345 Ludic Computing

Tutorial 4

Steering & Pathfinding

Simon Colton and Alison Pease
Computational Creativity Group
Department of Computing
Imperial College London

www.doc.ic.ac.uk/ccg

Question 1

An agent of unit mass is controlled by a Pursue behaviour with $t_{pred} = |\mathbf{q} - \mathbf{p}| / |\mathbf{v}|$ and $t_{lim} = 3$. It has velocity (1,1) at position (4,3) at t = 0. It has a max. speed of 4 and no max. force. The moving target is at (7,6) with a constant velocity (2,0).

a) In general, why do we calculate t_{pred}?

b) Under what (maybe false) assumptions does t_{pred} = time to target?

c) Calculate the force on the agent at t = 0

d) What will the agent and target's positions be after $\Delta t = 0.5$?

Answer 1

a) t_{pred} is how far the agent will look ahead to predict the target's movement.

b) $|\mathbf{q} - \mathbf{p}| / |\mathbf{v}|$ = *time to target* assuming i) the target doesn't move ii) the agent's speed does not change and iii) it heads directly in a straight line to the target.

c) $t_{pred} = |\mathbf{q}_0 - \mathbf{p}_0| / |\mathbf{v}_0| = |(7,6) - (4,3)| / |(1,1)| = 3$

$\mathbf{q}_{pred} = \mathbf{q}_0 + t_{pred}.\mathbf{v}_q = (7,6) + 3.(2,0) = (13, 6)$

$\mathbf{f}_0 = \text{Seek}(13,6) = s_{max}.\text{unit}(\mathbf{q}_{pred} - \mathbf{p}_0) - \mathbf{v}_0$
$= 4.\text{unit}((13,6) - (4,3)) - (1, 1)$
$= (2.795, 0.265)$ [to 3d.p.]

Answer 1

d) $\mathbf{v}_1 = \text{trim}((1, 1) + (2.795, 0.265), 4) = (3.795, 1.265)$

$\mathbf{p}_1 = \mathbf{p}_0 + \Delta t.\mathbf{v}_1 = (4, 3) + 0.5.(3.795, 1.265) = (5.898, 3.633)$

$\mathbf{q}_1 = \mathbf{q}_0 + \Delta t.\mathbf{v}_q = (7, 6) + 0.5.(2, 0) = (8, 6)$

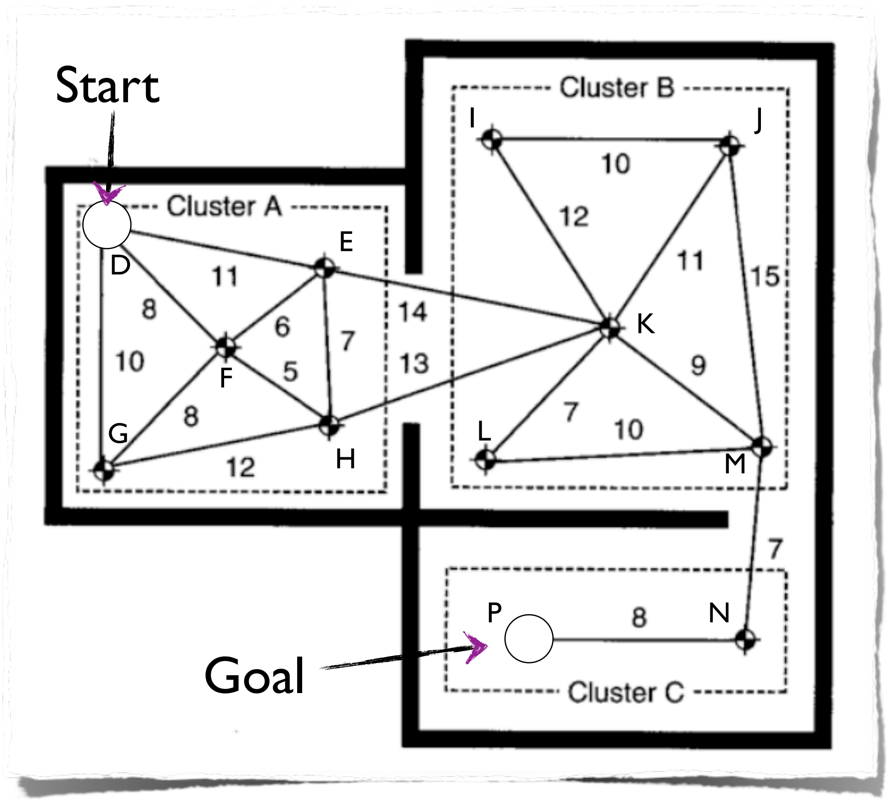
Question 2

On the following navgraph, find a path from D to P using the "estimated straight line distance" heuristic (see table) and...

a) A*

b) HPA*

And c) Which search method expands more nodes? What will happen if we add more rooms between the start and goal?



