

# Programming Assignment 2

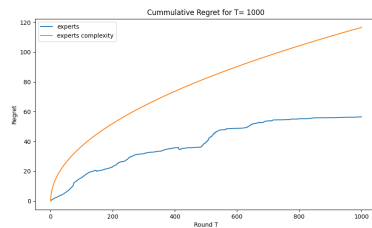
Ζαφειράκης Κωνσταντίνος 2019030035

## Εισαγωγή

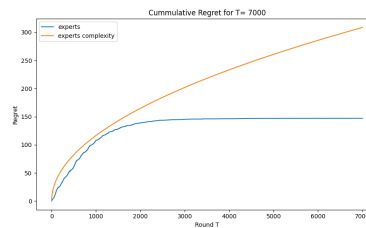
Ζητήθηκε να εφαρμόσουμε τον αλγόριθμο Multiply Weights Algorithm για την εύρεση του σέρβερ με το μικρότερο φόρτο σε ένα περιβάλλον με 30 σέρβερ και 7000 τιμές για κάθε σέρβερ. Ο αλγόριθμος MWA λειτουργεί με επαναληπτική προσαρμογή των βαρών που αποδίδονται σε ένα σύνολο πιθανών αποφάσεων, με βάση τα αποτελέσματα προηγούμενων επαναλήψεων. Τα βάρη πολλαπλασιάζονται με έναν συντελεστή που εξαρτάται από την απόδοση κάθε απόφασης στην προηγούμενη επανάληψη, με στόχο την αύξηση των βαρών που ανατίθενται σε αποφάσεις με καλές επιδόσεις και τη μείωση των βαρών που ανατίθενται σε αποφάσεις με κακές επιδόσεις. Ο αλγόριθμος αυτός εφαρμόστηκε σε περιβάλλον experts δηλαδή σε κάθε χρονική στιγμή είναι γνωστά τα φορτία όλων των σέρβερ όπου και είχε πολυπλοκότητα  $O(2 \cdot \sqrt{T \cdot \ln k})$  και σε περιβάλλον bandits όπου σε κάθε χρονική στιγμή είναι γνωστό μόνο το φορτίο του σέρβερ που έχει επιλεγεί με τα φορτία των άλλων σέρβερ άγνωστα όπου είχε πολυπλοκότητα  $O(\sqrt{k \cdot T \cdot \ln k})$ . Ακόμη εφαρμόστηκε ο αλγόριθμος UCB αλλά εφόσον στο πρόβλημα ζητούνταν να επιλέγονται οι χαμηλότερες τιμές και όχι οι υψηλότερες έτσι στο estimate  $\mu$  του UCB αφαιρούσαν οι τιμές των σέρβερ γιατί καθώς ο UCB από μόνος του βρίσκει τις μεγαλύτερες τιμές εάν έχει ως είσοδο τις αρνητικές τιμές η μεγαλύτερη θα είναι αυτή που έχει την μικρότερη απόλυτη τιμή άρα η μικρότερη τιμή που ζητείται.

## Expert Environment

Σε περιβάλλον Expert σε κάθε χρονική στιγμή επιλέγεται ένας σέρβερ με βάση την πιθανότητα που έχει η οποία είναι το βάρος του διαιρεμένο με τα βάρη όλων των σέρβερ. Με το που επιλεγεί υπολογίζεται το loss ως η διαφορά της τιμής του σέρβερ με την μικρότερη τιμή εκείνη την χρονική περίοδο και με βάση το loss υπολογίζεται το βάρος στην επόμενη χρονική στιγμή όλων των σέρβερ για το δικό τους loss. Το regret υπολογίζεται ως η διαφορά της τιμής του σέρβερ που επιλέχθηκε σε κάθε γύρο με την τιμή του σέρβερ που έχει το μικρότερο αθροιστικό Load συνολικά. Στα παρακάτω διαγράμματα φαίνεται το συνολικό regret για  $T=1000$  και  $T=7000$  παρατηρείται ότι για  $T=1000$  το regret είναι πρακτικά γραμμικό αυτό συμβαίνει καθώς ο αλγόριθμος κάνει ακόμη explore ενώ στο διάγραμμα  $T=7000$  όπως φαίνεται λίγο πριν από τον γύρο 2000 ο αλγόριθμος έχει μάθει και πλέον κάνει περισσότερο exploit έως ότου το regret σταθεροποιηθεί καθώς ο αλγόριθμος έχει βρει τον βέλτιστο σέρβερ. Εν τέλει Ο αλγόριθμος μαθαίνει και είναι sublinear



