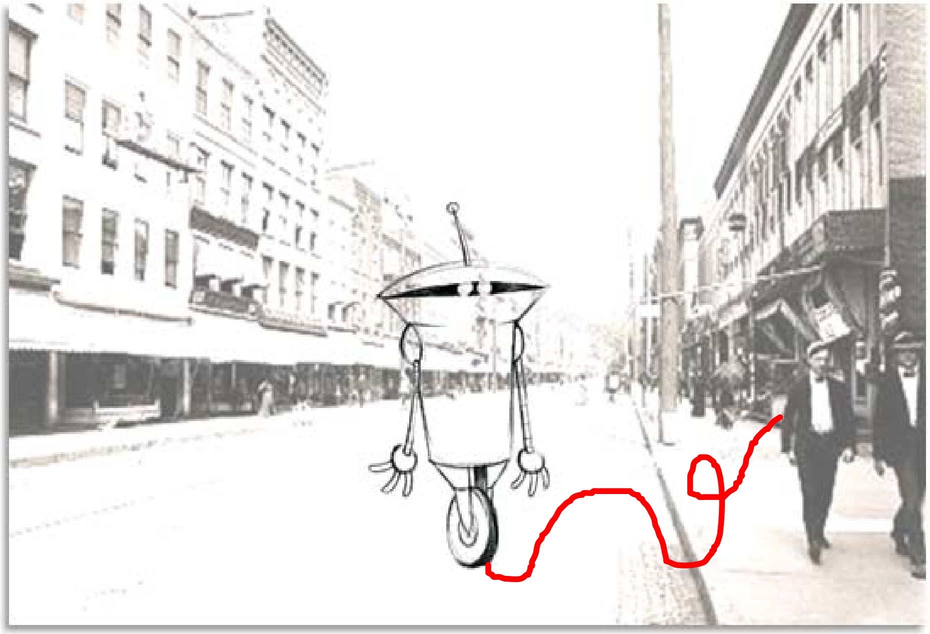
Lecture 18: Motion planning (5)
Graph search and PRMs



Topics:

- Graph search algorithms
  - Dijkstra
  - A*
- Probabilistic roadmaps
- Sampling strategies

Reading:

- Choset: 7, H
- LaVallle: 5

Motion planning

Given: q_{goal} , $q_{1:t}$

sometimes: map , q_0

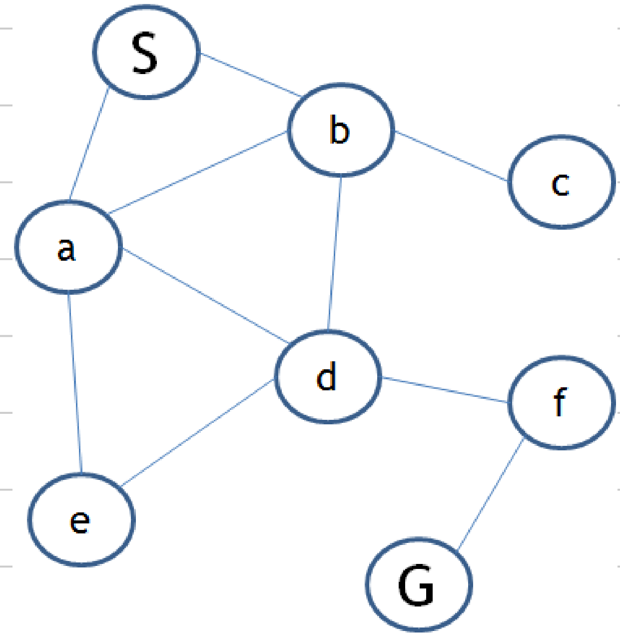
Find: $u_{1:t}$ s.t. $q_t = q_{goal}$

Graph search

```
FORWARD_SEARCH
Q.Insert(x_I) and mark x_I as visited
while Q not empty do
    x ← Q.GetFirst()
    if x ∈ X_G
        return SUCCESS
    forall u ∈ U(x)
        x' ← f(x, u)
        if x' not visited
            Mark x' as visited
            Q.Insert(x')
        else
            Resolve duplicate x'
return FAILURE
```

LV Fig 2.4

lec 17



FIFO LIFO

Breadth Depth

(BFS) (DFS)

