Artificial Intelligence

Assignment 6

Assignment due by: 6.12.2017, Discussion: 8.12.2017

Question 1 Simulated annealing (1+3+2=6 points)

(a) Explain with your own words in less than three lines the principle and the purpose of simulated annealing. (1 point)

(b) Using the following function,

$$E(x) = \exp\left(-\frac{1}{2}\left(\frac{x-0.2}{0.15}\right)^2\right) + 2\,\exp\left(-\frac{1}{2}\left(\frac{x-0.7}{0.1}\right)^2\right) \tag{1}$$

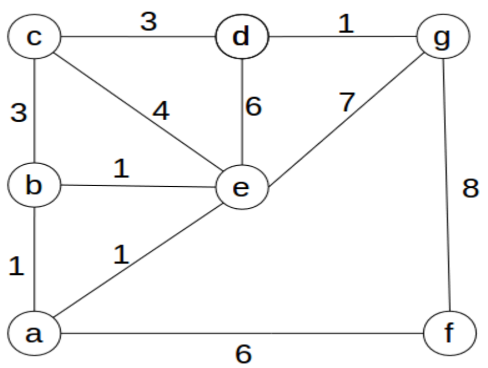
apply the first steps of simulated annealing in order to find the global maximum. Please fill in the table. The acceptance probability for this question is : $P(x_n+\Delta x_n, x_n) = \exp(-\frac{|E(x_n+\Delta x_n)-E(x_n)|}{T_n})$ if $E(x_n + \Delta x_n) < E(x_n)$, and 1 otherwise. If the move is accepted $x_{n+1} = x_n + \Delta x_n$, otherwise $x_{n+1} = x_n$. The starting point is $x_0 = 0.85$. You are required to use the sequence of steps Δx_n in the table to move along x. The sequence r_n displayed in the table should be used in place of the usual random numbers to carry out the algorithm (i.e. no need to generate your own random samples). (3 points)

n	x_n	Δx_n	$E(x_n)$	$E(x_n + \Delta x_n)$	T_n	$P(x_n + \Delta x_n, x_n)$	r_n
0	0.85	-0.1	0.65	1.77	1.6	1	0.3
1	0.75	-0.15	1.77		0.8		0.7
2		-0.5			0.4		0.01
3		0.1			0.2		0.1
4		0.2			0.1		0.8
5		0.05			0.05		0.6

(c) Considering the state reach in the fifth step and assuming $T_{n+1} = 0.5\,T_n$, do you think the algorithm is likely to reach the maximum of the function $E(x)$? Justify your answer. (2 points)

Question 2 Learning Real-Time A* (2+3+2=7 points)

Solve this question using the graph shown below. The initial node is a, the target node is g. In case you find more than one reasonable path, please follow an alphabetic order.



n	$h(n)$
a	3
b	3
c	3
d	1
e	1
f	1
g	0

(a) Find a path using the graph-search version of A*. Show the process to get such path.

(b) An agent that knows the possible actions in each state, as well as the outcome and cost of these actions is given. Apply LRTA* and fill in the table below, initialize $H(n)$ with $h(n)$.

a	s	s'	$H[a]$	$H[b]$	$H[c]$	$H[d]$	$H[e]$	$H[f]$	$H[g]$
-	-	-	3	3	3	1	1	1	0
-	-	a	3						
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(c) Applying the heuristic H found in task b, find a path using the graph-search version of A*. Show the process to get such path. Compare the number of iterations with task a. What can you conclude?

Question 3 Implementing Learning Real-Time A* (5+2 = 7 points)

The goal of this exercise is to implement Learning Real-Time A*. You may do so in either LISP or Java. If you decide to use LISP, the process is the same as in previous assignments. You should download either the template *graphsearch-lras.lisp* or *Graph.java*.

If you decide to use Java, you must hand in your source files AND a compiled jar that runs the algorithm on the start/goal specified on the command line and outputs the total path distance traveled and the path it took. The source must compile with a Java 8 compiler when I open the source directory with eclipse (feel free to edit *Graph.java* to change the package or put the heuristic in there, etc.). You may not use external libraries/dependencies for any Java assignments.

A note on terminology: in this graph, the actions possible at a node are the target cities that can be reached from there. Therefore, actions and their resulting states can be treated equivalently.

(a) Implement Learning Real-Time A* in LISP/Java, using the straight line heuristic. Test your program on the following paths: Kiel → Essen, Aachen → Frankfurt/Oder, Freiburg → Schwerin.

(b) Look at the paths traveled by the algorithm, are they optimal? If not, explain why.