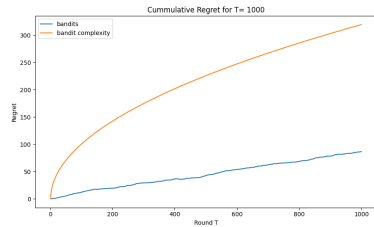
experts convergence  $T=1000$



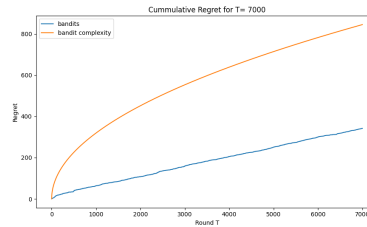
experts convergence  $T=7000$

## Bandit Environment

Οι διαφορές του περιβάλλοντος Bandit σε σχέση με το Expert είναι ότι στο Bandit ο σέρβερ σε κάθε χρονική στιγμή επιλέγεται με βάση την  $Q$  η οποία επηρεάζεται από την  $P$  και σε κάθε χρονική στιγμή ανανεώνεται μόνο ένα loss και ένα weight της επόμενης χρονικής στιγμής, αυτό του σέρβερ που έχει επιλεγεί. Προκύπτουν οι παρακάτω γραφικές παρατηρείται ότι ο αλγόριθμος χρειάζεται παραπάνω δεδομένα για να βρει τον βέλτιστο σέρβερ και να ομαλοποιηθεί το regret



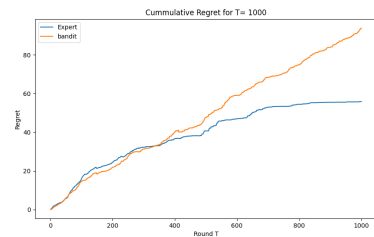
bandits convergence  $T=1000$



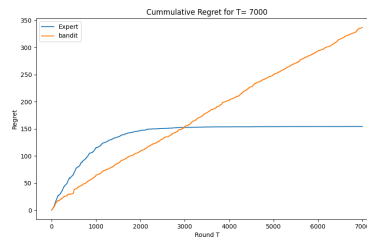
bandits convergence  $T=7000$

## Bandit - Expert Συγκριση

Παρακάτω συγκρίνονται οι MWA σε bandit και expert περιβάλλον και παρατηρείται ότι οι γραφικές για  $T=1000$  και  $T=7000$  διαφέρουν στην μορφή τους και για τα πρώτα 1000 rounds της  $T=7000$  δέν είναι τα ίδια με αυτά της  $T=1000$  αυτό συμβαίνει γιατί αλλάζοντας το  $T$  αλλάζουν και οι μεταβλητές  $\eta$ ,  $\epsilon$  που εξαρτιούνται από το  $T$ . Ακόμη βλέπουμε ότι και στις δύο γραφικές το expert είναι καλύτερο καθώς ομαλοποιείται πιο γρήγορα από το bandit μάλιστα για  $T=7000$  στο τέλος μένει σταθερό το regret αφού έχει βρεί τον βέλτιστο σερβερ σε αντίθεση με τον bandit ο οποίος είναι ακόμη σε φάση explore



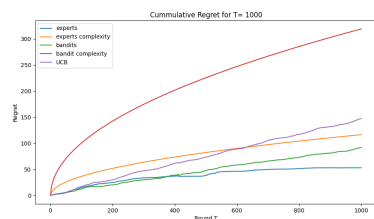
convergence  $T=1000$



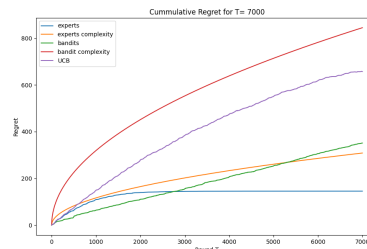
convergence  $T=7000$

## MWA - UCB Σύγκριση

Ο αλγόριθμος UCB έχει χειρότερη απόδοση απο τον MWA και σε περιβάλλον experts άλλα και σε bandit και για τις δύο περιπτώσεις  $T=1000$  και  $T=7000$  όπως φαίνεται και στα παρακάτω σχήματα καθώς το αθροιστικό regret του UCB είναι κατά πολύ μεγαλύτερο σε σχέση με τους άλλους δύο και σε καμία χρονική στιγμή δεν υπερτερεί ο UCB καθώς έχει το μεγαλύτερο explore phase. Ακόμη ο UCB θα ήθελε ακόμη μεγαλύτερο  $T$  για να βρει τον βέλτιστο σέρβερ και να κάνει exploit όπως ο MWA expert.



convergence  $T=1000$



convergence  $T=7000$