2ο Σετ Ασκήσεων: Ισορροπίες Nash σε Διμητρικά Παιχνίδια

Μπάλλας Μιχαήλ

Α.Μ. 1072599

up1072599@upnet.gr

Σούρλας Ζήσης

Α.Μ. 1072477

sourlas.zisis@upnet.gr

15/06/2023

Θεωρητικό Ερώτημα 1 (Γεννήτρια Παιχνιδιών)

Θ.Ε. 1.1

Εφόσον $x^T Ry = 0$ σημαίνει ότι ο παίκτης στήλης είτε έπαιξε αμιγώς μια στήλη η οποία στην πρώτη γραμμή περιέχει ένα 0,1-στοιχείο είτε ότι έπαιξε μια μικτή στρατηγική «μιξάροντας» μεταξύ στηλών που στην πρώτη γραμμή περιέχουν (0,1) και (0,0)-στοιχεία. Σε κάθε περίπτωση αποκλείεται να έπαιξε (είτε αμιγώς είτε μεικτά) σε στήλη που έχει (1,0) στοιχείο στην πρώτη γραμμή καθώς αυτό θα σήμαινε την ανάθεση πιθανότητας σε στήλη που αποδίδει όφελος στην παίκτρια γραμμής. Ωστόσο η γεννήτρια παιχνιδιών είναι κατασκευασμένη έτσι ώστε κάθε στήλη να έχει τουλάχιστον ένα (1,0) στοιχείο. Συνεπώς η παίκτρια γραμμής θα μπορούσε να παίξει σε μια γραμμή που να περιλαμβάνει τουλάχιστον ένα (1,0) στοιχείο σε στήλη που έχει επιλέξει ο παίκτης στήλης. Τέτοια γραμμή είναι εγγυημένο ότι υπάρχει. Σε αυτή την περίπτωση η ωφέλεια της παίκτριας γραμμής θα ήταν μεγαλύτερη του μηδενός και κατά συνέπεια στην παρούσα κατάσταση δεν βρίσκεται σε κατάσταση ισορροπίας.

Θ.Ε. 1.2

Για να ισχύει ότι $x^T Ry > 0$ θα πρέπει ο παίκτης στήλης να έχει επιλέξει είτε μια αμιγή στρατηγική σε στήλη που έχει (1,0)- στοιχείο στην πρώτη γραμμή (δηλαδή αυτή που επέλεξε η παίκτρια γραμμής) είτε μια αμιγή στρατηγική που να «μιξάρει» μεταξύ στηλών με (0,0), (0,1) και (1,0)- στοιχεία στην πρώτη γραμμή. Σε κάθε περίπτωση έχει αναθέσει κάποια (ή όλη) μάζα πιθανότητας σε στήλη που περιέχει (1,0) στοιχείο στην πρώτη γραμμή εξ ου και η θετική ωφέλεια για την παίκτρια γραμμής. Ο παίκτης στήλης θα μπορούσε να αυξήσει την ωφέλειά του (η οποία μπορεί να είναι θετική ή μηδέν) μεταθέτοντας την μάζα πιθανότητας που ανάθεσε σε στήλες με (1,0) και (0,0) στοιχεία σε στήλες με (0,1)-στοιχεία στην πρώτη γραμμή. Αν δε υπάρχει μόνο μία τέτοια στήλη θα μπορούσε να παίξει αμιγώς αυτή. Είναι επιπλέον εγγυημένο ότι υπάρχει τουλάχιστον ένα (0,1) στοιχείο στην πρώτη γραμμή και κατά συνέπεια αυτή η στρατηγική είναι πάντα εφικτή. Συνεπώς εφόσον ο παίκτης στήλης μπορεί να αυξήσει την ωφέλειά του δεν βρίσκεται σε κατάσταση ισορροπίας.

Θεωρητικό Ερώτημα 2 (Εντοπισμός Αυστηρά Κυριαρχούμενων Δράσεων)

Θ.Ε. 2.1

Για να ισχύει ότι για $i \in [m], R[i,j] = 0 \; \forall \; j \in [n]$ θα πρέπει να μην υπάρχει καμία στήλη με (1,0) στοιχείο στη συγκεκριμένη γραμμή. Σε διαφορετική περίπτωση για κάποια δράση j του παίκτη στήλης η παίκτρια γραμμής θα είχε θετική ωφέλεια. Εφόσον για κάθε στήλη υπάρχει ένα τουλάχιστον (1,0)- στοιχείο σε κάποια γραμμή, υπάρχει σίγουρα μικτή στρατηγική που κυριαρχεί επί της e_i. Αυτή προκύπτει «μιξάροντας» μεταξύ γραμμών οι οποίες έχουν (1,0)- στοιχεία σε διαφορετικές στήλες (πιθανώς και σε κάποιες ίδιες) με τέτοιο τρόπο ώστε για κάθε στήλη να υπάρχει μια γραμμή με (1,0) στοιχείο για την στήλη αυτή κάτι που είναι πάντα εφικτό δεδομένης της συγκεκριμένης

γεννήτριας παιχνιδιών. Για μια τέτοια στρατηγική έστω $x \in \Delta_m$ ισχύει ότι $x^T R y > 0, \forall\, y \in \Delta_n$. Κατά συνέπεια κυριαρχεί αυστηρά επί της e_i.

Θ.Ε. 2.2

Αντίστοιχα για να ισχύει ότι για $j \in [n]$, $C[i,j] = 0\ \forall\ i \in [m]$ θα πρέπει να μην υπάρχει καμία γραμμή με (0,1) στοιχείο στη συγκεκριμένη στήλη. Σε διαφορετική περίπτωση για κάποια δράση i της παίκτριας γραμμής, ο παίκτης στήλης θα είχε θετική ωφέλεια. Εφόσον για κάθε γραμμή υπάρχει ένα τουλάχιστον (0,1)- στοιχείο σε κάποια στήλη, υπάρχει σίγουρα μικτή στρατηγική που κυριαρχεί επί της e_j. Αυτή προκύπτει «μιξάροντας» μεταξύ στηλών οι οποίες έχουν (0,1)- στοιχεία σε διαφορετικές γραμμές (πιθανώς και σε κάποιες ίδιες) με τέτοιο τρόπο ώστε για κάθε γραμμή να υπάρχει μια στήλη με (0,1) στοιχείο για την γραμμή αυτή κάτι που είναι πάντα εφικτό δεδομένης της συγκεκριμένης γεννήτριας παιχνιδιών. Για μια τέτοια στρατηγική έστω $y \in \Delta_n$ ισχύει ότι $x^T C y > 0, \forall\, x \in \Delta_m$. Κατά συνέπεια κυριαρχεί αυστηρά επί της e_j.

Υλοποίηση 1

checkForPNE (m,n,R,C,x,y)

Είσοδοι:

m: Ο αριθμός των γραμμών του διμητρώου.

n: Ο αριθμός των στηλών του διμητρώου.

R: Το μητρώο ωφέλειας του παίκτη γραμμής.

C: Το μητρώο ωφέλειας του παίκτη στήλης.

Έξοδοι:

pne: Πίνακας ο οποίος σε κάθε του στοιχείο έχει αποθηκευμένο τον αριθμό γραμμής και στήλης όπου το διμητρικό παιχνίδι έχει ισορροπία NASH.

Περιγραφή:

Η συνάρτηση αρχικά διατρέχει για μια συγκεκριμένη γραμμή όλες τις στήλες της γραμμής και επιστρέφει την μεγαλύτερη τιμή της ωφέλειας της παίκτριας γραμμής. Την τιμή αυτή την εισάγει μια λίστα.Αυτή η διαδικασία γίνεται για όλες τις γραμμές.

Έπειτα για μια συγκεκριμένη στήλη η συνάρτηση διατρέχει όλες τις γραμμές της στήλης και επιστρέφει την μεγαλύτερη τιμή της ωφέλειας του παίκτη στήλης. Την τιμή αυτή την εισάγει σε μια άλλη λίστα. Επίσης αυτή η διαδικασία γίνεται για όλες τις στήλες .

Έπειτα διατρέχει πάλι για κάθε γραμμή όλες τις στήλες και ελέγχει αν κάποιο συγκεκριμένο κελί έχει τιμή μέγιστης ωφέλειας και για την παίκτρια γραμμής(για την στήλη που επέλεξε ο παίκτης στήλης) και για τον παίκτη στήλης(για την γραμμή που επέλεξε η παίκτρια γραμμής). Αν ισχύει αυτό εισάγουμε τις θέσεις του κελιού στον πίνακα pne.

Υλοποίηση 2

computeApproximationGuarantees(m,n,R,C,x,y)

Είσοδοι:

m: Ο αριθμός των γραμμών του διμητρώου.

n: Ο αριθμός των στηλών του διμητρώου.

R: Το μητρώο ωφέλειας του παίκτη γραμμής.

C: Το μητρώο ωφέλειας του παίκτη στήλης.

x: Διάνυσμα που αναπαριστά τη στρατηγική του παίκτη γραμμής.

y: Διάνυσμα που αναπαριστά τη στρατηγική του παίκτη στήλης.

Έξοδοι:

epsAPPROX: Η τιμή του ε για την ε-προσεγγιστική ισορροπία Nash που προκύπτει από την είσοδο.

epsWSNE: Η τιμή του ε για την ε-καλά στηριγμένη προσεγγιστική ισορροπία Nash που προκύπτει από την είσοδο.

Περιγραφή:

Η συνάρτηση αρχικά μετατρέπει όλες τις εισόδους σε numpy arrays.

