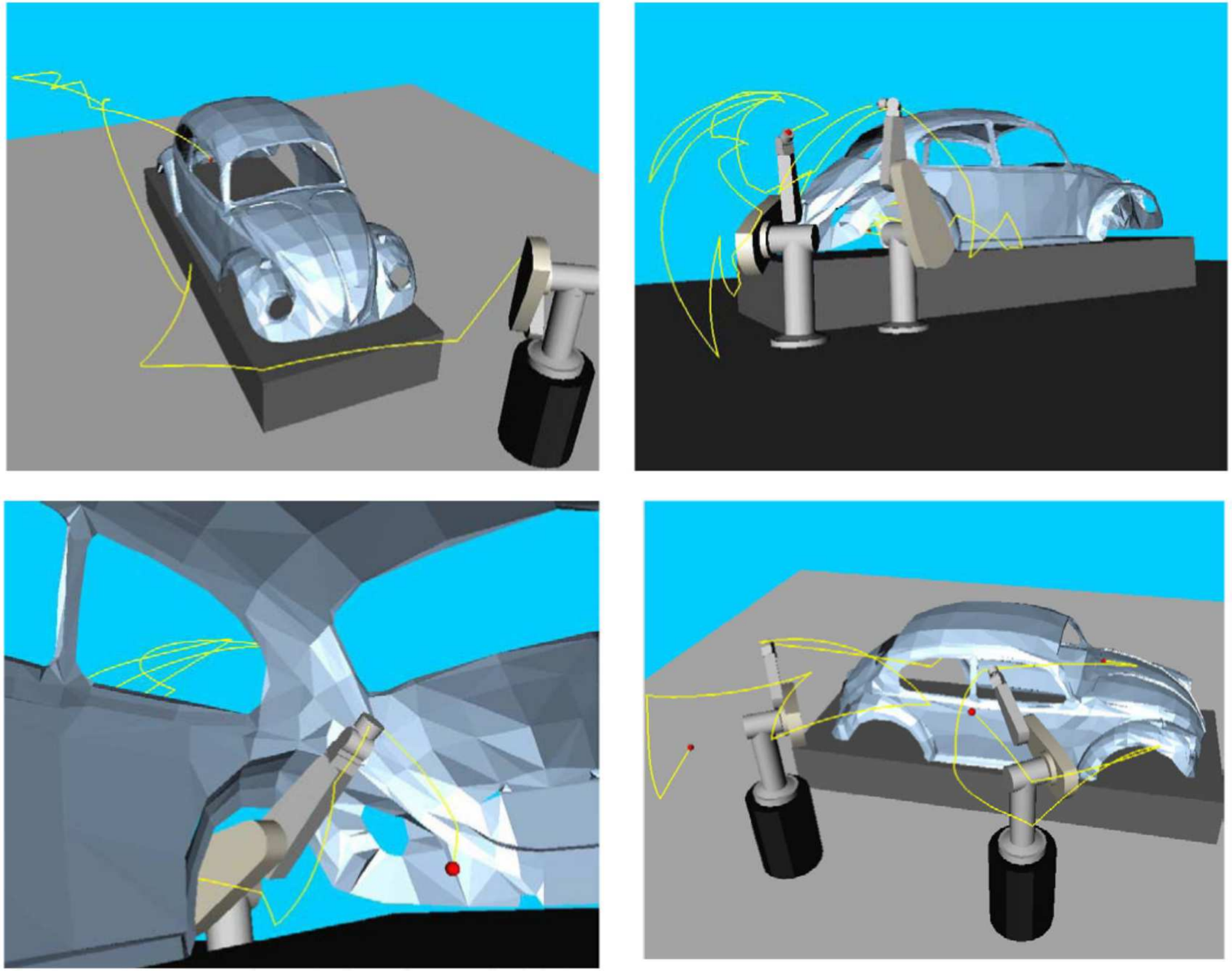
Path Planning using Ant Colony Optimisation

PhD of Mohd Murtadha Mohamad

(Universiti Teknologi Malaysia)

