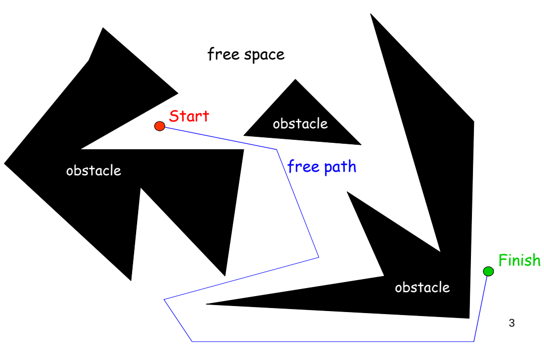
Motion Planning for a Point Robot (1/2)

Purposes

- Introduce simple algorithms with little geometric sophistication
- Present two extreme approaches: purely (sensor-based) reactive strategies and omniscient off-line planners
- Illustrate that motion planning requires predictive models

Problem

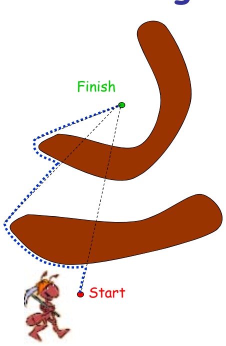


Bug Algorithms

Assumptions:

- The world is a two-dimensional plane
- The robot is modeled as a point
- The obstacles have bounded perimeters and are in finite number
- The robot has no prior knowledge of locations and shapes of the obstacles
- The robot senses perfectly its position (~GPS) and can measure traveled distance
- The robot's touch sensor can perfectly detect contact with an obstacle, allowing the robot to track the contour of the obstacle
- The robot has small computational power and small amount of memory, but can compute the direction toward the goal from its current position, as well as the distance between two points

Bug-0 Algorithm



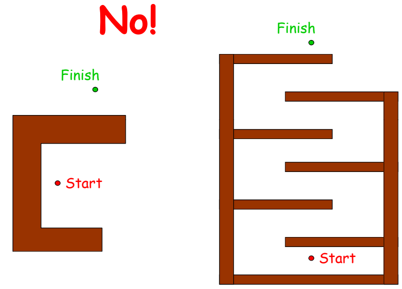
Bug-0

Repeat:

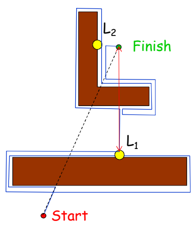
1. Head toward the goal
2. If the goal is attained then stop
3. If contact is made with an obstacle then follow the obstacle's boundary (toward the left) until heading toward the goal is possible again.

Is Bug-0 Guaranteed to Work?

No!



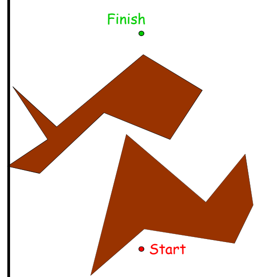
Bug-1 Algorithm



Bug-1:
Repeat:
1. Head toward the goal
2. If the goal is attained then stop
3. If contact is made with an obstacle then circumnavigate the obstacle, identify the closest point L_i to the goal in the obstacles' boundary, and return to this point by the shortest path along the obstacle's boundary

7

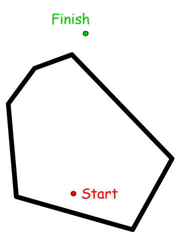
Path Followed by Bug-1?



Bug-1:
Repeat:
1. Head toward the goal
2. If the goal is attained then stop
3. If contact is made with an obstacle then circumnavigate the obstacle, identify the closest point L_i to the goal in the obstacles' boundary, and return to this point by the shortest path along the obstacle's boundary

8

Han Can Bug-1 Recognize that the goal is not reachable?



Bug-1:
Repeat:
1. Head toward the goal
2. If the goal is attained then stop
3. If contact is made with an obstacle then circumnavigate the obstacle, identify the closest point L_i to the goal in the obstacles' boundary, and return to this point by the shortest path along the obstacle's boundary
4. If the direction from L_i toward the goal points into the obstacle then the goal can't be reached. Stop

9

Distance Traveled T by Bug-1?