Για τον υπολογισμό της epsAPPROX υπολογίζει τα μητρώα x^TC και Ry δηλαδή τα μητρώα ωφελειών του παίκτη στήλης και γραμμής αντίστοιχα όπως αυτά διαμορφώνονται όταν ο αντίπαλός τους παίξει τη στρατηγική του. Στη συνέχεια υπολογίζει τα x^TCy και x^TRy δηλαδή τις αναμενόμενες ωφέλειες των παικτών. Έπειτα αφαιρεί από το μέγιστο των Ry, x^TC (δηλαδή τη μέγιστη ωφέλεια θα μπορούσε να έχει κάθε παίκτης έπειτα από την κίνηση του αντιπάλου του) τις αντίστοιχες αναμενόμενες ωφέλειες που υπολόγισε. Από αυτή την αφαίρεση προκύπτουν δύο ε (τα μικρότερα δυνατά για κάθε παίκτη) από τα οποία επιλέγεται το μεγαλύτερο (αυτό δηλαδή που καλύπτει και τους δύο παίκτες) ως η πρώτη έξοδος.

Για τον υπολογισμό της epsWSNE υπολογίζει όλες τις αναμενόμενες ωφέλειες των δύο παικτών έπειτα από την κίνηση του αντιπάλου τους για κάθε αμιγή στρατηγική που μπορούν να ακολουθήσουν. Από αυτές

επιλέγει την ελάχιστη και χρησιμοποιώντας αυτή κάνει την αφαίρεση και την επιλογή μέγιστου ε όπως προηγουμένως για να παράξει τη δεύτερη έξοδο.

Υλοποίηση 3

removeRistriclyDominatedStrategies(m,n,R,C)

Είσοδοι:

m: Ο αριθμός των γραμμών του διμητρώου.

n: Ο αριθμός των στηλών του διμητρώου.

R: Το μητρώο ωφέλειας του παίκτη γραμμής.

C: Το μητρώο ωφέλειας του παίκτη στήλης.

Έξοδοι:

m: Ο αριθμός των γραμμών του διμητρώου που προκύπτει από τις διαγραφές .

n: Ο αριθμός των στηλών του διμητρώου που προκύπτει από τις διαγραφές.

R: Το μητρώο ωφέλειας του παίκτη γραμμής που προκύπτει από τις διαγραφές.

C: Το μητρώο ωφέλειας του παίκτη στήλης που προκύπτει από τις διαγραφές.

Περιγραφή:

Η συνάρτηση αρχικά μετατρέπει τα μητρώα C και R σε numpy arrays.

Για κάθε γραμμή διατρέχει όλες τις στήλες και αν βρεθεί ότι για την παίκτρια γραμμής υπάρχει (σε όλες τις στήλες) μηδενική ωφέλεια διαγράφει την συγκεκριμένη γραμμή από τα μητρώα C και R και ενημερώνει το m. Έπειτα καλείται εκ νέου η συνάρτηση ώστε να διαγραφούν οι υπόλοιπες αυστηρά κυριαρχούμενες δράσεις που έμειναν ή νέες που προέκυψαν από τις διαγραφές.

Αν δεν υπάρχει άλλη γραμμή με μηδενική ωφέλεια για την παίκτρια γραμμής ελέγχει αντίστοιχα την κάθε στήλη και αν βρεθεί ότι για την συγκεκριμένη στήλη υπάρχει σε κάθε γραμμή μόνο μηδενική ωφέλεια για τον παίκτη στήλης διαγράφει την στήλη αυτή από τα μητρώα R και C και ενημερώνει το n. Έπειτα καλείται εκ νέου η συνάρτηση ώστε να διαγραφούν οι υπόλοιπες αυστηρά κυριαρχούμενες δράσεις που έμειναν ή νέες που προέκυψαν από τις διαγραφές.

Τέλος επιστρέφονται τα νέα μητρώα R,C και οι νέες διαστάσεις τους μετά τις διαγραφές.

Υλοποίηση 4

approxNEConstructionDMP(m,n,R,C)

Είσοδοι:

m: Ο αριθμός των γραμμών του διμητρώου.

n: Ο αριθμός των στηλών του διμητρώου.

R: Το μητρώο ωφέλειας του παίκτη γραμμής.

C: Το μητρώο ωφέλειας του παίκτη στήλης.

Έξοδοι:

x: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη γραμμής που υπολόγισε ο αλγόριθμος.

y: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη στήλης που υπολόγισε ο αλγόριθμος.

epsAPPROX: Η τιμή του ε για την ε-προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

epsWSNE: Η τιμή του ε για την ε-καλά στηριγμένη προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

Περιγραφή:

Η συνάρτηση επιλέγει μια τυχαία γραμμή του R και υπολογίζει τη μέγιστη τιμή του C (μέγιστη αναμενόμενη ωφέλεια του παίκτη στήλης) στη γραμμή αυτή. Στη συνέχεια επιλέγει τυχαία μια από τις στήλες του C που περιέχουν τη μέγιστη τιμή. Υπολογίζει τη μέγιστη τιμή του R στην στήλη αυτή (μέγιστη αναμενόμενη ωφέλεια του παίκτη γραμμής) και επιλέγει τυχαία μια από τις γραμμές που την περιέχουν. Τέλος αρχικοποιεί δύο διανύσματα με m και n μηδενικά αντίστοιχα και στο πρώτο προσθέτει 0.5 στις θέσεις που προέκυψαν προηγουμένως ως γραμμές του R (μεικτή στρατηγική παίκτη γραμμής εκτός αν βρέθηκε η ίδια γραμμής και τις δύο φορές) και θέτει ίση με 1 την θέση του δεύτερου διανύσματος που προέκυψε προηγουμένως ως στήλη του C (αμιγής στρατηγική παίκτη στήλης). Τα δύο διανύσματα επιστρέφονται ως έξοδοι μαζί με τις τιμές των ε που υπολογίζονται καλώντας την **computeApproximationGuarantees.**

approxNEConstructionDEL(m,n,R,C)

Είσοδοι:

m: Ο αριθμός των γραμμών του διμητρώου.

n: Ο αριθμός των στηλών του διμητρώου.

R: Το μητρώο ωφέλειας του παίκτη γραμμής.

C: Το μητρώο ωφέλειας του παίκτη στήλης.

Έξοδοι:

x: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη γραμμής που υπολόγισε ο αλγόριθμος.

y: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη στήλης που υπολόγισε ο αλγόριθμος.

epsAPPROX: Η τιμή του ε για την ε-προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

epsWSNE: Η τιμή του ε για την ε-καλά στηριγμένη προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

Περιγραφή:

Αρχικά η συνάρτηση υπολογίζει τις MAXMIN στρατηγικές των δύο παικτών καλώντας την **solveZeroSumGame** η οποία επιστρέφει δύο διανύσματα που αναπαριστούν τη στρατηγική κάθε παίκτη. Έπειτα υπολογίζει την εγγυημένη ωφέλεια τους πολλαπλασιάζοντας τις στρατηγικές που προέκυψαν με την τιμή του μητρώου ωφέλειας του παίκτη. Εάν η εγγυημένη ωφέλεια του παίκτη γραμμής είναι μικρότερη από αυτή του παίκτη στήλης τότε αντιμεταθέτει τους δύο πίνακες ωφέλειας (κάνοντας τον παίκτη γραμμής παίκτη στήλης και αντίστροφα) και τις διαστάσεις τους. Επανυπολογίζει τις MAXMIN στρατηγικές και τις εγγυημένες ωφέλειες και αν η εγγυημένη ωφέλεια του παίκτη γραμμής είναι μικρότερη ή ίση με το 2/3 επιστρέφει την στρατηγική για τον παίκτη γραμμής που προέκυψε από τον υπολογισμό της MAXMIN στρατηγικής του παίκτη στήλης και αντίστοιχα για τον παίκτη στήλης. Διαφορετικά υπολογίζει την μέγιστη αναμενόμενη ωφέλεια του παίκτη στήλης για την MAXMIN στρατηγική του παίκτη γραμμής και αν αυτή είναι μικρότερη του 2/3 επιστρέφει τις στρατηγικές που προέκυψαν από τον υπολογισμό της MAXMIN στρατηγική του παίκτη γραμμής. Σε διαφορετική περίπτωση επιλέγει τυχαία μία από τις στήλες του C που περιέχουν τη μέγιστη τιμή αναμενόμενης ωφέλειας για την MAXMIN στρατηγική του παίκτη γραμμής και αποθηκεύει τον δείκτη της. Έπειτα διατρέχει όλες τις γραμμές της στήλης ελέγχοντας αν υπάρχει γραμμή όπου R>1/3 και C>1/3. Όταν αυτή βρεθεί αποθηκεύει το δείκτη της και αρχικοποιεί δύο διανύσματα με m και n μηδενικά αντίστοιχα και στο πρώτο προσθέτει ένα στη θέση που δείχνει ο δείκτης γραμμών και στο δεύτερο ένα στη θέση που δείχνει ο δείκτη στήλης (αμιγείς στρατηγικές). Τέλος επιστρέφει τα δύο διανύσματα. Σε κάθε περίπτωση επιστρέφει μαζί με τις στρατηγικές και τα ε που υπολογίζονται καλώντας την **computeApproximationGuarantees**.

approxNEConstructionFPPBR(m,n,R,C)

Είσοδοι:

m: Ο αριθμός των γραμμών του διμητρώου.

n: Ο αριθμός των στηλών του διμητρώου.

R: Το μητρώο ωφέλειας του παίκτη γραμμής.

C: Το μητρώο ωφέλειας του παίκτη στήλης.

