**2ο Σετ Ασκήσεων: Ισορροπίες Nash σε Διμητρικά Παιχνίδια**

Μπάλλας Μιχαήλ

A.Μ. 1072599

up1072599@upnet.gr

Σούρλας Ζήσης

Α.Μ. 1072477

sourlas.zisis@upnet.gr

/05/2023

**Θεωρητικό Ερώτημα 1 (Γεννήτρια Παιχνιδιών)**

**Θ.Ε. 1.1**

Εφόσον 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 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 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 . 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 ισχύει ότι . Κατά συνέπεια κυριαρχεί αυστηρά επί της .

**Θ.Ε. 2.2**

Αντίστοιχα για να ισχύει ότι για 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 . 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 ισχύει ότι . Κατά συνέπεια κυριαρχεί αυστηρά επί της .

**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 αποθηκευμένo 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 σε έναν πίνακα maxR.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 έναν πίνακα maxC. Επίσης αυτή η διαδικασία γίνεται για όλες τις στήλες .

Έπειτα διατρέχουμε πάλι για κάθε γραμμή όλες τις στήλες και ελέγχουμε 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 και 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 και δηλαδή τις αναμενόμενες ωφέλειες των παικτών. Έπειτα αφαιρεί από το μέγιστο των (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 διατρέχουμε όλες τις στήλες και αν βρεθεί ότι για την παίκτρια γραμμής υπάρχει (σε όλες τις στήλες) μηδενική ωφέλεια πάμε και διαγράφουμε την συγκεκριμένη γραμμή από τα μητρώα C και R. Έπειτα καλείται η συνάρτηση αναδρομικά.

Αν δεν υπάρχει πια γραμμή με μηδενική ωφέλεια για την παίκτρια γραμμής ελέγχουμε αντίστοιχα την κάθε στήλη και αν βρεθεί ότι για την συγκεκριμένη στήλη υπάρχει σε κάθε γραμμή μόνο μηδενική ωφέλεια για τον παίκτη στήλης διαγράφουμε την στήλη αυτή από τα μητρώα R και C. Έπειτα καλείται η συνάρτηση αναδρομικά.

Αν τελικά καθώς διατρέχουμε τις γραμμές και έπειτα τις στήλες δεν βρεθεί καμία γραμμή με μηδενική ωφέλεια για την παίκτρια γραμμής και καμία στήλη με μηδενική ωφέλεια για τον παίκτη στήλης επιστρέφονται τα νέα μητρώα R,C και οι νέες διαστάσεις τους μετά τις διαγραφές.

**Υλοποίηση 4**

***αpproxNEConstructionDMP(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:**

**χ:** 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.**

***αpproxNEConstructionDEL(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:**

**χ:** 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:**

***αpproxNEConstructionFPPBR(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:**

**χ:** 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.**

***αpproxNEConstructionFPUNI(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:**

**χ:** 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 ***α*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: -**