Graph search - shortest path

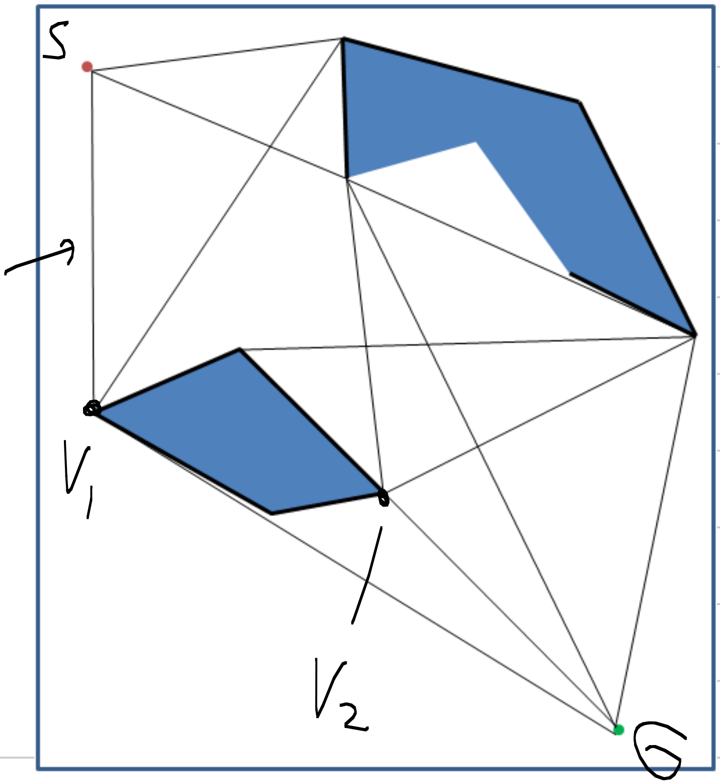
- Dijkstra's algorithm : weighted graphs

cost $l(e_{ij}) \geq 0$

- Q.getFirst = "cost to come"

