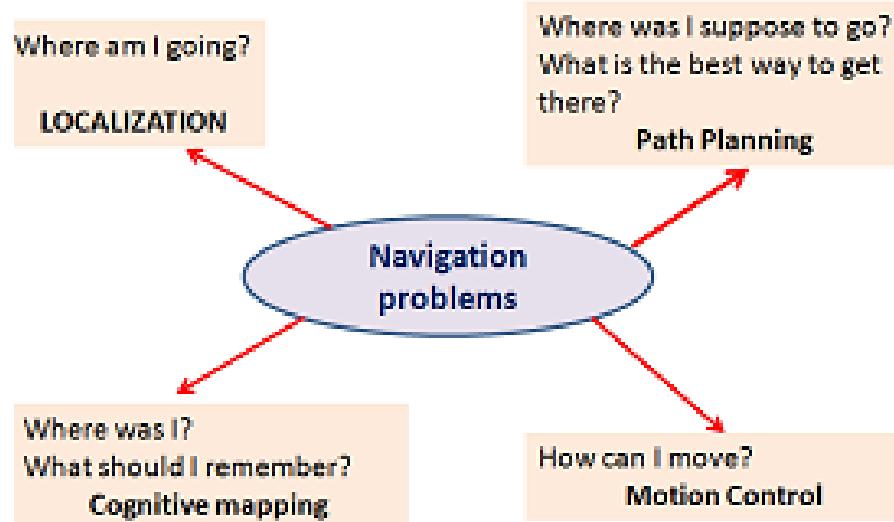


Fig. 1 Robot navigation problems.

Среди них:

- 1) Локализация;
- 2) **Планирование пути;**
- 3) Управление движением;
- 4) карта окружения.

Здесь планирование пути является самой сложной и важной задачей.

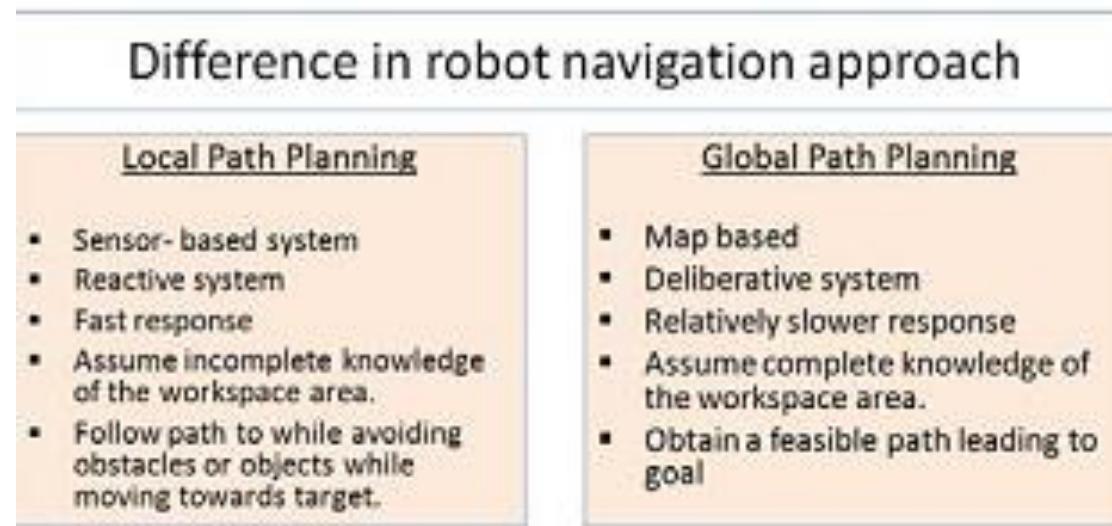
Основными компонентами при этом являются:

- 1) представление окружающей среды робота;
- 2) алгоритм построения пути.

Если окружающая среда полностью известна, то применяется глобальное планирование движения робота, которое можно выполнить заранее (в режиме offline).

С другой стороны, если окружающая среда не полностью определена, например, могут появляться препятствия в процессе движения робота, то используется локальное планирование пути робота (навигация), где в процессе движения работает сенсор по обнаружению препятствий. (в режиме online).

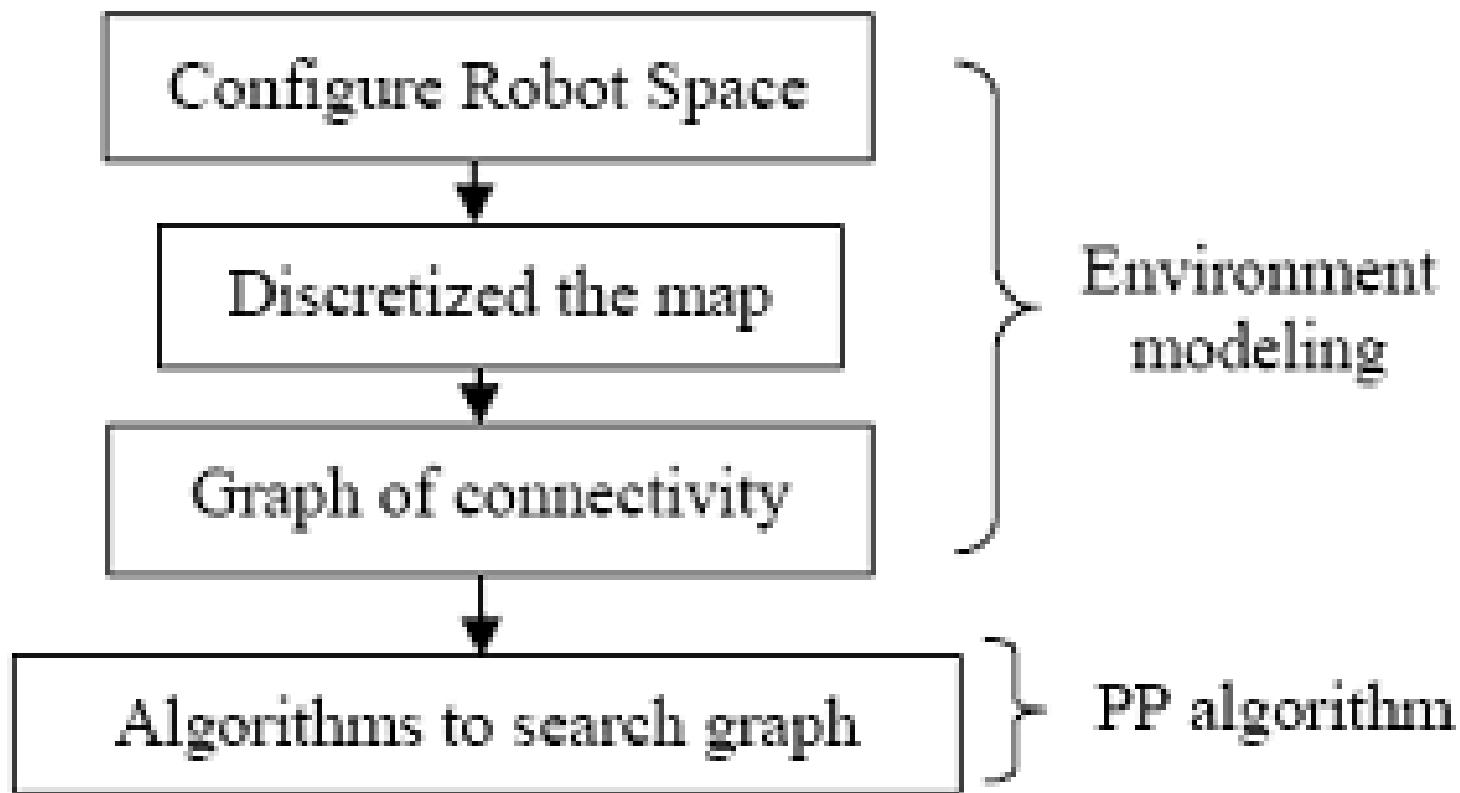
На следующем рисунке показаны отличия между этими задачами в статической и динамической среде.

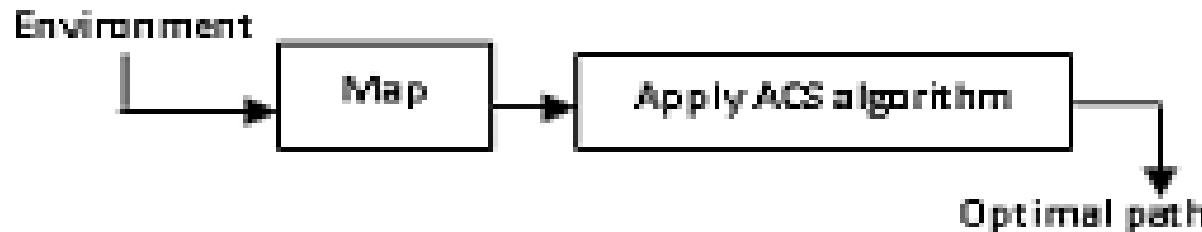


В глобальном планировании используется полное описание статической среды с препятствиями.

Целью является найти свободный от столкновений с препятствиями путь от начальной позиции до конечной.

На следующем рисунке представлены основные этапы этого процесса.



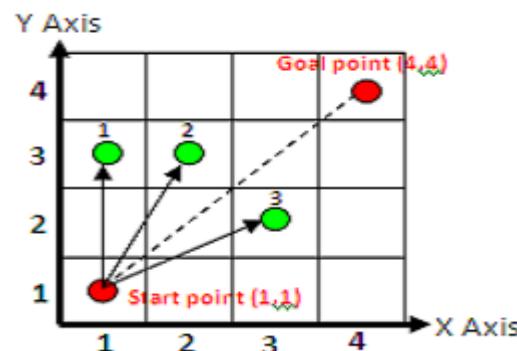


Далее рассмотрим решение задачи планирования пути на основе метода ACS.

Ниже представлена блок-схема этого алгоритма.

Алгоритм включает следующие шаги.

1. Начиная со стартовой вершины с (x,y) , координатами $(1,1)$ муравей движется от одной вершины к другой (реализуемой) вершине, как показано на следующем рисунке. Здесь возможны 3 варианта выбора, отмеченные зеленым: вершина(узел) 1 с координатами $(1,3)$, вершина 2 с координатами $(2,3)$ и вершина 3 с $(3,2)$.



2. Муравей должен выбрать следующую вершину с вероятностью, которая определяется правилом перехода.
3. Каждый раз, когда муравей переходит от одной вершины к следующей, локально происходит «испарение» искусственного феромона - его концентрация уменьшается путем умножения на число, меньшее 1.
4. Когда муравьи достигают цель, вычисляется фитнесс значение для каждого построенного пути-длина пути.
5. Эти значения используются при коррекции концентрации феромона для отрезков, вошедших в построенные пути. Это повторяется для каждого построенного пути. При этом отрезки, вошедшие в лучшие пути, получают большее приращение феромона.
6. Шаги 1-5 повторяются пока процесс не сойдется.
7. Процесс останавливается, когда все муравьи выбирают кратчайший путь.

Псевдокод алгоритма имеет вид.

C.Pseudo Code of ACS Algorithm for RPP

```
If iteration (tmax )=1,2,3,4,5,6,.....+n  
Else if ant(m)=1,2,3,4,5,6,.....+n  
Else if nodes(n)=1,2,3,4,.....+n  
Compute the probability of the m th ants next nodes  
Move to the next nodes by computed probability  
Store history of past location of nodes in an array  
If current location of nodes is equal to destination  
Break the nodes (n) loop  
End  
End  
Evaluate fitness and store path distance of m th ants  
Compute pheromone amount generated by m th ants  
End  
Update pheromone amount of the entire map  
End
```

Чаще всего, применяются следующие модификации муравьиных алгоритмов.