Path Planning



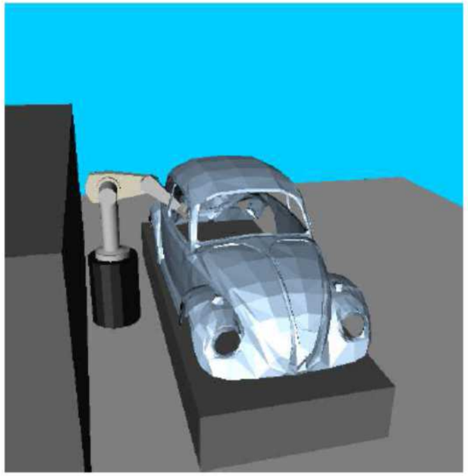
Configuration Space (C-Space)

- For the purposes of planning it is important to define a state space
- The state space for motion planning is a set of possible transformations that could be applied to a robot
- C-space is an abstraction of all the complicated kinematics models and transformations that we devised earlier
- It enables us to convert them into the general problem of computing a path from a start state to a goal state in a state space

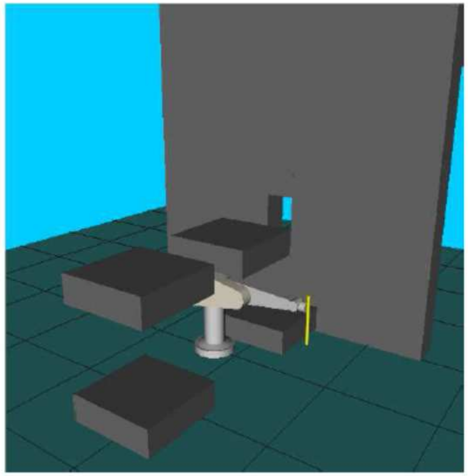
Probabilistic Road Maps (PRMs)

- A PRM is a discrete representation of a continuous configuration space (C-Space) generated by randomly sampling the free configurations of the C-space and connecting the points into a graph
- PRM constructs a roadmap of paths between points in C-Space
- A roadmap is a pre-computed undirected graph covering the entire free space
- Path planning is then applied by connecting the initial and goal configurations to points in the roadmap and then searching the roadmap for the path between these points

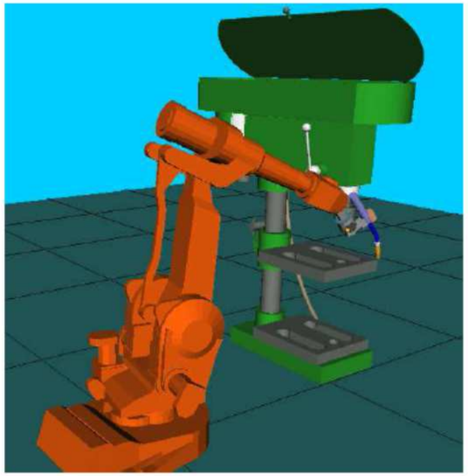
Possible Collision Examples



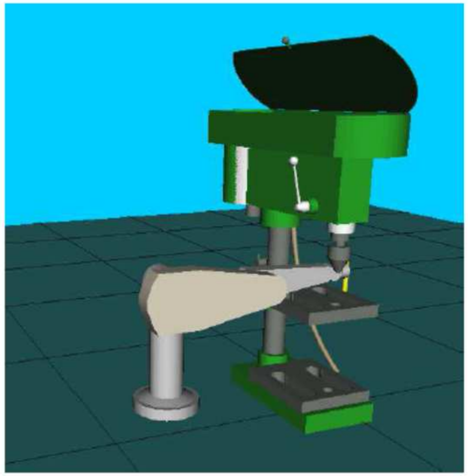
(a)



(b)



(c)