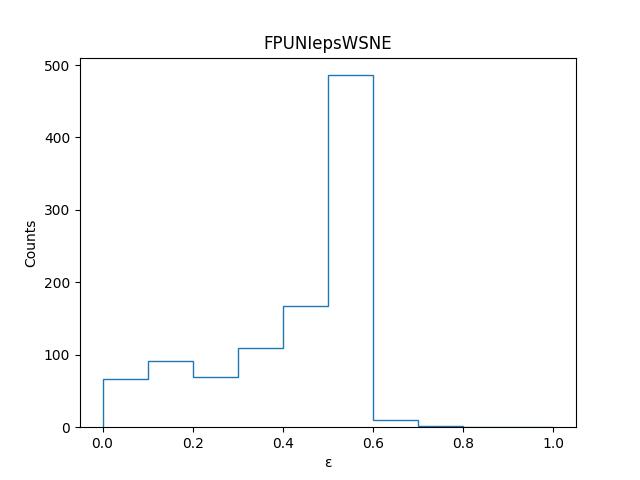
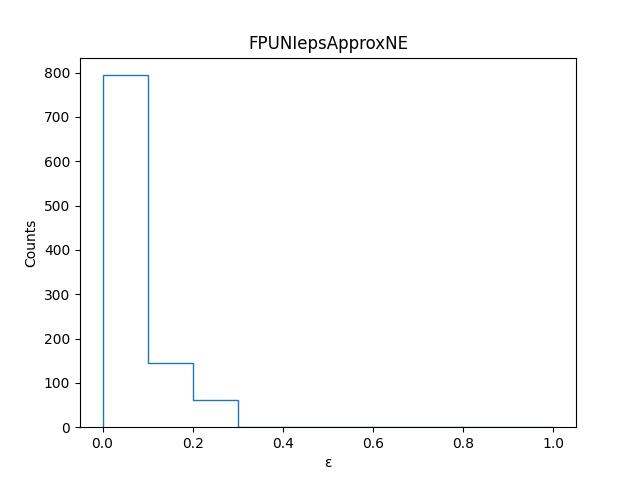
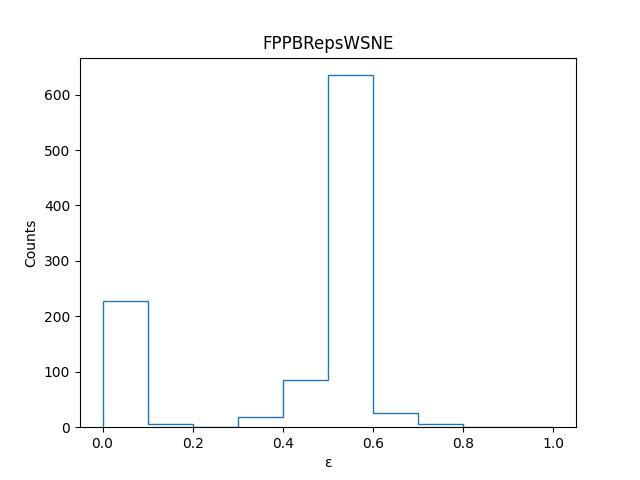
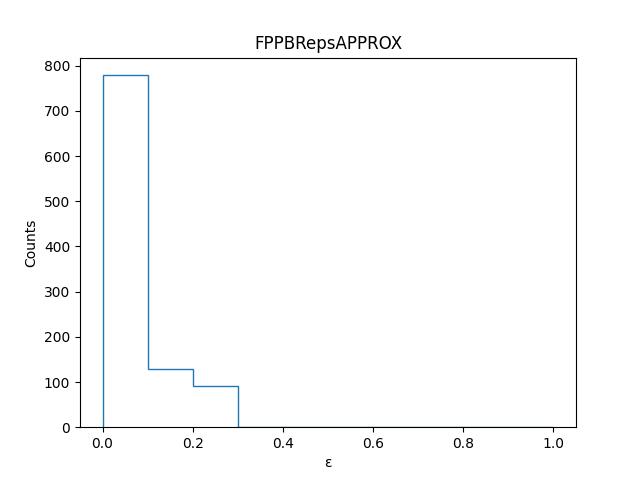
