MANU2480

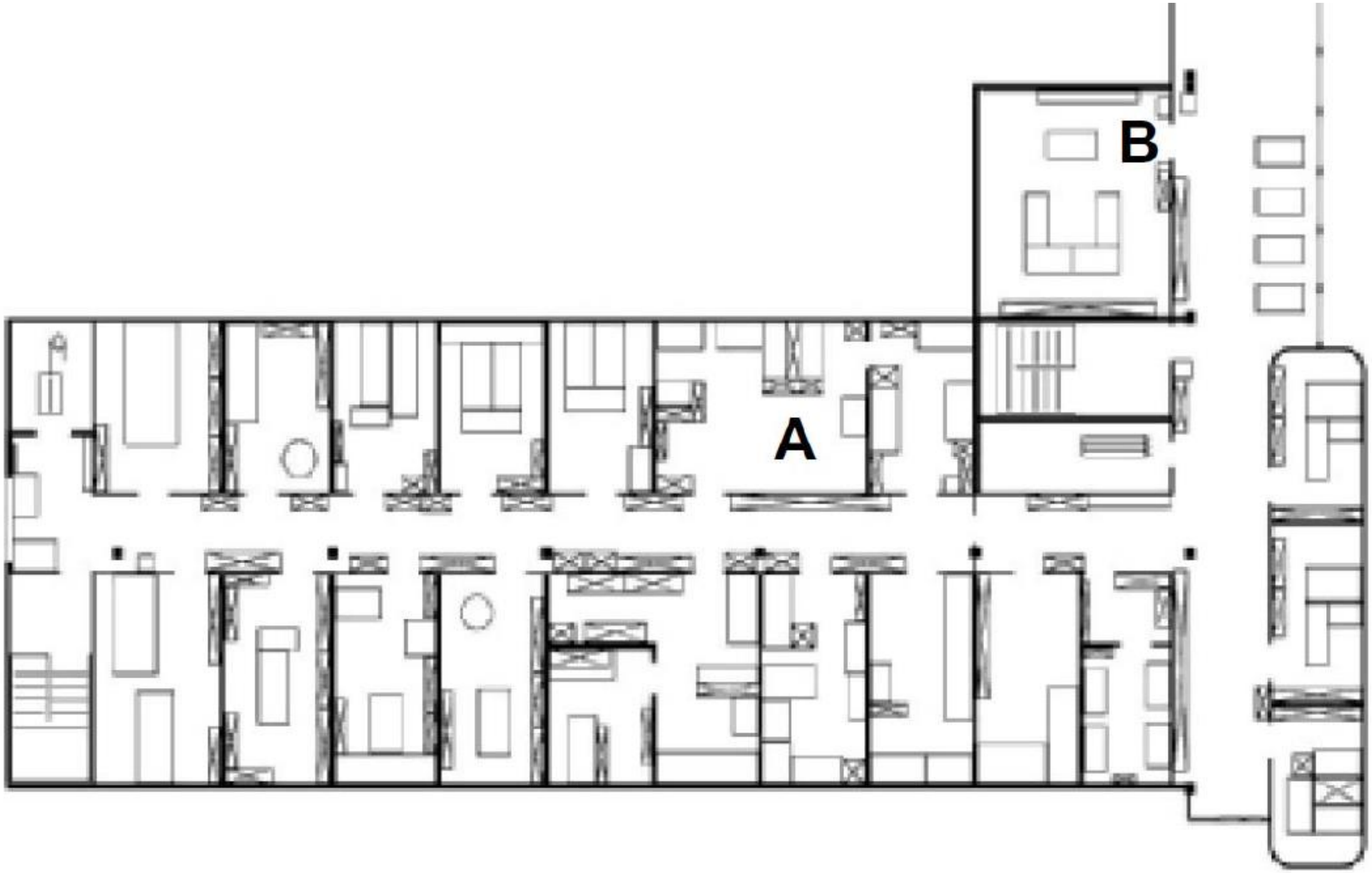
AUTONOMOUS SYSTEM

Path Planning

School of Science and Technology, RMIT Vietnam

RMIT UNIVERSITY

Autonomous Navigation Example

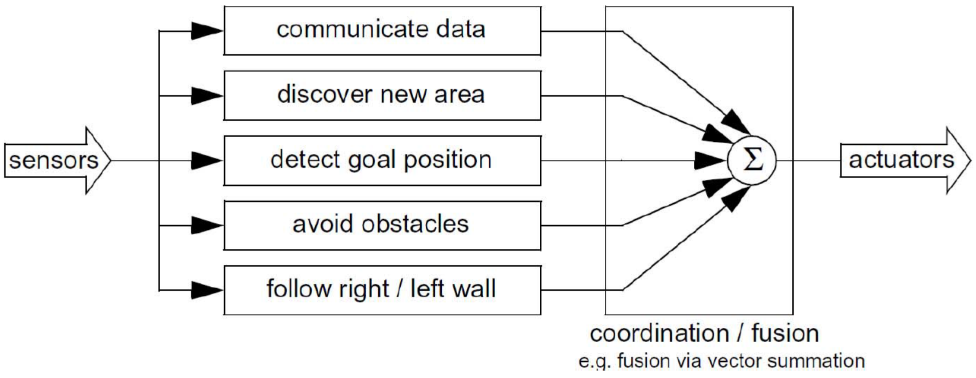


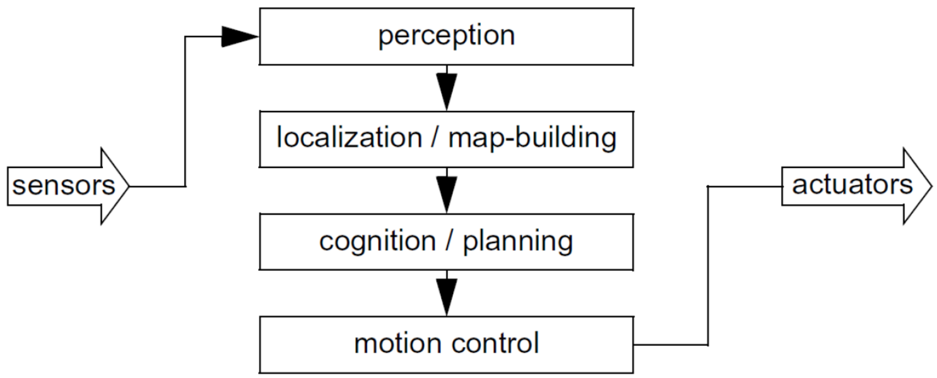
Path Planning Problem

Identify a path (trajectory) in the work space (physical space) from the initial position to the goal position avoiding all collisions with the obstacles.

Autonomous Navigation: Two Approaches

Behaviour-Based Navigation and Map-Based Navigation





Autonomous Navigation: Behaviour-Based Navigation

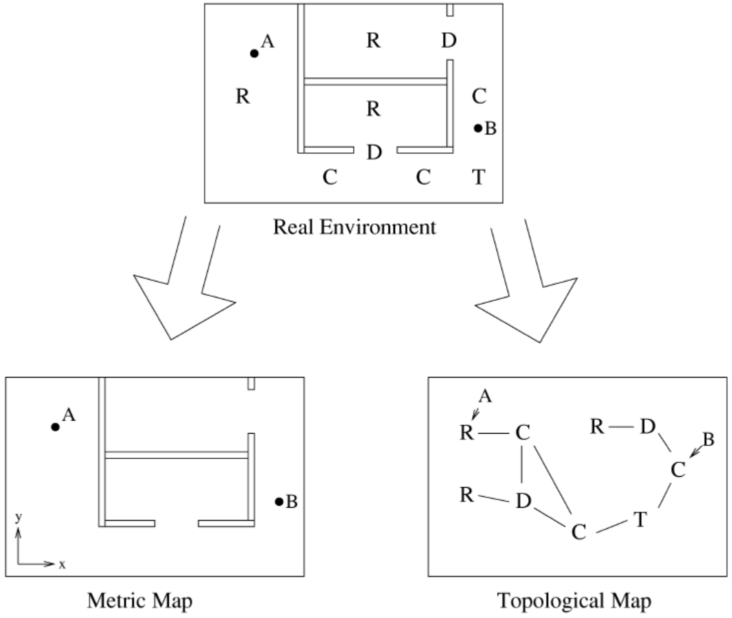
Behaviour-Based Navigation (Reactive Approach, Online)

Genetic algorithm (GA), Fuzzy Logic (FL), Neural Network (NN), Firefly Algorithm (FA), Particle Swarm Optimisation (PSO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO) algorithm, Artificial Bee Colony (ABC) algorithm, Cuckoo Search (CS) algorithm, Shuffled Frog Leaping (SFL) algorithm, and other miscellaneous algorithms.