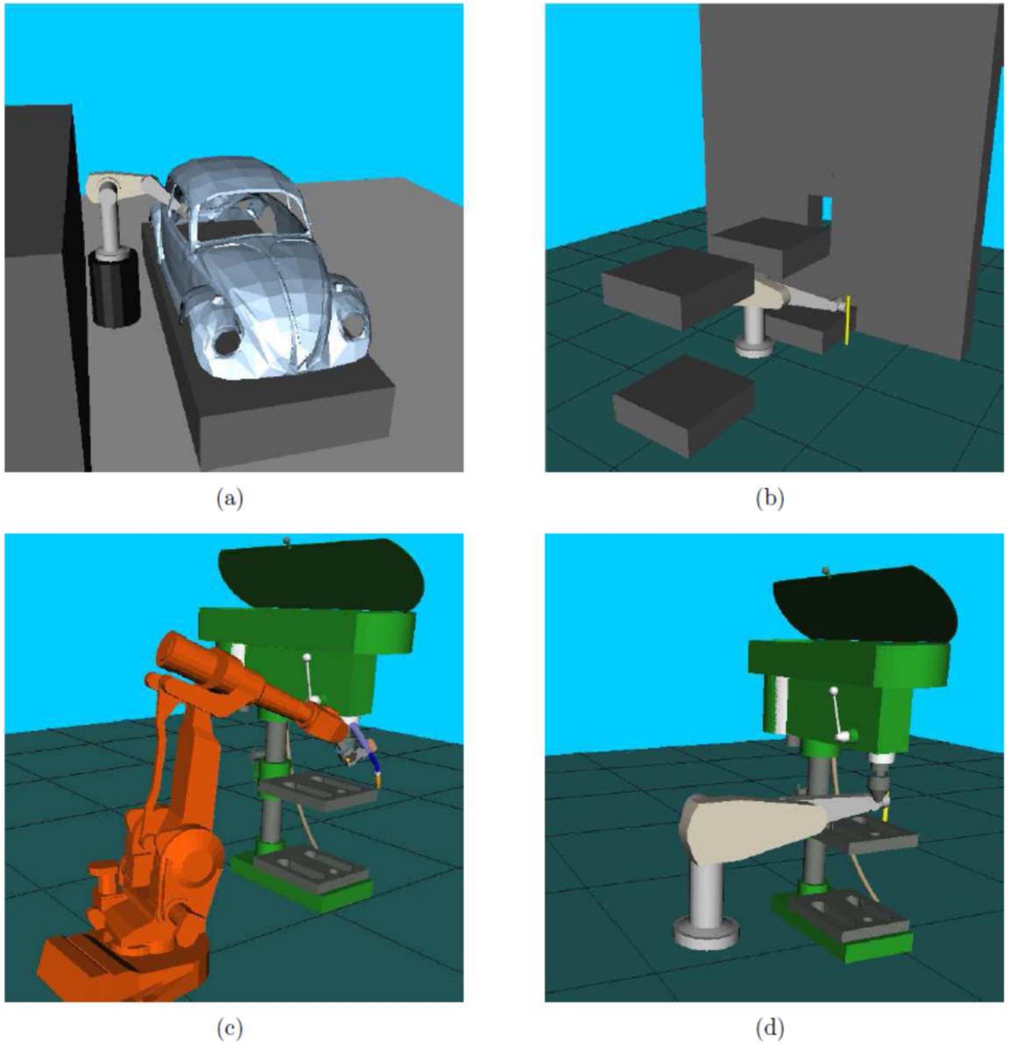


(d)

Collision Checking

- All nodes and paths are assumed to be collision-free during the pre-computation phase
- Once the shortest path has been found the nodes and path are checked for collisions
- If collisions have occurred the corresponding nodes and edges are removed
- The planner either finds the new shortest path or updates the roadmap and then searches for the new shortest path

