Motion Planning in Multi-Robot Systems

Kostas E. Bekris
Department of Computer Science
Rutgers University
07/16/2015
• It is difficult to provide a comprehensive coverage of all motion planning methods for multi-robot systems

• An effort was made to cover foundational work in the case of centralized solutions

• For decentralized solutions, the presentation highlights methods that the author has utilized in his research

• But the version of the presentation on the TC’s website can potentially be a live document that gets updated given your feedback
  – So, if you believe that a certain line of work should be highlighted here please contact Kostas Bekris (kostas.bekris @ cs.rutgers.edu)
Key question:

- **What information does an approach access?**
  - Global: Centralized approaches
  - Local: Decentralized approaches
Important In & Beyond Robotics

Multiple Direct Applications (including centralized methods)

- Warehouse management
- Transportation applications
- Controlling teams of robots in structured environments
- Digital entertainment
- Product assembly
- Combinatorial puzzles and pure scientific curiosity
Key questions for centralized approaches:

- What is the space that the method searches over for a solution?
  - Composite state space of all robots: Coupled approaches
  - Individual robot conf. space and coordination: Decoupled approaches

- What kind of guarantees can be provided?
  - Safety, Completeness, Optimality
Decentralized Approaches

Key questions for decentralized approaches:

- How does a local method access information about other robots?
  - Sensing or communication
  - Inference or shared information

- What kind of properties can be provided?
  - Collision Avoidance, Deadlock/Livelock Avoidance
Centralized – Coupled Planning
Key features of Coupled Approaches

• Consider the composite state space

\[ X = C^1 \times C^2 \times \ldots \times C^m \]

• Search can be performed with standard single-robot motion planning methods in X, e.g.,
  – combinatorial planners in low-dimensional cases,
  – sampling-based planners, [Svestka and Overmars, 1998]
  – optimization methods,
  – search (A*) etc.

• Then, it is possible to achieve the same properties as the algorithm achieves in the single-robot case

But… computational issues!
• **A complete algorithm** [Schwartz and Sharir, 1983]
  – Coordinating planar disk-robots: Exponential complexity in the number of robots

• **Exponential running time in some cases is unavoidable**
  – Rectangular robots in rectangular region: PSPACE-hard [Hopcroft, Schwartz and Sharir, 1984]
  – NP-hard for disc robots in a simple polygon workspace [Spirakis, Yap 1984]
  – For 2-3 robots, reduce number of DOFs but computing paths where the robots maintain contact [Aronov et al. 1999]
• A variation of the problem with interchangeable robots [Kloder and Hutchinson 2005]
  – Group of identical robots that need to reach a set of target positions

• Could it be that it is an easier challenge?
  – No, unit-square robots moving amidst polygonal obstacles and other variations are PSPACE-hard [Solovey, Halperin RSS 2015]

• Study of the disc robot case among polygonal obstacles:
  – Efficient solution when aiming for minimizing the longest robot path length [Turpin, Michael and Kumar 2013]
    • The space must be star-shaped surrounding each start and target position
  – This has been relaxed to less restrictive sparsity requirement [Adler et al. 2014]
  – Efficient algorithm also in the case of minimizing total path length [Solovey et al. RSS 2015]
Centralized – Decoupled Planning
Basic Idea

• First compute individual path for each robot
  – i.e., in the corresponding configuration space $C_i$

• Then consider plan interactions to produce a solution that is (hopefully) valid in the composite space $X$

• When successful…
  – They solve problems orders of magnitude faster than coupled alternatives!

• But when the pair-wise interactions are considered, the available choices are already constrained…
  – i.e., no completeness or optimality guarantees in the general case
Prioritized Planning

• Compute paths sequentially for different agents in order of priority
  – Higher-priority agents are considered moving obstacles for lower-priority one
  \[\text{Erdmann and Lozano-Perez, 1986}\]

• Choice of priorities has significant impact on solution quality \[\text{van den Berg and Overmars, 2005}\]

• Searching the space of priorities can improve performance \[\text{Bennewitz, Burgard, Thrun 2002}\]

Incremental methods:
• plan path for a robot, considering the paths of a subset of the other agents
• a plan-merging scheme coordinates actions to detect deadlocks
• when a circular dependency is detected, a couple planner is invoked

\[\text{Alami et al. 1995, Qutub et al. 1997}\]
Velocity Tuning

Two step approach:

- Fix paths for all agents and then in order of priority apply velocity tuning
  - i.e., select velocity for low priority agent along path so as to avoid collisions
  - treat high-priority agents as dynamic obstacles \([\text{Kant, Zucker } 1986]\)

Idea relates to coordination diagrams which were developed for dual-arm manipulation: \([\text{O’Donnell, Lozano-Perez } 1989]\)
\([\text{Simeon, Leroy, Laumond } 2002]\)