Autonomous Navigation: Map-Based Navigation

Map-Based Navigation (Classical Approach, Offline): Potential Field Planning method and Graph Search method.

Assumption: There exists a good enough map of the environment for navigation.



Procedure in Map-Based Path Planning

- We can distinguish between: (global) path planning, and (local) obstacle avoidance.
- **3 Steps in Map-Based Path Planning:**

➢ Step 1: Transformation of the map into a representation useful for planning.

➢ Step 2: Plan a path on the transformed map.

➢ Step 3: Send motion commands to controller.

- **Path-Planning is gauged by:**

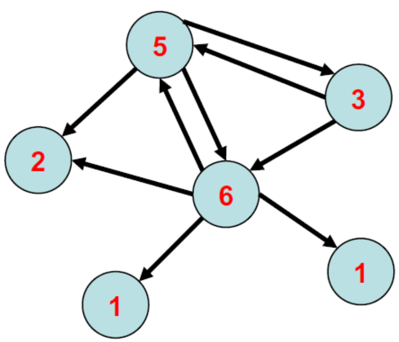
➢ *Completeness:* Guaranteed to find a solution? (if one exist)

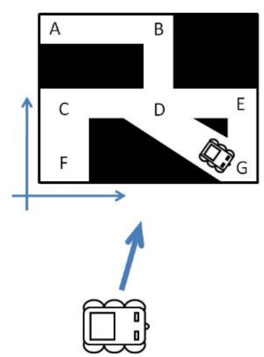
➢ *Optimality:* Finds the best Solution? (for example, shortest)

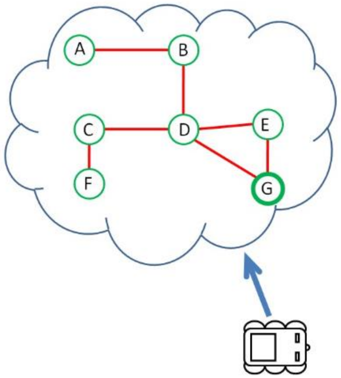
Graph Search Method

There are two main steps in Graph Search Method:

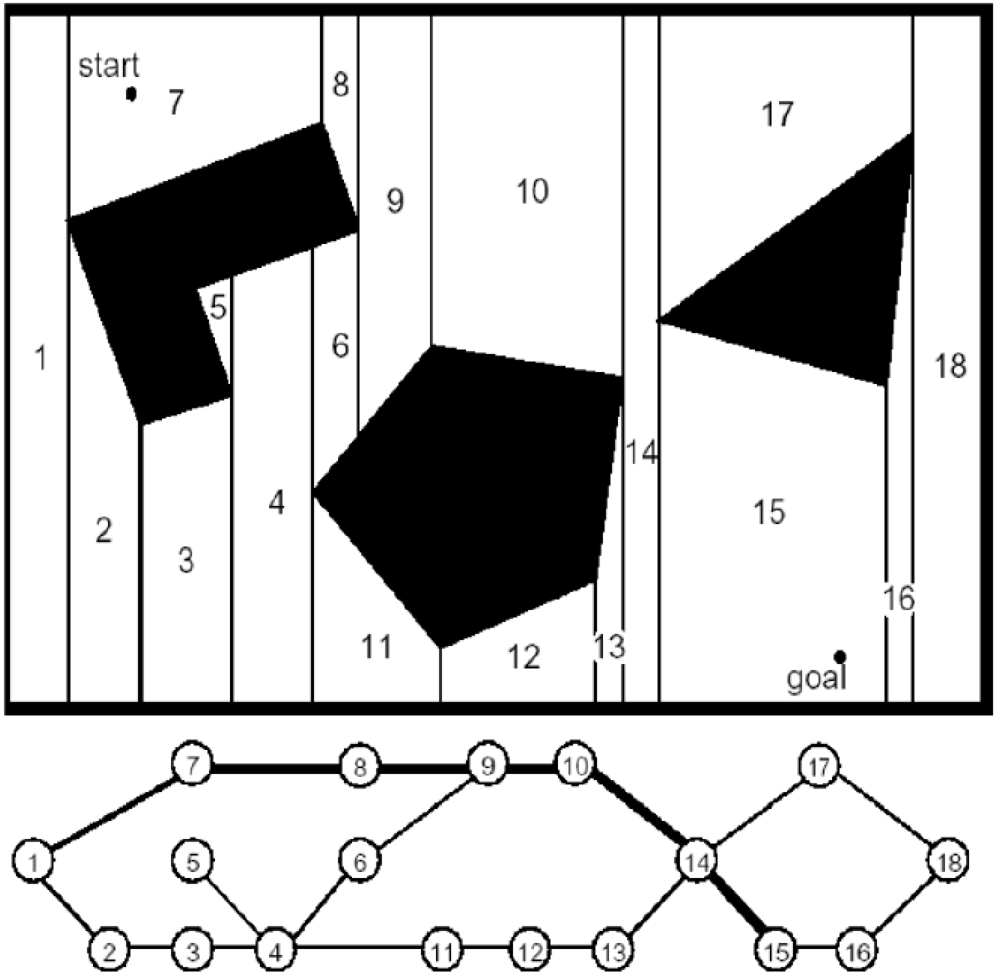
- *Graph Construction*, where nodes are placed and connected via edges.
- *Graph Search algorithm*, where the computation of an (optimal) solution is performed.