Έξοδοι:

x: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη γραμμής που υπολόγισε ο αλγόριθμος.

y: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη στήλης που υπολόγισε ο αλγόριθμος.

epsAPPROX: Η τιμή του ε για την ε-προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

epsWSNE: Η τιμή του ε για την ε-καλά στηριγμένη προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

Περιγραφή:

Η συνάρτηση διατηρεί δύο βασικές λίστες για κάθε παίκτη. Η μία αναπαριστά την τρέχουσα στρατηγική του παίκτη και η άλλη λειτουργεί ως ένα "ιστορικό στρατηγικών". Αρχικοποιεί τις στρατηγικές των δύο παικτών έτσι ώστε να παίζουν αμιγώς στην πρώτη γραμμή ή στην πρώτη στήλη αντίστοιχα. Στη συνέχεια εκτελεί 100 επαναλήψεις κατά τις οποίες υπολογίζει για κάθε παίκτη την πρώτη γραμμή/στήλη που περιέχει τη μέγιστη ωφέλεια στο μητρώο ωφελειών του παίκτη (Pure Best Response) όπως αυτό προκύπτει μετά την κίνηση του αντιπάλου (της προηγούμενης επανάληψης την οποία λαμβάνει από τη λίστα-ιστορικό). Θέτει την τρέχουσα στρατηγική του παίκτη ως αμιγή σε αυτή τη γραμμή/στήλη (1 στην αντίστοιχη θέση) και υπολογίζει την τελευταία θέση στη λίστα - ιστορικό του παίκτη ως το μέσο όρο των στρατηγικών που έχει υπολογίσει στις μέχρι στιγμής επαναλήψεις (συμπεριλαμβανομένης και της τελευταίας). Ως έξοδοι επιστρέφονται τα διανύσματα που είναι αποθηκευμένα στην τελευταία θέση της λίστας - ιστορικό του κάθε παίκτη μαζί με τις τιμές των ε που υπολογίζονται καλώντας την **computeApproximationGuarantees.**

approxNEConstructionFPUNI(m,n,R,C)

Είσοδοι:

m: Ο αριθμός των γραμμών του διμητρώου.

n: Ο αριθμός των στηλών του διμητρώου.

R: Το μητρώο ωφέλειας του παίκτη γραμμής.

C: Το μητρώο ωφέλειας του παίκτη στήλης.

Έξοδοι:

x: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη γραμμής που υπολόγισε ο αλγόριθμος.

y: Διάνυσμα που αναπαριστά την στρατηγική του παίκτη στήλης που υπολόγισε ο αλγόριθμος.

epsAPPROX: Η τιμή του ε για την ε-προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

epsWSNE: Η τιμή του ε για την ε-καλά στηριγμένη προσεγγιστική ισορροπία Nash που υπολόγισε ο αλγόριθμος.

Περιγραφή:

Λειτουργεί αντίστοιχα με την ***a*pproxNEConstructionFPPBR** με δύο βασικές διαφοροποιήσεις. Πρώτον, η αρχικοποίηση των κινήσεων των παικτών γίνεται έτσι ώστε να προκύπτει μια μεικτή στρατηγική που επιλέγει όλες τις στήλες/γραμμές ομοιόμορφα. Δεύτερον, σε κάθε επανάληψη υπολογίζει όλες τις Pure Best Responses και δημιουργεί την τρέχουσα στρατηγική έτσι ώστε να τις επιλέγει όλες ομοιόμορφα. Αυτό γίνεται θέτοντας 1 σε όλες τις θέσεις του διανύσματος που αντιστοιχούν σε PBR και διαιρώντας έπειτα το διάνυσμα με το πλήθος των PBRs.

Υλοποίηση 5

defaultExperiments()

Είσοδοι: -

Έξοδοι: -

Περιγραφή:

Η συνάρτηση ζητάει από τον χρήστη να επιλέξει ποιο πείραμα από τα 4 προεπιλεγμένα θέλει να εκτελέσει. Ανάλογα με την είσοδο θέτει κατάλληλα τις παραμέτρους του πειράματος και επαναληπτικά δημιουργεί παιχνίδια καλώντας την **generate_winlose_game_without_pne.** Σε κάθε επανάληψη αφαιρεί τις ισχυρά κυριαρχούμενες δράσεις του παιχνιδιού καλώντας την **removeStrictlyDominatedStrategies** και εκτελεί διαδοχικά τους 4 αλγορίθμους καλώντας τις κατάλληλες συναρτήσεις. Αποθηκεύει τα ε που επιστρέφουν οι αλγόριθμοι (πλήθος ε σε κάθε εύρος) σε δύο λίστες για κάθε αλγόριθμο καθώς και τα μητρώα R και C για το μέχρι στιγμής χειρότερο παιχνίδι για κάθε αλγόριθμο για κάθε ε. Τέλος με χρήση της **numpy.histogram** και της **pyplot** σχεδιάζει τα ιστογράμματα και τα

αποθηκεύει μαζί με τις προαναφερθείσες λίστες και μητρώα στον κατάλληλο φάκελο.

Κύριο πρόγραμμα

customExperiments()

Είσοδοι: -

Έξοδοι: -

Περιγραφή:

Η συνάρτηση επιτρέπει στον χρήστη να τρέξει πειράματα που ορίζει ο ίδιος τις παραμέτρους τους. Λειτουργεί παρόμοια με την **defaultExperiments** με τη διαφορά ότι οι παράμετροι του πειράματος πληκτρολογούνται από το χρήστη καθώς και την πρόσθετη δυνατότητα να επιλεγεί ένας μόνο ή όλοι οι προσεγγιστικοί αλγόριθμοι για την εκτέλεση του πειράματος. Στην περίπτωση που επιλεγεί ένας αλγόριθμος επιστρέφεται και το runtime του. Τέλος αποθηκεύει τα ίδια πράγματα με την **defaultExperiments** με εξαίρεση το πλήθος των ε ανά εύρος (το οποίο επιστρέφεται μόνο ως ιστόγραμμα και όχι ως μητρώο).

menu()

Είσοδοι: -

Έξοδοι: -

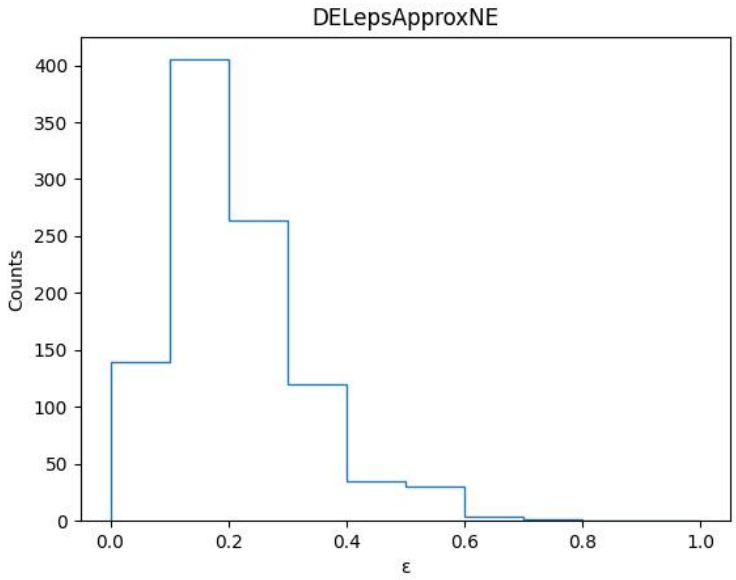
Περιγραφή:

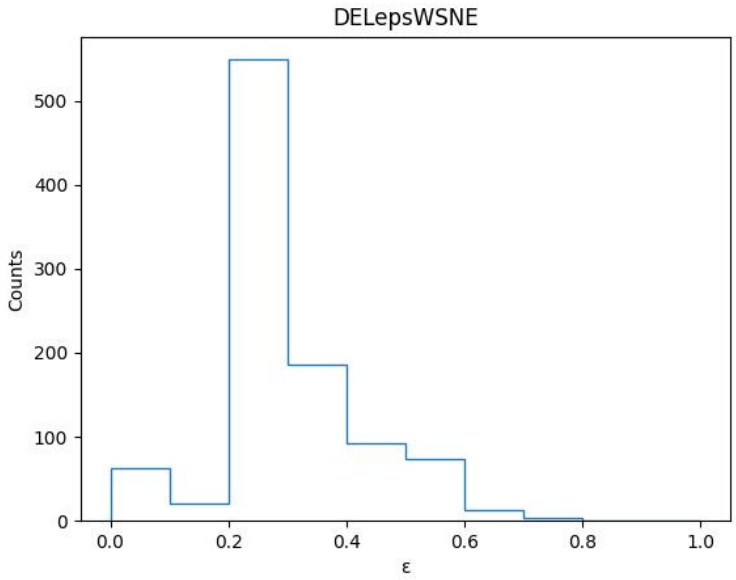
Επιτρέπει στον χρήστη να κάνει όλες τις ενέργειες που ζητούνται στην εκφώνηση εμφανίζοντας μενού επιλογής στο χρήστη και καλώντας τις κατάλληλες συναρτήσεις ανάλογα με την είσοδο. Για την φόρτωση και αποθήκευση διμητρικών παιχνιδιών δεν καλεί κάποια συνάρτηση αλλά πραγματοποιεί αυτές τις διαδικασίες στο εσωτερικό της.