TABLE I. OVERVIEW OF THE THREE SUCCESSFUL ANT COLONY ALGORITHM

Algorithm	Tour Construct	Evaporation	Pheromone	Updation
AS (Ant System) Dorigo et al. 1991,	random proportional rule	all arcs lowered with constant factor	$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$ $+ \sum_{k=1}^m \Delta\tau_{ij}$	deposit on all arcs visited by all ants
ACS (Ant Colony System) Dorigo and Gambardella (1997).	Pseudo random proportional rule	only arcs of the best-so-far tour are lowered	$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0$ where $\varphi \in (0,1)$ is the pheromone decay coefficient	deposit only on arcs of the best so-far tour
MMAS (MAX-MIN Ant System) Stützle and Hoos (2000)	random proportional rule	all arcs lowered with constant factor	$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{bestij}$ where $\Delta\tau_{bestij} = 1/L_{best}$	deposit only either by the iteration best-ant, or the best-sofar ant; interval [\underline{min} ; \underline{max}]

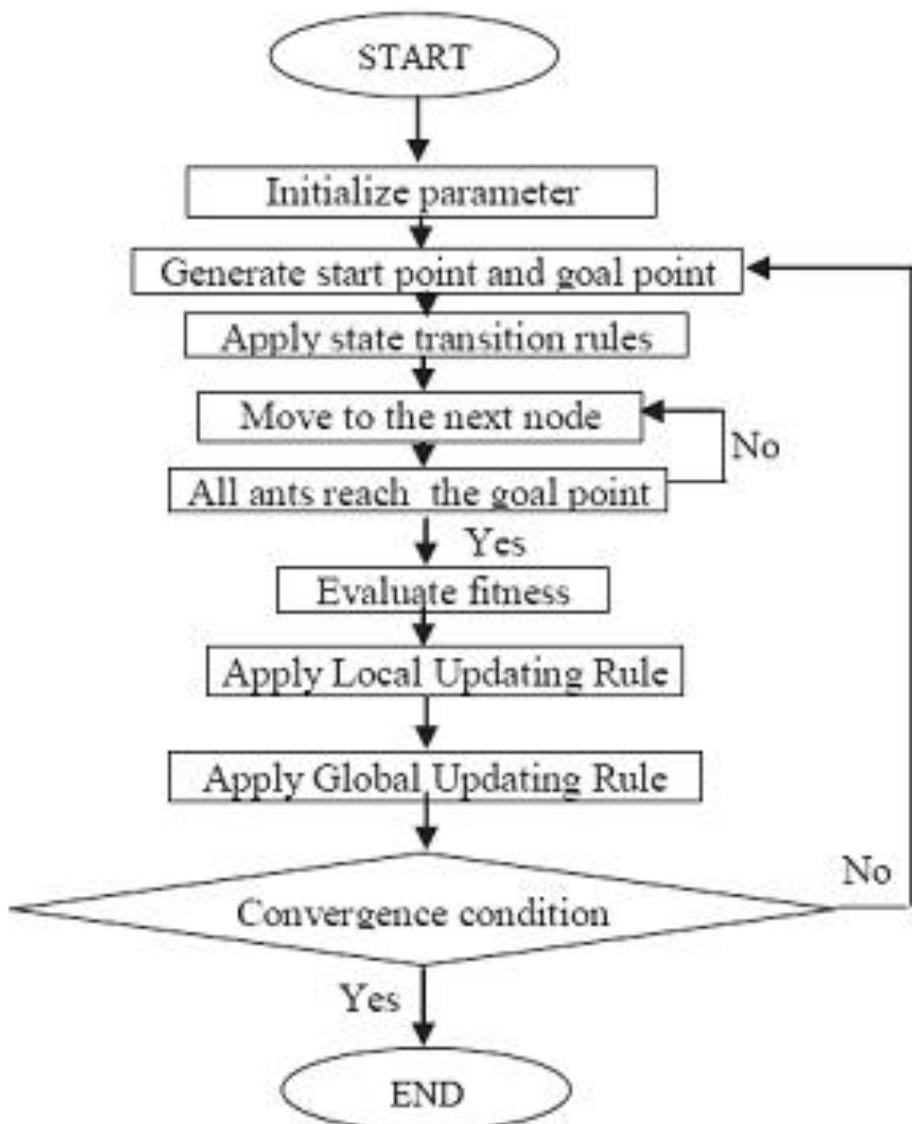
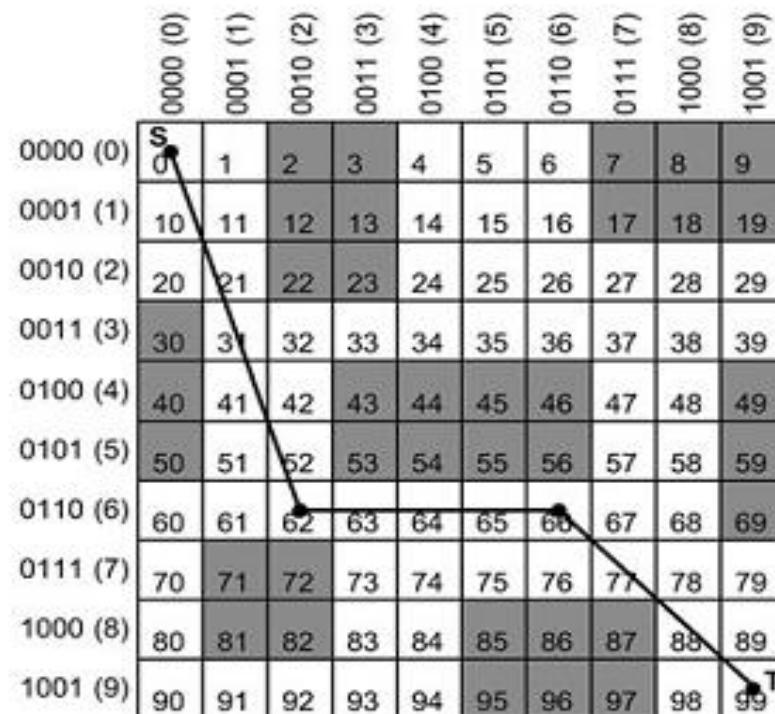


Fig. 10 Outline of ACO for RPP of a mobile robot

Рассмотрим снова представление окружения робота в виде «решетки» -сетки (grid).

The size of each cell has been determined according to the size of fixed and moving obstacles as well as the amount and limitation of robot's motion. Then rate of cell occupation is measured by each of these barriers at every moment. Fixed obstacles with different numbers, shapes and coordinates are in environment.



Путь от начальной точки S (левый верхний угол) до конечной точки- цели T (правый нижний угол)

В каждой клетке (вершине графа) робот имеет несколько возможных направлений движения.

Как мы видели ранее, возможные направления можно определить по-разному (с различной точностью).

На рис. представлены 2 варианта кодирования направлений движения

- а) 4 направления -робот может передвигаться по горизонталям и вертикалям.
- б) 8 направлений- робот может передвигаться также по диагоналям.

Для кодирования направлений используются целые числа (как показано на рис.): а)(0,1,2,3) либо б)(0,1,2,3,4,5,6,7)

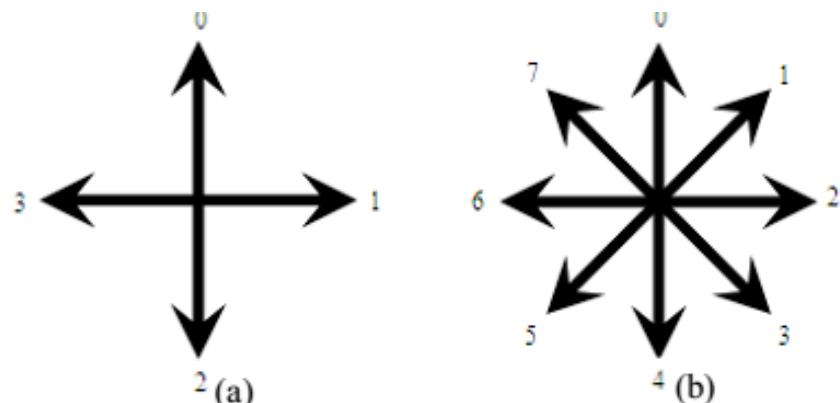


Figure 1. (a) 4-way movement (b) 8-way movement

Итак, напомним, что для решения задачи с помощью муравьиного алгоритма необходимо, прежде всего, определить:

- 1) Правила и вероятности перехода из текущей вершины в следующую;
- 2) Формулы коррекции концентрации искусственного феромона в зависимости от полученных результатов;
- 3) Параметры муравьиного алгоритма.

Каждый муравей, находящийся в текущей вершине, должен выбрать следующую вершину строящегося пути.

В зависимости от количества возможных направлений движений он может выбрать 3 (при 4 возможных направлениях) или 7 (при 8 возможных направлениях) вершин, соответствующим соседним клеткам (ячейкам) дискретной сетки.

Этот выбор (вероятность варианта) производится в соответствии со следующей формулой

$$P_i^k(t) = \begin{cases} \frac{\tau_i^\alpha(t) \times \eta_i^\beta(t)}{\sum_{l=1}^8 \tau_l^\alpha(t) \times \eta_l^\beta}, & \text{if } \text{node } \notin \text{Black List} \\ 0 & \text{if } \text{node } \in \text{Black List} \end{cases},$$

где ““Black List” определяет список запрещенных (для посещения) вершин в текущей позиции;

$\tau_i(t)$ - концентрация феромона в вершине ;

$\eta_j(t)$ - эвристический фактор, определяемый в данном случае $\frac{1}{h_i}$ - разность угла (смены направления движения);

α - коэффициент, определяющий степень влияния феромона;

β - коэффициент, определяющий степень влияния эвристического фактора.

После того, как каждый искусственный муравей нашел свой путь (потенциальное решение) в муравьином алгоритме выполняется коррекция концентрации феромона для вершин, вошедших в построенные пути.

При этом лучшие (короткие) пути должны премироваться!

Вершины, вошедшие в лучшие пути должны больше повысить концентрацию феромона.

Кроме этого, моделируется процесс испарения феромона.

Реализуется это путем использования соответствующих формул коррекции концентрации феромона

$$\begin{aligned}\tau_{ij}(t+1) &= \tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} &= -\lambda\tau_{ij}(t) + \begin{cases} \frac{Q}{num} & \text{for nodes of the best ant} \\ 0 & \text{otherwise} \end{cases}\end{aligned}$$

, где

$\tau_{ij}(t)$ - концентрация феромона между вершинами i и j на предыдущей итерации;

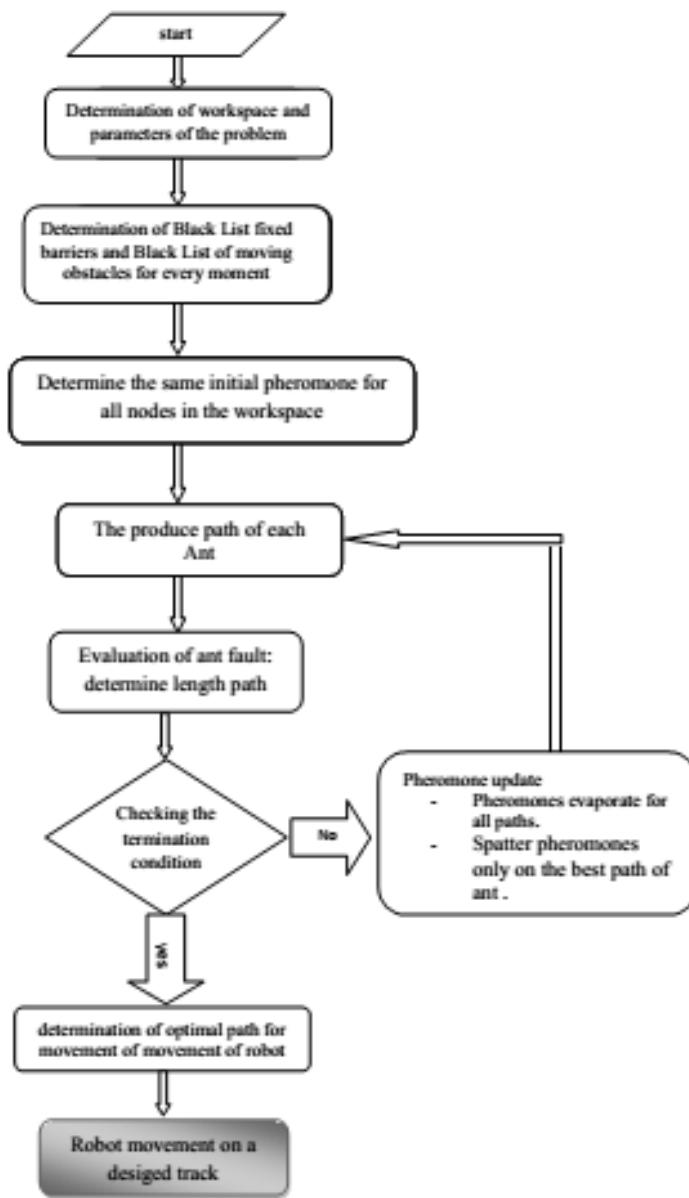
$\tau_{ij}(t+1)$ - концентрация феромона между вершинами i и j на следующей итерации;

Q - количество феромона, оставляемого на вершинах пути, вошедшего в путь лучшего (элитного) муравья:

num – число муравьев;

λ - коэффициент испарения феромона.

