Problem Set no. 4

Given: December 26, 2019

Exercise 4.1 Suppose that the randomized strategy iteration algorithm (the "dual" algorithm on slide 7, lecture 5) is run on a game G with an initial strategy σ of player 0. An action a in σ is said to be *frozen* iff $Val^{-}(a) < Val(\sigma)$, i.e., if all strategies that do not use a are inferior to σ. Suppose that k actions $a_1, a_2, \ldots, a_k$ in σ are frozen. Consider the subgame G' obtained by removing from G all the alternatives of $a_1, a_2, \ldots, a_k$, and considering the states from to which actions $a_1, a_2, \ldots, a_k$ belong as states of player 1. Let σ' be the strategy of player 0 in G' obtained by removing $a_1, a_2, \ldots, a_k$ from σ. Show that the distribution of the number of improving switches performed by the algorithm on (G, σ) is identical to this distribution on (G', σ'). In particular, the expected number of improving switches performed is the same.

Exercise 4.2 Suppose that the expected number of improving switches performed by the "dual" randomized algorithm is bounded by $f(m_0, n_0)$, where m_0 is the number of actions and n_0 is the number of states of player 0. Suppose that evaluating a strategy σ of player 0 takes $h(m_1, n_1, n_0)$ time, where m_1 and n_1 are the number of actions and states of player 1. Suppose that checking whether an action a is an improving switch with respect to σ, after σ was evaluated, takes $O(1)$ time. Obtain an upper bound on the expected total running time of the algorithm.

Exercise 4.3 Obtain an upper bound on the number of improving switches performed by the "primal" randomized algorithm. (See slide 6 of lecture 5.) More specifically, let $g(m, n)$ be a function that satisfies the following recurrence relation: $g(m, n) = g(m-2, n-1) + \frac{1}{n}\sum_{i=1}^{n}(1 + g(m-i, n))$, if $m \geq 2n$, $g(m, n) = g(2(m-n), m-n)$, if $m < 2n$, and $g(m, 0) = 0$. Prove by induction that $g(m, n)$ is an upper bound on the number of improving switches performed by the algorithm on any game with n states and m actions. (Hint: For any action $a \in A$, let $Val^{+}(a)$ be the value of the optimal strategy that uses a. Let $a_1, a_2, \ldots, a_n$ be the actions in the current strategy such that $Val^{+}(a_1) \leq Val^{+}(a_2) \leq \cdots \leq Val^{+}(a_n)$. Continue along the lines of the analysis of the "dual" randomized algorithm.)

Exercise 4.4 Let $f(m, n) = f(m-1, n) + \frac{1}{m-n}\sum_{i=1}^{n}(1 + f(m-2i, n-i))$, if $m \geq 2n$, $f(2n-1, n) = f(2n-2, n-1)$, $f(m, 0) = 0$ be the recurrence relation obtained by analysing the "dual" algorithm. (a) What is $f(m, 1)$? (b) Estimate the asymptotic behavior of $f(m, 2)$.

Exercise 4.5 Show that the following algorithm is a non-recursive version of the "dual" randomized algorithm given in class: Choose a random permutation $a_1, a_2, \ldots, a_{m-n}$ on the actions that are not used in the initial strategy σ. For $i = 1, 2, \ldots, m-n$ check whether a_i is an improving switch with respect to σ. If no improving switch is found, then σ is an optimal strategy. Otherwise, if a_i is the first improving switch, let $\sigma' = \sigma[a_i]$, and let a' be the action in σ that a_i replaced. Let $a'_1, a'_2, \ldots, a'_i$ be a random permutation of $a_1, a_2, \ldots, a_{i-1}, a'$. Continue in the same manner with σ' and the permutation $a'_1, a'_2, \ldots, a'_i, a_{i+1}, \ldots, a_{m-n}$ on the actions not used by σ'.

Exercise 4.6 Consider the following non-recursive randomized algorithm for binary games, i.e., games in which each state has exactly two actions. Choose a random permutation $s_1, s_2, \ldots, s_n$ on the states of the game. If σ is the current strategy, then for $i = 1, 2, \ldots, n$ check whether switching the action used in s_i yields an improvement. If no improvement is found after considering all states, then σ is optimal. Otherwise, let σ' be the strategy obtained by performing a switch on s_i, where i is the smallest index for which the switch is improving. Continue with σ' using the *same* permutation $s_1, s_2, \ldots, s_n$. Show that:

(a) The algorithm is not identical to the recursive randomized algorithm for binary games.

(b) Nevertheless, the expected number of improving switches performed by the two algorithm on every binary game is the same.