Estimated distances

	D	E	F	G	H	I	J	K	L	M	N	P
P	28	21	21	23	14	25	26	16	8	15	8	0
N	36	27	29	32	23	27	22	16	15	7	0	
M	33	22	26	31	21	20	15	9	10	0		
L	22	11	13	19	7	15	18	7	0			
K	25	14	19	25	13	12	11	0				
J	30	21	26	33	25	10	0					
I	20	9	16	25	15	0						
H	13	7	5	12	0							
G	10	14	8	0								
F	8	6	0									
E	11	0										
D	0											

Answer 2a

Initial Paths = {D (28)}

Expand D: Paths = {DF (29), DE (32), DG (33)}

Expand DF: Paths = {DFH (27), DE (32), DG (33)}

Expand DFH: Paths = {DE (32), DG (33), DFHK (42)}

Expand DE: DEK cheaper than DFHK, so Paths = {DG (33), DEK (41)}

Expand DG: Paths = {DEK (41)}

Expand DEK: Paths = {DEKL (40), DEKM (49), DEKI (62), DEKJ (62)}

Expand DEKL: Paths = {DEKM (49), DEKI (62), DEKJ (62)}

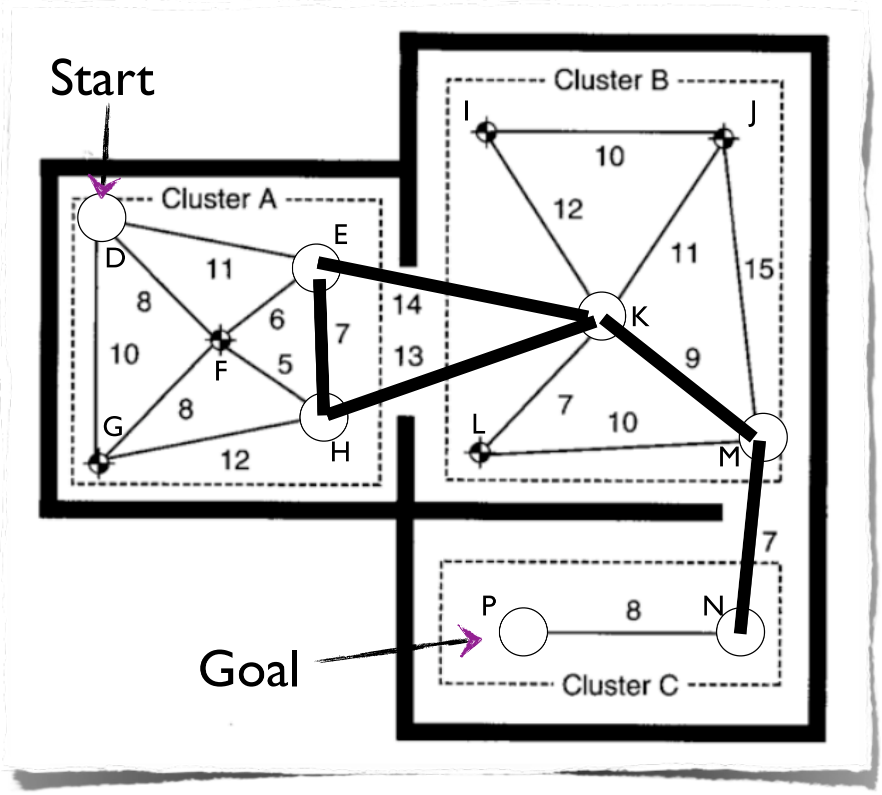
Expand DEKM: Paths = {DEKMN (49), DEKI (62), DEKJ (62)}

Expand DEKMN: solution found DEKMNP (49)

The A* search expanded 9 nodes.