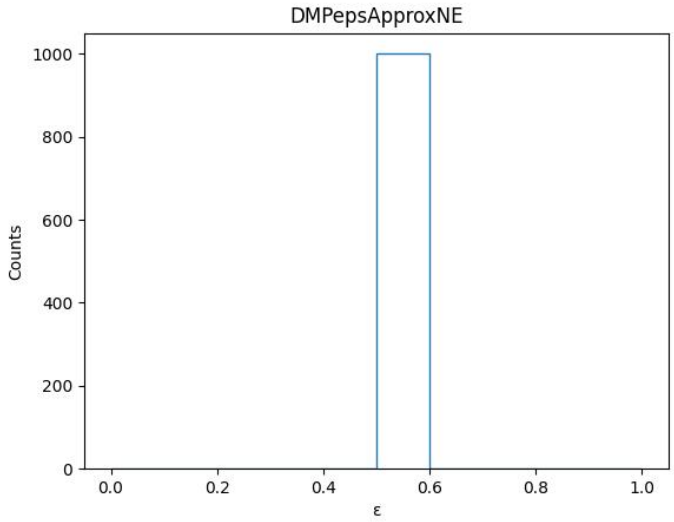
Πειραματική Αξιολόγηση

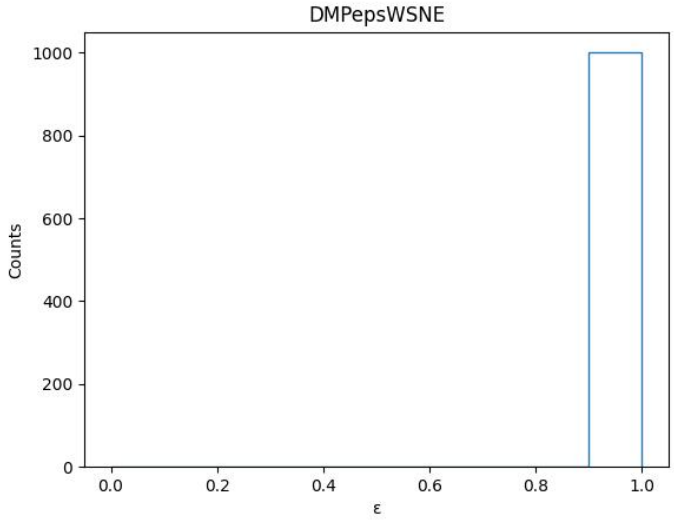
Αποτελέσματα

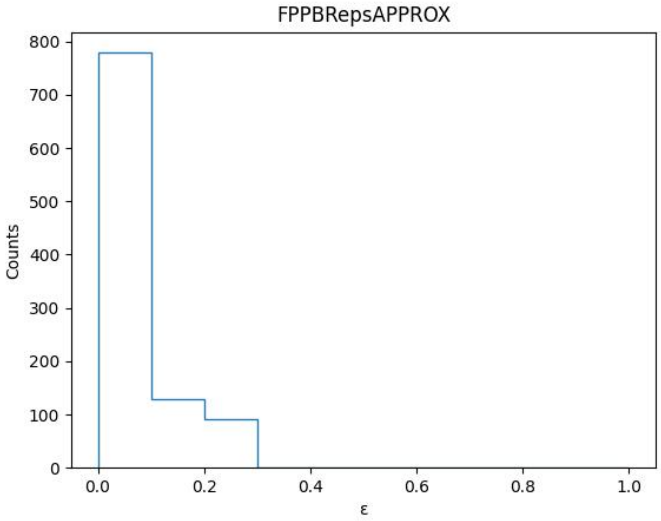
Πείραμα Π1

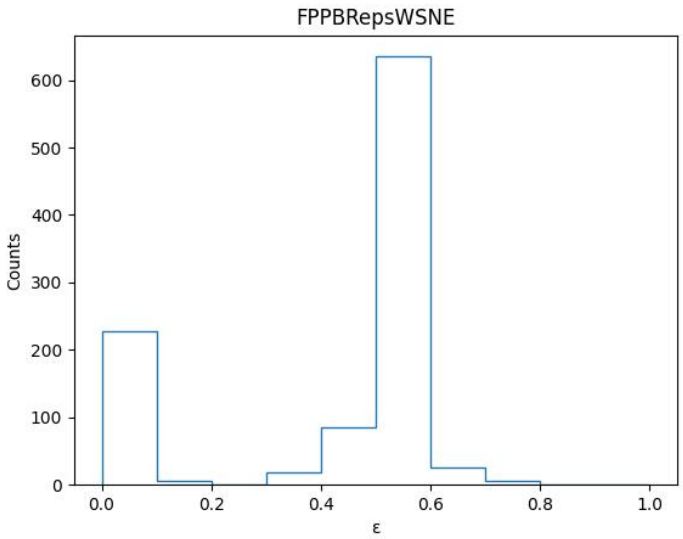


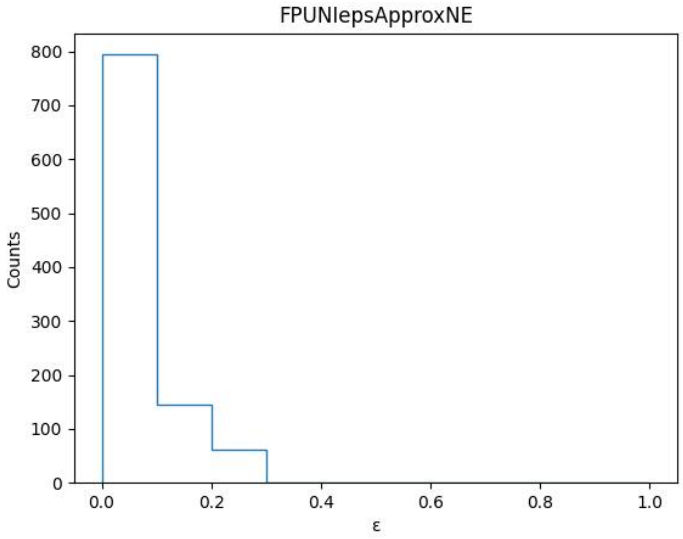


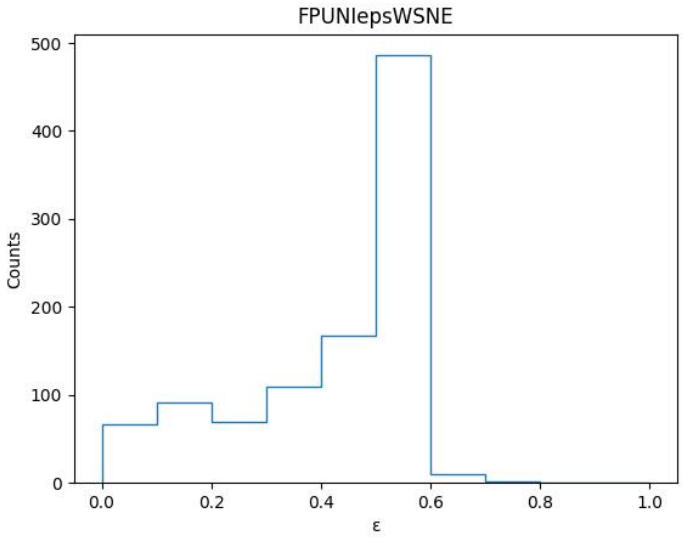

DMPepsApproxNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0


DMPepsWSNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0

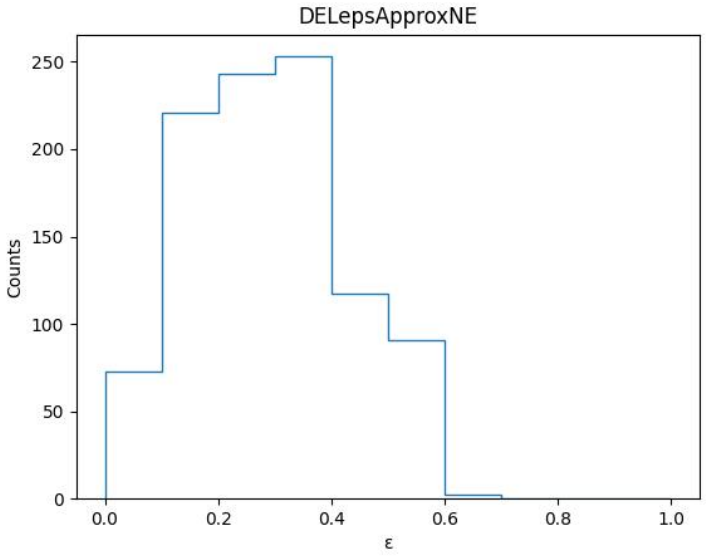

FPPBRepsAPPROX
Counts
ε
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0