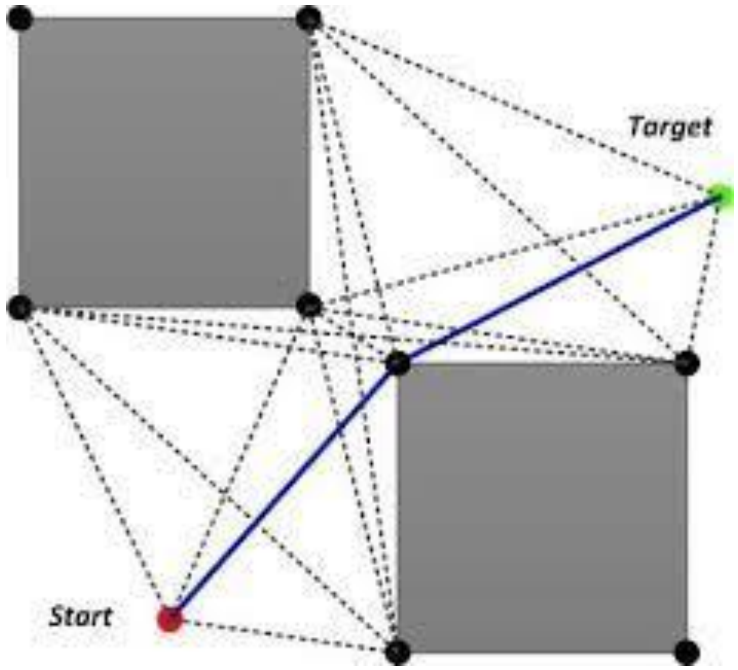


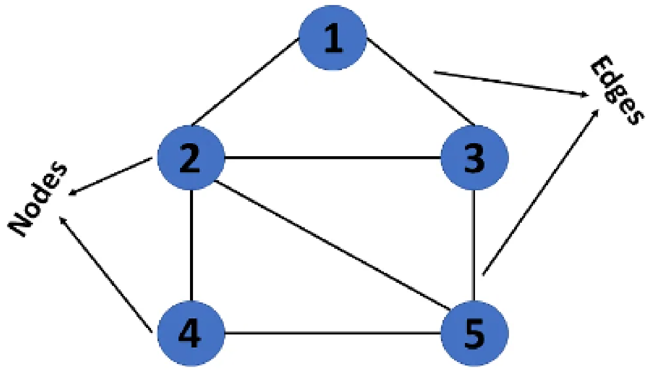


Graph Construction: Cell Decomposition



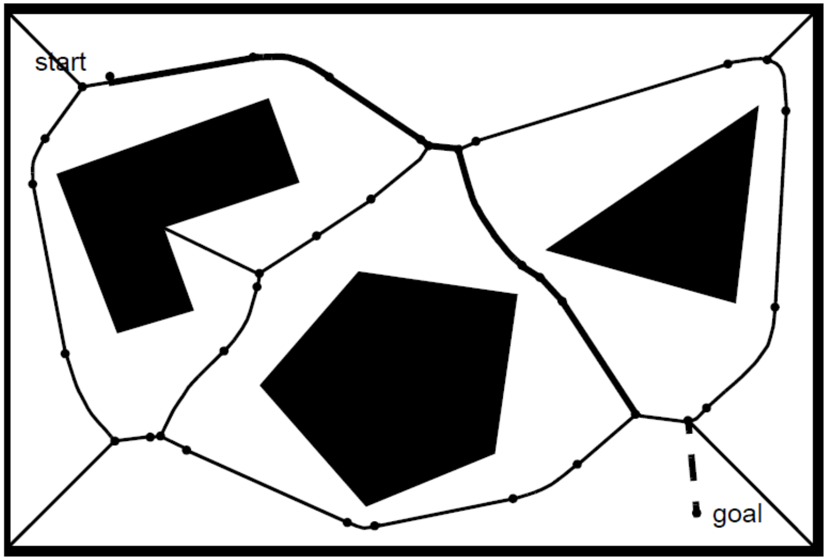
Graph Construction: Visibility Graph

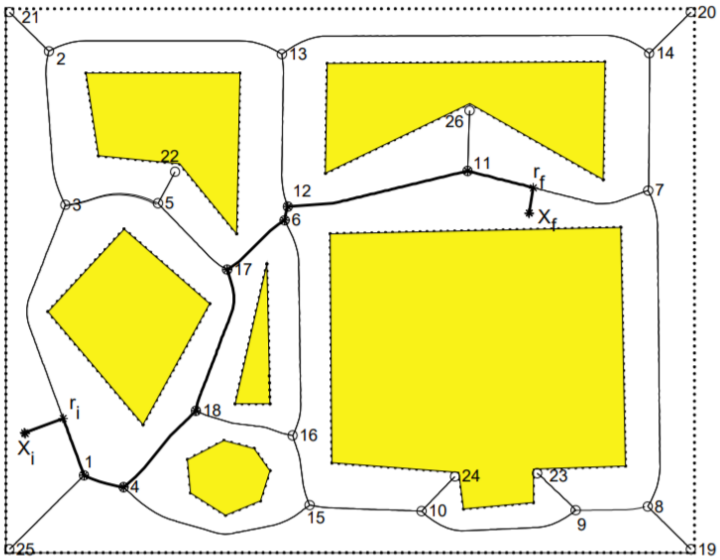




- Visibility graph is a graph of “intervisible” locations, typically for a set of points and obstacles.
- Each node in the graph represents the locations and each edge represents a visible connection between them.

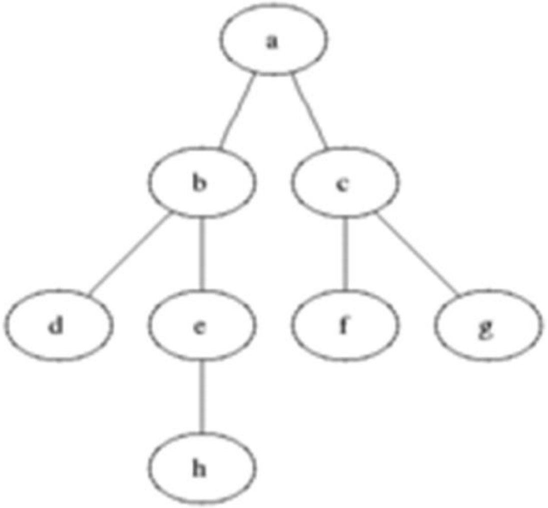
Graph Construction: Voronoi diagram





The Voronoi diagram consists of the lines constructed from all points that are equidistant from two or more obstacles.

Graph Search Algorithm

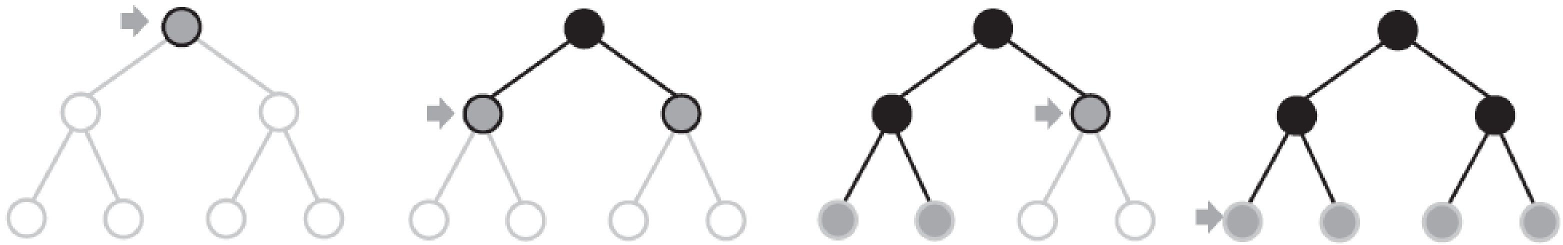


In general, path searching algorithms proceed as follows.