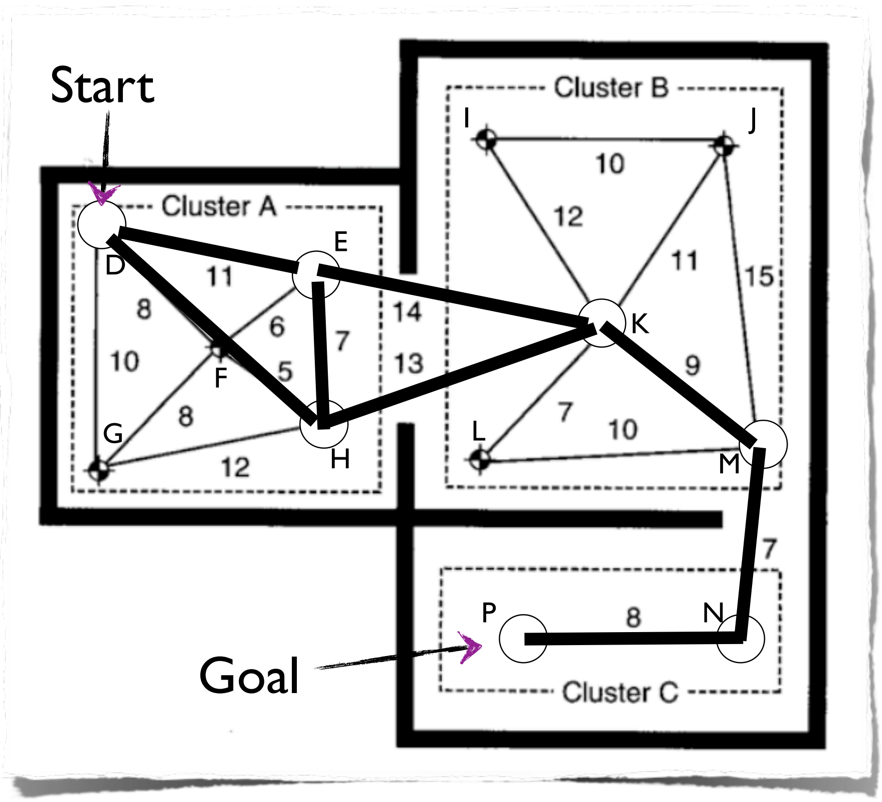
Here we find DEK is a cheaper path to K than our old path DFHK, so DFHK is replaced by DEK

2b



The abstract graph

2b



The abstract graph with start & goal inserted

Answer 2b

Inserting D and P into abstract graph expands 4 nodes. The search in the abstract graph proceeds as follows...

Initial Paths = {D (28)}

Expand D: Paths = {DFH (27), DE (32)}

Expand DFH: Paths = {DE (32), DFHK (42)}

Expand DE: DEK cheaper than DFHK, so Paths = {DEK (41)}

Expand DEK: Paths = {DEKM (49)}

Expand DEKM: Paths = {DEKMN (49)}

Expand DEKMN: solution found DEKMNP (49)

Hence HPA* expands 10 nodes in total.

Answer 2c

- A* expanded 9 nodes and HPA* 10 nodes.
- As more rooms are added A* will encounter more intermediate nodes than HPA*, whereas the cost of inserting the start/goal into HPA*'s abstract graph will remain fixed. So HPA* will tend to be better for bigger maps.

Question 3

a) How many bytes would it take to store the entries for a 'next node' look-up table for Q2's navigation graph? (with a constant number of bits per node)

b) How many if we use a next edge table instead?

c) How many for an edge look-up table for the HPA* abstract graph, and separate tables for the 3 clusters?

Answer 3

a) 12 node labels. 12 x 12 = 144 entries. Need 4 bits to store each node label. So we need 144 x 0.5 = 72 bytes

b) Max. node degree is 6, so it only takes 3 bits per entry. 144 x 3 / 8 = 54 bytes

c)

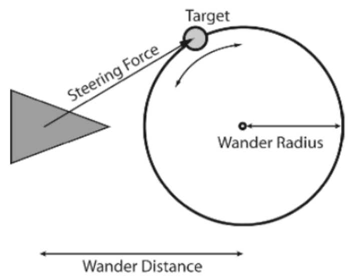
	Abs	**A**	**B**	**C**
Max. degree	3	4	4	1
Bits per entry	2	2	2	0
Node Labels	5	5	5	2
Entries	25	25	25	4
Bits	50	50	50	0

It only takes 150 bits (18.75 bytes) for the HPA* look-up tables

Question 4

The Wander steering behaviour is parameterised by the wander radius and distance, and the target jitter.

For each parameter, what effect does increasing it have on the movement of the agent?



Answer 4

- Increasing radius: wider and slower turns
- Increasing distance: narrower and slower turns
- Increasing jitter: faster turns