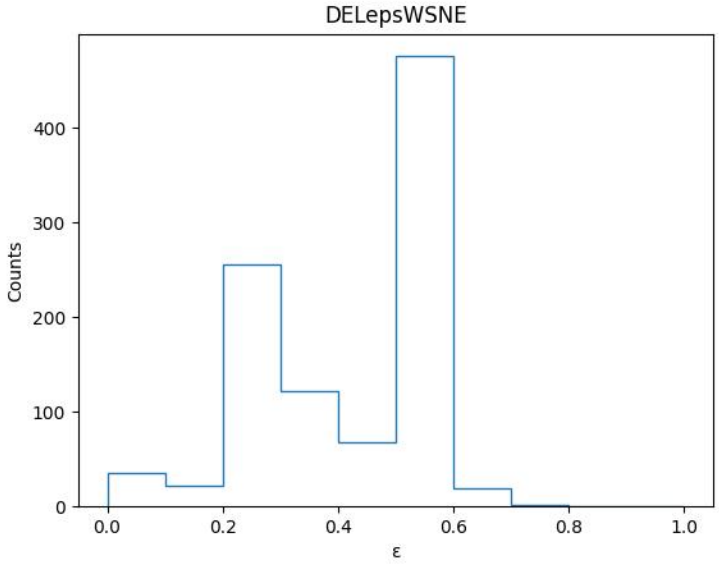

FPPBRepsWSNE
Counts
ε
0
100
200
300
400
500
600
0.0
0.2
0.4
0.6
0.8
1.0

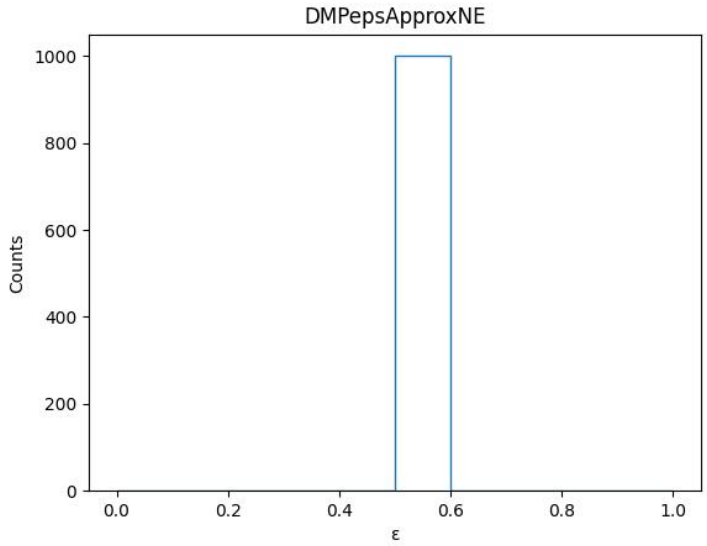

FPUNIepsApproxNE
Counts
ε
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0

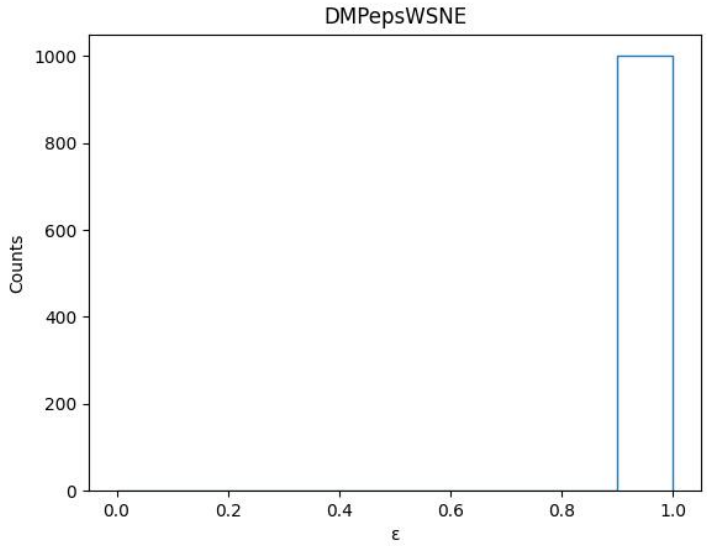

FPUNIepsWSNE
Counts
ε
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0

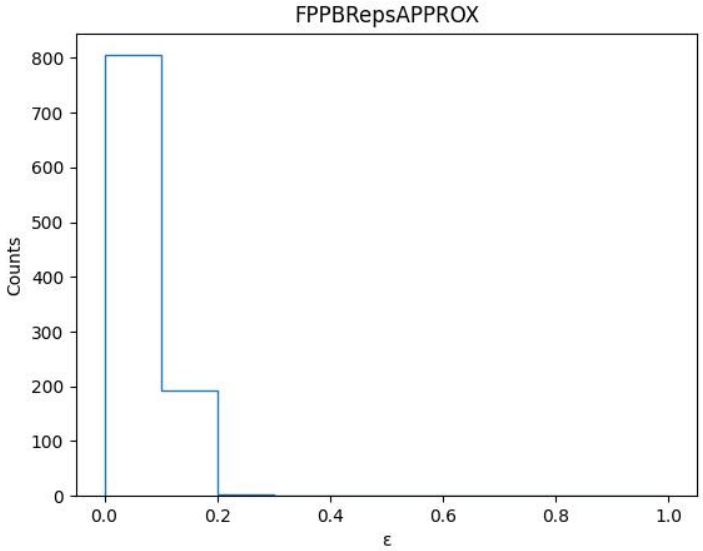
Πείραμα Π2

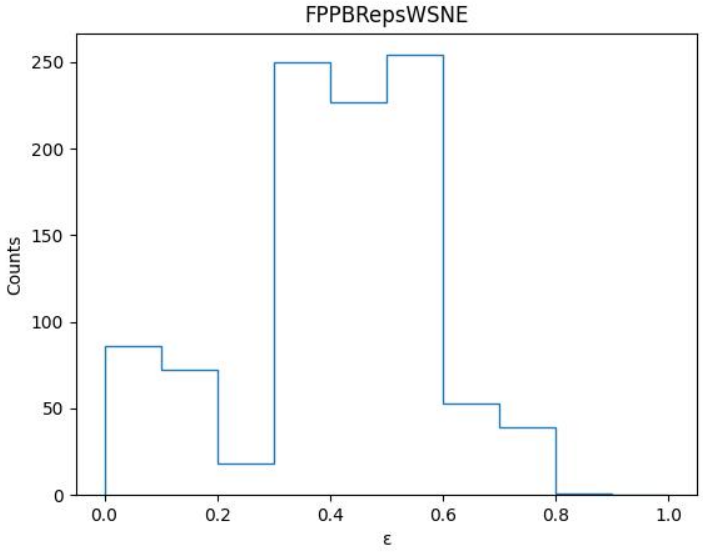

DELepsApproxNE
Counts
ε
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
1.0

