Planning with Concurrency in Continuous-time Stochastic Domains (thesis proposal)

Håkan L. S. Younes

Thesis Committee:
Reid G. Simmons (chair)
Geoffrey J. Gordon
Jeff Schneider
David J. Musliner, Honeywell
Aim of Thesis Research

- Policy generation for continuous-time stochastic systems with concurrency

Sampling-based Probabilistic Verification

Success
Motivating Example

- Dependable workstation cluster
  - Components may fail at any point in time
  - Single repair agent

![Diagram showing a dependable workstation cluster with a left and right cluster, a switch, and a backbone.](image-url)
Motivating Example

- Planning problem:
  - Find repair policy that maintains satisfactory quality of service

```
<table>
<thead>
<tr>
<th>L_1</th>
<th>L_2</th>
<th>L_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_1</td>
<td>R_2</td>
<td>R_3</td>
</tr>
</tbody>
</table>
```

Left cluster

Right cluster
Motivating Example

all working

Left cluster

Right cluster
Motivating Example

L₂ fails

all working → L₂ failed

3.6

L₁, L₂, L₃

Left cluster

Switch ➔ Backbone ➔ Switch

R₁, R₂, R₃

Right cluster
Motivating Example

L₂ fails → repair L₂

all working

3.6

L₂ failed

1.5

repairing L₂

Left cluster

Right cluster

Switch

Backbone

Switch

R₁

R₂

R₃
Motivating Example

L₂ fails → repair L₂ → left switch fails

all working

L₂ failed

repairing L₂

repairing L₂ switch failed

3.6

1.5

2.8

L₁

L₂

L₃

Left cluster

Switch

Backbone

Switch

R₁

R₂

R₃

Right cluster
Motivating Example

L₂ fails → repair L₂ → left switch fails → L₂ repaired

all working → L₂ failed → repairing L₂ → repairing L₂ switch failed → switch failed

L₁, L₂, L₃ (Left cluster) → Switch → Backbone → Switch → R₁, R₂, R₃ (Right cluster)

3.6 1.5 2.8 4.2
Why Challenging?

- Continuous-time domain:
  - State transitions can occur at any point in time along a continuous time-axis
- Concurrent actions and events:
  - Left switch fails while repairing $L_2$
- Non-Markovian:
  - Weibull distribution for time to hardware failure
Semi-Markovian?

- Each component can be viewed as a semi-Markov process

![Diagram showing the states of working, repairing, and failed with probability distributions: Uniform(3,8), Weibull(2), and Uniform(1,2).]
Concurrent Semi-Markov Processes

- L_2 fails
- repair L_2
- left switch fails
- L_2 repaired

- all working
- L_2 failed
- repairing L_2
- repairing L_2 switch failed
- switch failed

3.6 1.5 2.8 4.2

generalized semi-Markov process (GSMP)
Potential Application Domains

- Multi-agent systems
- Process plant control
  - Startup/shutdown procedures
- Robotic control
  - Mars rover
  - Large degree-of-freedom humanoid
Planning Problem

- **Input:**
  - Domain model (GSMP)
  - Initial state
  - Goal condition

- **Output:**
  - Policy mapping states to actions
Maintain three interconnected workstations for 24 hours with probability at least 0.9
Approach

- Generate, Test, and Debug (GTD) (Simmons 1988)
The GTD Paradigm

- **Generate** initial plan making simplifying assumptions
- **Test** if plan satisfies goal condition
- **Debug** plan if not satisfactory
My Adaptation of GTD

- **Generate**
  - assume we can quickly generate plan for simplified problem, or start with null-plan

- **Test**
  - Sampling-based probabilistic verification

- **Debug**
  - Analysis of sample execution paths
Approach: Schematic View

- Generate
- Test: plan, sample paths
- Debug: best plan
- Compare: old plan, sample paths

Initial plan: Test
New plan: Debug
Sample paths: Compare
Test Step

- Verify goal condition in initial state
  - Use *simulation* to generate sample execution paths
  - Use *sequential acceptance sampling* (Wald 1945) to verify probabilistic properties
Debug Step

