ALLEGRO: Belief-based Programming in Stochastic Dynamical Domains

Vaishak Belle, Hector J. Levesque

OBJECTIVE

- *High-level program execution*, e.g., GOLOG, FLUX, are attractive and influential alternatives to planning
- especially with incomplete information, where plans with branches and loops are required
- lots of success, e.g., cognitive robotics

$$\textbf{loop}: \quad \textbf{if } \neg Empty(queue) \textbf{ then } (\pi p)\, selectRequest(p);\ pickupCoffee;\ bringCoffee(p) \textbf{ else } wait$$

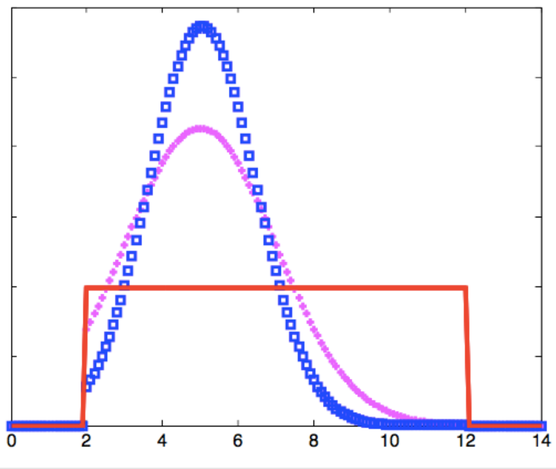
Major criticism: action & knowledge model very unrealistic for actual robots!

- *noisy* actions + sensors
- noise models use *continuous distributions*

Need to rethink GOLOG, both at specification and implementation level

PREGO

- Bacchus, Halpern and Levesque proposed a general account for noisy actions, sensors and degrees of belief in the situation calculus; IJCAI-13 our generalization to continuous distributions



- AAAI-14: a *projection system* called PREGO, where you specify basic action theory and *query beliefs*

```
(define-fluents h)
(define-ini-p-expr '(UNIFORM h 2 12))
(define-ss-exprs h
    (fwd z) '(max 0 (- h ,z)))
(define-l-exprs
    (sonar z) '(GAUSSIAN ,z h 4.0))
```

Query mechanism:

```
> (eval-bel (< h 3) ((fwd 4)))
0.5
```

LIMITATIONS

PREGO is Regression-based, appropriate for automated planning

But *no support for programs* (don't know outcomes/sensed values in advance!)

Iterative programs = regression with an *infeasible number of integrals*!

ALLEGRO = ALGOL IN PREGO

- Modeler specifies basic action theory (BAT), including probabilistic beliefs and noise models
- Expressive programing language:

prim	primitive programs;
(begin $prog_1 \ldots prog_n$)	sequence;
(if *form* $prog_1$ $prog_2$)	conditional;
(let $((var_1\ term_1) \ldots (var_n\ term_n))$ *prog*)	assignments;
(until *form prog*)	until loop.

where *form* stands for formulas:

$$form ::= (\circ\ term_1\ term_2) \mid (\bullet\ form_1\ form_2) \mid (\texttt{not}\ form)$$

where $\circ \in \{<, >, =\}$, $\bullet \in \{\texttt{and}, \texttt{or}\}$, and *term* stands for terms:

$$term ::= (\texttt{exp}\ term) \mid number \mid fluent \mid var \mid (\diamond\ term_1\ term_2) \mid (\texttt{if}\ form\ term_1\ term_2)$$

To get close to the wall, for example:

```
(until (> (bel (and (>= h 2) (<= h 6))) .8)
    (until (> (conf h .4) .8) (sonar))
    (let ((diff (- (exp h) 4)))
       (nfwd diff)))
```

- cf. paper for semantics of ALLEGRO programs, interpreted as situation-suppressed formulas. Successful termination after σ is expressed as:

$$\mathcal{D} \cup \mathcal{E} \cup \mathcal{F} \models Do(\delta, S_0, do(\sigma, S_0)).$$

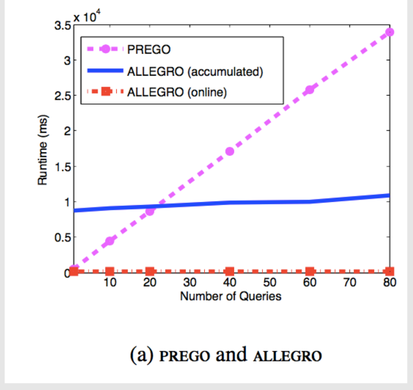
INTERPRETER

- ALLEGRO requires a novel kind of *interpreter* and *correctness theorem*, because worlds and action outcomes are possibly uncountably many!
- Suppose we sampled worlds and action outcomes.
- Suppose program δ terminated after sequence σ.

THM: $\mathcal{D} \models Bel(\phi, do(\sigma, S_0)) = u$ iff $\lim_{n\to\infty} INTERPRETER[(\texttt{bel}\ \phi), \delta, \mathcal{D}_0] = u$.

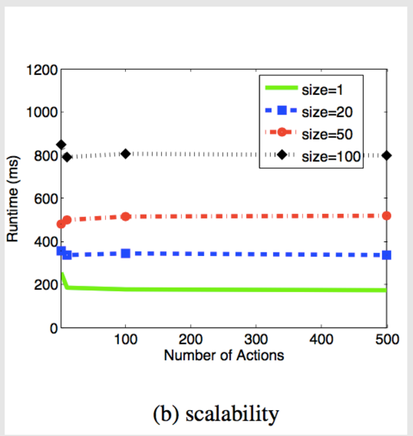
EMPIRICAL EVALUATIONS

- Regression $\Rightarrow$ many integrals:



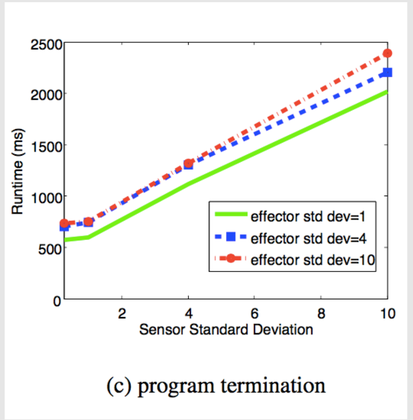
(a) PREGO and ALLEGRO

- ALLEGRO scales well:



(b) scalability

- Program termination can be studied wrt error models of sensors and actuators:



(c) program termination

CONCLUSIONS

- A new variant of GOLOG over noise and belief
- Different from all other "probabilistic" GOLOG variants, e.g. DTGOLOG, that do not handle unobservable nondeterminism, noisy sensors, belief change, and continuous distributions
- Techniques and empirical results demonstrate promise of proposal

REFERENCES

V. Belle and H. J. Levesque. *Reasoning about continuous uncertainty in the situation calculus.* IJCAI, 2013.

V. Belle and H. J. Levesque. *Reasoning about probabilities in dynamic systems using goal regression.* UAI, 2013.

V. Belle and H. J. Levesque. *How to progress beliefs in continuous domains.* KR, 2014.

V. Belle and H. J. Levesque. *PREGO: An Action Language for Belief-Based Cognitive Robotics in Continuous Domains.* AAAI, 2014.

vaishak@cs.kuleuven.be, hector@cs.toronto.edu