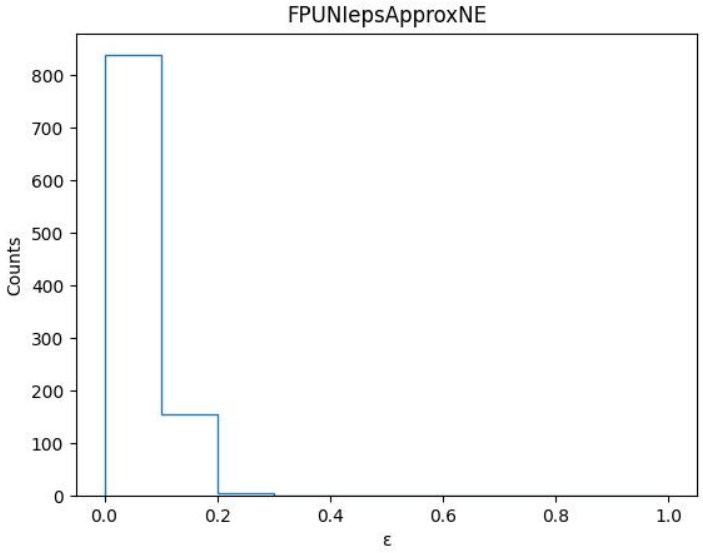

DELepsWSNE
Counts
ε
0
100
200
300
400
0.0
0.2
0.4
0.6
0.8
1.0

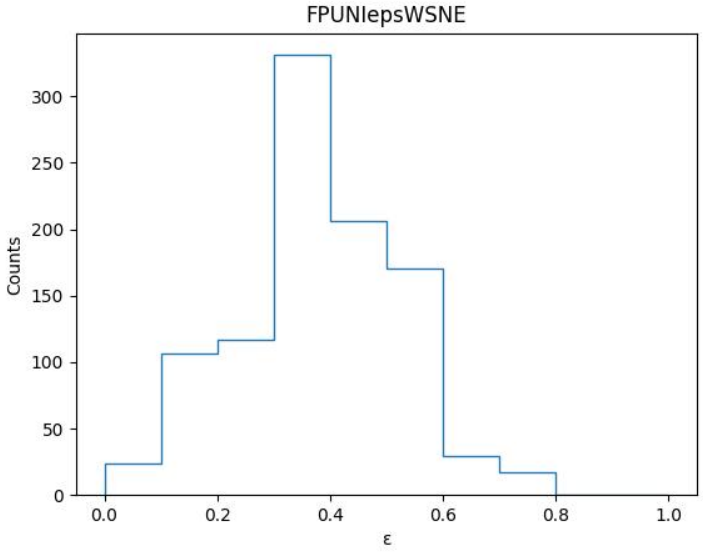

DMPepsApproxNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0


DMPepsWSNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0

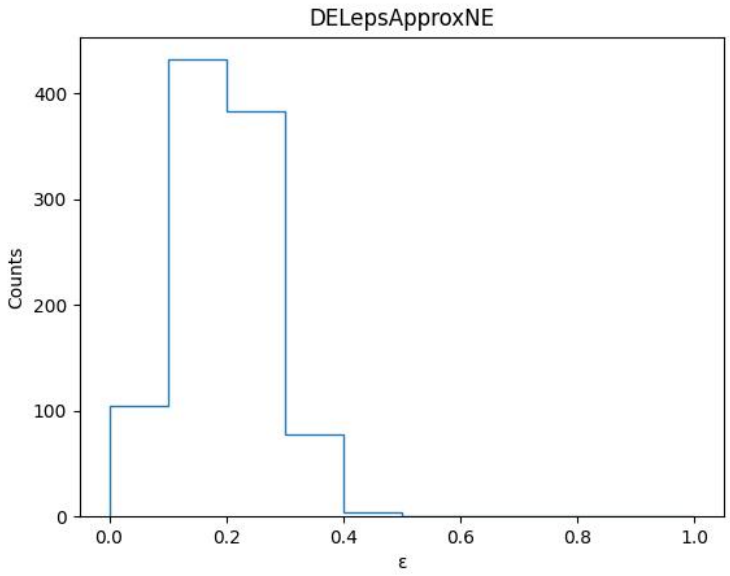

FPPBRepsAPPROX
Counts
ε
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0

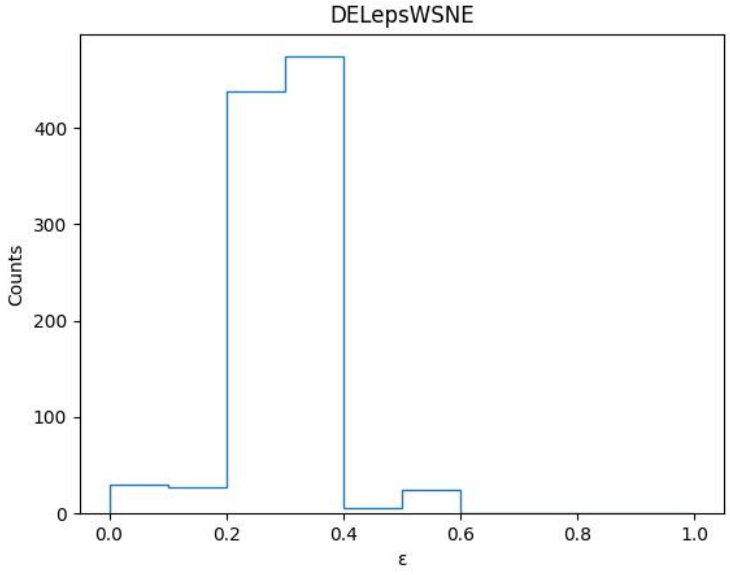

FPPBRepsWSNE
Counts
ε
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
1.0

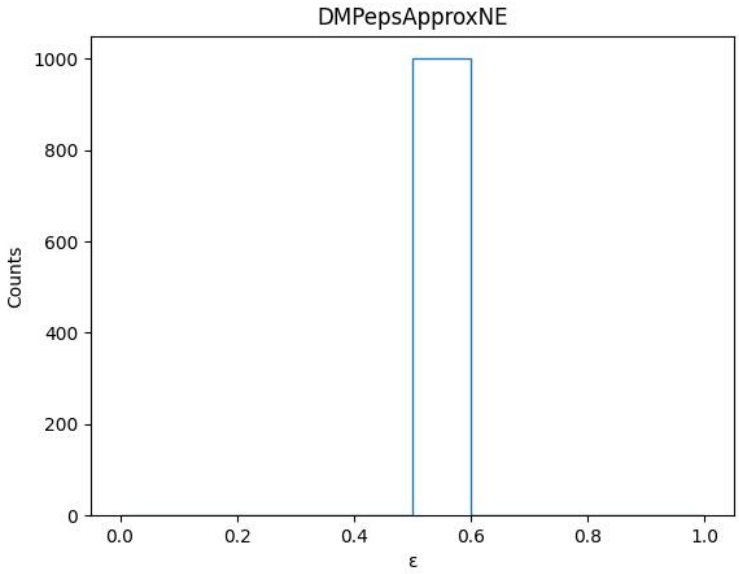

FPUNIepsApproxNE
Counts
ε
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0

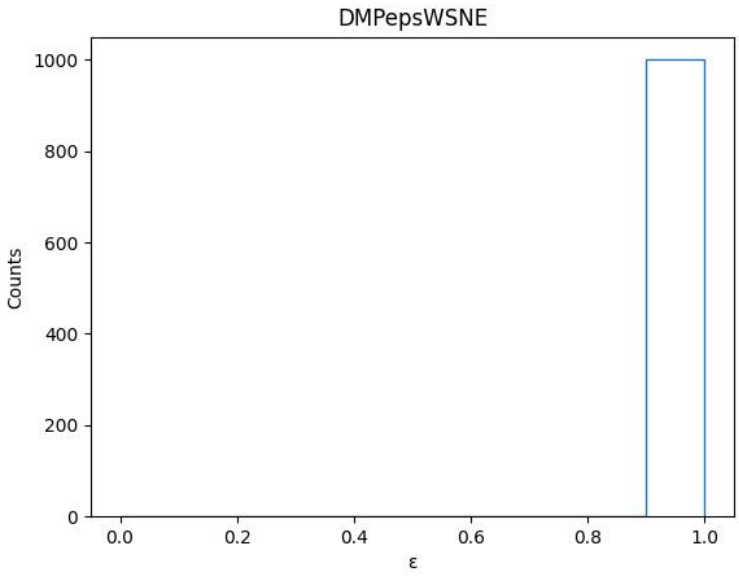

FPUNIepsWSNE
Counts
ε
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0

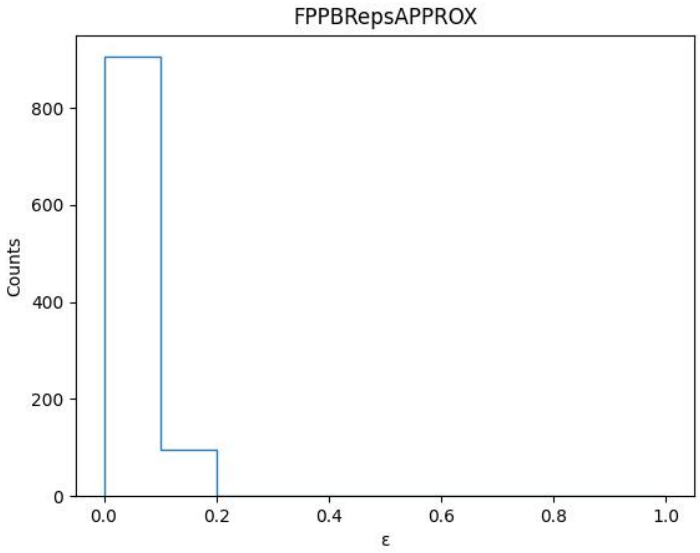
Πείραμα Π3

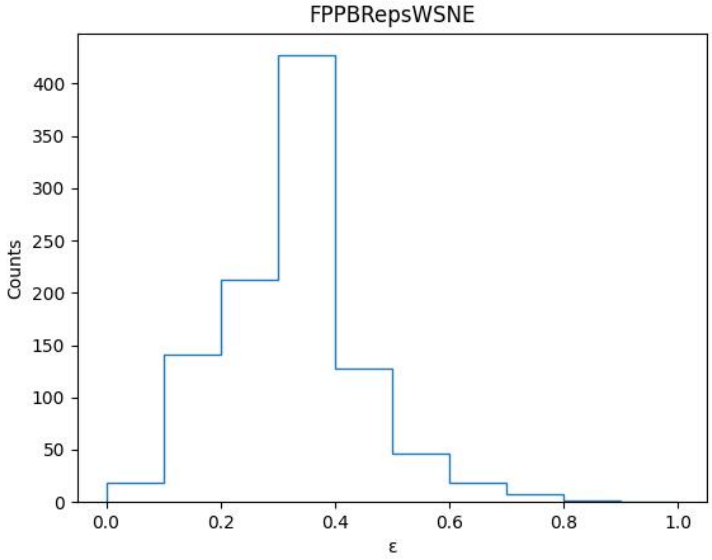

DELepsApproxNE
Counts
0
100
200
300
400
0.0
0.2
0.4
0.6
0.8
1.0
ε

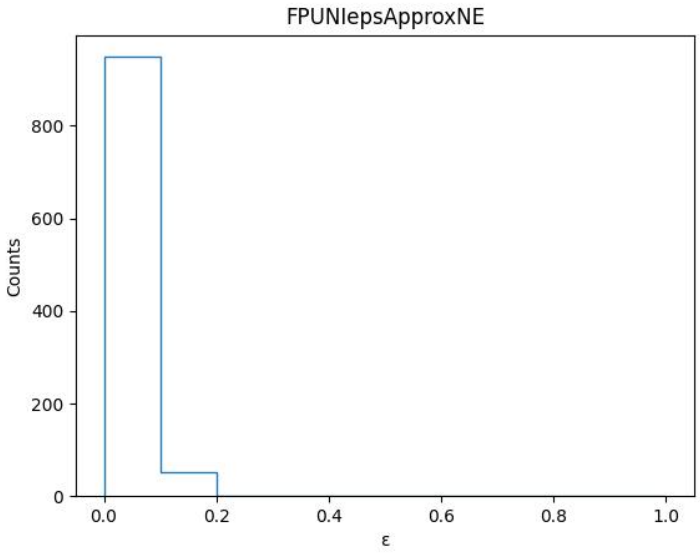

DELepsWSNE
Counts
0
100
200
300
400
0.0
0.2
0.4
0.6
0.8
1.0
ε

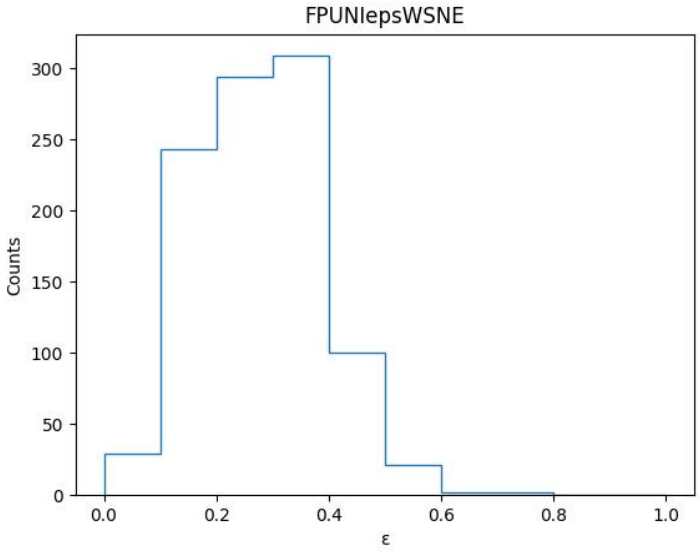

DMPepsApproxNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0


DMPepsWSNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0

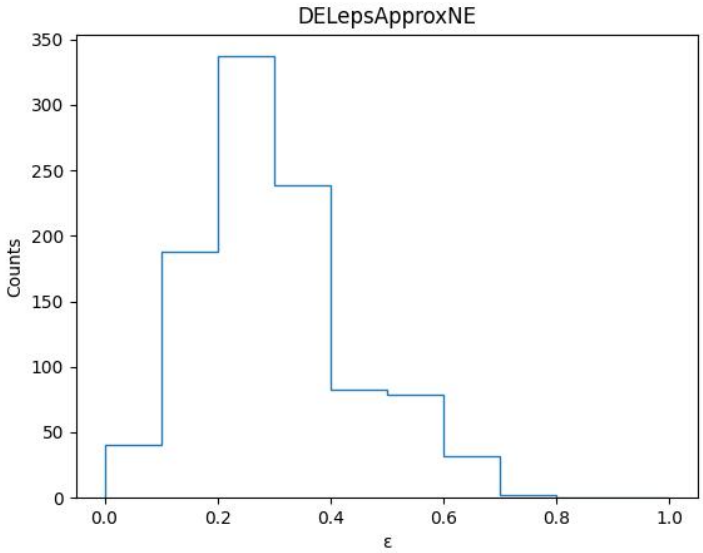

FPPBRepsAPPROX
Counts
ε
0
200
400
600
800
0.0
0.2
0.4
0.6
0.8
1.0

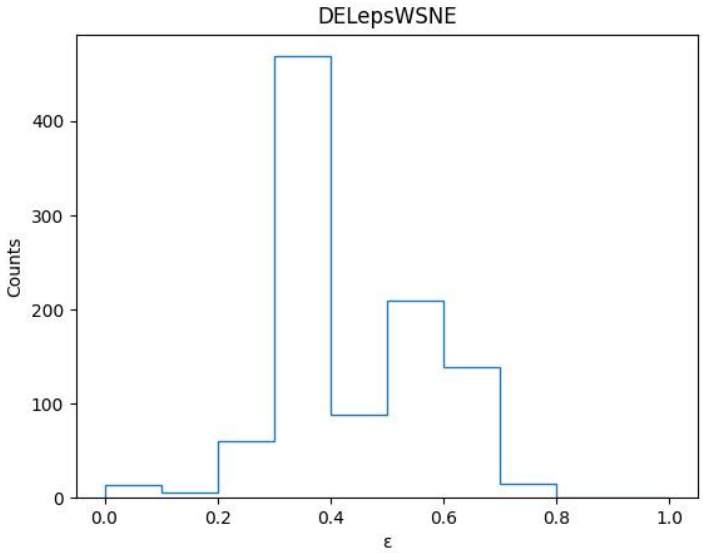

FPPBRepsWSNE
Counts
ε
0
50
100
150
200
250
300
350
400
0.0
0.2
0.4
0.6
0.8
1.0

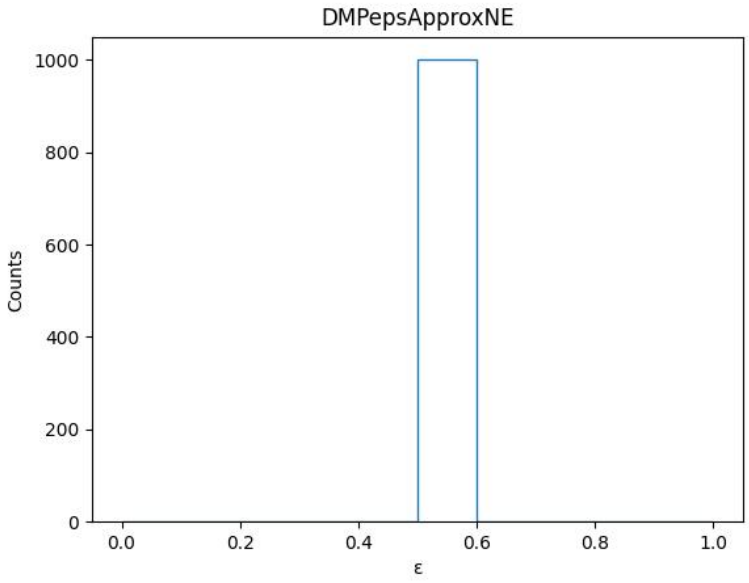

FPUNIepsApproxNE
Counts
ε
0
200
400
600
800
0.0
0.2
0.4
0.6
0.8
1.0

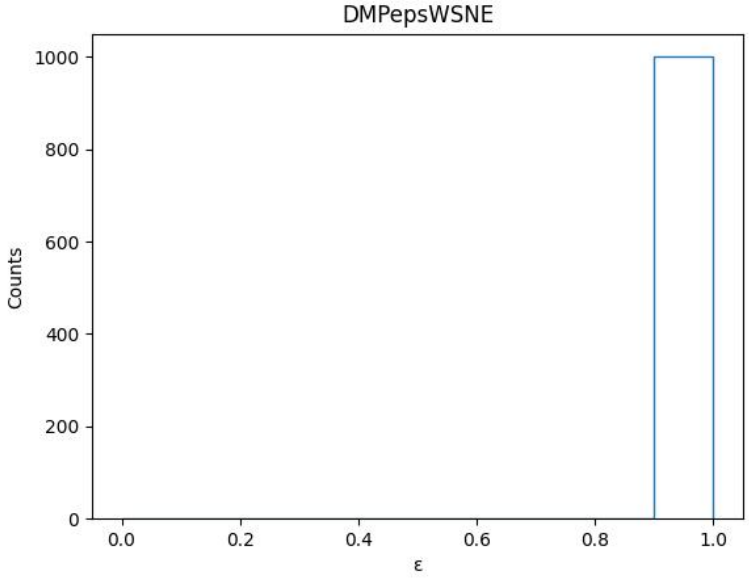

FPUNIepsWSNE
Counts
ε
0
50
100
150
200
250
300
0.0
0.2
0.4
0.6
0.8
1.0

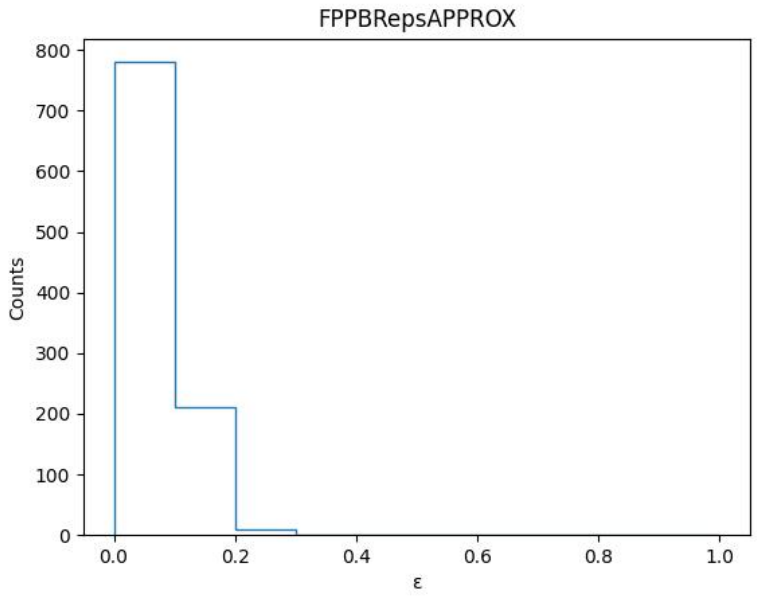
Πείραμα Π4

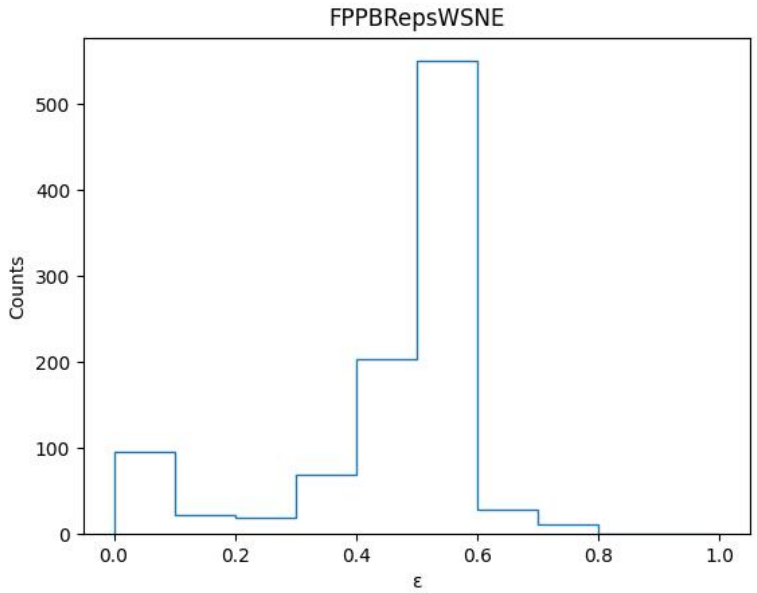

DELepsApproxNE
Counts
ε
0
50
100
150
200
250
300
350
0.0
0.2
0.4
0.6
0.8
1.0

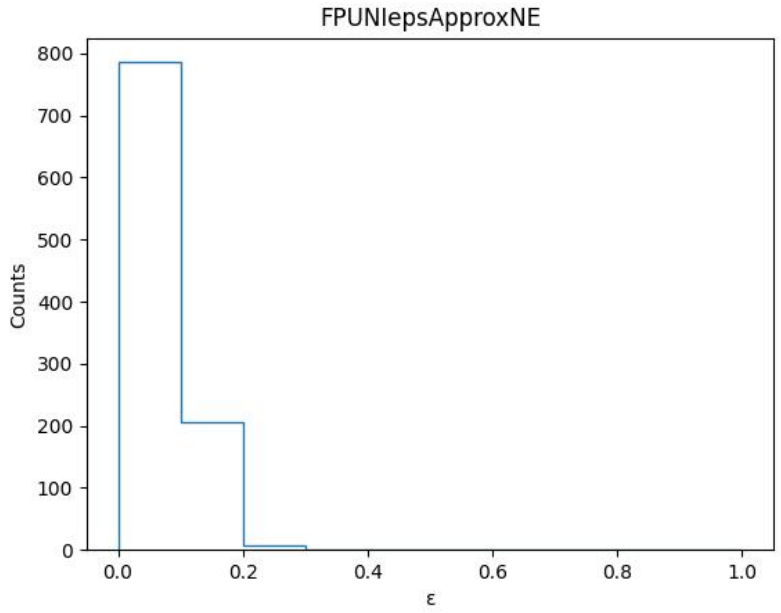

DELepsWSNE
Counts
ε
0
100
200
300
400
0.0
0.2
0.4
0.6
0.8
1.0

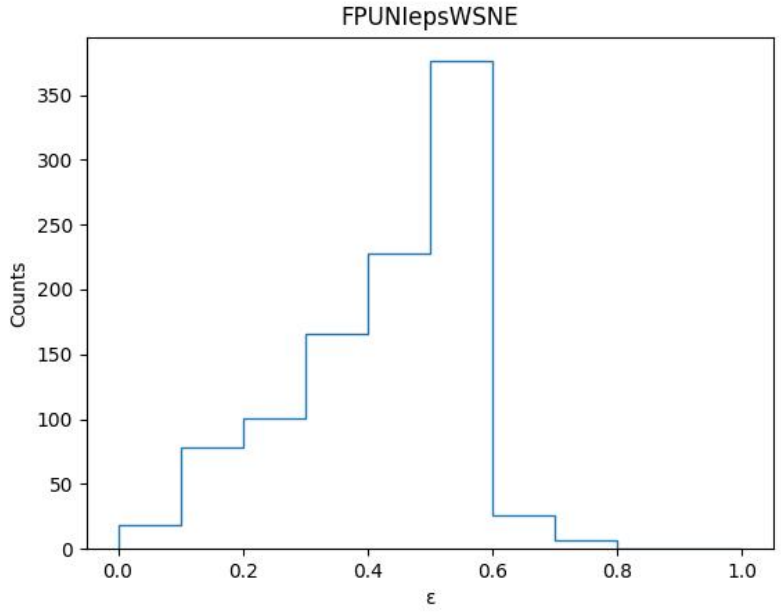

DMPepsApproxNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0


DMPepsWSNE
Counts
