# Homework 2

## Artificial Intelligence

## CS 540 Section 3

Ritvik Upadhyaya

# Contents

# 1 Genetic Algorithms

## 1.a Mating Events

Case 1:
Mating between chips 4 and 6: 0000110011 & 0111110010

Case 2:
Mating between chips 1 and 3: 0100010010 & 1101110011

Case 3:
Mating between chips 5 and 6: 1000001111 & 0100110001

## 1.b Mutation Events

Mutation events are rare occurances of errors during the copying of chromosomes that can result in the following characteristics:

- changes that make the produced chip/organism worse off or "weaker" than the original one.
- changes that make the produced chip/organism better off or "stronger" than the original one.
- changes that make the produced chip/organism neither better nor worse off than the original one.

These events are used to get one or more members of a population out of a local minimum or maximum space and discover new potential minimum and maximum spaces that might be better.

## 1.c GA vs Hillclimbing Algorithm

The fact that unlike in traditional hillclimbing algorithms, mutation and crossover move the population away from the local optima in genetic algorithms. This is the reason why genetic algorithms do not stay at the local optima and are also not susceptible to it.

# 2 Simulated Annealing

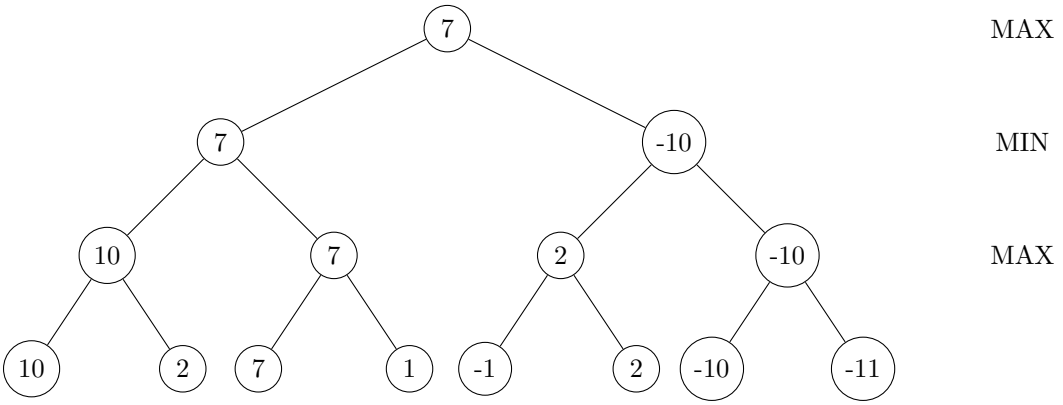
## 2.a Table

Table 1: Question 2

| # of Iterations | ΔE | ΔE>0 | Temperature (T) | $e^{E/T}$ (Random #) | If $e^{E/T}$ > Random number | Decision |
|---|---|---|---|---|---|---|
| 1 | 1 | Yes | 1.8 | 1.743 | Yes | Go to 3 |
| 2 | 2 | Yes | 1.62 | 3.436 | Yes | Go to 1 |
| 3 | 0 | Yes | 1.458 | 1 | Yes | Go to 1 |
| 4 | -3 | No | 1.312 | 0.101 | No | Stay at prev (1) |
| 5 | 2 | Yes | 1.181 | 5.44 | Yes | Go to 2 |
| 6 | -1 | No | 1.063 | 0.390 | No | Stay at prev (2) |
| 7 | -2 | No | 0.957 | 0.123 | No | Stay at prev (2) |
| 8 | -1 | No | 0.861 | 0.312 | No | Stay at prev (2) |

# 3 Game Playing

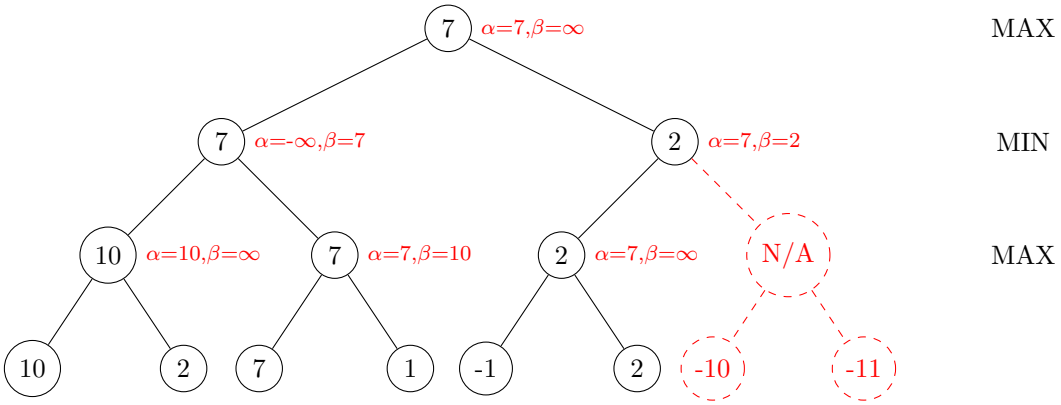
## 3.a Minimax




The tree shown above contains all the values in the corresponding nodes in accordance to Minimax algorithm.

## 3.b $\alpha$-$\beta$ Pruning

Alpha-Beta Pruning Tree




**Note:** The pruned branches are shown in red dotted lines and curves.

## 3.c Use of $\alpha$-$\beta$ Pruning

We use $\alpha$-$\beta$ Pruning in order to prevent the algorithm from traversing unnecessary branches of the search tree. This helps us to not waste time and use the time to perform a deeper search at the same time (reduce the effective depth of the search tree).