Planning with Time and Scheduling – Three Examples

P@trik Haslum

NICTA/ANU

Advanced Topics in AI, 2007

Australian Government
Department of Communications, Information Technology and the Arts
Australian Research Council

NICTA Members

ANU
THE AUSTRALIAN NATIONAL UNIVERSITY

UNSW
THE UNIVERSITY OF NEW SOUTH WALES

NSW
First for Business
Department of State and Regional Development

ACT GOVERNMENT

Victoria
The Place To Be

THE UNIVERSITY OF MELBOURNE

The University of Sydney

Queensland Government

Griffith UNIVERSITY

QUT
Queensland University of Technology

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

NICTA Partners

Summary

- 1st Examle: TP4
  - Makespan-optimal planning.
  - Temporal regression.
- 2nd Example: LPGP & Crikey
  - Separating planning and plan scheduling (almost).
  - "Expressive" temporal planning (PDDL2.1).
- 3rd Example: HSTS/Europa
  - Constaint-Based Scheduling.
  - Planning with an activity/constraint model.

TP4

- Simple temporal planning ("TGP semantics"):
  - Actions have *duration*, $dur(a) > 0$.
  - For action a to execute over $[s, t]$:
    - preconditions hold at s;
    - preconditions and effects *not interfered with* (not added/deleted by any other action) over $[s, t]$;
    - effects can be relied on only at t.
- Temporal regression.
  - The "regression cut" property.
- h^2 heuristic for makespan.

Digression: Temporal Constraint Networks

- Temporal Constraint Network (TCN):
  - Variables representing *time points*
  - Disjunctive interval constraints on differences:

    $$(t_j - t_i \in [l_1, u_1]) \vee \ldots \vee (t_j - t_i \in [l_n, u_n])$$

  - Consistency checking is NP-hard
  - Reasonably efficient meta-CSP approach.
- Simple Temporal Network (STN):
  - Single interval constraint between any two variables:

    $$d_{i,j}^{min} \leqslant t_j - t_i \leqslant d_{i,j}^{max}$$

  - Constraint are *linear* – consistency checking is tractable.
  - More efficient: all-pairs shortest path on *distance graph*.
- Dechter, R., Meiri, I., & Pearl, J., AIJ, 1991.

LPGP, Crikey & DEP+

- PDDL2.1 temporal planning:
  - Actions have conditions *at start*, *at end* and *over all* (interior of execution interval).
  - Actions have *instantaneous* effects *at start* and *at end*.
  - Compatibility constraints at a time point are the same as in simple temporal planning.
  - States have duration > 0: conditions must be separated from establishing effects by a positive amount of time.
- Separating planning and scheduling:
  - Sequential plan in the space of *events*: sets of actions starting and ending.
  - Maintain temporal constraints as STN/LP to ensure *schedulability*.
  - Makespan optimality only *in the limit* – not in practice.

Constraint-Based Scheduling

- Variables:
  - $start(A)$, $end(A)$, $\forall A$ – absolute or relative.
- Constraints:
  - Duration: $d_{min}(A) \leqslant end(A) - start(A) \leqslant d_{max}(A)$
  - Precedence ("A before B): $end(A) < start(B)$.
  - Unary resource:
    $(end(A) < start(B)) \vee (end(B) < start(A))$.
  - Cumulative: $\forall t, \left(\sum_{start(A) \leqslant t \leqslant end(A)} req(A)\right) \leqslant cap(R)$.
- Alternative resources and (optional) set-up activities cause *disjunctions* – hard to deal with efficiently.
- Specialised propagators for certain classes of resource constraints more efficient than general (disjunctive) formulation.

HSTS/Europa

- Activity/constraint model:
  - Timelines – state variables.
  - Tokens – variable value over *time interval* ($[s,t](v = x)$).
  - Compatibility constraints
    $[s,t](v = x) \rightarrow ([u_1, v_1](v'_1 = y)) \vee \ldots \vee ([u_n, v_n](v'_n = z))$
- No distinction between states and actions!
- Search:
  - Branch on disjunctive compatibilities/token placement.
  - Maintain consistency by STN.
  - Solution – consistent and all compatibilities satisfied.
- Requires (domain-specific) search control for efficiency.
- Recent work on domain-independent heuristics.