ε
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0


FPPBRepsAPPROX
Counts
ε
0
100
200
300
400
500
600
700
800
0.0
0.2
0.4
0.6
0.8
1.0


FPPBRepsWSNE
Counts
ε
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0





Συμπεράσματα

DMP

Παρατηρούμε ότι ανεξαρτήτως πειράματος ο DMP πάντοτε πετυχαίνει εAPPROXNE = 0.5 και εWSNE=1.

Το πρώτο είναι απόλυτα λογικό καθώς εφόσον ο παίκτης στήλης παίζει αμιγώς μια στήλη η οποία αποτελεί PBR στην μία από τις δύο γραμμές που επιλέγει ο παίκτης γραμμής (και τις οποίες παίζει ισοδύναμα) έχει μέγιστη αναμενόμενη ωφέλεια 1. Αφού όμως η δεύτερη γραμμή που επιλέγει ο παίκτης γραμμής είναι PBR στην κίνηση αυτή, η αναμενόμενη

ωφέλεια του παίκτη στήλης γίνεται 0.5 (δηλαδή διαφορά 0.5 σε σχέση με την μέγιστη αναμενόμενη ωφέλεια) καθώς μόνο με 0.5 πιθανότητα παίζει ο παίκτη γραμμής την κίνηση η οποία αποδίδει ωφέλεια στον παίκτη στήλης.

Το δεύτερο είναι επίσης αναμενόμενο καθώς η στρατηγική που προτείνεται από τον αλγόριθμο για τον παίκτη γραμμής δεν είναι καλά υποστηριζόμενη. Η μία εκ των δύο δράσεων του παίκτη γραμμής αποφέρει μηδενική ωφέλεια καθώς η αντίδραση του παίκτη στήλης είναι PBR σε αυτή, πράγμα που για τη συγκεκριμένη γεννήτρια παιχνιδιών σημαίνει μηδενική ωφέλεια για τον παίκτη γραμμής. Οπότε ισχύει πάντοτε ότι $\max_{i\in[m]}\{(Ry)_i\} - \min_{i\in support(x)}\{(Ry)_i\} = \varepsilon_{WSNE} = 1$.

DEL

Παρατηρούμε ότι για κάθε πείραμα στη συντριπτική πλειοψηφία των περιπτώσεων το ε_{WSNE} είναι μικρότερο του 0,7 όπως αναμένεται καθώς ο αλγόριθμος θεωρητικά θα πρέπει να επιστρέφει εγγυημένα $\varepsilon_{WSNE} < \frac{2}{3}$. Ωστόσο παρατηρούμε ορισμένες (λίγες) ακραίες τιμές μεγαλύτερες του αναμενόμενου 2/3 ένα φαινόμενο το οποίο δυστυχώς αδυνατούμε να εξηγήσουμε. Το φαινόμενο είναι εντονότερο στο Πείραμα Π4. Στα Πειράματα Π1 και Π2 το φαινόμενο ελαττώνεται ενώ στο Π3 δεν παρατηρείται καθόλου. Τέλος παρατηρούμε ότι το ε_{APPROX} είναι μικρότερου του ε_{WSNE} κάτι απόλυτα λογικό καθώς το δεύτερο είναι πιο αυστηρή μετρική.

FICTIOUS PLAY PBR

