Evaluating Coverage Based Intention Selection

Max Waters, Lin Padgham and Sebastian Sardina

RMIT University, Melbourne, Australia
Overview

• This paper tackles intention selection in BDI agents

• A selection mechanism based on goal coverage was proposed at AAMAS in 2012\(^1\)

• This work implements and empirically evaluates it

• Analysis of results reveals a powerful selection mechanism based on the idea of progressability

BDI Agents

Events
Prompt a response from the agent

Plans
A strategy to respond to an event, of the form:

\[ e: cc \leftarrow p \]

Coverage of event
% of states with plans available

Intentions
Committed strategies
BDI Agents

Events
Prompt a response from the agent

Plans
A strategy to respond to an event, of the form:

\[ e: cc \leftarrow p \]

Coverage of event
% of states with plans available

Intentions
Committed strategies

How to provide infrastructure support for intelligent intention selection?
Intention selection

How to choose which intention to progress next?

Issues: intention interference and unexpected changes

Objective: maximize successfully executed intentions

Existing approaches

Simple: first-in-first-out (FIFO) and round-robin (RR)

Meta-level programming

Domain info: priorities, deadlines, dependencies,

Challenge: intelligent, domain-independent intention selection
Using coverage for domain-independent intention selection

Coverage-based selection was proposed in AAMAS 2012

Opportunistically execute the most vulnerable intention
• Vulnerability is measured through coverage

Coverage of a goal: % of states with plans available
Lower the coverage → more vulnerable intention

Calculating coverage
• Calculated from plans’ context conditions
• Can be calculated off-line before execution, and without extra information from the programmer
Coverage-based scheduling

C1: a variation on the AAMAS 2012 proposal

- Select the **progressable** intention with the **lowest coverage**
- **Progressable**: has an applicable plan
- Pre-emptive: change focus if necessary
- All are unprogressable? Failure recovery

C1 is compared experimentally with FIFO and RR under different levels of coverage and environmental dynamism
Experimental setup

Agent with ten concurrent intentions in a dynamic environment

Automated test generation
• Simple binary structure allows for bulk generation of test cases

Preparatory effects
A plan brings about a condition which is required by a later plan

Coverage gaps
• Remove a branch
• Add p-effect to parent plan
• Change probability of proposition to alter gap size
Experimental setup

Agent with ten concurrent intentions in a dynamic environment

Automated test generation
• Simple binary structure allows for bulk generation of test cases

Preparatory effects
A plan brings about a condition which is required by a later plan

Coverage gaps
• Remove a branch
• Add p-effect to parent plan
• Change probability of proposition to alter gap size
Experimental setup

The dynamic environment

• Dynamism rate – 0 \leq d \leq 1
• At each step, the variables referred to by context conditions are re-sampled with probability $d$

Test runs

• Comparison of C1, FIFO, 1-step RR
• 100,000 test runs for each algorithm
• Each test run has a randomly selected dynamism and coverage
• For each test run, the proportion of successfully executed intentions is recorded (success rate)
Coverage results

- On average, C1 improves on FIFO by 13pp, RR by 24.5pp
- Never detrimental to the success rate
- Most benefit when environment is dynamic and goals have low coverage
- Low coverage and high dynamism:
  - C1 improves on FIFO by up to 60pp
  - C1 improves RR by up to 62pp
Coverage results

- On average, C1 improves on FIFO by 13pp, RR by 24.5pp
- Never detrimental to the success rate
- Most benefit when environment is dynamic and goals have low coverage
- Low coverage and high dynamism:
  - C1 improves on FIFO by up to 60pp
  - C1 improves RR by up to 62pp

RR’s switching makes it prone to failure even in low dynamism environments
Coverage results

- On average, $C_1$ improves on FIFO by 13pp, RR by 24.5pp.
- Never detrimental to the success rate.
- Most benefit when environment is dynamic and goals have low coverage.
- Low coverage and high dynamism:
  - $C_1$ improves on FIFO by up to 60pp.
  - $C_1$ improves RR by up to 62pp.
Progressability

C1 has two key features:
1. Prioritizing by coverage
2. Progressability checking

Two questions:
1. How much success is due to coverage prioritization, and how much is due to progressability checking?
2. Can progressability checking improve FIFO or RR?

\(\text{FIFO}^\text{LA}\) and \(\text{RR}^\text{LA}\): variations of FIFO and RR which change focus when an intention becomes unprogressable
Progressability results

- Overall:
  - $FIFO^{LA}$ improves on FIFO by 12pp
  - $RR^{LA}$ improves on RR by 18pp
- Benefit of 5pp even with high coverage and low dynamism
- With low coverage and high dynamism:
  - $FIFO^{LA}$ improves on FIFO by up to 48pp
  - $RR^{LA}$ improves on RR by up to 40pp
Progressability results

- Overall:
  - $\text{FIFO}^{\text{LA}}$ improves on FIFO by 12pp
  - $\text{RR}^{\text{LA}}$ improves on RR by 18pp
- Benefit of 5pp even with high coverage and low dynamism
- With low coverage and high dynamism:
  - $\text{FIFO}^{\text{LA}}$ improves on FIFO by up to 48pp
  - $\text{RR}^{\text{LA}}$ improves on RR by up to 40pp
Progressability results

- Overall:
  - \( \text{FIFO}^{\text{LA}} \) improves on FIFO by 12pp
  - \( \text{RR}^{\text{LA}} \) improves on RR by 18pp
- Benefit of 5pp even with high coverage and low dynamism
- With low coverage and high dynamism:
  - \( \text{FIFO}^{\text{LA}} \) improves on FIFO by up to 48pp
  - \( \text{RR}^{\text{LA}} \) improves on RR by up to 40pp

(a) \( \text{FIFO}^{\text{LA}} \) vs FIFO

(b) \( \text{RR}^{\text{LA}} \) vs RR
Further benefit of coverage

- On average, C1 improves on \( \text{FIFO}^{\text{LA}} \) by 1.2pp, and \( \text{RR}^{\text{LA}} \) by 5.3pp
- Most improvement over \( \text{FIFO}^{\text{LA}} \) in low-coverage, high-dynamism tests
- Most Improvement over \( \text{RR}^{\text{LA}} \) in low-coverage tests
- Improves RR even in low dynamism environments
Further benefit of coverage

- On average, C1 improves on FIFO$^\text{LA}$ by 1.2pp, and RR$^\text{LA}$ by 5.3pp
- Most improvement over FIFO$^\text{LA}$ in low-coverage, high-dynamism tests
- Most improvement over RR$^\text{LA}$ in low-coverage tests
- Improves RR even in low dynamism environments
Further benefit of coverage

- On average, C1 improves on FIFO$^{LA}$ by 1.2pp, and RR$^{LA}$ by 5.3pp
- Most improvement over FIFO$^{LA}$ in low-coverage, high-dynamism tests
- Most improvement over RR$^{LA}$ in low-coverage tests
- Improves RR even in low dynamism environments
Conclusions

Progressability
✓ Very easily implemented
✓ Increases the success rate
✗ Introduces pauses in execution
✗ Postpones failure recovery mechanisms
   ▷ Use in conjunction with (e.g.) priorities

Coverage
✓ An effective priority measure when goal coverage is low and the environment is unpredictable
✓ Standard BDI languages have information needed to implement
✗ Not trivial to implement
Further work

- Experimentation on more 'realistic' goal-plan trees

- Hybrid intention selection mechanisms
  - Combine progressability with check for failure recovery

- Further uses for coverage
  - Prioritize by expected gain in coverage