Блок-схема муравьиного алгоритма в целом представлена на следующем рисунке.



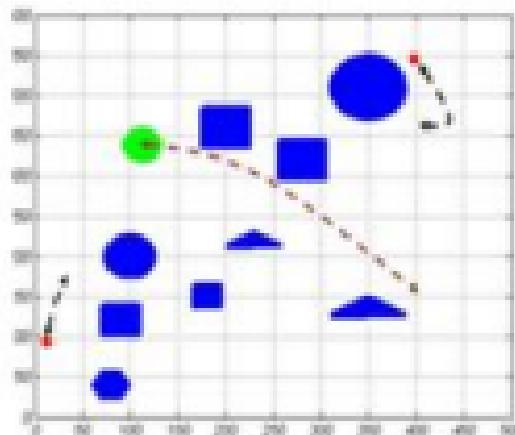
Параметры этого МА показаны в таблице

Parameter	Value
Iteration	30~50
Number of Ants	20~30
Number of Candidate ants	5
Initial Pheromone of All Features	0.5
Evaporation factor (ρ)	0.1
Deposition amount (τ)	0.5
Pheromone trail coefficient, α	1
heuristic coefficient, β	0.5

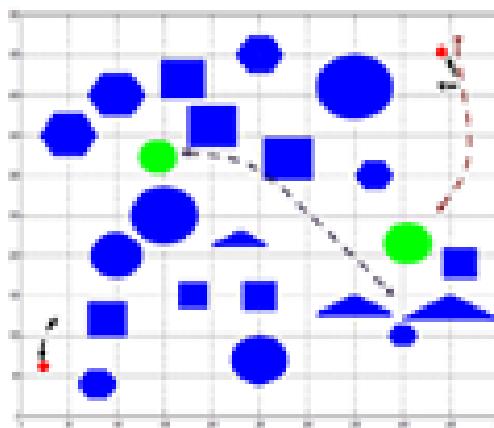
ЭКСПЕРИМЕНТЫ

Программная реализация выполнена в МАТЛАБ (R2011)

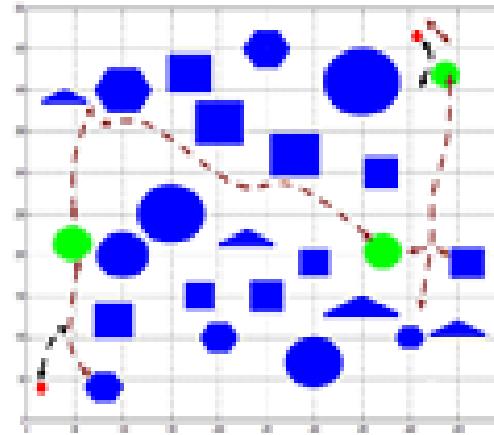
Эксперименты проводились разной сложности средах, например, представленных в



(a)



(b)



(c)

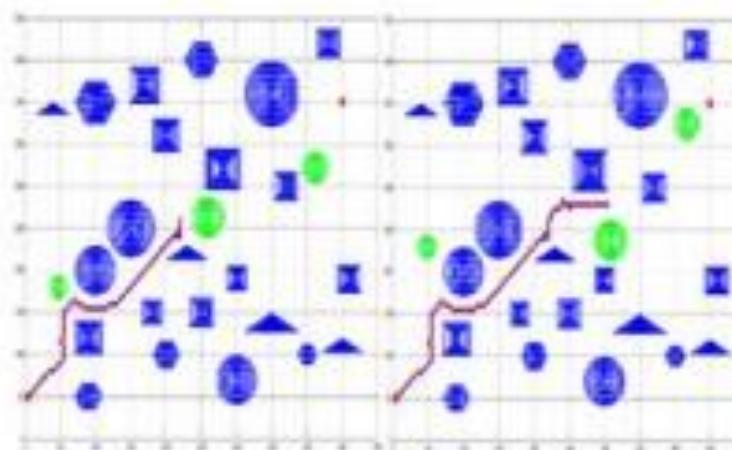
Figure 3-2: (a) relatively simple workspace, (b) relatively complex workspace and (c) quite complex workspace for the robot.

- a) относительное простая среда;
- b) относительно сложная среда;
- c) сложная среда.

Кроме этого, использовались различные начальные и конечные точки в этих средах.

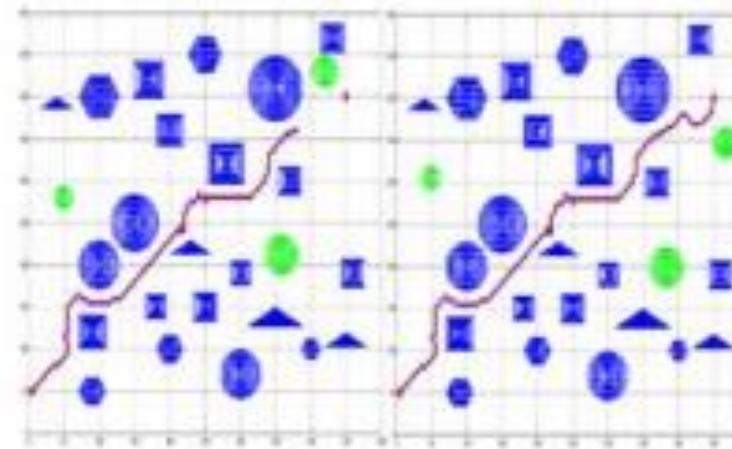
Результаты МА для сложной среды (длина лучшего пути $L=913,82$ см.)

Начальная точка (5, 50), конечная (400, 450)



(a)

(b)

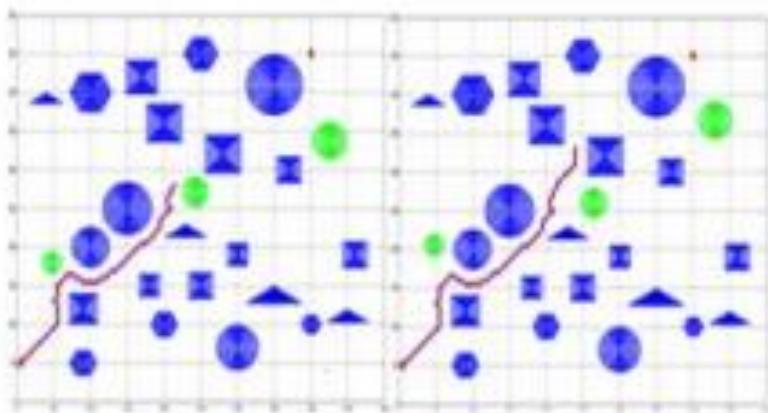


(c)

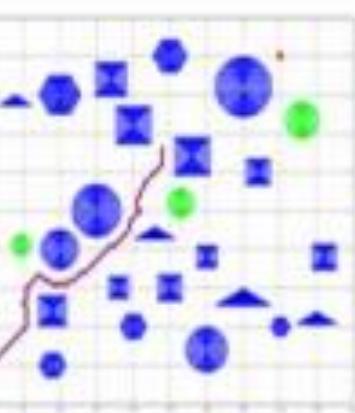
(d)

СРАВНЕНИЕ С ГЕНЕТИЧЕСКИМ АЛГОРИТМОМ

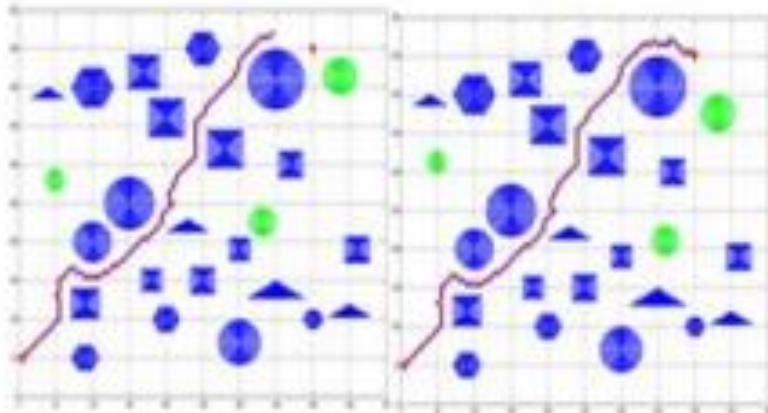
На тех же данных проведены эксперименты построения пути робота с помощью генетического алгоритма (построен лучший путь 921,03см.)



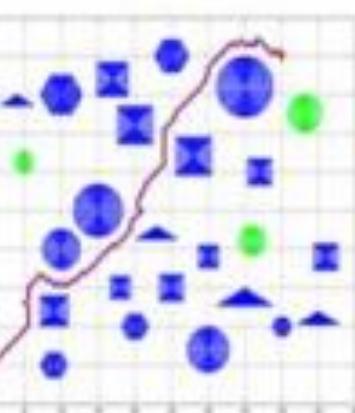
(a)



(b)



(c)



(d)

В следующей таблице даны сравнительные характеристики построенных путей

workspaces complexity	starting point	Target point	Ant colony	genetic
Relatively simple workspaces	(5,50)	(400,450)	610.34	618.05
Relatively simple workspaces	(20,55)	(460,455)	655.22	659.69
Relatively simple workspaces	(30,65)	(465,460)	648.62	652.08
Relatively complex workspaces	(5,50)	(400,450)	919.45	923.04
Relatively complex workspaces	(20,55)	(460,455)	711.28	719.48
Relatively complex workspaces	(30,65)	(465,460)	697.80	707.09
Complex workspaces	(5,50)	(400,450)	913.82	921.03
Complex workspaces	(20,55)	(460,455)	718.59	722.28
Complex workspaces	(30,65)	(465,460)	698.51	703.79

Результаты показывают, что МА имеют лучшие характеристики, чем ГА

СРАВНЕНИЕ ПО ВРЕМЕНИ ПОСТРОЕНИЯ ПУТЕЙ

Представленная ниже таблица показывает, что МА требует гораздо меньше времени на построение пути, чем ГА

workspaces complexity	Relatively simple workspaces		Relatively complex workspaces		Complex workspaces	
Algorithm	Ant colony	Genetic	Ant colony	Genetic	Ant colony	Genetic
Time(s)	47.721	1522.03	648.69	1865.28	972.65	2560.42

При этом, мощность популяции (число муравьев и хромосом) была одинаковой.

Отметим, что число возможных направлений движения робота влияет на эффективность МА.

Чаще всего используется 8 возможных направлений.

Отметим, что эта концепция позволяет легко учитывать данные сенсора при возникновении новых препятствий.

В этом случае направление, по которому обнаружено новое препятствие, (временно) исключается из рассмотрения.