- Lower bound?
- Upper bound?

Distance Traveled T by Bug-1?

- Lower bound?
  $T \geq D$
  (where D is the straight-line distance from Start to Finish)
- Upper bound?

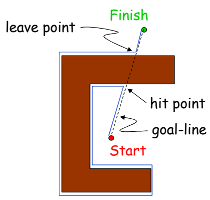
11

Distance Traveled T by Bug-1?

- Lower bound?
  $T \geq D$
  (where D is the straight-line distance from Start to Finish)
- Upper bound?
  $T \leq D + 1.5 \times \Sigma P_i$
  (where ΣP_i is the sum of the perimeters of all the obstacles)

12

Bug-2 Algorithm



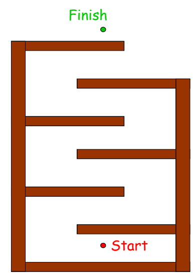
Bug-2:

Repeat:

1. Head toward the goal along the goal-line
2. If the goal is attained then stop
3. If a hit point is reached then follow the obstacle's boundary (toward the left) until the goal-line is crossed at a leave point closer to the goal than any previous hit point on the same side of the goal in the goal-line

13

Path Followed by Bug-2?



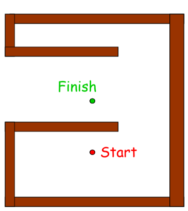
Bug-2:

Repeat:

1. Head toward the goal along the goal-line
2. If the goal is attained then stop
3. If a hit point is reached then follow the obstacle's boundary (toward the left) until the goal-line is crossed at a leave point closer to the goal than any previous hit point on the same side of the goal in the goal-line

14

Path Followed by Bug-2?



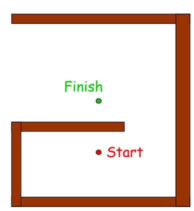
Bug-2:

Repeat:

1. Head toward the goal along the goal-line
2. If the goal is attained then stop
3. If a hit point is reached then follow the obstacle's boundary (toward the left) until the goal-line is crossed at a leave point closer to the goal than any previous hit point on the same side of the goal in the goal-line

15

Path Followed by Bug-2?



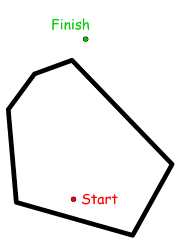
Bug-2:

Repeat:

1. Head toward the goal along the goal-line
2. If the goal is attained then stop
3. If a hit point is reached then follow the obstacle's boundary (toward the left) until the goal-line is crossed at a leave point closer to the goal than any previous hit point on the same side of the goal in the goal-line

16

Han Can Bug-2 Recognize that the goal is not reachable?



Bug-2:

Repeat:

1. Head toward the goal along the goal-line
2. If the goal is attained then stop
3. If a hit point is reached then follow the obstacle's boundary (toward the left) until the goal-line is crossed at a leave point closer to the goal than any previous hit point on the same side of the goal in the goal-line

17

Distance Traveled T by Bug-2?

- **Lower bound?**

  $T \geq D$

  (where D is the straight-line distance from Start to Finish)

- **Upper bound?**

18

Distance Traveled T by Bug-2?

- **Lower bound?**

  $T \geq D$

  (where D is the straight-line distance from Start to Finish)

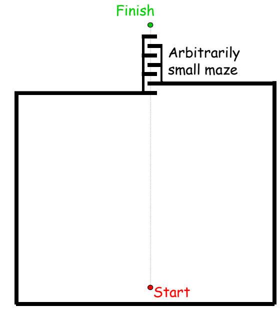
- **Upper bound?**

  $T \leq D + 0.5 \times \Sigma n_i P_i$

  (where the sum Σ is taken over all the obstacles intersected by the goal-line, P_i is the perimeter of intersected obstacle i, n_i is the number of times the goal-line intersects obstacle i)

19

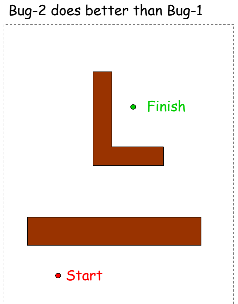
Worst Case for Bug-2?



