Multi-Robot Planning

Jan Faigl

Department of Computer Science
Faculty of Electrical Engineering
Czech Technical University in Prague

Lecture 11

B4M36UIR – Artificial Intelligence in Robotics

Overview of the Lecture

- Part 1 – Multi-Robot Systems (MRS)
- Part 2 – Multi-Robot Planning
- Part 3 – MRS domains and tasks
- Part 4 – Swarm and modular robots

Part I

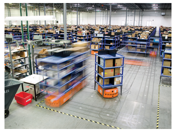
Part 1 – Multi-Robot Systems (MRS)

Multi-Robot Systems (MRS) - Intro

- Formed by individual robots (agents) capable of perceiving the environment by their sensors, communicating with other agents, and changing the environment by their actions.

(A. Farinelli et al., Trans. on Syst. Man and Cyber., 2004)

- Challenges in MRS scenarios:
  - **Path/Motion planning**
    How to find path for multiple robots?
  - **Collision avoidance**
    How to find obstacle-free path?
  - Dynamic obstacles in the environment
    How to execute the plans deadlock-free?
  - Limited communication radius
  - Physical limitations of the robot
  - Reliability of (centralized) MRS
  - And others ...



Kiva Systems (Amazon warehouse)

A busy traffic intersection

Multi-Robot vs. Single-Robot

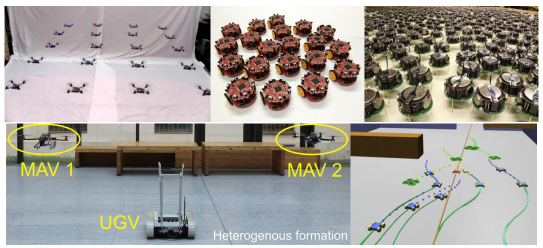
- Pros.
  - **Parallel task execution** - actions can be done in parallel
  - **Improved robustness** - failure of an individual should not affect the whole team
  - **Wider range of applications** - some tasks cannot be solved by a single robot or some specialization of the robot is needed (heterogeneous teams)
- Cons.
  - **Interference** - the robots may interfere and disturb each other, there is an uncertainty about intentions of other robots
  - **Communication** - there is a limited communication bandwidth between robots
  - **Maintenance** - multiple robots are harder to maintain

Multi-Robot Systems - taxonomy (part 1)

- Cooperation
  - **Cooperative** - robots cooperate to achieve joint goal
  - **Competitive** - robots compete to best fulfill their own self-interest, i.e., robots can cooperate or form coalitions if that is in their own self-interest
- Communication
  - **Implicit** - the information is transmitted through the environment
  - **Explicit** - the information is transmitted directly between robots
- Organization
  - **Centralized** - global coordination and planning
  - **Hierarchical** - army model - hierarchy of leaders
  - **Decentralized** - local coordination, the global pattern of behavior is an emergent property

Multi-Robot Systems - taxonomy (part 2)

- Team composition
  - **homogeneous** - all robots have identical hardware and software
  - **heterogeneous** - robots differ either in sensory-actuator capabilities or in the software control procedures
  - **swarms** - large number of usually homogeneous robots, local control, little to no explicit communication



Part II

Part 2 – Multi-Robot Planning

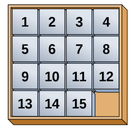
Multi-Robot Path Planning on Discrete Graphs (MPP)

MPP problem definition:

- $G = (V, E)$ is a connected undirected simple graph where
  - $V = v_i$ is a the vertex set,
  - $E = \{(v_i, v_j)\}$ is the edge set.
- $\mathcal{R} = R_1, \cdots, R_m$ is a set of m robots.
- Robots moves at discrete time steps.
- Each robot R_i is associated with an *start* and *goal* configuration (q_i^s, q_i^g)
- MPP can be transformed to **boolean satisfiability problem** (3SAT).
- Finding optimal solution is NP-complete

(J. Yu, "Optimal Multi-Robot Path Planning on Graphs: Structure and Computational Complexity", Robotics and Automation Letters, 2016)

- Pebble motion problems - more "pebbles" can occupy one vertex



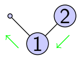
Example of MPP - 15 puzzle

Multi-Robot Path Planning on Discrete Graphs (MPP)

9-puzzle example (J. Yu, Robotics and Automation Letters, 2016)

Initial configurations — Desired goal configurations

Possible moves of two robots:



Impossible moves of two robots:

Multi-Robot Path Planning on Discrete Graphs (MPP)

- Advantages of MPP on a discrete graph
  - Simple formulation
- Limitations of MPP on a discrete graph
  - A unit speed is assumed (one edge per time step)
  - A robot body is not considered
  - Some problems are hard to discretize
  - Even relatively small MPP instance can be computationally intractable