Есть работы, в которых используется большее число возможных направлений.

Такое повышение «степени свободы» дает дополнительные возможности.

Например, на следующем рисунке показано существенно большее число возможных направлений движения робота из текущей вершины.

Здесь текущая позиция робота расположена в центре – A(x,y).

При стандартном подходе робот может выбрать для перехода из A в 8 соседних вершин (B,C,D,E,F,G,H,I).

$S_{(x+3,y-3)}$			$A_{(x,y+3)}$			$T_{(x+3,y+3)}$
	$X_{(x-2,y+2)}$	$W_{(x-1,y+2)}$	$J_{(x,y+2)}$	$U_{(x+1,y+2)}$	$V_{(x+2,y+2)}$	
	$Y_{(x-2,y+1)}$	$I_{(x-1,y+1)}$	$H_{(x,y+1)}$	$G_{(x+1,y+1)}$	$T_{(x+2,y+1)}$	
$L_{(x-1,y)}$	$M_{(x-2,y)}$	$B_{(x-1,y)}$	$A_{(x,y)}$	$F_{(x+1,y)}$	$L_{(x+2,y)}$	$Z_{(x+3,y)}$
	$N_{(x-2,y-1)}$	$C_{(x-1,y-1)}$	$D_{(x,y-1)}$	$E_{(x+1,y-1)}$	$S_{(x+2,y-1)}$	
	$O_{(x-2,y-2)}$	$P_{(x-1,y-2)}$	$K_{(x,y-2)}$	$Q_{(x+1,y-2)}$	$R_{(x+2,y-2)}$	
$\theta_{(x-3,y-3)}$			$\beta_{(x,y-3)}$			$\delta_{(x+3,y-3)}$

«Усиленный» вариант допускает выбор (переход) уже 16 вершин (B,C,D,E,F,G,H,I,K,L,J,M,O,R.
V,X).

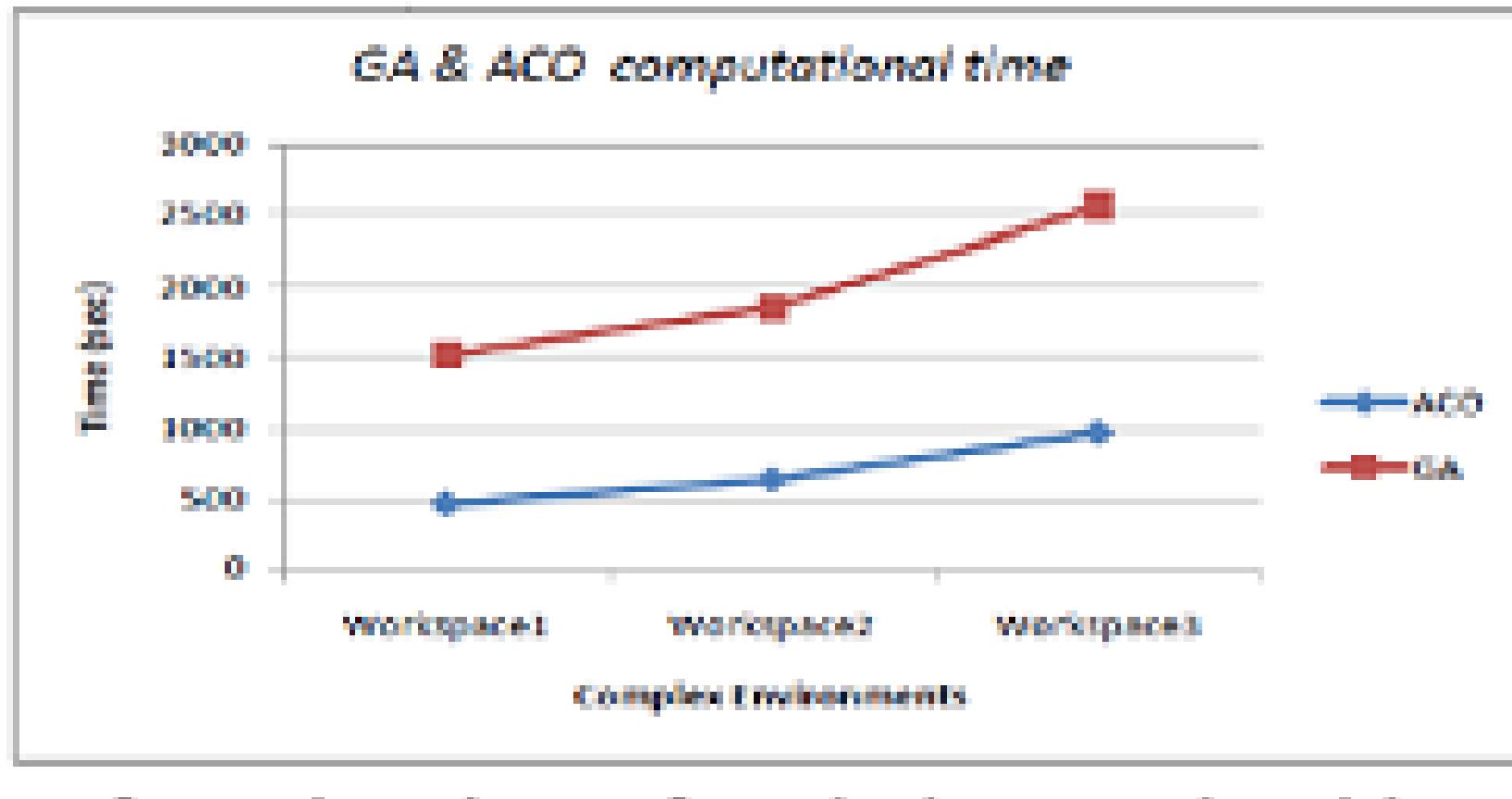
Есть еще более «сильные» варианты выбора:

24 вершины (B,C,D,E,F,G,H,I,K,L,J,M,O,R,V,X,W,Y,N,,P,U,T,S,Q).

и даже 30 вершин (B,C,D,E,F,G,H,I,K,L,J,M,O,R,V,X,W,Y,N,,P,U,T,S,Q,1,2,3,4,5,6,7,8).

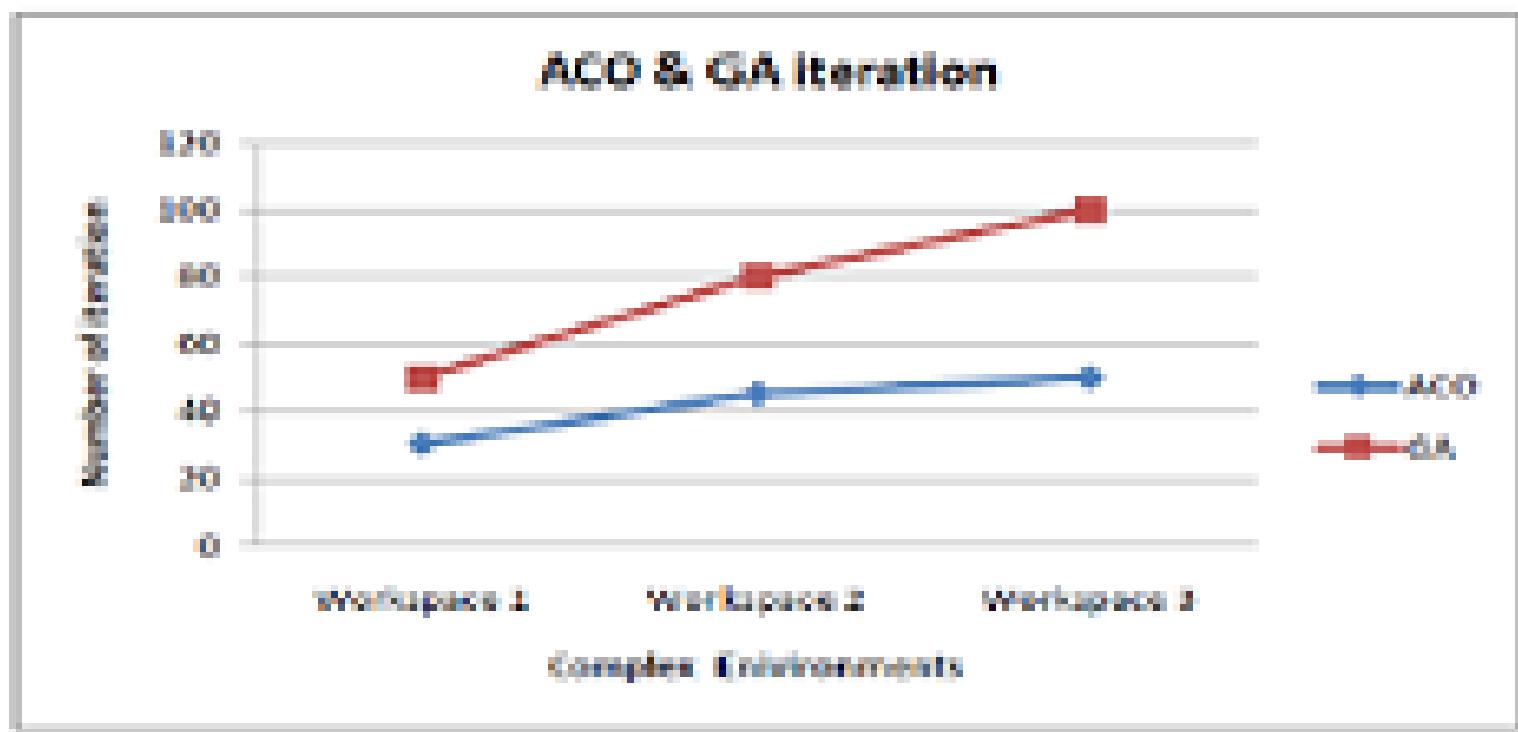
Число возможных переходов повышает возможности алгоритма, но конечно и усложняет его...

СРАВНЕНИЕ ПО ВРЕМЕНИ

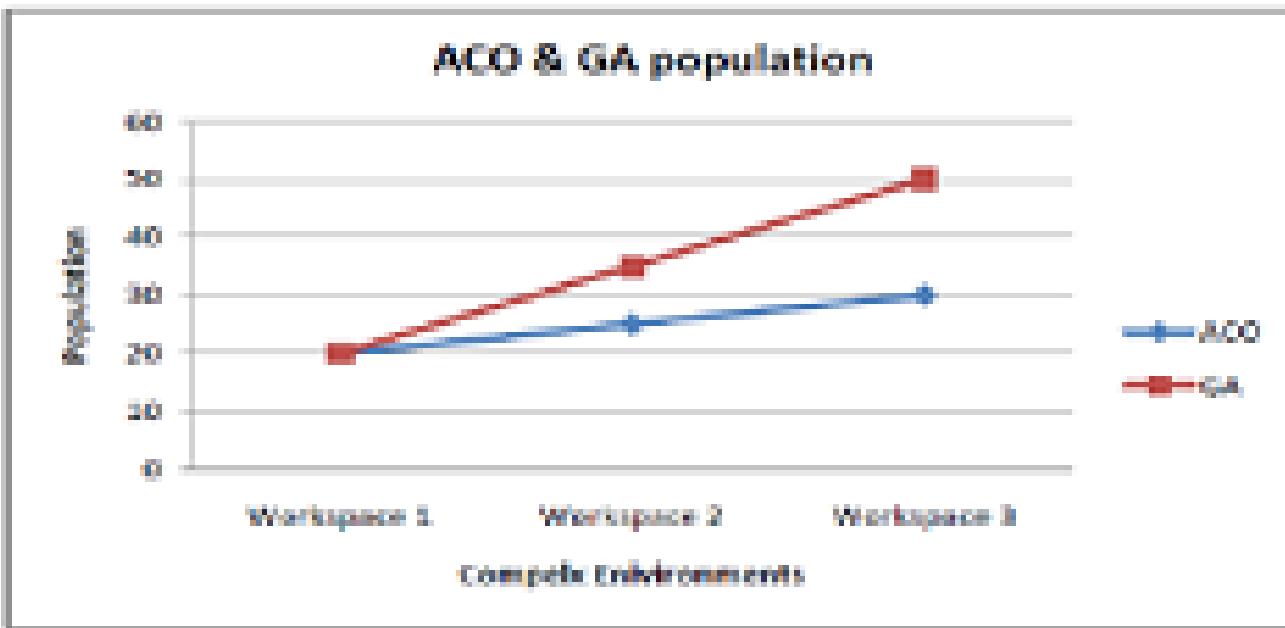


Следует отметить, что МА требует меньшее число итераций и меньшую мощность популяции для сходимости, что показывает следующая таблица

<i>workspaces complexity</i>	<i>Relatively simple workspaces</i>		<i>Relatively complex workspaces</i>		<i>Complex workspaces</i>	
Evolutionary Algorithms	30	Genetic	45	Genetic	50	Genetic
Iterations	30	50	45	80	50	100
Population	20	20	25	35	30	50



Comparison of number of iterations in the ant colony and ge



Comparison of population in the ant colony and genetic algorithms.

