Robust Execution of Contingent, Temporally Flexible Plans

Stephen Block
Andreas Wehowsky, Brian Williams

Robustness to Disturbances

Robustness to...
• Temporal uncertainty: Temporally flexible mission plan
• Execution uncertainty: Dispatchable execution
• Communication latency: Distributed architecture
• Plan failure: Contingent mission plan

Robustness to Execution Uncertainty

Robustness to...
• Temporal uncertainty: Temporally flexible mission plan
• Execution uncertainty: Dispatchable execution

Problem: Plan is either...
• Brittle to temporal execution uncertainty
• Overly conservative to ensure success

Solution: Dispatchable execution...
• Postpone scheduling until execution time

Assignment of execution times

Maintenance of temporal flexibility

Execution time

Planning time

Hardware Commands

Reactively schedule execution times

Hardware Commands

Real World Autonomous Agents

• Coordinate multiple agents
• Provide robustness

“Drive to rock”

Simple temporal constraint: \( l \leq t^+ - t^- \leq u \)

Hardware command
Robustness to Execution Uncertainty

- Temporal uncertainty: Temporally flexible mission plan
- Execution uncertainty: Dispatchable execution

Least commitment planning allows the executive to use temporal flexibility to respond to uncertainties at run time.

Robustness to Communication Latency

- Temporal uncertainty: Temporally flexible mission plan
- Execution uncertainty: Dispatchable execution
- Communication latency: Distributed architecture

Problem: Centralized architecture introduces communication bottleneck at master agent.

Solution: A distributed architecture evens out the communication requirements.

Agent Structuring

- Contingent temporarily flexible plan
- Maintenance of temporal flexibility
- Reduced computational complexity
- Avoids communication bottleneck
- Hardware

Distributed Architecture

- Plan distribution
- Maintenance of temporal flexibility
- Execution time
- Temporally flexible plan
- Dispatching
- Hardware Commands

Problem: Centralized architecture introduces communication bottleneck at master agent.

Solution: Dispatchable execution...
- Plan reformulation to compile the plan to a form for easy dispatching
- Dispatching to schedule and execute activities
Robustness to Plan Failure

- Temporal uncertainty: Temporally flexible mission plan
- Execution uncertainty: Dispatchable execution
- Communication latency: Distributed architecture
- Plan failure: Contingent mission plan

Temporal Plan Network (TPN)

Contingencies encode choices between alternate threads of execution
Interleaved Candidate Generation and Consistency Checking

1. Generate candidate plans through distributed search on the TPN
2. Test the generated plans for temporal consistency

Implemented using a message passing scheme ...

- **findfirst** Initial search for a consistent set of choice variable assignments
- **findnext** Search for a new consistent assignment, to achieve global consistency
- **fail** No consistent set of choice variable assignments was found
- **ack** A consistent set of choice variable assignments was found

Interleaved and concurrent

Candidate Generation

Choice node selects subnetwork and sends findfirst message

Depth first search to generate candidates

Consistency Checking

Nodes perform consistency checking and reports failure to choice node

Distributed Bellman Ford to test candidates

Candidate Generation

Choice node selects previously unselected subnetwork and sends findfirst message
Consistency Checking

Nodes perform consistency checking and report success to choice node.

Candidate Generation

Candidate generation and consistency checking are interleaved and concurrent.

Nodes send findfirst messages to their children. Search progresses at increasing depth in parallel.

Consistency Checking

Nodes perform consistency checking and report to their parent. Checking progresses at decreasing depth in parallel.

Time Complexity Analysis

<table>
<thead>
<tr>
<th></th>
<th>Centralized Planner</th>
<th>Distributed Planner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate Generation</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Temporal Consistency</td>
<td>NE ≈ N²</td>
<td>No</td>
</tr>
<tr>
<td>Overall Time Complexity</td>
<td>Exponential</td>
<td>Exponential</td>
</tr>
</tbody>
</table>

N: Number of nodes
E: Total number of edges
e: Number of local edges
C: Size of domain of choice variables

Worst case complexity of candidate generation corresponds to a plan entirely composed of choice nodes.

Questions?