Comp3620/Comp6320 Artificial Intelligence
Tutorial 6: Heuristics, Regression, and Partial-Order Planning

May 15-18, 2018

Exercise 1 (Delete Relaxation)

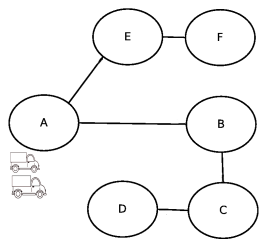


Figure 1: Delivering Problem

Consider the following delivery problem. Two driverless trucks (T1 and T2) can autonomously deliver the products that customers have requested. The products are initially at the depot (location A), and our trucks can reach the customers identified by their locations (B,C,D,E,F) by following suitable paths connecting adjacent locations. Just two out of these five customers (F and D) have to be served. Assume for simplicity that the two trucks have always enough products on-board. Ignore any cost that could be associated to the actions. A typed STRIPS description of the problem is as follows:

- Types: Locations ={A,B,C,D,E,F}, Trucks={T1,T2}
- Propositions:
  - at(t,x), $x \in$ Locations, $t \in$ Trucks
  - delivered(x) $x \in$ Locations
  - adjacent(x,y), $x, y \in$ Locations
- Actions:
  - go(t,x,y), $t \in$ Trucks, $x, y \in$ Locations
    - preconditions: {at(t,x), adjacent(x,y)}
    - add-list: {at(t,y)}
    - del-list: {at(t,x)}
  - deliver(t,x), $t \in$ Trucks, $x \in$ Locations
    - preconditions: {at(t,x)}
    - add-list: {delivered(x)}
    - del-list: $\emptyset$

· set every actions delete list = $\emptyset$

· Once a proposition becomes true, it stays true forever.

· Relaxed plan for a problem P

P^+ is the delete relaxation of P.

→ any solution to P^+

Optimal Relaxed Plan

< go(T_1, A, E)
go(T_1, E, F)
deliver(T_1, F)
go(T_2, A, B)
go(T_2, B, C)
go(T_2, C, D)
deliver(T_2, D) >

h^* = True cost of optimal plan (for P)

h^+ = True cost of optimal delete-relaxed plan (for P^+)

$7 = h^+ \le h^* = 7+5 = 12$

3 ways to write a plan:

① totally ordered
② partially ordered
③ parallel

partially ordered plan: → suggested plan

{go(T1, A, E) < go(T1, E, F)

go(T1, E, F) < ~~g~~deliver(T1, F)

{go(T1, A, E), go(T1, E, F), ...}

also relaxed-version

full answer needs "moving back"

T2 AB T2 BC

T2 BC T2 CD

T2 CD T2 D

- Initial state: {at(T1,A),at(T2,A)} ∪ { adjacent(x,y) |$x, y \in$ Locations, there is an edge between x and y in the graph of Figure1}
- Goal: {delivered(F),delivered(D),at(T1,A),at(T2,A)}

Questions:

ignores all the delete-lists.

1. Explain the concepts of the delete relaxation of a problem P and of a relaxed-plan for P.
2. Write the optimal relaxed plan for this problem. What are the values of h^+ and h^* at the initial state?
3. Write a partially ordered plan and a parallel plan solving the problem.

parallel: < {go(T2, A, B), go(T1, A, E)}, {go(T2, B, C), go(T1, E, F)}, ... { }, ..., {go(T2, B, A)} >

in step order, within step, no order required.

Exercise 2 (Regression Planning)

Consider a small propositional STRIPS planning problem with a set of propositions $P = \{p, q, r, s\}$, and actions $a_1 \dots a_6$. The preconditions and effects of the actions are described in the following table:

Not everything in a contributes, but at least contributes one.

relevant ✓

not ✗

Action	PRE	EFF$^+$	EFF$^-$	Action	PRE	EFF$^+$	EFF$^-$
a_1 ✓	$\{p, q\}$	$\{r\}$	$\{p\}$	a_4 ✓	$\{r\}$	$\{s\}$	$\{q\}$
a_2 ✗	$\{q\}$	$\{p\}$	$\{q\}$	a_5 ✗	$\{r, s\}$	$\{q\}$	$\{r\}$
a_3 ✓	$\{p, q\}$	$\{s\}$	$\{q\}$	a_6 ✗	$\{s\}$	$\{r\}$	$\{s\}$

① $g \cap Eff^+(a) \neq \emptyset$

② $g \cap Eff^-(a) = \emptyset$

Questions: Action a is relevant to the goal g: a must produce a proposition in g but not delete anything in g.

1. Let g be a goal in regression (backward search) planning, and a be a *propositional* STRIPS action such that $a = \langle \text{PRE}(a), \text{EFF}^+(a), \text{EFF}^-(a) \rangle$. State the condition under which a is relevant to g and explain how to compute the regression of g through a.
2. For each of the 6 actions described above, state whether the goal $\{r, s\}$ can be regressed through that action, and if yes, what the result is.

$g = \{r, s\}$, relevant: a_1 ⇒ $\{s, p, q\}$ repeat

① subtract $Eff^+(a)$ from g

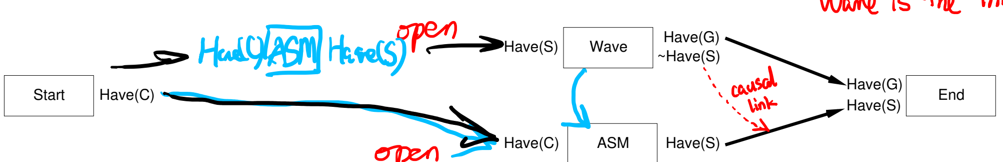
② add $Eff^+(a)$ to g.

⇒ until g is a subset of your updating "initial steps"

Exercise 3 (Partial-Order Planning)

Examine the following partial plan. As in the previous tutorial, the goal is to have a sheep and a goat. The *ASM* (automated sheep machine) operator yields a sheep if you have an ASM card. The *Wave* operator waves your magic wand to turn a sheep into a goat.

Wave is the threat action



1. Which conditions are open (indicate both the condition and the operator for which it is open).
2. List all threats (indicate both the causal link threatened and the threatening action).
3. State how those threats can be resolved.

{1. move D after C
{2. move D before B
3.

D

B p → p C

Order wave before ASM

4. Draw the final plan that the plan-space planning algorithm would produce. Include all new operators, causal links, and ordering constraints required (except those implied by the causal links).
5. Suppose you are executing the resulting plan. What conditions must be true just prior to executing the *Wave* operator in order for the plan to reach the goal?

1. open preconditions.

Have (S) for wave

Have (C) for ASM