- Analyze sample execution paths to find reasons for failure
  - **Negative** sample paths provide information on how a plan can fail
Generic Repair Procedure

1. Select some state along some negative sample path
2. Change the action planned for the selected state

Develop heuristics to make informed state/action choices
Hill-climbing Search for Satisfactory Plan

1. Compare repaired plan to original plan
   - Fast sampling-based comparison
2. Select the better of the two plans
Approach Summary

- Plan verification using discrete event simulation and acceptance sampling
- Plan repair using sample path analysis
  - Heuristics to guide repair selection
- Hill-climbing search for satisfactory plan
  - Fast sampling-based plan comparison
Related Work

- Probabilistic planning
- Planning with concurrency
- Policy search for POMDPs
- Probabilistic verification
Probabilistic Planning

- Anytime synthetic projection (Dummond & Bresina 1990)
  - Modal temporal logic for goal specification
  - Generate initial solution path
  - “Robustify”: Find solution paths for deviations to ideal path
Planning with Concurrency

- CIRCA (Musliner et al. 1995)
  - Policy generation for timed automata
  - Non-deterministic model
- Concurrent temporally extended actions (Rohanimanesh & Mahadevan 2001)
  - Decision-theoretic approach
  - SMDP Q-learning
Policy Search for POMDPs

- Pegasus (Ng & Jordan 2000)
  - Estimate value function by generating sample execution paths
- GPOMDP (Baxter & Bartlett 2001)
  - Simulation-based method for estimating gradient of average reward
Probabilistic Verification

- PRISM (Kwiatkowska et al. 2002)
  - Symbolic probabilistic model checking
  - Requires Markov model
Previous Work

- Heuristic deterministic planning
- Sampling-based probabilistic verification
- Initial work on sample path analysis
Heuristic Deterministic Planning

- VHPOP: A heuristic partial order planner
  - Distance-based heuristics for plan ranking
  - Novel flaw selection strategies
  - “Best Newcomer” at 3rd International Planning Competition
Sampling-based Probabilistic Verification

- Verifying CSL properties of discrete event systems
- Sampling-based vs. symbolic methods:
  - Sampling-based approach...
    - uses less memory
    - scales well with size of state space
    - adapts to difficulty of problem (sequential)
  - Symbolic methods...
    - give exact results
    - handle time-unbounded and steady-state properties
Tandem Queuing Network

- $\text{M/Cox}_2/1$ queue sequentially composed with $\text{M/M/1}$ queue
- Each queue has capacity $n$
- State space of size $O(n^2)$
Tandem Queuing Network (property)

- When empty, probability is less than 0.5 that system becomes full within $t$ time units:
  - $\neg \Pr_{\geq 0.5}(\text{true } U \leq t \text{ full})$ in state “empty”
Tandem Queuing Network (results)

\[ \neg \Pr_{\geq 0.5}(\text{true} \ U \leq \text{full}) \]

\[ \alpha = \beta = 10^{-2} \]
\[ \delta = 10^{-2} \]
Tandem Queuing Network
(results)

\[ \neg \Pr_{\geq 0.5}(true \leq t \text{ full}) \]

- $\alpha = \beta = 10^{-2}$
- $\delta = 10^{-2}$

Verification time (seconds)

- $n=255$ (symbolic)
- $n=31$ (" )
- $n=3$ ("")
- $n=255$ (sampling)
- $n=31$ ("")
- $n=3$ ("")
Symmetric Polling System

- Single server, $n$ polling stations
- Stations are attended in cyclic order
- Each station can hold one message
- State space of size $O(n \cdot 2^n)$
Symmetric Polling System (property)

- When full and serving station 1, probability is at least 0.5 that station 1 is polled within $t$ time units:
  - $\Pr_{\geq 0.5}(\text{true } U \leq t \text{ poll1})$ in state "full, srv 1"
Symmetric Polling System (results)

Pr_{\geq 0.5}(true U_{\leq t} poll1)