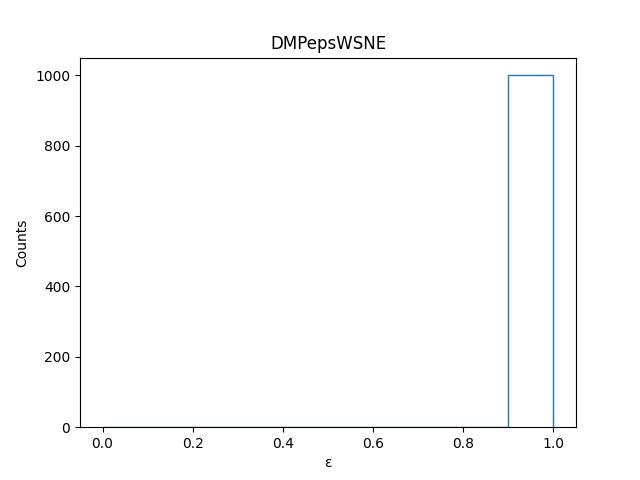
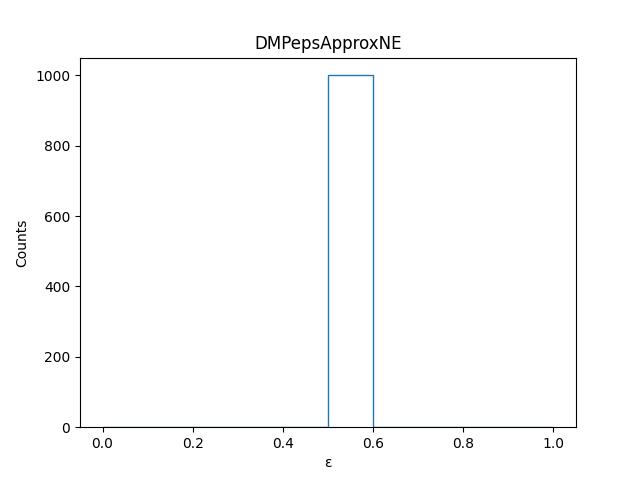
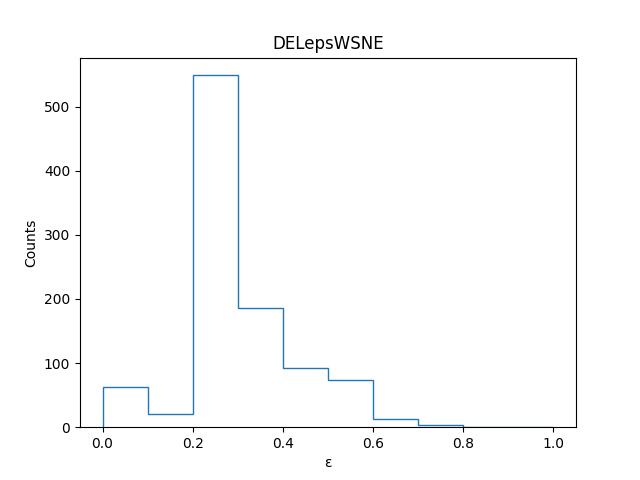
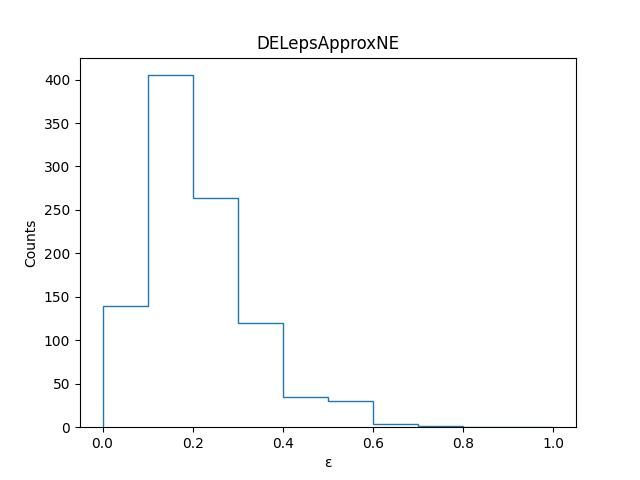
**Έξοδοι: -**