Multi-Robot Motion Planning - part 1

- **Fundamental problem in MRS**
- Formal notation:
  - There is a set of $m > 1$ robots $\mathcal{R} = R_1, \cdots, R_m$,
  - each operating in a configuration space $\mathcal{C}_i$, for $1 \le i \le m$,
  - let $\mathcal{C}_i^f \in \mathcal{C}_i$ be each robot's free space,
  - and $\mathcal{C}_i^o = \mathcal{C}_i \setminus \mathcal{C}_i^f$ be each robot's occupied space.
  - The **composite configuration space** $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_m$ is Cartesian product of each robot's configuration space.
  - A composite configuration $Q = (q_1, \cdots, q_m) \in \mathcal{C}$ is m-tuple of robot configurations.
  - For two robots $R_i, R_j, i \neq j$, let $I_i^j(q_j) \in \mathcal{C}_i$ be the set of configurations of robot R_i that lead into collision with robot R_j at configuration q_j.
  - Then the **composite free space** is defined as $\mathcal{C}^f \in \mathcal{C}$ consists of configurations $Q = (q_1, \cdots, q_m)$ subject to:
    - $q_i \in \mathcal{C}_i^f$ for every $1 \le i \le m$,
    - $q_i \notin I_i^j(q_j), q_j \notin I_j^i(q_i)$ for every $1 \le i \le j \le m$.
  - The **composite obstacle space** is then defined as $\mathcal{C}^o = \mathcal{C} \setminus \mathcal{C}^f$.

(S. M. LaValle, "Planning Algorithms", Cambridge University Press, 2006)

Multi-Robot Motion Planning - part 2

- **The problem**:
  - Set of **Start configurations** $S = (q_1^s, \cdots, q_m^s) \in \mathcal{C}^f$
  - Set of **Goal configurations** $G = (q_1^g, \cdots, q_m^g) \in \mathcal{C}^f$
  - Find a continuous trajectory $\tau_i : [0, 1]$ for each robot R_i, for $1 \le i \le m$, without collisions with obstacles and other robots, minimizing a cost function c, such that: $\tau_i(0) = q_i^s$ and $\tau_i(1) = q_i^g$
  - The selection of a **cost function** c is subject to optimization criteria, e.g.:

1. **Min Total Time**
   minimize $\sum_{i=1}^m t_i$
2. **Min Makespan**
   minimize $\max_{1 \le i \le m} t_i$
3. **Min Total Distance**
   minimize $\sum_{i=1}^m l_i$
4. **Min Max Distance**
   minimize $\max_{1 \le i \le m} l_i$

where t_i and l_i are the trajectory τ_i duration and length, respectively

Multi-Robot Motion Planning - Approaches

- **Centralized planning** – planning directly in the composite configuration space.
  - **Coupled planning** – direct planning in the composite configuration space
  - **Assembly planning** – determining a sequence of motions that assembles the parts
  - **Decoupled planning** – planning of each trajectory separately (Prioritized planning, Pairwise cooperation)
- **Decentralized planning** – each robot plans its own trajectories and solves collision situations as they appear

Centralized Planning - Coupled Planning

Planning directly in the composite configuration space
$\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_m$

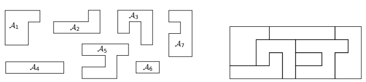
- Utilizes standard path planning methods, such as random-sampling based approaches or grid-based planners
- m robots with d DOFs are assumed as a single robot with $m \cdot d$ DOFs
- **Complete**, i.e., it always find a solution (if exists)
- **Complexity** $\approx \exp(m \cdot d)$
- Becomes computationally intractable even for small number of robots.
- Finding optimal solution is NP-complete

  (J. Yu, "Optimal Multi-Robot Path Planning on Graphs: Structure and Computational Complexity", Robotics and Automation Letters, 2016)
- Note, for **unlabeled case**, when there is no explicit mapping $S \to G$, i.e., you do not care which robot is on particular goal, the complexity is polynomial

  (M.Turpin et al., "Goal assignment and trajectory planning for large teams of interchangeable robots", Autonomous Robots, 2014)

Centralized Planning - Assembly Planning

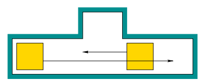
- The task is to assembly final product from multiple parts.
- A single part is moved at a time.
- Result of the planning is a sequence of paths for individual parts.
- **Planning is started from the final configurations backwards.**



Courtesy of (S. M. LaValle, 2006)