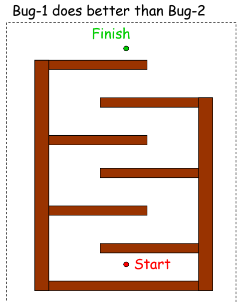
20

Which one --- Bug-1 or Bug-2 --- does better?

Bug-2 does better than Bug-1

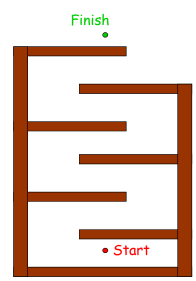


Bug-1 does better than Bug-2



21

Variant of Bug-2



Bug-2':

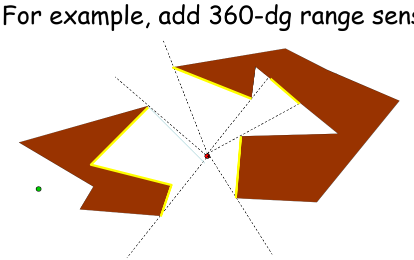
Repeat:

1. Head toward the goal along the goal-line
2. If the goal is attained then stop
3. If a hit point is reached then follow the obstacle's boundary (toward the left) until the goal-line is crossed at a leave point that has not been visited yet

22

Bug Extensions

- Add more sensing capabilities
- For example, add 360-dg range sensing



23

Planning requires models

- Bug algorithms don't plan ahead. They are not really motion planners, but "reactive motion strategies"
- To plan its actions, a robot needs a (possibly imperfect) **predictive model** of the effects of its actions, so that it can choose among several possible combinations of actions

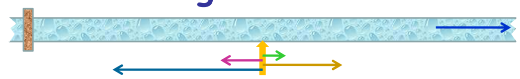
24

Notion of Competitive Ratio

- Bug algorithms are examples of online algorithms where a robot discovers its environment while moving
- The competitive ratio of an online algorithm A is the maximum over all possible environments of the ratio of the length of the path computed by A by the length of the path computed by an optimal offline algorithm B that is given a model of the environment
- What is the competitive ratio of Bug-1 and Bug-2 relative to an algorithm always computes the shortest path?

25

The Bridge-River Problem



- Problem:
  A lost hiker reaches a river. There is a bridge across the river, but it is not known how far away it is, or if it is upstream or downstream. The hiker is exhausted and wishes to find the bridge while minimizing path length.
- Solution:
  The optimal solution consists of moving alternatively in the upstream and downstream directions, exploring 1 distance unit downstream, then 2 units (from the original starting position) upstream, then 4 downstream, and continuing in powers of 2 until the bridge is found.
- What is the competitive ratio of this method?

26

The Bridge-River Problem

- Calculation of competitive ratio:
  - Let us number the moves $1, 2, 3, ..., i, ...$
  - After move i the hiker stands 2^{i-1} units away from the starting position S, downstream if i is odd, and upstream otherwise.
  - In the worst case, the bridge is at distance $d = 2^{k-1} + \varepsilon$ from S, for an arbitrarily small $\varepsilon > 0$ and some $k \geq 1$. In this case, the hiker does not find the bridge at move k, and must perform move $k+1$ and then a fraction of move $k+1$.
  - Each unsuccessful move $i = 1, 2, ..., k+1$ leads the hiker to travel a round-trip distance of $2\times 2^{i-1}$.
  - So, overall the hiker travels:

$$2\times(2^0+2^1+2^2+...+2^{k-1}+2^k)+d = 2\times(2^{k+1}-1)+2^{k-1}+\varepsilon < 9d$$

  - The competitive ratio is bounded by 9.
  - This bound is a tight. For any $r < 9$, there exists d such that the hiker travels more than $r \times d$.

27