Extended to systems with more complex dynamics \([\text{Peng and Akell } 2005]\)
Example Use of Velocity Tuning

Scheduling Pick-and-Place Tasks for Dual-arm Manipulators using Incremental Search on Coordination Diagrams

HUMANOIDS 2015 Video Submission

Andrew Kimmel, Athanasios Krontiris, Kostas Bekris
Rutgers University
• More flexible solutions if the robots are not constrained on individual paths but on entire roadmaps [Ghrist, O’Kane and LaValle 2005]
  – Give rise to interesting coordination spaces (cube complexes)
  – Makes more sense to aim for Pareto optimal solutions

• Similar idea:
  – Try to compute multiple diverse paths first for each agent [Green, Kelly 2007] [Knepper, Mason 2009][Voss, Moll, Kavraki 2015]
  – Or make sure you are covering many different homotopic classes [Bhattacharya, Kumar, Likhachev 2010]
Centralized
Discrete Case and New Insights
Remove the complexity of reasoning about the geometry

- Employ a graph-based abstraction

The problem is studied in many different communities under different names:

- Multi-agent Planning
- Cooperative Path Finding
- Pebble Motion on a Graph
- Multi-agent Navigation

Finding optimal solutions is an NP-complete problem \cite{Ratner and Warmuth, 1986}
Fast but Incomplete Methods

- Computationally efficient.
- Decoupled framework.
- No guarantees for
  - Completeness.
  - Path Quality.

- Dynamic prioritization and windowed search
  [Silver 2005]

- Spatial abstraction with heuristic computation
  [Sturtevant and Buro 2006]

- Use of a flow network with replanning
  [Wang and Botea 2008]

- Smart direction maps that learns movements
  [Jansen and Sturtevant 2008]
Suboptimal but Complete Methods

- Still efficient: polynomial running time.
- They will find a solution if one exists.
- They do not provide optimal paths.

- Specific topologies
  [Peasgood et al. 2008][Surynek 2009]

- Slideable grid-based problems
  [Wang and Botea 2011]

- Complete on trees
  [Khorshid et al. 2011]

- “Push and Swap”: Polynomial-time solution on graphs with two empty vertices
  [Luna and Bekris 2011]

“Push and Swap” Software Package Available:
Scales up to Thousands of Agents
• Polynomial time feasibility test algorithm for graphs [Kornhauser et al. 1984][Roger and Helmert 2012]

• Linear time feasibility algorithm on trees [Auletta et al. 1999]

• Linear algorithm for graphs with two blanks [Goraly and Hassin 2010]
Evaluating Feasibility

Finding Suboptimal Paths

Linear Time!

Finding an Optimal Path

Cubic Time

NP-hard

Interesting Disparity

Finding an Optimal Path

NP-hard

Krontiris, Luna, Bekris SoCS ‘13

Yu, ‘13

Extension to simultaneous motion

[Yu, Rus, WAFR ‘14]
New Optimal Methods

- Provide path quality guarantees.
- Coupled framework - often A*-based.
- Great recent progress but... scalability conditional to the hardness of the problem

- Optimal decoupling
  [van den Berg et al. RSS 2009]

- Working on independent subproblems
  [Standley 2010, Standley and Korf 2011]

- Subdimensional expansion search space
  [Wagner and Choset 2011, 2013]

- Conflict-based Search
  [Sharon, Stern, Sturtevant 2012, 2015]

- Cast challenge to another NP-hard problem
  – Linear Programming [Yu, LaValle 2013]
  – Or other formal methods [Erdem et al. 2013, Surynek 2012]
Integrating sampling-based algorithms with pebble graph solvers to address continuous challenges [Solovey and Halperin WAFR 12]

We have recently transferred the idea in the context of rearranging multiple movable bodies with a manipulator [Krontiris, Shome, Dobson, Kimmel Bekris Humanoids 2014] [Krontiris, Bekris RSS 2015]
Rearranging Similar Objects With A Manipulator: a non-monotone benchmark


[Krontiris, Shome, Dobson, Kimmel Bekris Humanoids 2014]
Krontiris, Bekris RSS 2015
Multi-Arm Manipulation

[Koga, Latombe 1994]
[Cohen, Philips, Likhachev RSS 2014]
[Dobson, Bekris IROS 2015]

Planning handoffs and stable grasps
Decentralized Approaches
It is possible to employ reactive collision avoidance methods

- No need to employ communication

e.g. Reciprocal Velocity Obstacles

[van den Berg, Lin, Manocha ‘08]
Deconfliction for First-order Systems

Reciprocal Velocity Obstacles
[van den Berg, Lin, Manocha ‘08]

Extended to address team coherence constraints
[Kimmel, Bekris AAMAS ‘12]
• A prototypical motion coordination challenge
  – Agent A must decide whether to move down Corridor 1 or 2.
  – Similarly, Agent B will need to decide the same.

