Lista 3 - Inteligência Artificial Avançada INF05004

Gustavo Prolla Lacroix
00338354
gustavo.lacroix@ufrgs.br

Henrique Lorentz Trein
00341853
henrique.trein@ufrgs.br

Maio 2025

1

(a) Provide planning tasks in positive normal form with the following characteristics, or justify why no such a task exists:

i) A task Π_1 that is unsolvable, but with Π_1^+ having an optimal plan length of 2.

Let the set of operators of Π_1 be $O = \{o_1, o_2\}$ where $o_1 = \langle T, a \wedge \neg b, 1\rangle$, $o_2 = \langle a, b \wedge \neg a, 1\rangle$, and let $s_0 = \emptyset$ and $s^* = \{a, b\}$.
Clearly Π_1 has no solution since o_2 removes what o_1 adds (and o_1 removes what o_2 adds), and the objective state requires what both of them add.
The relaxed task Π_1^+ has the set of operators $O^+ = \{o_1^+, o_2^+\}$ where $o_1 = \langle T, a, 1\rangle$, $o_2 = \langle a, b, 1\rangle$. Which has an optimal solution $\pi^+ = \langle o_1^+, o_2^+\rangle$ with cost 2.

ii) A task Π_2 with optimal plan length of 2, but such that Π_2^+ is unsolvable.

No such task is possible. By the Relaxation Lemma we know that if π is a sequence of operators in Π that leads to a goal, then π^+ is a sequence of applicable operators that leads to a goal in Π^+ .

iii) A task Π_3 with set of operators $O = \{o_1, o_2, o_3\}$ (to be specified by you) such that Π_3^+ has an optimal plan of length 4.

Let $o_1 = \langle T, a, 1\rangle$, $o_2 = \langle a, b, 1\rangle$, and $o_3 = \langle b, c, 2\rangle$, with $s_0 = \emptyset$ and $s^* = \{a, b, c\}$.

iv) An infinite family of planning tasks $P = \{P_1, P_2, \dots\}$ (e.g. the definition of P_i is parametrized by the value of the integer parameter i) such that the optimal plan length of each task P_i increases with the value of i, but the optimal plan length of any P_i^+ is always 1.

Let P_i, for $i \geq 1$, be such a planning task with:
$s_0 = \{x_0\}$
$s^* = \{x_i\}$
and O containing the following operators:
$o_0 = \langle x_0 \wedge x_1, x_i, 0\rangle$
$o_1 = \langle T, \neg x_0 \wedge x_1, 1\rangle$
$o_2 = \langle x_1, x_2, 1\rangle$
$\dots$
$o_i = \langle x_{i-1}, x_i, 1\rangle$
The optimal solution of P_i is exactly i, while the optimal solution of any P_i^+ is exactly 1. In both planning tasks the only applicable operator from s_0 is o_1. In the original task, o_1 removes the starting variable x_0, which blocks using o_0 to acquire x_i in the next step, forcing the application of the next $i-1$ operators. In the relaxed task o_1^+ does not remove x_0 which allows the application

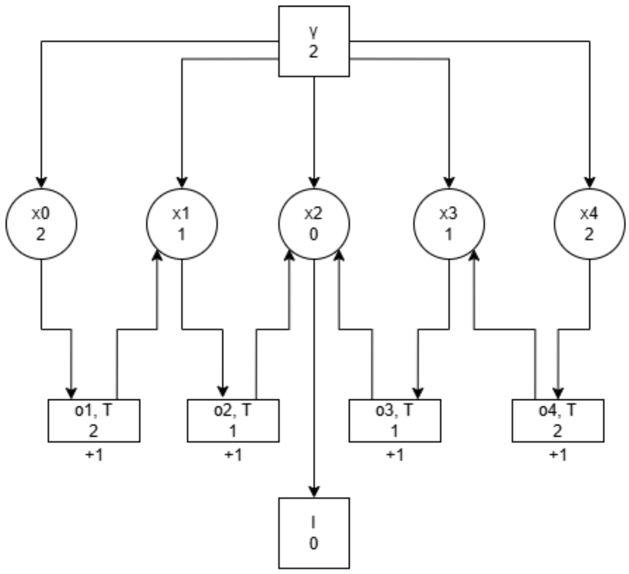
of o_0.

(b) Take the simple instance of the *Visitall* domain (from the International Planning Competition) in directory `visitall-untyped`, and make sure you understand the problem. What is the optimal solution value $h^*(I)$? What is the value of $h^+(I)$? Draw the full Relaxed Task Graph corresponding to the instance, and label each node with the final cost that results from (manually) applying the algorithm seen in class for computing $h^{\max}$. What is the value of $h^{\max}(I)$? Finally, label the graph again, but with the costs that result from h^{add} instead of $h^{\max}$.