Centralized Planning - Decoupled Planning

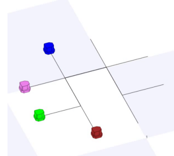
- Planning for each robot $R_i, 1 \le i \le m$ with d DOFs separately
- Coordination of particular plans is done later
- **Not complete, not optimal**
- **Complexity** $\approx m \exp(d)$



Courtesy of (S. M. LaValle, 2006)

Methods of plan coordination:

- **Prioritized planning**
  - Each robot is assigned with a priority
  - Plans are constructed according to priorities
  - Cannot prevent deadlocks
  - (https://www.youtube.com/watch?v=dFm-JJhyuv0)
- **Pairwise cooperation**
  - Planning in **coordination space** - Robot configuration is considered one-dimensional (position on a trajectory in time)
  - Coordinations are incrementally solved for all the robots



Decentralized planning

- Each robot plans its own trajectory and resolves possible collision with other vehicles
- Both implicit and explicit communication types can be considered
- Collision situations are resolved as they appear

Collision resolution methods

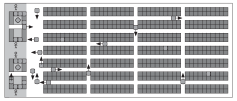
- Based on the priority - earliest collision is solved first
- Based on the shortest trajectory prolongation

Kiva robots in Amazon warehouses

- Amazon acquired Kiva Systems for $775 million in 2012
- Grid-based planning problem
- Restricted areas for human operators
- Classical A* planning
- Task allocation problem
- About 30,000 robots in 2016
- Open-source example

(https://github.com/oliehoek/kiva)

Kiva Systems (Amazon warehouse)



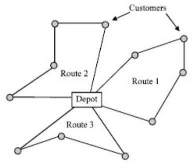
An example of Amazon warehouse layout

((P. R. Wurman et al., "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses", AI Magazine , 2008)

Vehicle routing problem (VRP)

Special case of a graph-based Multi-Robot planning with multiple goals.

- First introduced by Dantzig and Ramser in 1959.
- Generalization of the classical TSP with multiple vehicles.
- Problem definition
  - n customers.
  - m vehicles with maximal capacities a_k.
  - A single depot (both initial and final positions of vehicles).
  - Costs $d_{i,j}$ between the given cities.
  - Find a set of routes with a minimal total cost.



(Souce: K. Ghoseiri et al., 2009)

Vehicle routing problem

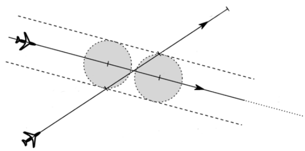
- **Possible formulations**
  - Vehicle Routing Problems with **Pickup and Delivery** (VRPPD) - pickup and delivery locations are defined
  - Vehicle routing problem with **LIFO** - similar to VRPPD but with stack loading
  - Vehicle Routing Problem with **Time Windows** (VRPTW) - times windows of visits are limited
  - **Capacitated** Vehicle Routing Problem (CVRP) - each vehicle has its capacity (Uncapacitated VRP is also called mTSP)
  - **Open** Vehicle Routing Problem (OVRP) - return to depot is not required

Plan execution

- **Reactive control**
  - Agent follows the shortest path to the target
  - Collision situations are resolved as they appear
  - Cannot prevent deadlocks or infinite loops
- **Deliberative control**
  - Planning of coordinated trajectories for all the robots
  - Agents execute the path in an incremental way
  - Guarantees deadlock free execution (if plans are executed precisely)

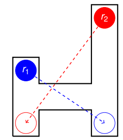
Reactive control

- Each agent plans its own path
- Each agent detects possible collisions
- Resolution of the conflicts:
  - set of evasion maneuvers,
  - adapting speed and heading,
  - complete re-planning



Plan execution environments with external disturbances

- Plan generated for each of m robot (agents) with d-DOF globally by existing approaches
- External disturbance such as
  - human operator
  - hardware failure
  - or any other unknown obstacle
- Resolution of the conflicts:
  - **ALLSTOP** - stop all robots, deadlock-free but ineffective
  - **reactive** - solve collision situations reactively, cannot avoid dead-locks
  - **RMTRACK** - execute plans according to coordination diagram, i.e., do not change order at crossings



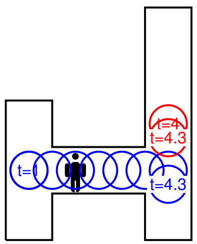
(M.Čáp, IROS, 2016)

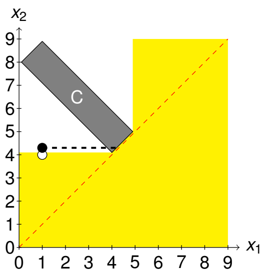
(https://www.youtube.com/watch?v=29YRLJpB90Q)

RMTRACK - Plan execution in coordination space

(M.Čáp, IROS, 2016)

Coordination space shows mutual collisions of trajectories τ_i and τ_j with respect to the time scale