  – Assume employment of RVOs for safety purposes
  – How can we achieve progress?
    • No communication, only observe the other agents
For each agent, the cost of each action $\alpha$ is defined as $C(\alpha)$, the length of the corresponding path to the goal.
Let $I_i$ represent the interaction cost for action $a_i$ given the observed state of the other agent.

- Represents whether the other agent is along the corresponding path.
2 Greedy Agents

Corridor Environment

The red line is the solution trajectory.
The light blue lines are the Velocity Obstacles.

[Kimmel, Bekris DARS 2014]
Deconfliction for First-Order Systems

[Pallotino, Scordio, Frazzoli, Bicchi ‘07]
Deconfliction for First-order Systems

[Krontiris, Bekris IROS ’11]

30 Airplanes
Safety Concerns (ICS)

• Safety becomes a concern in decentralized planning
  – Independently plan paths that are pairwise collision-free
• For systems with dynamics, e.g., inertia
  – Also avoid inevitable collision states

• How can communication help?
  – i.e., couple choices in terms of safety considerations
Safe Multi-Robot Motion Coordination

Initial state $x(t_{N+1})$

Goal $V_B$

Plan $A_1$

Plan $A_2$

Plan $A_3$

Current contingency for C

Current contingency for B

States $x(t_{N+2})$

Bekris, Tsianos, Kavraki '07, '09

Goal $V_A$

Goal $V_C$
Safe Multi-Robot Motion Coordination

Initial state \( x(t_{N+1}) \)

\[ \text{plan } A_1 \]

\[ \text{plan } A_2 \]

\[ \text{plan } A_3 \]

Goal \( V_B \)

Goal \( V_A \)

Goal \( V_C \)

[Bekris, Tsianos, Kavraki ’07,’09]
Safe Multi-Robot Motion Coordination

Goal $V_B$

Initial state $x(t_{N+1})$

[Bekris, Tsianos, Kavraki ’07,’09]
Safe Multi-Robot Motion Coordination

[ Bekris, Tsianos, Kavraki '07,'09 ]
If the requirements are satisfied: Safety is guaranteed

How can we implement the requirements for coordination?

Alternative solutions:
1. Global priority scheme

Problem: Low priority vehicles do not have time to compute a solution
Effect: Vehicles result often in contingency plans

2. Cooperative Action Selection

Can the planning framework be integrated with a balanced, scalable coordination scheme and guarantee safety?

[Bekris, Tsianos, Kavraki ROBOCOMM ’07] Best Student Paper Award
Selection of Contingencies

- Problem of priorities:

  Frequent selection of contingency plans

- Casted the problem as Distributed Constrained Optimization and used a message-passing algorithm (belief propagation based)

<table>
<thead>
<tr>
<th></th>
<th>Rooms</th>
<th>Labyrinth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td># Vehicles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prioritized</td>
<td>3.61 %</td>
<td>24.5 %</td>
</tr>
<tr>
<td>Max-plus</td>
<td>0.98 %</td>
<td>2.26 %</td>
</tr>
<tr>
<td></td>
<td>1.35 %</td>
<td>8.42 %</td>
</tr>
<tr>
<td></td>
<td>3.04 %</td>
<td>4.84 %</td>
</tr>
</tbody>
</table>
Asynchronous Operation

[Bekris, Grady, Moll, Kavraki - IJRR '12]

• Safety challenge:
  - Guarantee that there is a safe path $\pi^i_*$ to select in every planning cycle

• Challenges vs synchronous operation:
  - States cannot be accompanied by timestamps
  - No guarantee messages arrive in order
Motion Planning Approaches

**Properties**
- Classical approaches
- Suboptimal but complete solutions that are tractable
- New optimal planners (opportunistic decoupling)
- Exciting applications: manipulation

**Approaches**
- Centralized
- Decentralized

**Optimality**
- Coupled vs Decoupled

**Completeness**
- Is it safe?
- Does it avoid deadlocks/livelocks?
- What are its information requirements?

**Safety**
- How efficient solutions with formal guarantees can we achieve with limited information requirements?
Thank you for your attention!

Primary Contributors

- Push and Swap approach
- Communication-less Motion Coordination
- Dual-arm scheduling
- Deconfliction approach
- Pebble graph solvers
- Manipulation applications

Our research efforts have been supported by:
- the National Science Foundation (NSF),
- the National Aeronautics and Space Administration (NASA),
- the Department of Defense (ONR & DoD TARDEC),
- the National Aeronautics and Space Administration (NASA),
- the Department of Defense (ONR & DoD TARDEC),