Представленные результаты показывают, что МА для задачи построения пути робота является предпочтительней по всем параметрам: качеству решения, времени поиска, числу итераций и т.п.

ГИБРИДНЫЙ МУРАВЬИНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ ПУТИ

Используем также, как и в предыдущем случае (гибридный ГА), в качестве начального приближения субоптимальный путь, построенный с помощью алгоритма Дейкстры.

ОБЩИЙ ПОДХОД

1. Определение модели свободного пространства для мобильного робота на основе **MAKLINK** теории.
2. Использование алгоритма Дейкстры для поиска субоптимального пути, свободного от столкновений.
3. Использование генетического алгоритма для построения глобально оптимального пути.

МОДЕЛЬ СВОБОДНОГО ПРОСТРАНСТВА

1. Высота препятствий и объектов среды игнорируется.
2. Присутствуют несколько заданных препятствий в форме многоугольников (среда также определяется многоугольниками).
3. Для того, чтобы избежать столкновений с препятствиями, границы каждого препятствия расширяются (окаймляются).

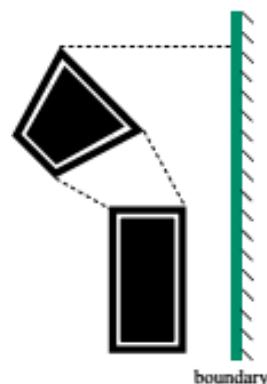
МОДЕЛЬ СВОБОДНОГО ПРОСТРАНСТВА

1. Применение **MAKLINK** теории для формирования свободного пространства..



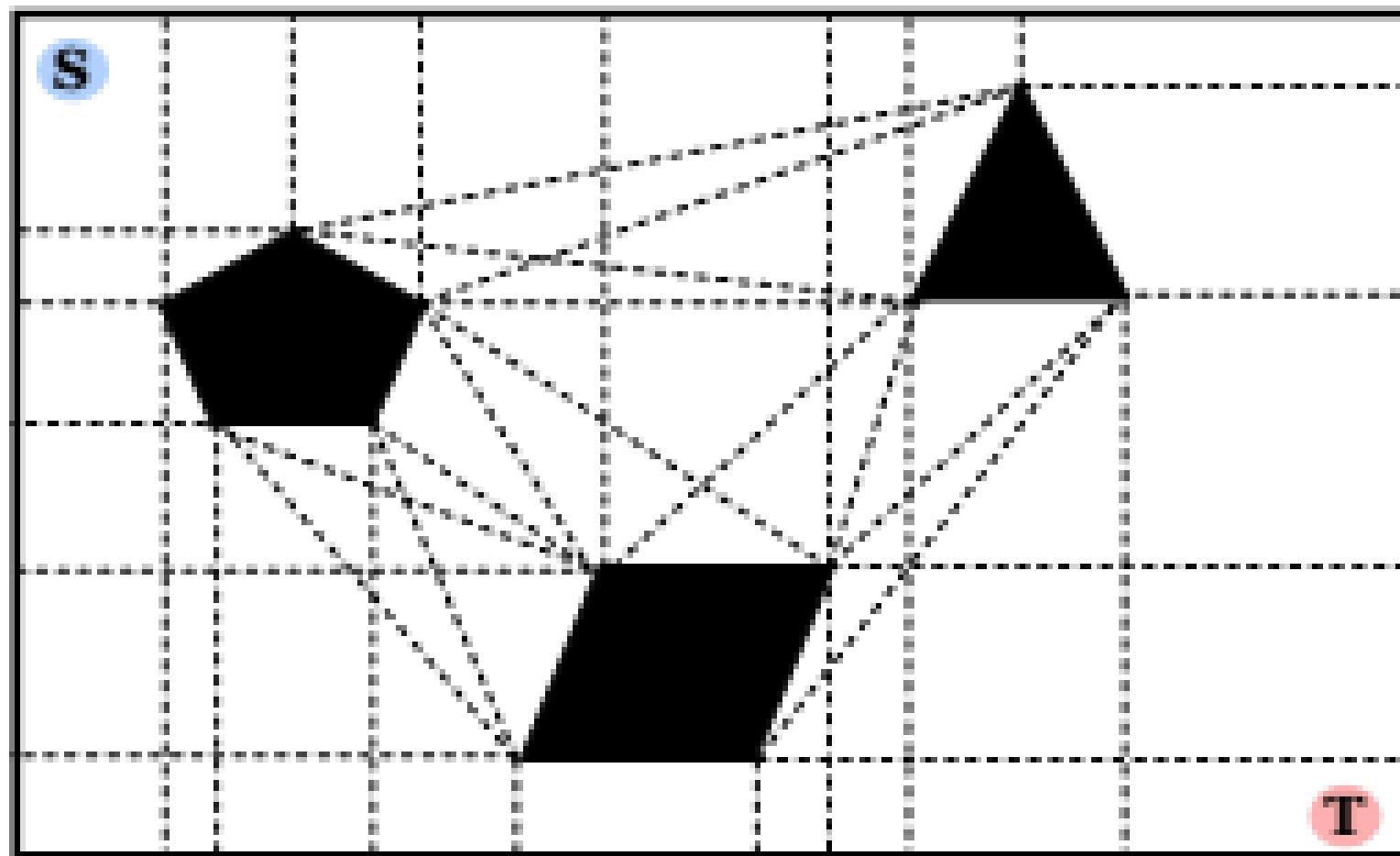
2. Свободная **MAKLINK** линия:

- Конечные точки (линии) – вершины окаймленного препятствия или расположены на границе (среды).
- Каждая **MAKLINK** линия не может пересекать ни одно окаймленное препятствие.



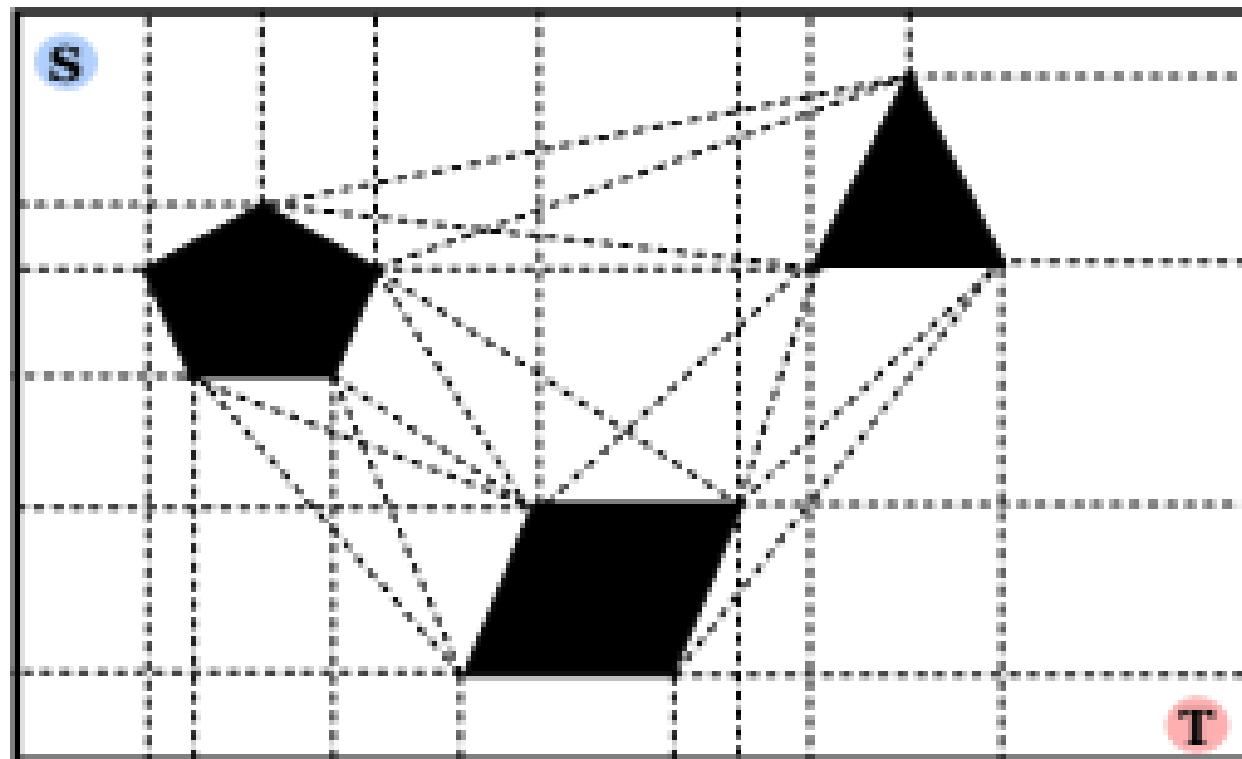
ПОСТРОЕНИЕ MAKLINK ГРАФА

1. Находим все линии, которые соединяют угол, принадлежащий многоугольному препятствию, с углами другими препятствий (включая текущее препятствие)