- The starting node is first checked to determine if it is also the goal node.
- The search is extended to other nodes in the neighborhood of the current node.
- One of the neighbor nodes is selected (***how the nodes are selected depends on the used search algorithm and its cost function***), and if it is not the goal node, then the search is also extended to the neighbor nodes of this new node.
- This procedure is continued until the solution is found or until all graph nodes have been investigated.

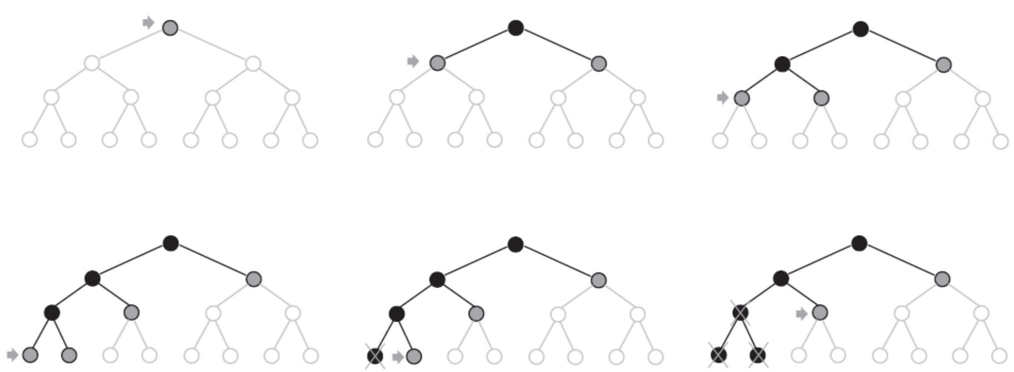
Graph Search Algorithm: Breadth-First Search

- It first explores the shallow nodes, that is, the nodes that are closest to the starting node.
- All nodes that can be accessed in k steps are visited before any of the nodes that require k + 1 steps.



Graph Search Algorithm: Depth-First Search

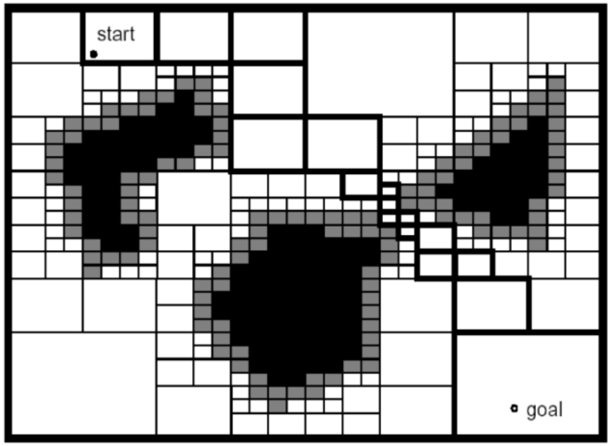
The search is first extended in the depth. The node that is the farthest away from the starting node is used to extend the search. The search continues in the depth until the current node has no further successors. The search is then continued with the next deepest node whose successors have not been explored yet.

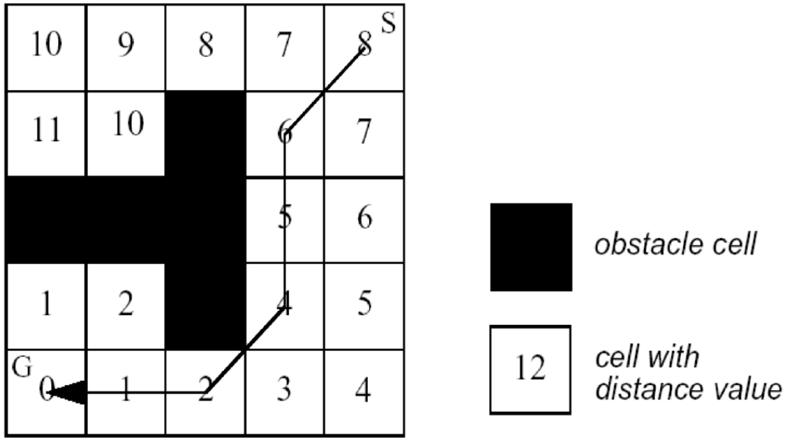


Dijkstra's Algorithm: A Variant of BFS

This algorithm is similar to breadth-first search, except that edge costs may assume any positive value and the search still guarantees solution optimality.