Part III

Part 3 – MRS domains and tasks

MRS domains and tasks

- Data collection planning (https://www.youtube.com/watch?v=5MPSAReNzJU)
- Exploration (https://www.youtube.com/watch?v=tqMcK5YzwZc)
- Inspection
- Coverage
- Monitoring
- Pickup and delivery
- Pursuit evasion
- Cooperative transportation
- Multi-robot sensor fusion

Part IV

Part 4 – Swarm and modular robots

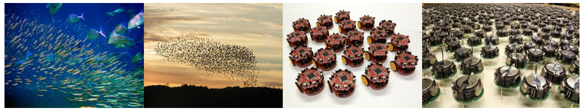
Swarm robotics

An approach to **coordination** of (usually a large number of) robots in a **distributed** and **decentralized** way. A plain set of rules at individual level can produce a large set of complex behaviors at the swarm level that emerges from interactions between the robots and interactions of robots with the environment.

(Y. Tan, Z. Zheng, Defense Technology, 2013)

- Nature inspired, e.g., social insects, fish, birds, herding mammals
- Properties
  - **Homogenity** - agents in a swarm are homogeneous robots, as such, they are assumed to be interchangeable
  - **Locality** - agents can observe only part of the system within a certain range. Decisions depend on current neighborhood.
  - Little to no explicit communication - swarms in nature are decentralized

(S. Jha et al., Anim. Behav., 2006)



Swarm robotics - behavior model

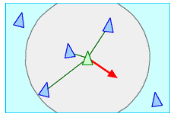
How to describe the control policies in swarms?

- A distributed behavioral model - **boids**

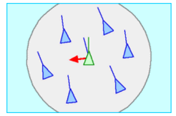
(C.Reynolds, SIGGRAPH, 1987)

- Introduces three basic steering maneuvers based on local neighbors (flockmates)

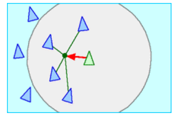
Separation - steer to avoid local flockmates



Alignment - steer towards an average heading of local flockmates



Cohesion - steer to the average position of the flockmates



Further complex behaviors can be developed, e.g., avoidance, following, aggregation, dispersion, homing

Swarm robotics - applications

- **Collective Movement** - how can an uncoordinated group of robots move from one place to another

(M.Saska et al., ICRA, 2014)

- **Distributed sensing** - swarms are very effective in **Source search** missions

(J. E. Hurtado et al., JIRS, 2004)

- **Cooperative transportation**

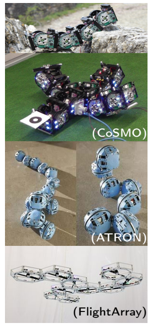
(C. R. Kube et al., RAS, 2000)

- **Collective mapping** - e.g. area coverage, shoveling

(M.Saska et al., JIRS, 2014)

Modular robots

- Composed of elementary mechatronic modules that can assemble to form body of various shapes
- Pros.
  - Adaptability to various operation conditions
  - Failure recovery by ejecting or replacing broken modules
- Cons.
  - Complicated mechatronic design
  - Complicated development of locomotion strategies
- Locomotion control principles
  - **Self-reconfiguration** - repeatedly disconnecting and re-connecting modules
  - **Joint-controlled locomotion** - controlling individual limbs of the robot



Modular robots - Joint-controlled locomotion

How to develop new locomotion rules for a robot with variable morphology?

1. Each module is an individual entity - MPP
2. The whole robot is an individual entity - Planning with motion primitives - require synthesis of new gaits for each topology
   - Often used - CPG controllers developed by genetic algorithms (GA)
   - Leads to high-dimensional parameter optimization - crucial role of cost function
     - Greedy optimization - early iterations of GA does not provide ability to solve the problem which leads to a blind random search

       (H.Lipson et al., Nature, 2000)
     - Ranking quality and novelty of found solutions - low-performing solutions may help in solving other task (crippling walking robot)

       (A.Cully et al., Evolutionary Computation, 2016)

       (https://www.youtube.com/watch?v=2aTIL_c-qwA)
     - Random sampling with CPGs as motion primitives - combination of motion primitives may lead to feasible solutions

       (V.Vonásek et al., SSCI, 2016)

       (https://www.youtube.com/watch?v=4KNDk2jjUGs)

Summary of the Lecture

Topics Discussed

- MRS systems and their taxonomy
- Multi-robot path planning
- Multi-robot motion planning
  - Centralized approaches (Coupled, Assembly, Decoupled)
  - Decentralized approaches
- Vehicle routing problem
- Swarm robotics
- Modular robots
- Next: Game Theory in Robotics

Thank you for your attention!