assign cost $C(v_i)$ to each node $v_i \in V$

$l(e_{s,v_1})$

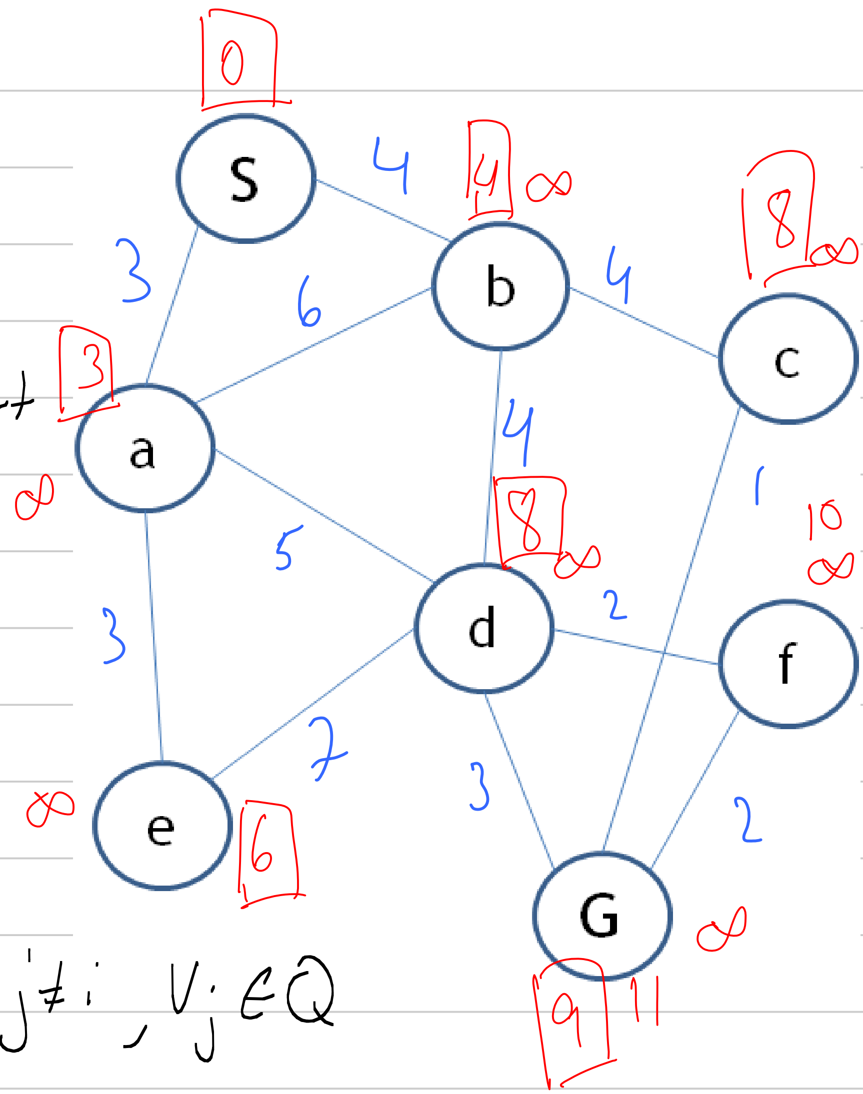


Algorithm:

Init: $C(V_{start}) = 0$

$C(V_i) = \infty \qquad V_i \neq V_{start}$

$Q = V$



While $Q \neq \emptyset$ and $V_{goal} \in Q$

Choose V_i s.t. $C(V_i) \leq C(V_j)$

$\forall j \neq i,\ V_j \in Q$

$\forall e_{ij} \in E$ if $C(V_j) > C(V_i) + l(e_{ij})$

then $C(V_j) = C(V_i) + l(e_{ij})$, $pred(V_j) = V_i$

remove V_i from Q $\quad (Q = Q \setminus V_i)$

Graph search - shortest path

- A* :

cost $\ell(e_{ij}) \geq 0$

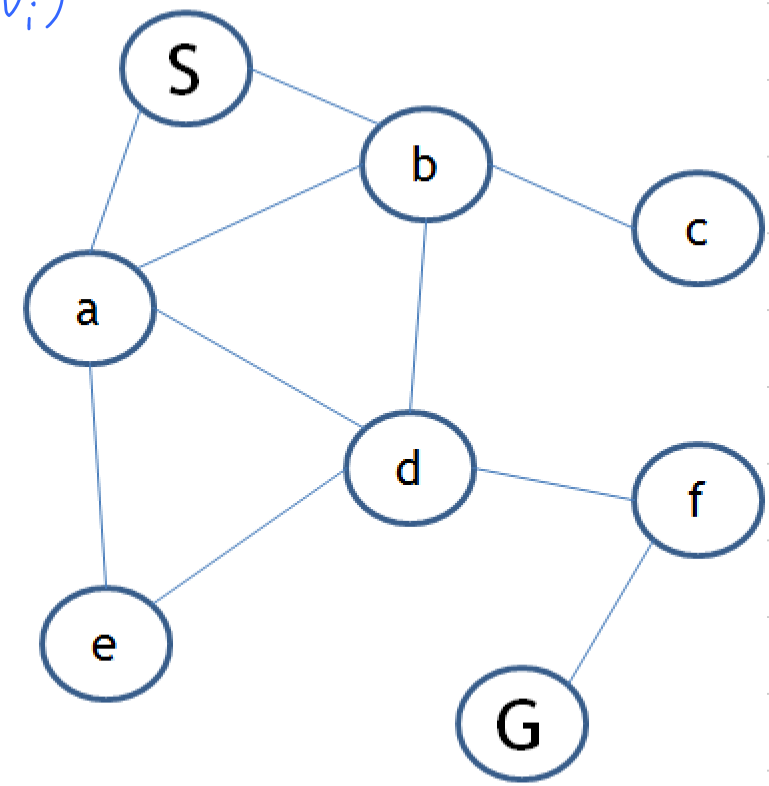
$v_{start} \to v_i$

$c(v_i)$

Q. get first = "cost to come" + "cost to go"

Under approximation

$v_i \to v_{goal}$



Algorithm: $c(v_i)$, $h(v_i)$ = heuristic "cost to go"
<u>under</u> approximation

$$f(v_i) = c(v_i) + h(v_i)$$

Init: $f(v_{start}) = 0 + h(v_{start})$

$f(v_i) = \infty$ $\quad i \neq start$

While $Q \neq \emptyset$ and $v_{goal} \in Q$

Choose v_i s.t. $f(v_i) \leq f(v_j)$ $\forall j \neq i$, $v_j \in Q$

$\forall e_{ij} \in E$ if $f(v_j) > c(v_i) + l(e_{ij}) + h(v_j)$

then $f(v_j) = c(v_i) + l(e_{ij}) + h(v_j)$

$pred(v_j) = v_i$, $Q = Q \setminus v_i$

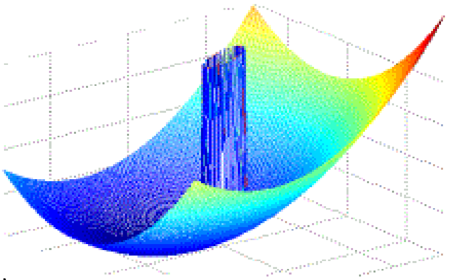
Localization

Motion Planning

"Missing info"