\alpha=\beta=10^{-2}
\delta=10^{-2}

Verification time (seconds) vs Size of state space

- T=40 (symbolic)
- T=20 (")
- T=10 ("")
- T=40 (sampling)
- T=20 ("")
- T=10 ("")
Symmetric Polling System (results)

\[ \Pr_{\geq 0.5}(true \text{ U} \leq t \text{ poll1}) \]

\[ \alpha = \beta = 10^{-2} \]
\[ \delta = 10^{-2} \]

Verification time (seconds)
Symmetric Polling System (results)

\[ \alpha = \beta = 10^{-10} \]
\[ \alpha = \beta = 10^{-8} \]
\[ \alpha = \beta = 10^{-6} \]
\[ \alpha = \beta = 10^{-4} \]
\[ \alpha = \beta = 10^{-2} \]

\[ \Pr_{\geq 0.5}(\text{true } U \leq t \text{ poll1}) \]

Verification time (seconds)

\[ n=10 \]
\[ t=50 \]
Initial Work on Sample Path Analysis

\[ \Pr_{\geq 0.9}(\text{safe } U \leq 100 \text{ goal}) \]
Proposed Work

- Heuristic plan repair
- Conjunctive probabilistic goals
Heuristic Plan Repair

- Identify local repair operations for GSMP domain models
- Develop distance-based heuristics for stochastic domains to guide repair selection
- Make use of positive sample execution paths to rank plan repairs
- Evaluate different search strategies, such as hill-climbing and simulated annealing
Heuristics to Guide Action Selection

\[ \Pr_{\geq 0.9}(\text{safe } U^{\leq 100} \text{ goal}) \]

Diagram:
- Safe
- \(\neg\text{safe}\)
- Goal

Alternative path
Utilizing Positive Sample Execution Paths

- Positive sample paths show how a plan can succeed
- Analyze positive sample paths to identify positive behavior

Pr ≥ 0.9 (safe $U \leq 100$ goal)
Evaluating Alternative Search Strategies

- Simulated annealing
  - Avoid getting stuck in local optimum
  - Use confidence from plan comparison to probabilistically select a plan
Conjunctive Probabilistic Goals

- Add support for more expressive goal conditions
  - \( \text{Pr}_{\geq \theta}(\rho) \land \text{Pr}_{\geq \theta'}(\rho') \)
- Can be used to express goal priorities
- Challenges:
  - How to compare two plans
  - Multiple goals to account for when repairing a plan
There is sufficient information in sample execution paths obtained during plan verification to enable efficient and effective repair of plans for continuous-time stochastic domains.
Validation

- Construct benchmark domains
  - Develop probabilistic PDDL
  - Extend benchmarks for deterministic temporal planning
- Gain insight into what makes a problem difficult (empirical and theoretical analysis)
- Make sure interesting plans can be generated in reasonable amount of time
## Timetable

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring 2003</td>
<td>Develop benchmark domains</td>
</tr>
<tr>
<td>Summer 2003</td>
<td>Identify repair operations</td>
</tr>
<tr>
<td></td>
<td>Implement initial version of planning system</td>
</tr>
<tr>
<td>Fall 2003</td>
<td>Develop distance-based heuristics for stochastic domains</td>
</tr>
<tr>
<td>Winter 2003/04</td>
<td>Experiment with different search strategies</td>
</tr>
<tr>
<td>Spring 2004</td>
<td>Add support for conjunctive probabilistic goals</td>
</tr>
<tr>
<td>Summer 2004</td>
<td>Theoretical analysis</td>
</tr>
<tr>
<td>Fall 2004</td>
<td>Final experimental validation</td>
</tr>
<tr>
<td></td>
<td>Writing of thesis</td>
</tr>
<tr>
<td>Winter 2004/05</td>
<td>Finalizing thesis</td>
</tr>
<tr>
<td></td>
<td>Thesis defense</td>
</tr>
</tbody>
</table>
Possible Extensions

- Generation of non-stationary policies
- Partial observability
- Continuous-space models
- Decision-theoretic planning