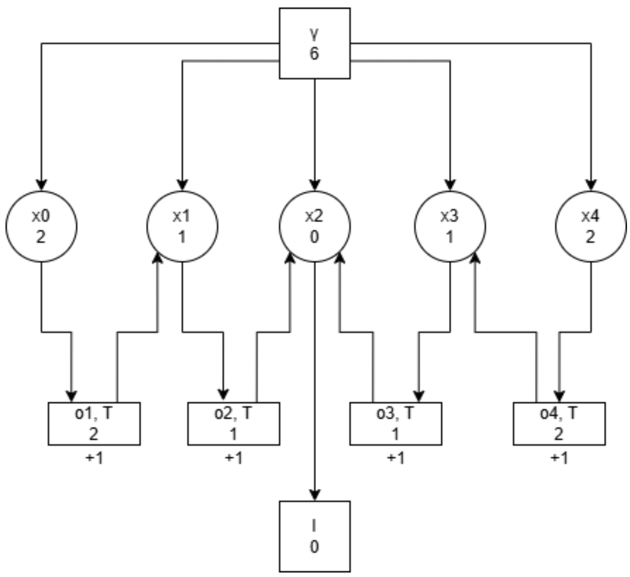
(i) The optimal solution value $h^*(I)$ is 6 because the robot needs to traverse the grid linearly.

(ii) The value of $h^+(I)$ is 4 because the robot doesn't need to revisit cells that have already been visited.

(iii) Relaxed Task Graph for $h^{\max}(I) = 2$:



(iv) Relaxed Task Graph for $h^{\text{add}}(I) = 6$:



2

Table 1 compares number of expansions and time of a blind A* search (A^* columns) and an A^* search that prunes states that are not relaxed solvable (A^{*+} columns). Empty values for plan cost indicate there is no solution. If the search did not terminate with the allocated time and memory constraints, the highest f-value reached is reported.

Table 1: Pruning non relaxed-solvable states

	Plan Cost		Expansions		Time	
Task	A^{*+}	A^*	A^{*+}	A^*	A^{*+}	A^*
castle-5-4-10-cards.pddl	-	-	0	0	0.00	0.18
castle-2-2-8-cards.pddl	4	4	8	8	0.00	0.18
castle-5-4-7-cards.pddl	-	-	6	12	0.02	0.29
castle-3-3-8-cards.pddl	9	9	84	84	0.00	0.21
castle-4-3-5-cards.pddl	15	15	263	263	0.02	0.23
castle-9-6-5-cards.pddl	-	-	206	661	0.14	0.48
castle-12-7-3-cards.pddl	-	-	2520	4585	1.71	0.80
castle-5-4-9-cards.pddl	19	19	7350	7350	0.62	0.30
castle-16-9-1-cards.pddl	-	-	4382	22005	7.70	1.52
castle-6-4-7-cards.pddl	26	26	24022	24146	2.32	0.39
castle-7-5-4-cards.pddl	29	35	282579	3680953	60.60	15.68
castle-8-5-9-cards.pddl	32	39	199741	4839625	60.68	20.12
castle-10-6-7-cards.pddl	28	39	83983	9428904	60.99	60.57

Table 2 compares the initial heuristic value and the plan cost for the planopt and FastDownward implementations of h^{add} and h^{FF} on a GBFS search with the optimal solution of both the task and the relaxed task (Π^* and Π^{+*} respectively). Same comments as above about column values apply.

Table 2: Comparing heuristics

	h^{padd}		h^{add}		h^{pFF}		h^{FF}			
Task	h_0	$c(\pi)$	h_0	$c(\pi)$	h_0	$c(\pi)$	h_0	$c(\pi)$	Π^{+*}	Π^*
castle-2-2-8-cards.pddl	6	4	6	4	4	4	4	4	4	4
castle-3-3-8-cards.pddl	53	9	53	9	11	9	9	9	9	9
castle-4-3-5-cards.pddl	161	15	161	15	21	15	19	15	14	15
castle-5-4-7-cards.pddl	182	-	182	-	28	-	25	-	18	-
castle-5-4-9-cards.pddl	232	21	232	21	25	22	25	21	19	19
castle-6-4-7-cards.pddl	555	26	555	26	36	28	35	30	23	26
castle-7-5-4-cards.pddl	595	40	595	40	42	36	41	38	26	35
castle-8-5-9-cards.pddl	1704	46	1704	46	49	46	48	43	28	39
castle-9-6-5-cards.pddl	2001	-	2001	-	57	-	56	-	34	-
castle-10-6-7-cards.pddl	2257	82	2257	82	63	88	62	72	36	50
castle-12-7-3-cards.pddl	3348	-	3348	-	79	-	79	-	42	-
castle-16-9-1-cards.pddl	10258	-	10258	-	109	-	109	-	60	-
castle-5-4-10-cards.pddl	-	-	-	-	-	-	-	-	-	-