"Discrete" cell decomposition, Roadmap

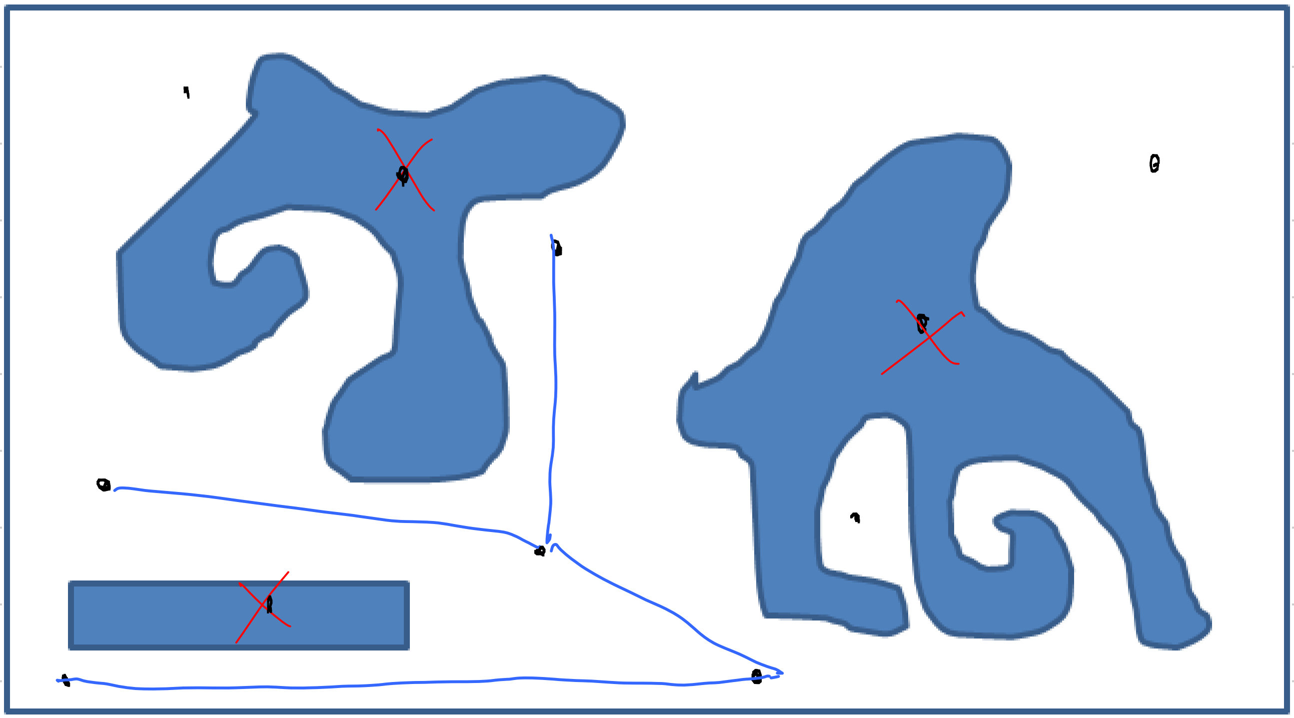
"Continuous" potential / navigation func.



"Samples" PRMs, RRTs

Basic idea:

- Sample points in Q
- check whether points are in Q_{free}
- Connect points in Q_{free}



probabilistic / resolution complete

= complete as # of sample goes to ∞

Building a PRM (Probabilistic Road Map)

given: ability to check whether a point is in Q_{free}, n (stopping condition)

find: PRM: $G = (V, E)$

Init: $V = \emptyset$ $E = \emptyset$

while $|V| < n$

$\boxed{\text{sample}}$ $q \in Q$

if $q \in Q_{free}$

then $V = V \cup \{q\}$

$\forall q \in V$

$\boxed{N_q = \text{set of neighbors}}$

$\forall q' \in N_q$

if $\boxed{(q, q') \in Q_{free}}$ and $(q, q') \notin E$

then $E = E \cup \{(q, q')\}$