Possible Collision Examples



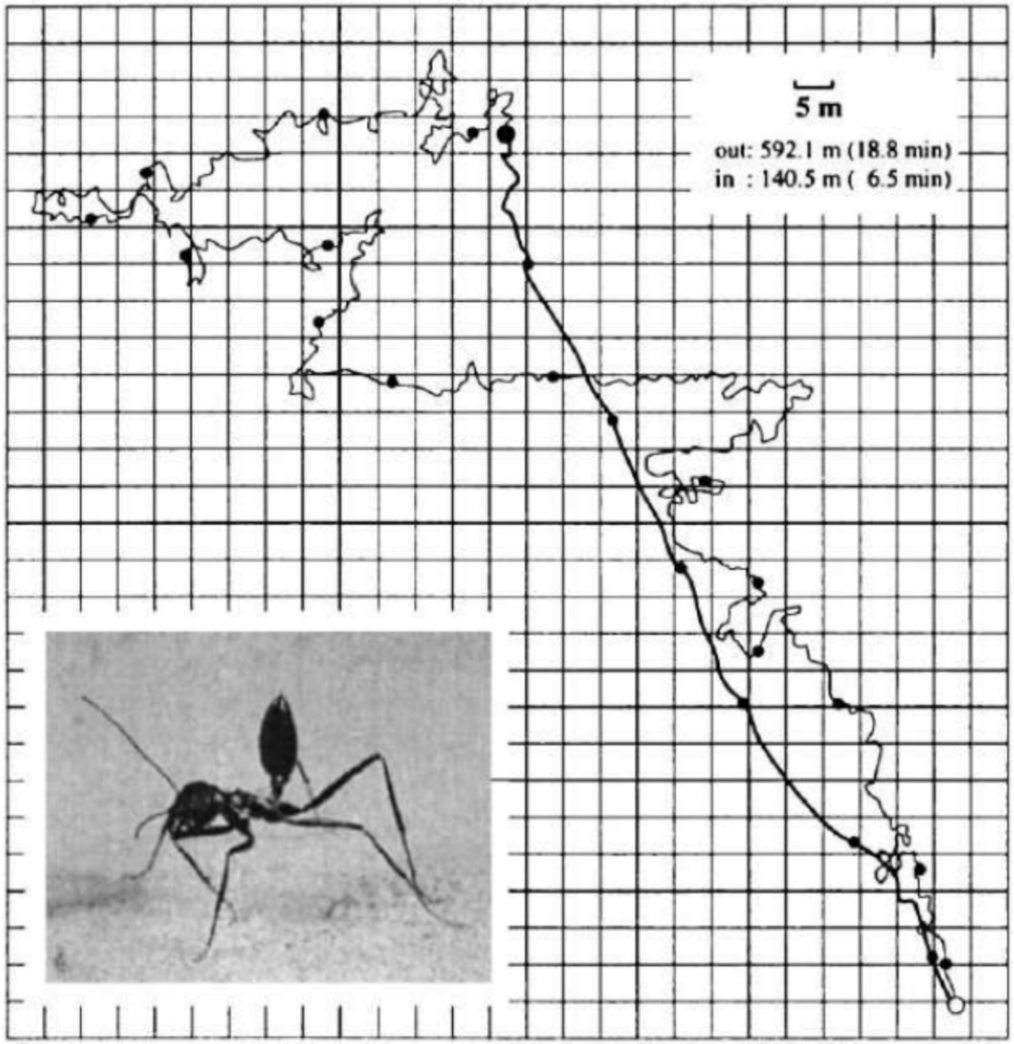
Single-query Bi-directional path planning with Lazy collision-checking (SBL-PRM)

- Instead of pre-computing a roadmap covering the entire free space SBL-PRM uses the two input query configurations (Start and Goal) to explore as little of the C-Space as possible
- The bi-directional approach explores the robot's free space by building a roadmap comprising two trees rooted at the query configurations
- Lazy collision checking means that the collision tests are delayed along the edges of the roadmap until they are absolutely needed

Ant Colony Optimisation (ACO)

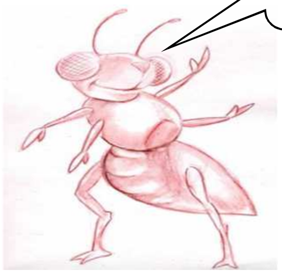
- ACO is a swarm intelligence approach to solving optimisation problems
- It is based on the behaviour of ants when finding optimal paths to food sources
- It has been successfully used to optimise the Travelling Salesman Problem (TSP)
- We can combine the ant behaviour of ACO with the roadmap characteristics of PRM to find paths through C-Space

Foraging Ant Motion Planning for Articulated Robots

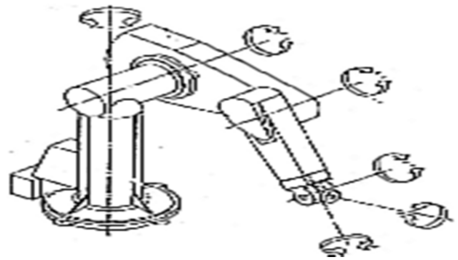


Ant Colony Optimisation in Robot Motion Planning

ACO can be used to optimise robot motion planning.



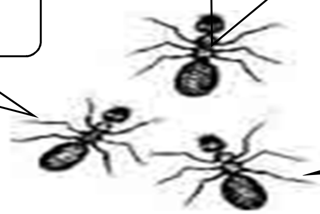
ACO permits a reduction in the number of intermediate configurations which need to be assessed, thus improving efficiency of planning and eventual execution.