Παρατηρούμε ότι ο ο αλγόριθμος επιστρέφει σε όλα τα πειράματα $\varepsilon_{APPROX} < 2.5$ με την πλειονότητα των περιπτώσεων να βρίσκεται στο διάστημα (0,0.1]. Για τη συγκεκριμένη υλοποίηση επιλέχθηκε αριθμός επαναλήψεων του αλγορίθμου ίσος με 100. Μεγαλύτερος αριθμός επαναλήψεων σημαίνει καλύτερη σύγκλιση του αλγορίθμου σε μια προσεγγιστική ισορροπία και κατά συνέπεια μικρότερα ε.

Όσον αφορά το ε_{WSNE} αυτό έχει πάντοτε τιμές μικρότερες του 0.8 με την πλειονότητα των περιπτώσεων να βρίσκονται στο διάστημα [0.4, 0.6]. Παρατηρούμε ότι το ε_{WSNE} είναι αρκετά μεγαλύτερο από το ε_{APPROX}. Αυτό ο οφείλεται στο ότι ενώ ο αλγόριθμος συγκλίνει σταδιακά προς μια ισορροπία Nash, δεν είναι απόλυτα «σίγουρος» για τις δράσεις που επιλέγει με αποτέλεσμα να αποδίδει πιθανότητες (έστω και πολύ μικρές) σε δράσεις που δεν προσφέρουν σημαντική ωφέλεια στον παίκτη. Αυτό έχει ως αποτέλεσμα οι δράσεις αυτές να βρίσκονται στα διανύσματα υποστήριξης με αποτέλεσμα να συμπεριλαμβάνονται στον υπολογισμό του ε_{WSNE} και να αυξάνουν την τιμή του.

FICTIOUS PLAY UNIFORM

Παρατηρούμε ότι όσον αφορά το ε_{APPROX} παρατηρούμε ότι έχουμε σχεδόν πανομοιότυπα αποτελέσματα με την άλλη παραλλαγή του αλγορίθμου πράγμα απόλυτα λογικό καθώς πρόκειται κατ' ουσία για τον ίδιο αλγόριθμο.

Ωστόσο στο σε σχέση με το ε_{WSNE} παρατηρούμε ότι ενώ κινείται στο ίδιο εύρος με την άλλη παραλλαγή του αλγορίθμου, η τιμές του τείνουν να κατανέμονται κάπως ποιο ομοιόμορφα. Αντιθέτως στον FP PBR παρατηρούμε ότι υπάρχει συγκέντρωση των τιμών σε συγκεκριμένα εύρη. Αυτό το φαινόμενο εκτιμούμε ότι οφείλεται στην διαφορά των δύο παραλλαγών. Ο UNIFORM αλγόριθμος επιλύει καταστάσεις ισοπαλίας μεταξύ PBR με ομοιόμορφη κατανομή πιθανοτήτων οδηγώντας το support διάνυσμα να περιλαμβάνει δράσεις που ακόμα και όταν δεν είναι προτιμητέες, οι επιπτώσεις τους δεν είναι τόσο μεγάλες λόγω του ομοιόμορφου «απλώματος» πιθανοτήτων και από μεριάς του αντιπάλου. Ο PBR αλγόριθμος επιλύει αυτές τις καταστάσεις με επιλογή της πρώτης γραμμής / στήλης οδηγώντας σε μεγαλύτερη πόλωση σε συγκεκριμένα εύρη ανάλογα με το πόσο καλά «συμβαδίζουν» οι δράσεις των αντιπάλων.