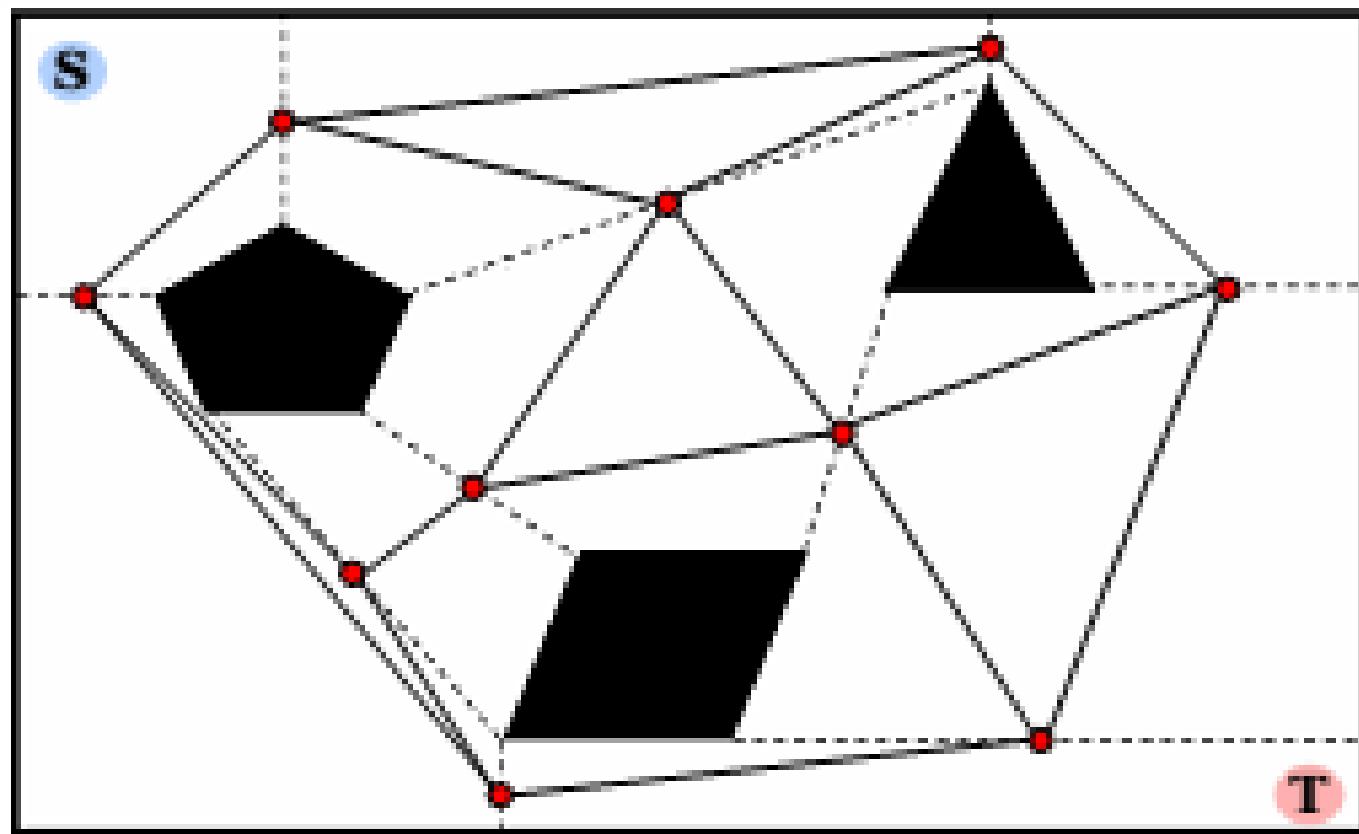
ПОСТРОЕНИЕ MAKLINK ГРАФА

2. Удаляем избыточные свободные линии, чтобы построить свободное пространство, у которого ребра – свободные линии, стороны препятствий(многоугольников) и стены границы среды. Свободное пространство должно быть выпуклым многоугольником и иметь максимальную площадь.



ПОСТРОЕНИЕ MAKLINK ГРАФА

3. Находим средние точки оставшихся свободных линий, используем их в качестве вершин пути, нумеруем их. Соединения средних точек, которые принадлежат той же выпуклой области, образуют сеть.



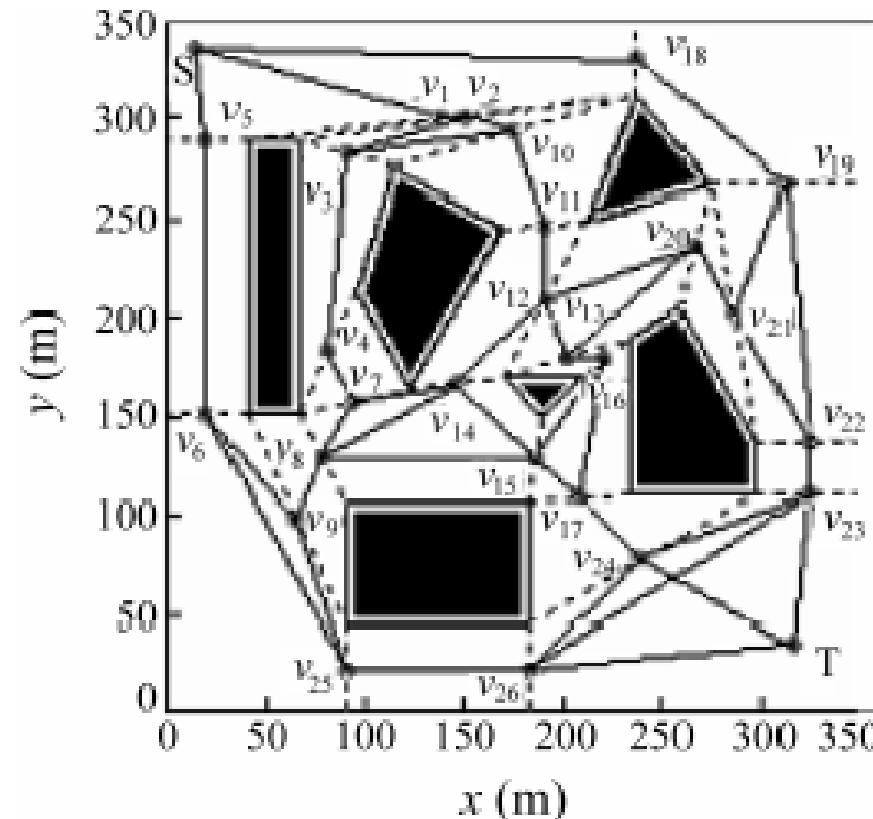
ПОСТРОЕНИЕ ПУТИ

1. Модель свободного пространства

Пусть V_1, V_2, \dots, V_l -- средние точки свободных **MAKLINK** линий;

l -- количество свободных **MAKLINK** линий в **MAKLINK** графе.

Для нашего примера $l=26$. $V_0=S$.

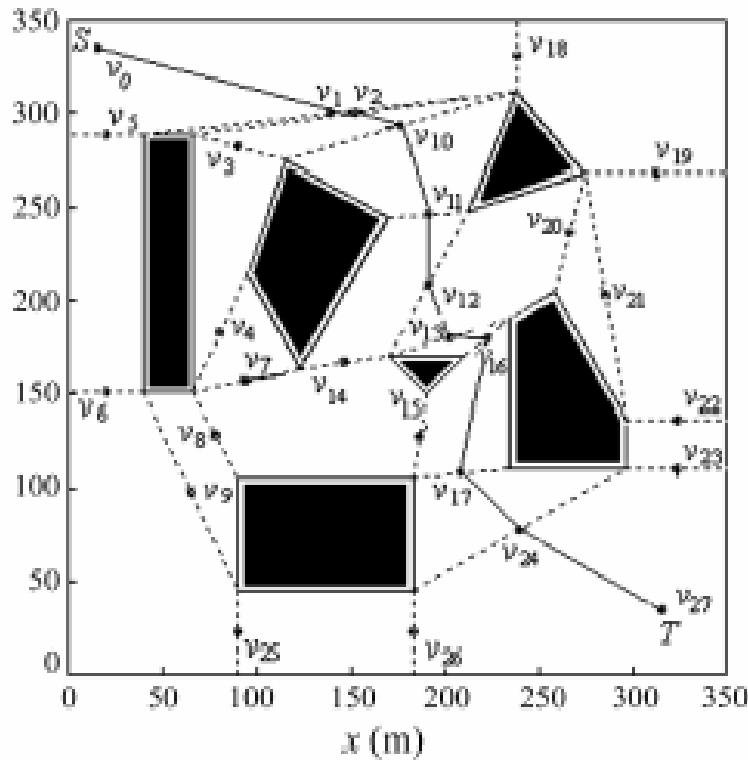


АЛГОРИТМ ДЕКСТРЫ

2. Используем алгоритм Дейкстры. Находим субоптимальный путь

$S \rightarrow v_1 \rightarrow v_2 \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_{12} \rightarrow v_{13} \rightarrow$
 $v_{16} \rightarrow v_{17} \rightarrow v_{24} \rightarrow T$

Длина пути=507.692м. Это только субоптимальный путь (по построению) – он проходит через средние точки свободных **MAKLINK** линий.

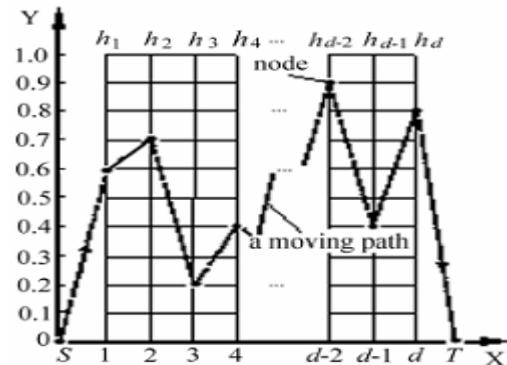


Как и ранее, будем искать оптимальный вектор параметров (смещений относительно средних точек

свободных линий) $\{h_1^*, h_2^*, \dots, h_d^*\}$, который дает минимальную длину пути

$$L = \sum_{i=0}^d \text{length} \{P_i(h_i), P_{i+1}(h_{i+1})\}$$

При этом, используем дискретизацию с некоторым шагом h_i , например, $h_i=0,1$



Тогда муравьиный алгоритм при пересечении каждой свободной линии должен выбрать величину (дискретную!) смещения h_i , относительно средней точки свободной линии.

Т.е. аналогом выбора следующей вершины в графе является выбор дискретного значения смещения h_i .

В такой постановке легко использовать муравьиный алгоритм.

Для применения муравьиного алгоритма к решению некоторой задачи необходимо, прежде всего,

Определить:

- 1. Правило перехода на следующую вершину графа (величину смещения h_i) .*
- 2. Формула коррекции искусственного феромона.*
- 3. Тип и параметры муравьиного алгоритма.*

В данном случае мы будем использовать вариант муравьиного алгоритма – «система муравьиных колоний» (СМК) (ant colony system – ACS).

Прежде всего, необходимо ввести концентрацию искусственного феромона τ_{ij} , который определяет «привлекательность» перехода от варианта смещения h_i к варианту h_j

Предположим, что в начальный момент времени $t=0$ все «узлы» (варианты смещений) имеют одинаковую малую концентрацию феромона τ_0

$$\tau_{ij}(0) = \tau_o \quad (i = 1, 2, \dots, d; \quad j = 0, 1, 2, \dots, 10)$$

При этом i определяет номер пересекаемой свободной линии (точки пересечения),
а j - величину смещения на этой линии

Определим количество искусственных муравьев - m

Каждый из этих муравьев пытается найти оптимальный путь (представляет потенциальное решение).

Муравьи действуют независимо друг от друга,
информация передается только через изменение концентрации феромона.

Из любого «узла» h_i в следующий h_{i+1} каждый муравей переходит в одно и тоже время.
Поэтому, если все муравьи вышли из начальной точки S одновременно, то они
прибывают на каждую линию h_i ($i=1,2,\dots,d$) также одновременно и на конечную
точку T в том числе.

Построение одного тура (пути от начальной точки S до конечной T) представляет 1
итерацию муравьиного алгоритма.