SBL-PRM multi-goal path planning is modelled in hierarchical form. A roadmap (RM) is built from nodes and edges in C-space. The SBL algorithm is applied to find a path between the initial and goal configurations. The multi-goal problem is tackled using the TSP approach.

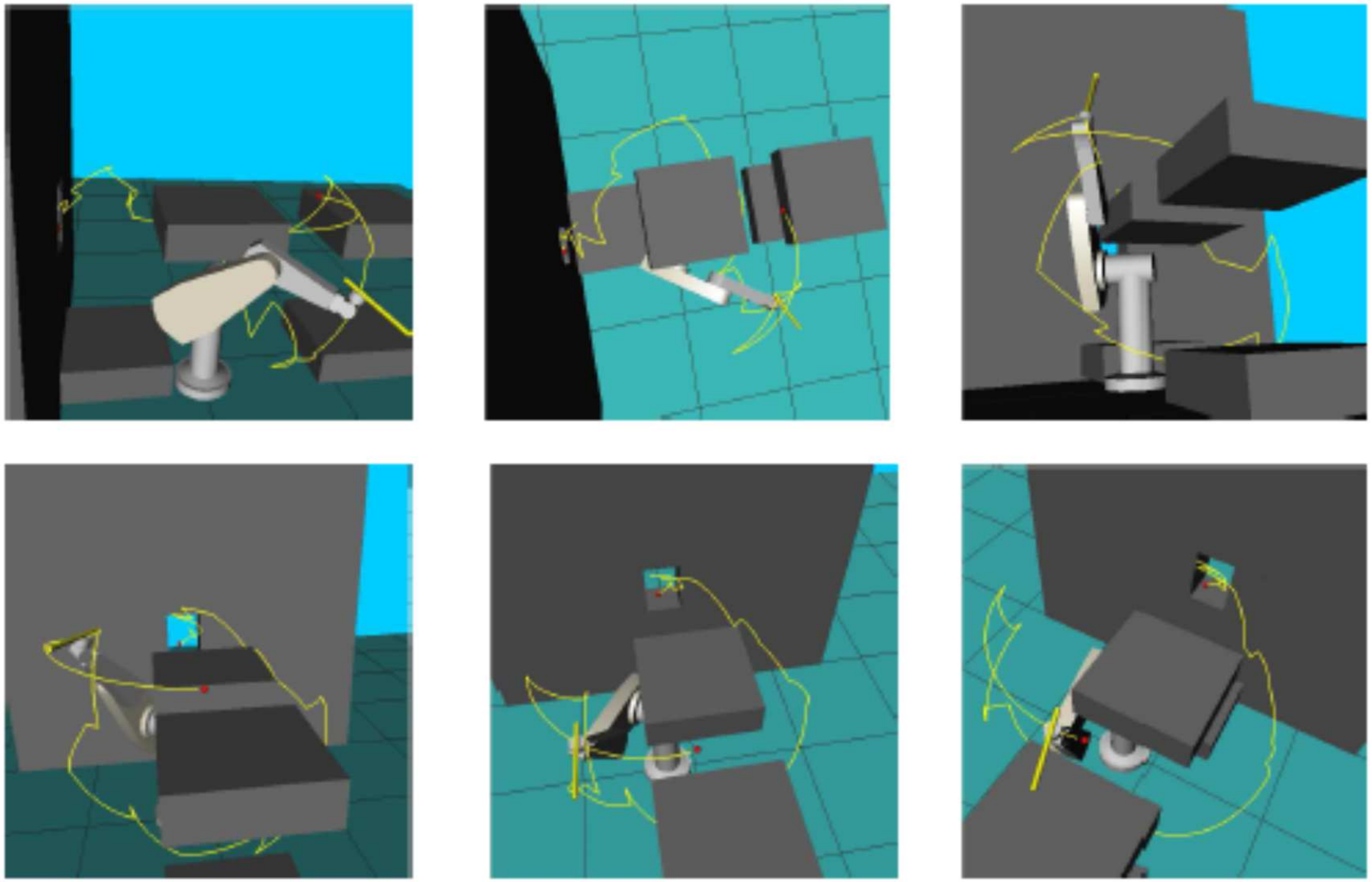
Replace RM with trail-ants (T-ANTS).

Replace SBL with foraging-ants (F-ANTS).



Optimise top level by applying ACO to TSP.

Snapshots along a Path for Example (b)



ACO Planning System