Marc Tomasek
Marius Hobbhahn

1	2	3	Σ

Übungsblatt Nr. 6

(Abgabetermin 06.12.2017)

Aufgabe 1

a

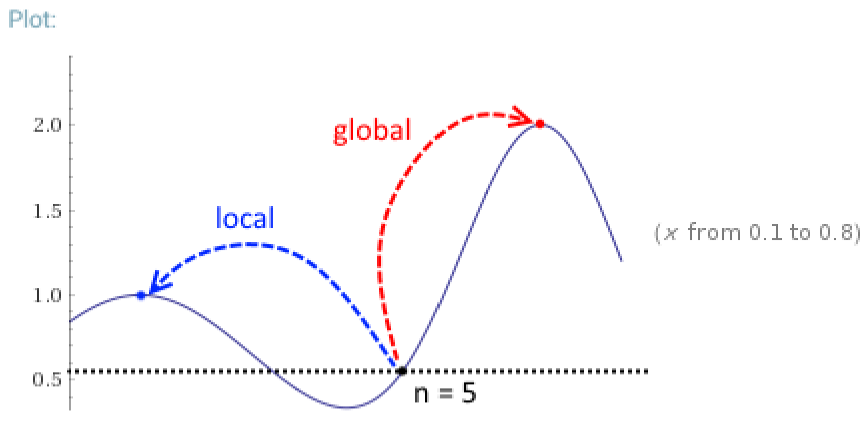
Simulated annealing is a hillclimbing variant which always allows 'uphill' moves, but also allows 'downhill' moves with a probability p, which is exponentially reduced with a decreasing 'temperature' T. It can sometimes find the global maximum where normal hillclimbing can't.

b

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	1.24	0.8	0.52	0.7
2	0.75	-0.5	1.77	0.95	0.4	0.13	0.01
3	0.25	0.1	0.95	0.61	0.2	0.19	0.1
4	0.35	0.2	0.61	0.72	0.1	1	0.8
5	0.55	0.05	0.72	1.24	0.05	1	0.6

c

We are on the fifth step at the point $(0.55, 0.72)$.



Because of $T_{n+1} = 0.5 \cdot T_n$, acceptance of values $E(x_{n+1}) < E(x_n)$ will become more and more rare, which leads to a low probabilty of moving away from the global maximum.
The chance of choosing values $E(x_{n+1}) > E(x_n)$ is uninfluenced and remains 1, which likely leads to a steady ascent towards the global maximum at $(0.7, 2.0)$ if we get small step-sizes.
If in the unlikely event of a big negative step-size, for example $\Delta x_n = -0.3$ (where $E(x_{n+1}) > E(x_n)$), we could end up on the slopes of the local maximum at $(0.2, 1.0)$, which could lead to us not finding the global maximum, if no similar event occurs to get us back on the other slope (within our allowed runtime).

If our runtime is infinite, we will always find the slopes of the global maximum from the local

maximum with a step-size $0.38 < \Delta x_n < 0.61$.
In conclusion, it is likely that we find the global maximum.

Aufgabe 2

a

Tabelle 1: Graph-search A*

explored nodes (and their parent)	**frontier f-Values**	**best node**
a	e = 2, b = 4, f = 7	e = 2
a, e(a)	b = 4, f = 7, c = 8, d = 8, g = 8	b = 4
a, e(a), b(a)	c = 7, f = 7, d = 8, g = 8	c = 7
a, e(a), b(a), c(b)	f = 7, d = 8, g = 8	f = 7
a, e(a), b(a), c(b), f(a)	d = 8, g = 8	d = 8
a, e(a), b(a), c(b), f(a), d(c)	g = 8	g = 8
a, e(a), b(a), c(b), f(a), d(c), g(d)	-	-

The path from a to g is as follows: $a \to b \to c \to d \to g$ and is of length 8
This path is preferred to $a \to e \to g$ due to alphabetical order when selecting the next node for expansion.

b

The last row describes the final heuristic

Tabelle 2: a describes edge taken from current node s' to next node s

a	s'	s	H[a]	H[b]	H[c]	H[d]	H[e]	H[f]	H[g]
-	a	-	3	3	3	1	1	1	0
(a,e)	a	e	3	3	3	1	**4**	1	0
(e,a)	e	a	**4**	3	3	1	4	1	0
(a,b)	a	b	4	**4**	3	1	4	1	0
(b,a)	b	a	**5**	4	3	1	4	1	0
(a,b)	a	b	5	**5**	3	1	4	1	0
(b,e)	b	e	5	5	3	1	**6**	1	0
(e,a)	e	a	**6**	5	3	1	6	1	0
(a,b)	a	b	6	**6**	3	1	6	1	0
(b,c)	b	c	6	6	**4**	1	6	1	0
(c,d)	c	d	6	6	4	1	6	1	0
(d,g)	d	g	6	6	4	1	6	1	0

c

Tabelle 3: Heuristic $H(n)$

n	$H(n)$
a	6
b	6
c	4
d	1
e	6
f	1
g	0

Tabelle 4: Graph-search A^* with heuristic $H(n)$

explored nodes (and their parents)	**frontier f-Values**	**best node**
a	b = 7, e = 7, f = 7	b
a, b(a)	e = 7, f = 7, c = 8	e
a, b(a), e(a)	f = 7, c = 8, d = 8, g = 8	f
a, b(a), e(a), f(a)	c = 8, d = 8, g = 8	c
a, b(a), e(a), f(a), c(b)	d = 8, g = 8	d
a, b(a), e(a), f(a), c(b), d(c)	g = 8	g
a, b(a), e(a), f(a), c(b), d(c), g(d)	-	-

Similar to the worse heuristic, it also took 6 steps to calculate the path from a to g.
The result is the same as the previous calculation $a \to b \to c \to d \to g$ with a length of 8.