ПРАВИЛО ПЕРЕХОДА

В процессе движения каждый муравей в позиции h_{i-1} должен выбрать «узел» j из 11 возможных вариантов (см.рисунок) смещения h_i в соответствии со следующим правилом перехода:

$$j = \begin{cases} \arg \max_{u \in A} \{[\tau_{iu}(t)][\eta_{iu}]^\beta\}, & \text{if } q \leq q_o \\ J, & \text{if } q > q_o \end{cases}, \quad \text{где}$$

A - представляет множество (выбора): $\{0, 1, \dots, 10\}$;

$\tau_{iu}(t)$ - концентрация феромона узла n_{iu}

$\eta_{iu}(t)$ - представляет «видимость» узла n_{iu} , которая вычисляется в соответствии с:

$$\eta_{ij} = \frac{1.1 - |y_{ij} - y^*_{ij}|}{1.1}$$

y_{ij} - у-координата узла n_{ij}

y_{ij}^* - значение, соответствующее построенному оптимальному пути в предыдущей итерации

β - регулируемый параметр, который определяет относительную важность параметра

«видимости» η_{iu} по сравнению с концентрацией феромона $\tau_{iu}(t)$

q - случайное число, которое выбирается с равной вероятностью из $[0,1]$;

$$q_0 \text{ -- регулируемый параметр} \quad 0 \leq q_0 \leq 1$$

$$P_{ij}^k(t)$$

J –узел, который выбирается с вероятностью в соответствии с формулой «рулетки»

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)].[\eta_{ij}]^\beta}{\sum_{w=0}^{10} [\tau_{iw}(t)].[\eta_{iw}]^\beta} \quad (1)$$

КОРРЕКЦИЯ КОНЦЕНТРАЦИИ ФЕРОМОНА В ИТЕРАЦИИ

После каждой итерации (построения множества путей) выполняется (глобальная) коррекцию концентрации феромона в зависимости от полученного результата в соответствии со следующей формулой

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}(t)$$

$$0 < \rho < 1$$

$$\Delta \tau_{ij}(t) = \frac{1}{L^+} \quad (2)$$

Где L^+ -длина лучшего пути, построенного лучшим муравьем в этом туре

Здесь узел n_{ij} входит в лучший построенный путь L^+

Чем короче(лучше) путь, тем большую коррекцию концентрации феромона получают элементы (узлы), вошедшие в этот путь

КОРРЕКЦИЯ КОНЦЕНТРАЦИИ ФЕРОМОНА ПРИ ПРОХОЖДЕНИИ УЗЛА

После прохождения каждого узла n_{ij} выполняется (локальная) коррекция концентрации феромона в соответствии со следующей формулой

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_o \quad (3)$$

Когда узел посещается муравьями несколько раз с данной локальной коррекцией, это ведет к уменьшению уровня феромона, связанного с этим узлом

Это ведет к расширению пространству поиска (способствует построению новых неизведанных путей)

МУРАВЬИНЫЙ АЛГОРИТМ ПОСТРОЕНИЯ ПУТИ

1. Построить модель свободного пространства на основе MAKLINK теории и найти субоптимальный путь, свободный от столкновений, $S \rightarrow P1 \rightarrow P2 \rightarrow \dots \rightarrow Pd \rightarrow T$ тс применением алгоритма Дейкстры.
2. Определить параметры алгоритма; число искусственных муравьев – m , значения параметров β, q_0, ρ, τ_0 ;
для каждого муравья k ($k=1,2,\dots,d$) определить одномерный массив Path_k – путь с d элементами, в которых у-координаты d узлов (не включая точки S и T) такой, что k -й муравей проходит каждой итерации в указанном порядке.
Массив Path_k используется для обозначения пути k -го муравья.
3. Установить счетчик итераций (поколений) $t=1$ и определить максимальное число итераций NC ;
Поместить m муравьев в начальную точку S .

4. Установить $i=1$.

5. Установить $k=1$.

6. Выбрать узел на линии h_i для муравья k , используя формулу (1)

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)].[\eta_{ij}]^\beta}{\sum_{w=0}^{10} [\tau_{iw}(t)].[\eta_{iw}]^\beta}$$

Передвинуть муравья k в этот узел и запомнить y -координату этого узла в i -м элементе массива Path_k ;

Выполнить локальную коррекцию концентрации феромона согласно формуле (3)

$$\tau_{ij}(t) \leftarrow (1 - \rho).\tau_{ij}(t) + \rho.\tau_o$$

7. Установить $k \leftarrow k+1$. Если $k \leq m$, переход на п.6, иначе продолжать.

8. Установить $i \leftarrow i+1$. Если $i \leq d$,
переход на п.5, иначе продолжать.

9. Передвинуть каждого муравья от его текущего положения к глобальной цели T .
10. Для каждого муравья k ($k=1,2,\dots,m$):
- в соответствии с массивом Path_k получить значения параметров $\{h_1^k, h_2^k, \dots, h_d^k\}$.
 - в соответствии с полученными значениями $\{h_1^k, h_2^k, \dots, h_d^k\}$ определить положения точек пути $P_1^k, P_2^k, \dots, P_d^k$ на соответствующих MAKLINK линиях;
 - Вычислить длину пути L_k , построенного k -м муравьем в соответствии с формулой

$$L = \sum_{i=0}^d \text{length}\{P_i(h_i), P_{i+1}(h_{i+1})\}$$

11. Сравнить полученные m путей и найти самый короткий путь T^t на текущей итерации t . Сравнить T^t с T^+ - оптимальным путем, который был получен за предыдущие $(t-1)$ итераций и обозначить лучший путь через T^+ ; запомнить значения параметров смещения h , соответствующие T^+ , в $\{h_1^*, h_2^*, \dots, h_d^*\}$.

12. Установить каждый элемент Path_k в 0 для $k=1,2,\dots,m$.

13. Выполнить глобальную коррекцию концентрации феромона в соответствии с формулой (2)

$$\tau_{ij}(t) \leftarrow (1 - \rho).\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t)$$

$$0 < \rho < 1$$

$$\Delta\tau_{ij}(t) = \frac{1}{L^+}$$

$$t \leftarrow t + 1.$$

14. Установить .

Если $t < NC$ все m муравьев не построили одинаковый тур, поместить всех муравьев в начальную точку S вернуться на шаг 4) алгоритма.

Если $t < NC$ и все m муравьев построили одинаковый тур или

$T = NC$, то оптимальным путем является путь T^+ ,

а оптимальными значениями параметров являются $\{h_1^*, h_2^*, \dots, h_d^*\}$.

Cton.

ПАРАМЕТРЫ

Выбор значений параметров β, q_0, ρ влияют на сходимость муравьиного алгоритма и качество полученного решения.

Экспериментально были найдены следующие значения этих параметров, которые обеспечили хорошие сходимость результаты:

$$\beta = 2, q_0 = 0.85, \rho = 0.1, \tau_0 = 0.0002, m = 10, NC = 200$$

В результате построен оптимальный путь **L=440.233** метра, представленный ниже.

Заметим, что построение начиналось с субоптимального пути длиной **L=507.692** метра.

Оптимальное множество параметров при этом

$$\{h_1^*, h_2^*, h_3^*, h_4^*, h_5^*, h_6^*, h_7^*, h_8^*, h_9^*\} = \{0.2, 0.1, 0.0, 0.0, 0.5, 0.5, 0.0, 1.0, 0.7\}.$$

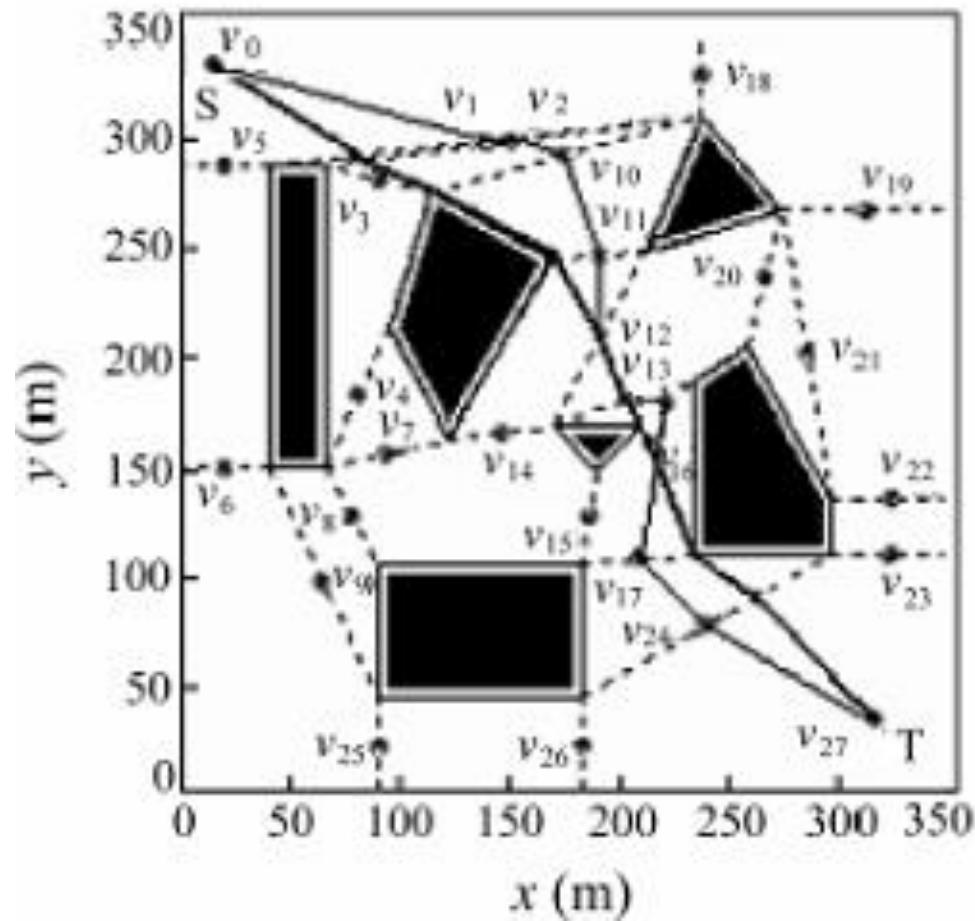
Заметим, что точность решения зависит от величины дискретизации (размера шага смещения h_i).

Например, если уменьшить шаг дискретизации и взять 20 (а не 10) точек, то можно улучшить результат с помощью того же муравьиного алгоритма и построить путь лучше с **длиной L=439.372** метра, который определяется соответствующими значениями параметров смещения:

$$\{h_1^*, h_2^*, h_3^*, h_4^*, h_5^*, h_6^*, h_7^*, h_8^*, h_9^*\} = \{0.20, 0.10, 0.00, 0.00, 0.50, 0.45, 0.00, 1.00, 0.65\}.$$

РЕЗУЛЬТАТЫ

Приведем полученные результаты



СРАВНЕНИЕ МУРАВЬИНЫХ И ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

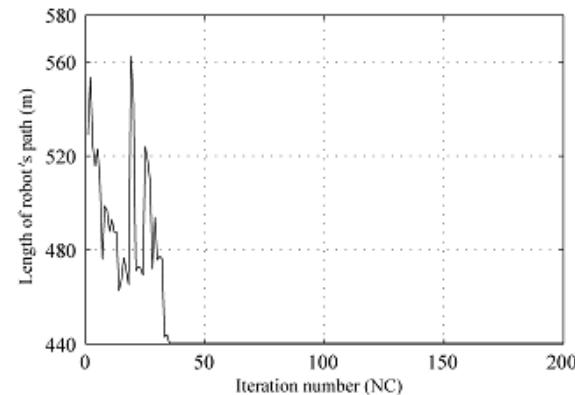
Интересно сравнить результаты построения путей с применением генетических и муравьиных алгоритмов.

