COMP3121: Assignment 3 – Q2

Gerald Huang

$z5209342$

Updated: July 16, 2020

Given the values for R_a, R_b, P_a, P_b, S_a and S_b, we will first give a general algorithm that minimises the number of losses, which in turn maximises the number of points won, and finally *show* that this produces an optimal solution. That is, *no other algorithm can be any better than our current algorithm.*

General algorithm

Let N be the number of rounds played between Player A and Player B. The idea is that we want to *minimise* the number of losses. We do this by maximising the number of wins and then playing draws if possible. As such, we reserve as many paper for R_a, scissors for P_a and rock for S_a. In each round, Player B will play their most optimal hand possible given the plays that A has made. B applies a greedy search to find a play that wins the round. For example, if A plays a rock, then B will apply a greedy search that looks for a paper in their hand. If that search returns a match, then B will play that hand which wins the round. However, if B doesn't find a match, then the next best option is for B to draw so that they do not lose any points. B will apply another greedy search to find a rock. If that search returns a match, then B will play that hand which draws the round. If this search fails, then the only option is to lose that round in which case, B will play scissors. We will be guaranteed at least one scissors because we attain the equality $R_b + P_b + S_b = N$. So if there are no more beneficial plays, there will *always* be a losing hand to play. This will be the basis for our optimality proof.

Proof of optimality

We will now prove that this algorithm produces an optimal solution. It should be noted that this algorithm may not be *the only* optimal algorithm but no other strategy will exceed this algorithm. We proceed with a **proof by contradiction**.

Define our algorithm to be algorithm X played with N rounds. Assume that there exists a strategy X^* that is *more* optimal than X. Then, at some round $k > 0$, a loss in X must have been either a win or a draw in X^*, or a draw in X must have been a win in X^*. We will define our solution space produced by X to be $S_N = \{s_1, s_2, \dots, s_k, \dots, s_N\}$. Let k be the index of the first iteration where X finds the suboptimal solution. Then define the solution space produced by X^* to be $T_N = \{s_1, s_2, \dots, s_{k-1}, s'_k, s'_{k+1}, \dots, s'_N\}$ since this means that our algorithm produced optimal choices for the first $(k-1)$ iterations. We'll consider the two different scenarios where X^* is more optimal than X.

- **Case 1**: X draws and X^* wins. Then there must have been a round $0 < n < k$ that we didn't play our most optimal hand because at the kth round, it must have been possible to win and our greedy algorithm didn't pick up. However, the previous $(k-1)$ hands have been optimal as per our algorithm. This is a contradiction and so in this case, X^* is only as good as X.
- **Case 2**: X loses while X^* either draws or wins. Again, this assumes that there has been no optimal play

possible which yielded in a loss in X. But this cannot be possible since it follows that X^* made a more optimal move. Since X and X^* played optimal hands, then it is that X^* would have traded its win and draw now for a loss later on which ultimately yields in a net change of zero. This works because the sum of all of R_b, S_b and P_b is N. As such, a win or draw now would lead to a loss later on. This contradicts the optimality of X^* so in both cases, X^* is *only as good* as X.

In both of these cases, we see that X^* is only as good as X. Since X^* is optimal, then it follow that X is also optimal.