CS540: HW2 (P2)

Keith Funkhouser
wfunkhouser@cs.wisc.edu

February 22nd, 2016

(a) There are n successors: one corresponding to each boolean value which can be "flipped".

(b) There are 2^n total states. If we let $\langle x_1, x_2, \dots x_n \rangle$ denote a given space, where each $x_i \in \{0, 1\}$, then x_1 has two possible values, as do x_2 through x_n. Since they each take values independently, there are $2 \times 2 \times \cdots \times 2 = 2^n$ total states.

(c) From a starting state of $\langle 0, 1, 0, 0 \rangle$ (with value 3), the possible successor states and their values are:

- $\langle 1, 1, 0, 0 \rangle$: 4
- $\langle 0, 0, 0, 0 \rangle$: 4
- $\langle 0, 1, 1, 0 \rangle$: 3
- $\langle 0, 1, 0, 1 \rangle$: 4

Hence, the successor state selected by the hill-climbing algorithm will a random selection between any of $\langle 1, 1, 0, 0 \rangle$, $\langle 0, 0, 0, 0 \rangle$, or $\langle 0, 1, 0, 1 \rangle$.

A global optimal solution will not always be found. For example, since ties are broken randomly, if $\langle 0, 0, 0, 0 \rangle$ is selected, then the successor states and the number of clauses they satisfy is as follows:

- $\langle 1, 0, 0, 0 \rangle$: 4
- $\langle 0, 1, 0, 0 \rangle$: 3
- $\langle 0, 0, 1, 0 \rangle$: 4
- $\langle 0, 0, 0, 1 \rangle$: 4

Hence, the initial state $\langle 0, 0, 0, 0 \rangle$ leads to a local optimum of 4 and does not reach a global optimal solution of 5, achived by $\langle 1, 1, 0, 1 \rangle$.

(d) Starting from an initial state of $\langle 1, 0, 1 \rangle$ (which is not a goal state, as it satisfies only 3 of the 4 clauses), the successor states and the number of clauses they satisfy are:

- $\langle 0, 0, 1 \rangle$: 3
- $\langle 1, 1, 1 \rangle$: 3
- $\langle 1, 0, 0 \rangle$: 3

Hence, the state $\langle 1, 0, 1 \rangle$ is a local optimum in the hill-climbing space.

(e) Since *up to* 2 variables can change their values, we will consider two cases.

In the first case, we change only one variable, and there are exactly n new successor states generated as described in part (a).

In the second case, we change exactly two variables, and there are exactly $\binom{n}{2}$ ways to pick which two variables will be changed.

Hence, there are $n + \binom{n}{2} = n + \frac{n!}{2!(n-2)!} = n + \frac{n(n-1)}{2}$ successor states that can be generated.