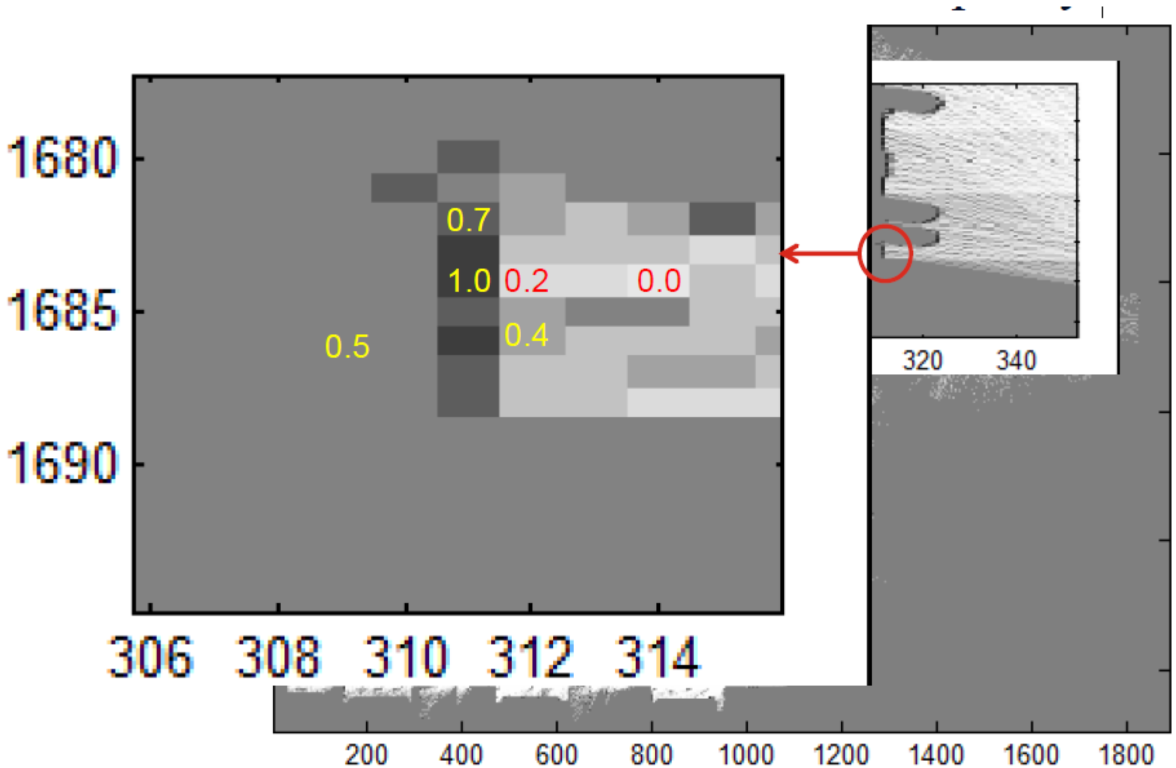
Consequently, not only the best path from the start node to the goal is computed, but also all lowest cost paths from any starting position in the graph to the goal node.





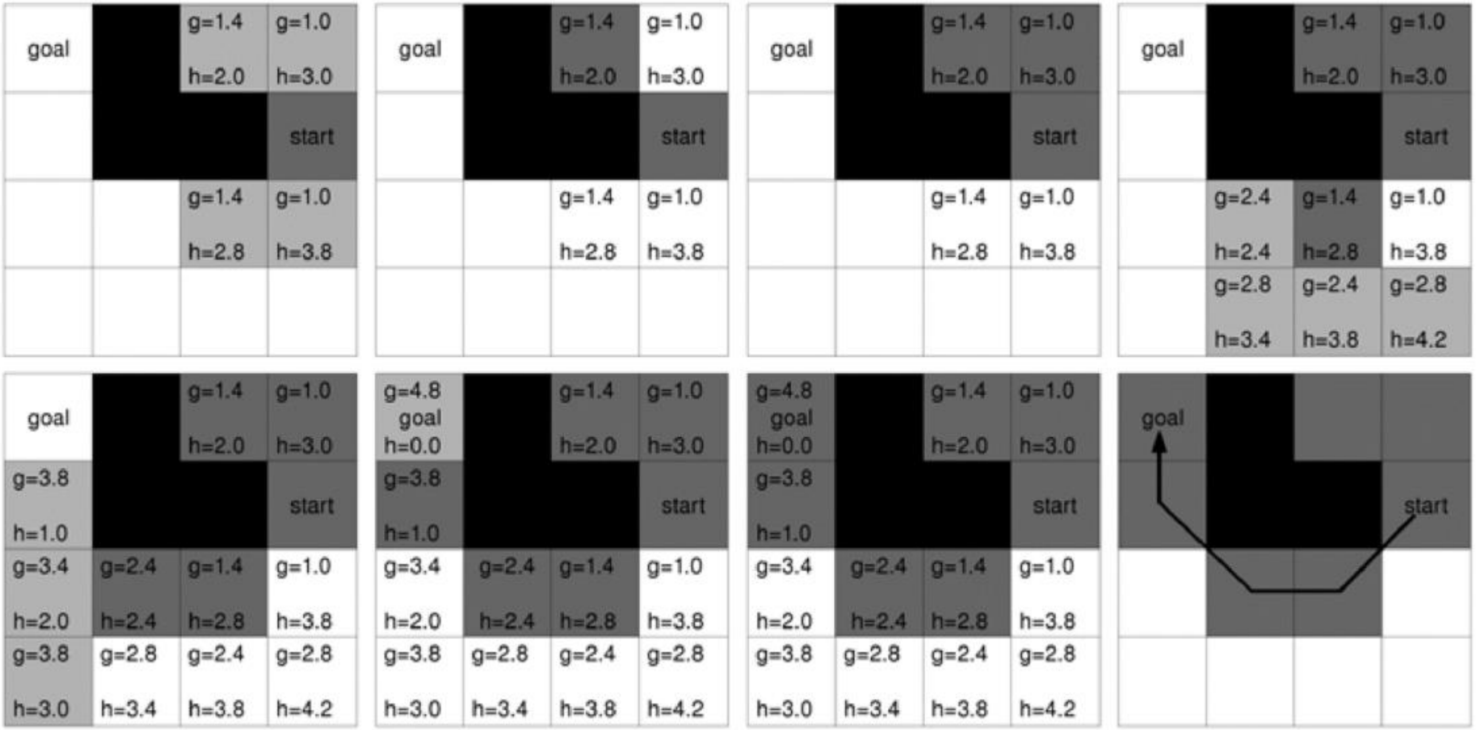
A* Algorithm

Useful with occupancy grid maps in which each cell is associated with a measure of occupancy.



A* Algorithm

- Similar to Dijkstra's algorithm, except that it uses the occupancy measure $h(n)$.
- The cost based on which the cells are evaluated is not just the distance $g(n)$, but also involves the occupancy measure, i.e. $f(n) = g(n) + \varepsilon . h(n)$.