Параметры ГА следующие: кодирование вещественное, отбор элитарный, размер популяции $m=50$, вероятность кроссинговера $p_c=0.6$, вероятность мутации $p_m=0.05$, константа кроссинговера

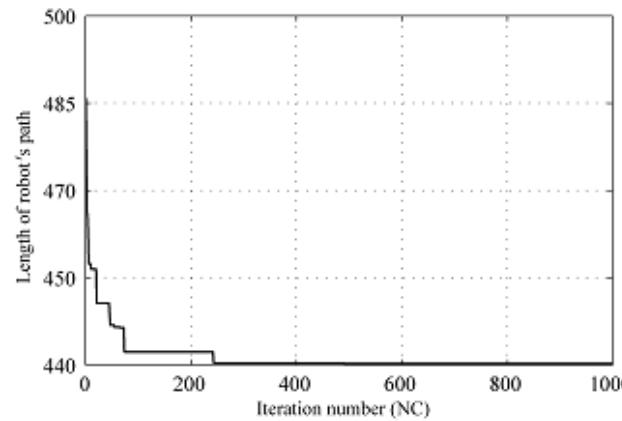
	ACS Algorithm	Real-coded GA
Average CPU time per iteration (sec.)	0. 00059	0. 00067
Average number of iterations needed for convergence	175	912
Average CPU time needed for obtaining optimal solution (sec.)	0.1033	0.6110

СКОРОСТЬ СХОДИМОСТИ АЛГОРИТМОВ

1.Муравьиный –дает оптимум на 38 итерации



3. Генетический – дает оптимум на 240 итерации



СРАВНЕНИЕ ИЗМЕНЧИВОСТИ ПОЛУЧЕННЫХ РЕШЕНИЙ

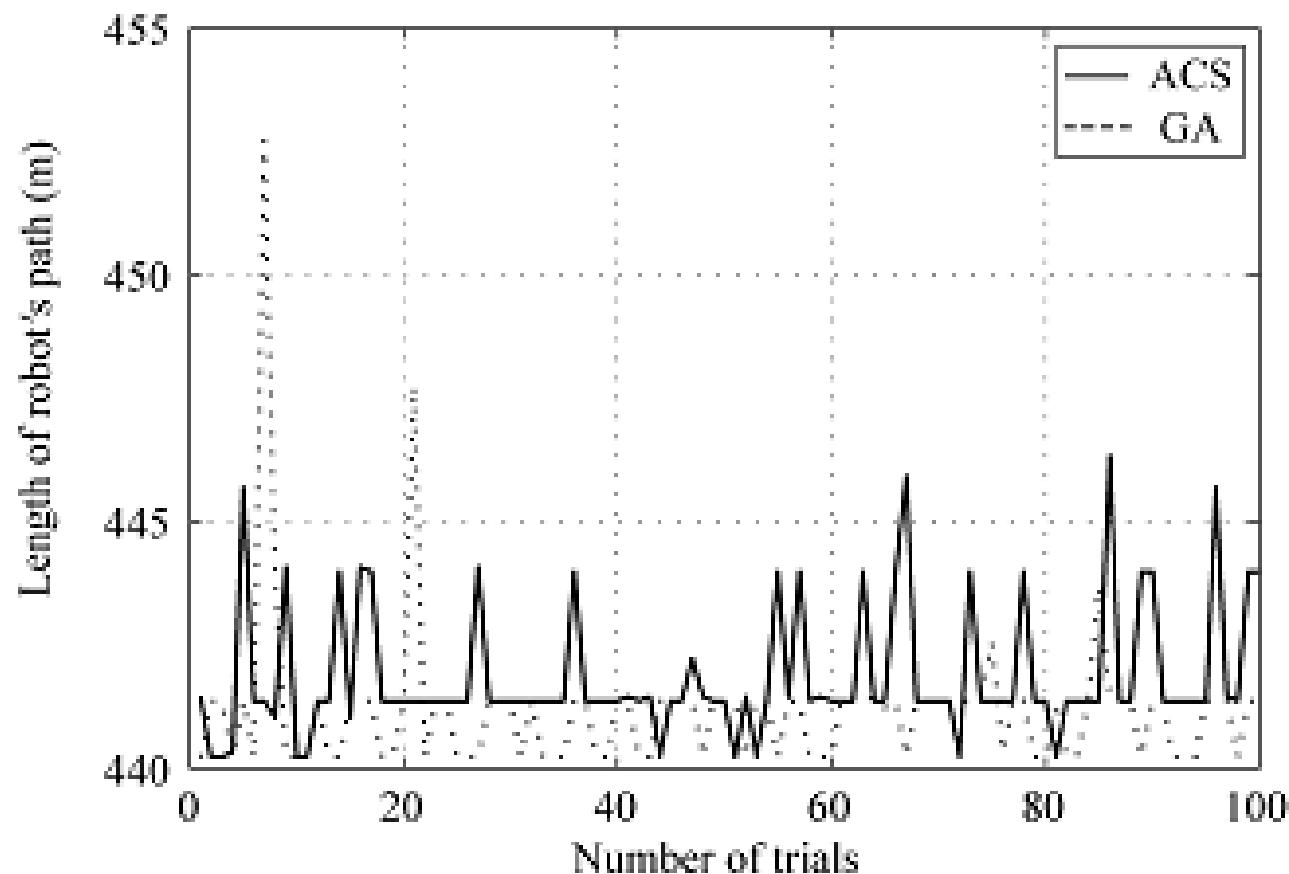


Fig. 8 Comparison of variations in the optimal solutions of both methods (100 trials)

СРАВНЕНИЕ ОТКЛОНЕНИЙ

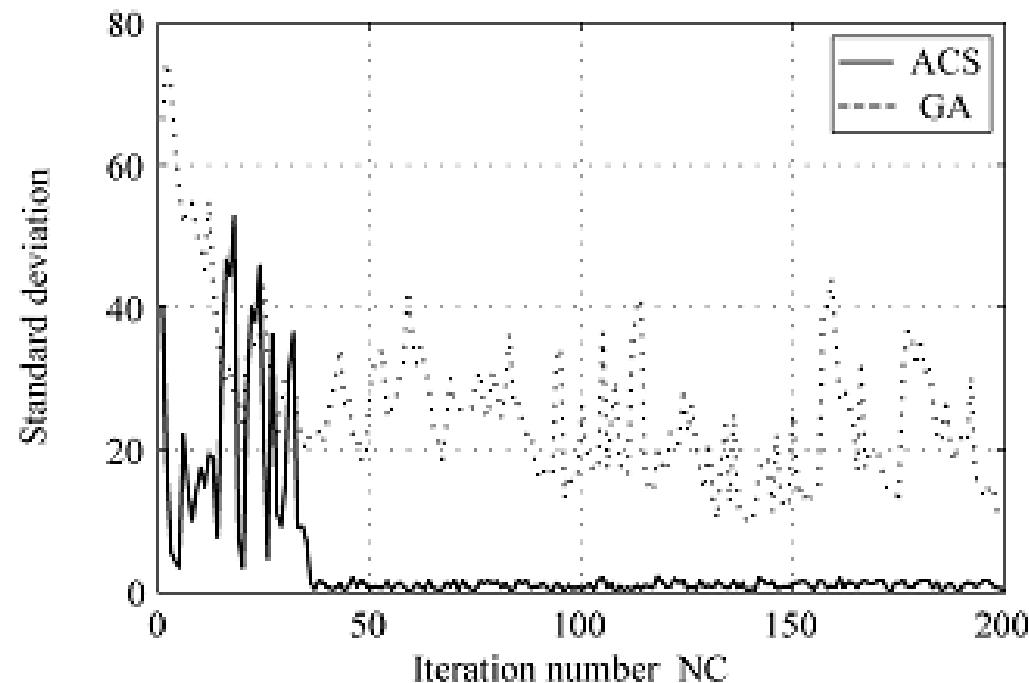


Fig. 10 Convergence tendency of standard deviation σ using both algorithms

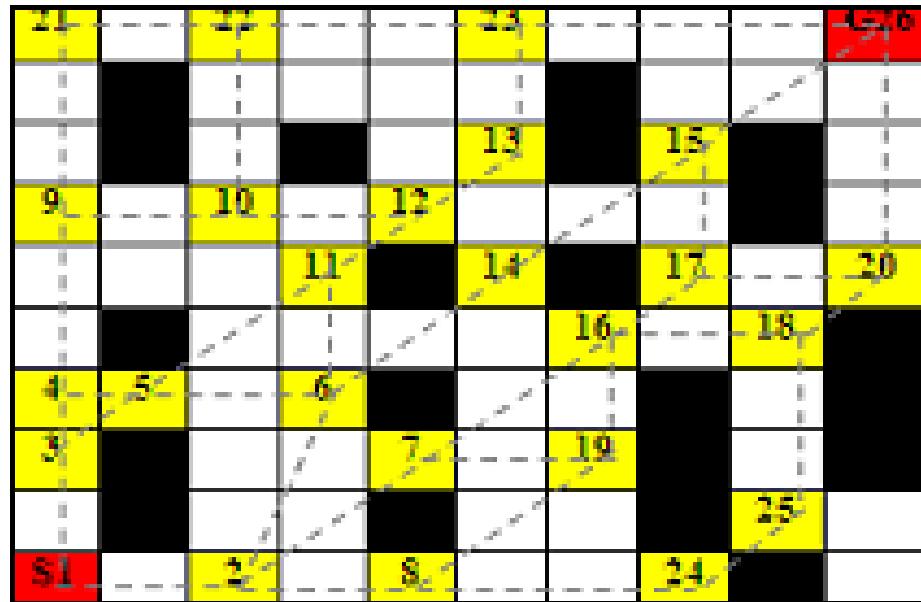
СРАВНЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ЭФФЕКТИВНОСТИ

Table 1 Comparison of computation efficiency of both algorithms

Algorithm	Average CPU time per iteration (sec.)	Average number of iterations needed for convergence	Average CPU time needed for obtaining optimal solution (sec.)
ACS algorithm	0.00059	175	0.1033
Real-coded GA	0.00067	912	0.6110

МОДЕЛЬ СВОБОДНОГО ПРОСТРАНСТВА

Рассматривается дискретная сетка, которая представляет окружение робота



Черным отмечены препятствия, белым свободные клетки

Желтым отмечены вершины, которые может пересекать робот.

Модель свободного пространства представляет

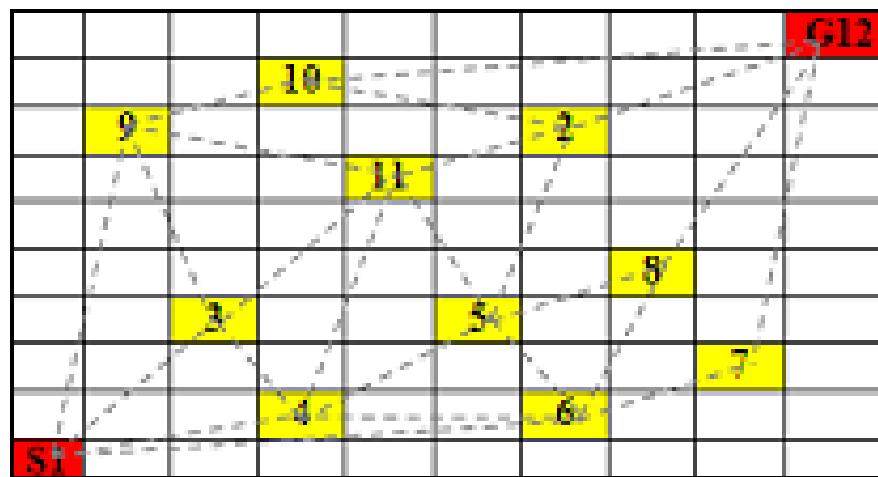


Fig. 4. Simple free space map

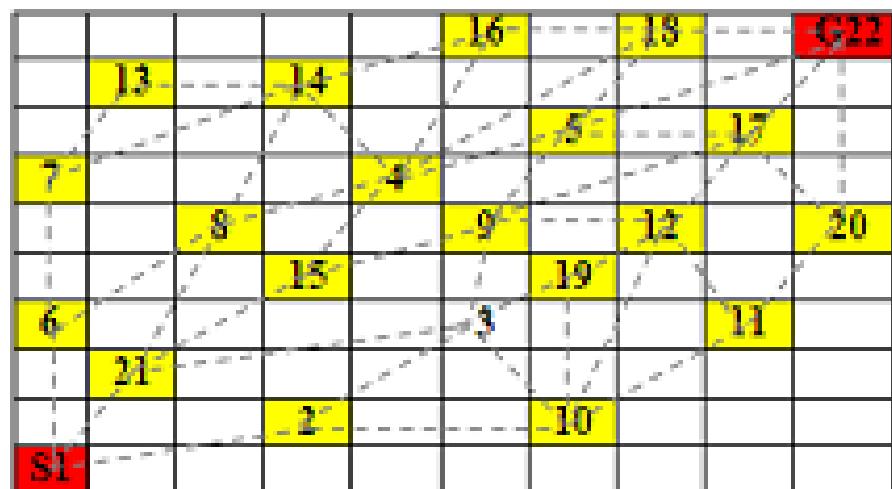


Fig. 5. Average free space map

	51	57	52	53	54	55		56	G63
44	58	48	45		47	60	43	49	50
38	39	40		41	59	42		43	
31		32	33		34		35	36	37
	27	28		29		30		25	
20		21	22		23		24		26
	14			13		16	17		19
10		11	12		15			15	
	61	62	3	4		6	7		9
SE		2			5			8	

Fig. 6. Complex free space map