Please watch after class

- Breadth First Search: https://www.youtube.com/watch?v=oDqjPvD54Ss
- Depth First Search: https://www.youtube.com/watch?v=7fujbpJ0LB4
- Dijkstra's Algorithm: https://www.youtube.com/watch?v=GazC3A4OQTE
- A* Algorithm: https://www.youtube.com/watch?v=ySN5Wnu88nE

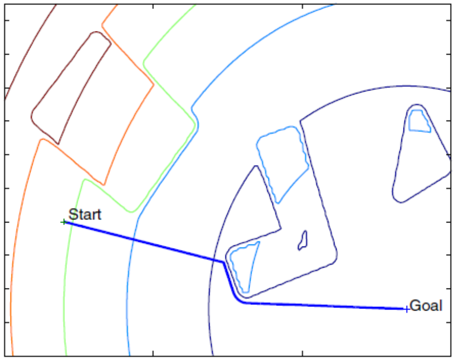
Potential Field Planning method

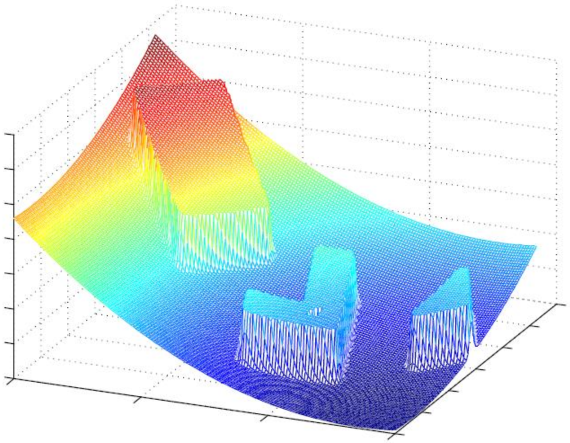
- Robot is treated as a point under the influence of an artificial potential field.
- Operates in the continuum.

➢ Generated robot movement is similar to a ball rolling down the hill.

➢ Goal generates attractive force

➢ Obstacle are repulsive forces.





Potential Field Planning method

- Generation of potential field function $U(x, y)$
  - attracting (goal) and repulsing (obstacle) fields.
  - summing up the fields.
  - functions must be differentiable.
- Generate artificial force field $F(x, y)$
  - $F(x, y) = -\nabla U = -\begin{bmatrix} \frac{\partial U}{\partial x} & \frac{\partial U}{\partial y} \end{bmatrix}^\top$
- The force is composed of:
  - attractive forces (depending on distance from the goal); and
  - repulsive forces (depending on distances from obstacles).

By using a potential field for environment presentation the robot can reach the goal point simply by following the negative gradient of the potential field.

RMIT UNIVERSITY

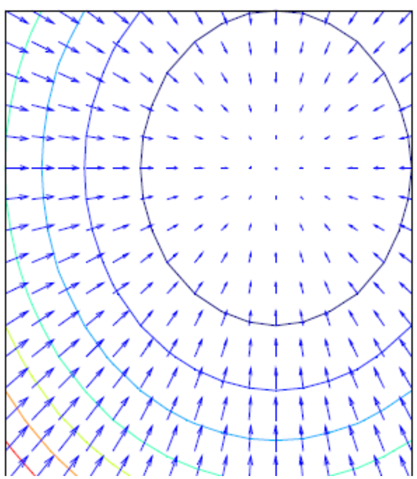
Potential Field Planning method: Attractive Force

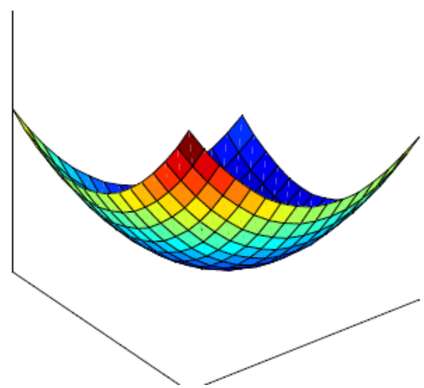
- Robot's location: $[x \ \ y]^{\mathsf{T}}$, Goal point: $[x_G \ \ y_G]^{\mathsf{T}}$
- Attractive potential field:

$$U_a(x,y) = \frac{1}{2}\, k_a[(x - x_G)^2 + (y - y_G)^2]$$

  - A parapolic function of the Euclidean distance from the goal point.

- Attractive force: $F_a(x,y) = \begin{bmatrix} -\dfrac{\partial U_a}{\partial x} \\ -\dfrac{\partial U_a}{\partial y} \end{bmatrix} = \begin{bmatrix} k_a(x_G - x) \\ k_a(y_G - y) \end{bmatrix}$





Potential Field Planning method: Repulsive Force

- Consider an obstacle whose closest point to the robot is $[x_o\ y_o]^\mathsf{T}$.
- Distance from robot to the obstacle = $\rho(x,y) = \sqrt{(x-x_o)^2+(y-y_o)^2}$
- A proper potential field should:
  - increase abruptly when ρ becomes small, otherwise
  - becomes negligible or zero.
- An example of a proper potential function:

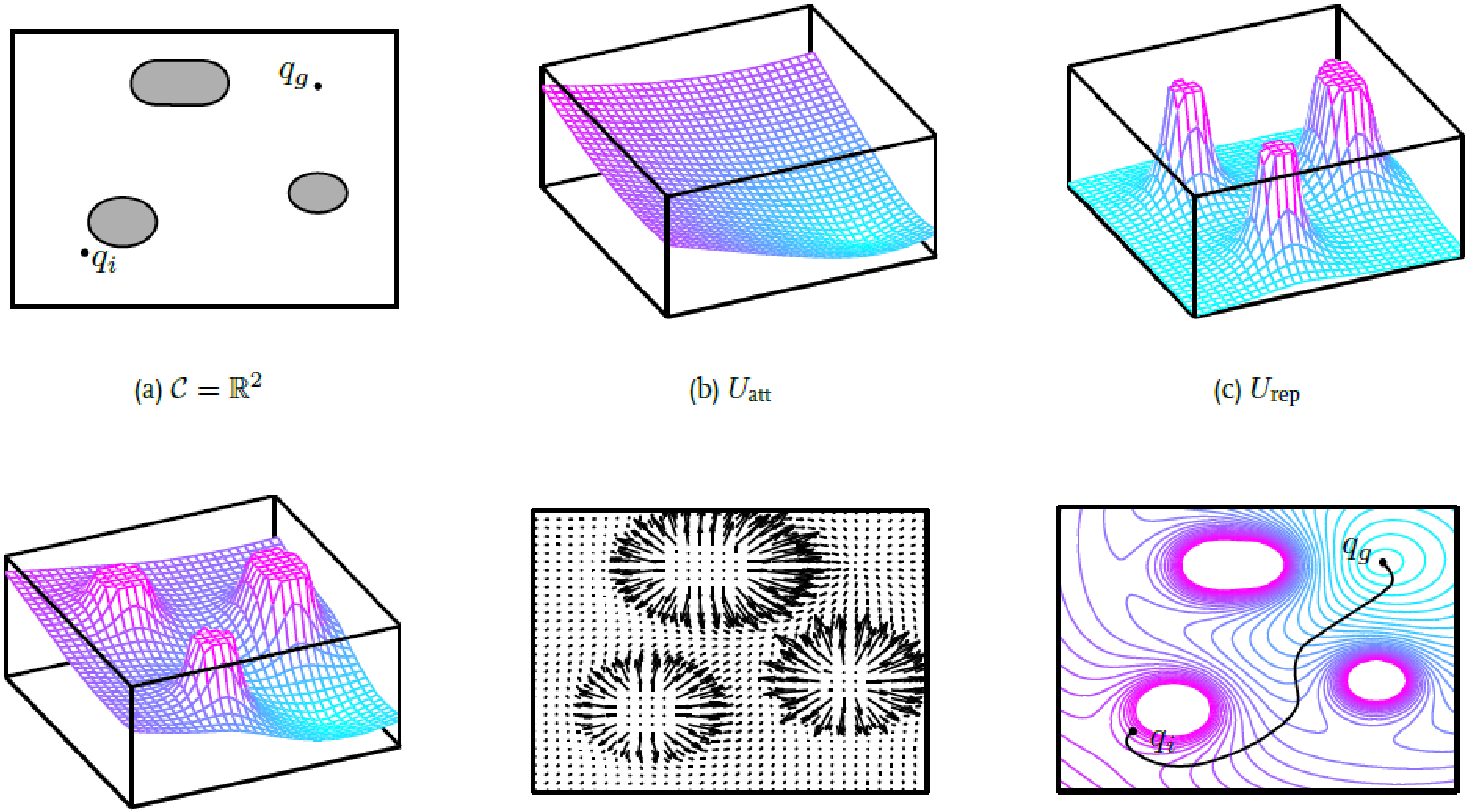
$$U_r(x,y) = \begin{cases} \frac{1}{2} k_r \left(\frac{1}{\rho(x,y)} - \frac{1}{\rho_0}\right)^2 & if\ \ \rho(x,y) \le \rho_0 \\ 0 & otherwise \end{cases}$$

- The resulting repulsive force (just for $\rho(x,y) \le \rho_0$):

$$F_r(x,y) = \begin{bmatrix} -\partial U_r/\partial x \\ -\partial U_r/\partial y \end{bmatrix} = k_r \left(\frac{1}{\rho(x,y)} - \frac{1}{\rho_0}\right) \times \frac{1}{\rho(x,y)^3} \begin{bmatrix} x - x_o \\ y - y_o \end{bmatrix}$$

- If $\rho(x,y) > \rho_0$ then the repulsive force is zero.

Potential Field Planning method



(a) $\mathcal{C} = \mathbb{R}^2$ (b) U_{att} (c) U_{rep}

(d) U_{tot} (e) $\vec{F}(\mathcal{C})$ (f) $\Phi(\mathcal{C})$

Figure 4.9: For a $\mathcal{C} = \mathbb{R}^2$ with three $\mathcal{C}$-obstacles (a), the attractive potential U_{att} towards the goal (b) and repulsive potential U_{rep} to obstacles (c), sum up to a total potential U_{tot} field as shown in Figure d. The gradient vector orientations over the field $\vec{F}(\mathcal{C})$ are displayed in Figure e. The equipotential contours of the total potential $\Phi(\mathcal{C})$ and a path that is generated by following the negative gradient of the combined potentials is depicted in Figure f.

Motion Planning for Mobile Robots - A Guide by S.A.M. Coenen

Thank you for your attendance :D

RMIT UNIVERSITY

Reference

- *MATHWORKS official tutorial.*
- *Lecture slides from RMIT Melbourne Autonomous System course, delivered by Prof Reza Hoseinnezhad.*
- *Introduction to Autonomous Mobile Robots by Roland Siegwart and Illah R. Nourbakhsh.*

RMIT UNIVERSITY

Copyright Claim

RMIT UNIVERSITY