**Περιγραφή:**

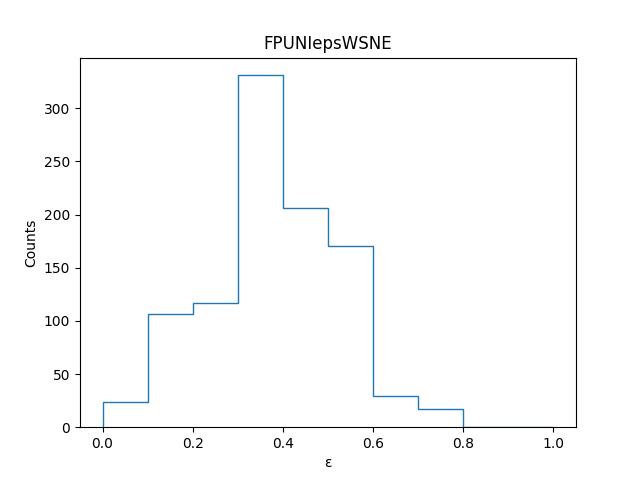
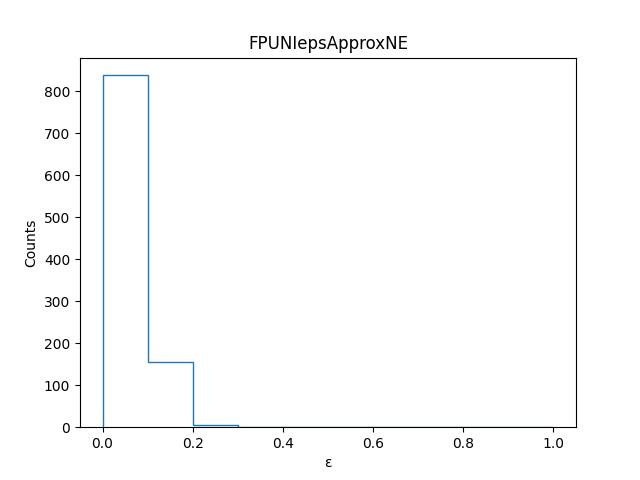
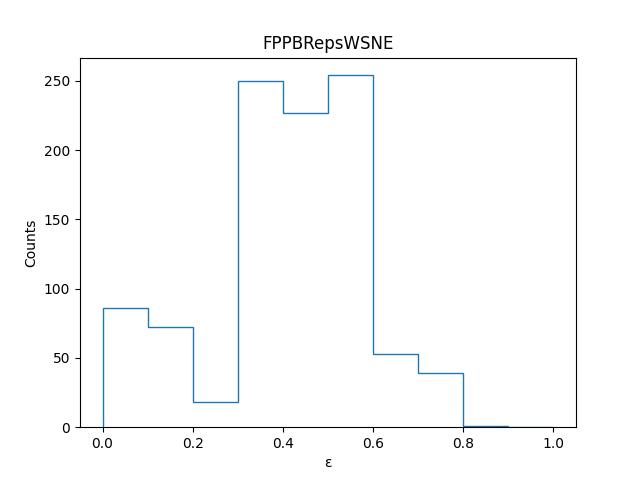
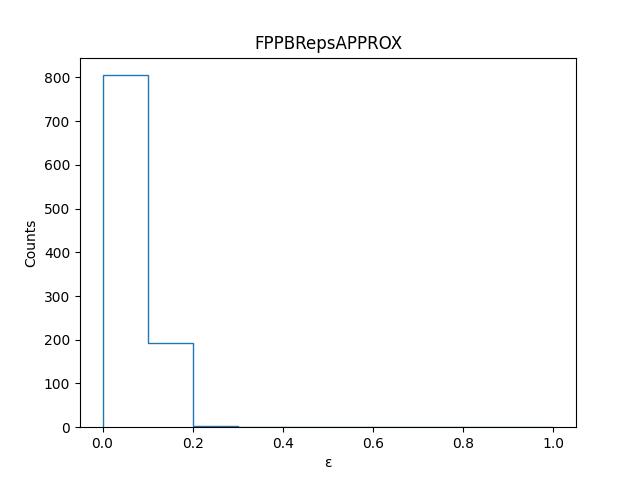
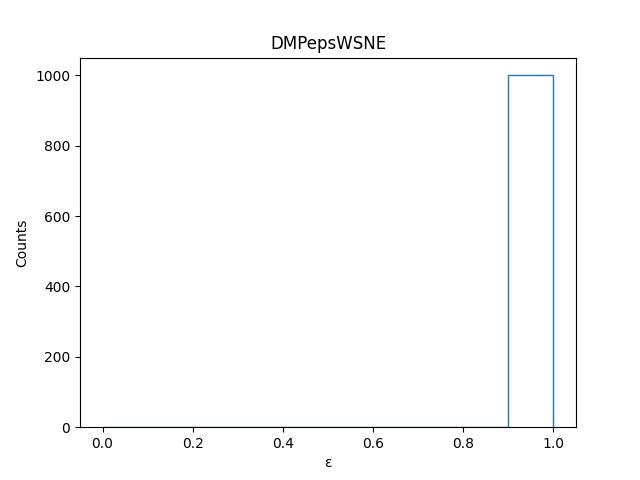
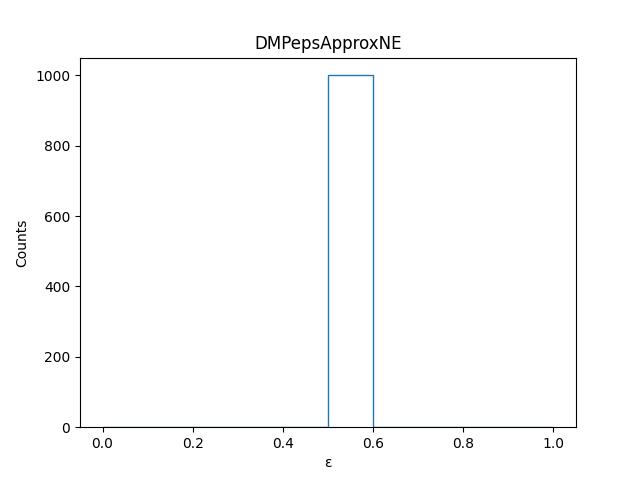
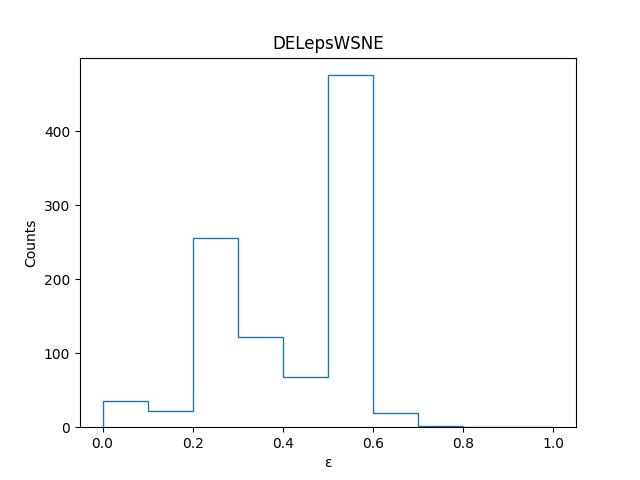
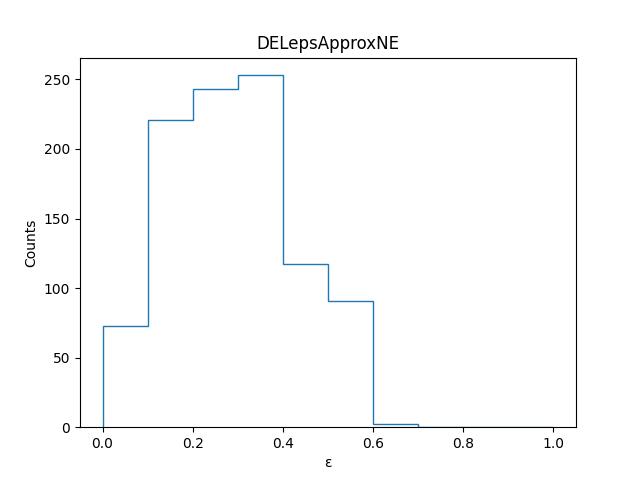
Η συνάρτηση ζητάει από τον χρήστη να επιλέξει ποιο πείραμα από τα 4 προεπιλεγμένα θέλει να εκτελέσει. Ανάλογα με την είσοδο θέτει κατάλληλα τις παραμέτρους του πειράματος και επαναληπτικά δημιουργεί παιχνίδια καλώντας την **generate\_winlose\_game\_without\_pne.** Σε κάθε επανάληψη αφαιρεί τις ισχυρά κυριαρχούμενες δράσεις του παιχνιδιού καλώντας την **removeStrictlyDominatedStrategies** και εκτελεί διαδοχικά τους 4 αλγορίθμους καλώντας τις κατάλληλες συναρτήσεις. Αποθηκεύει τα ε που επιστρέφουν οι αλγόριθμοι 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.

**Αξιολόγηση Πειραμάτων**

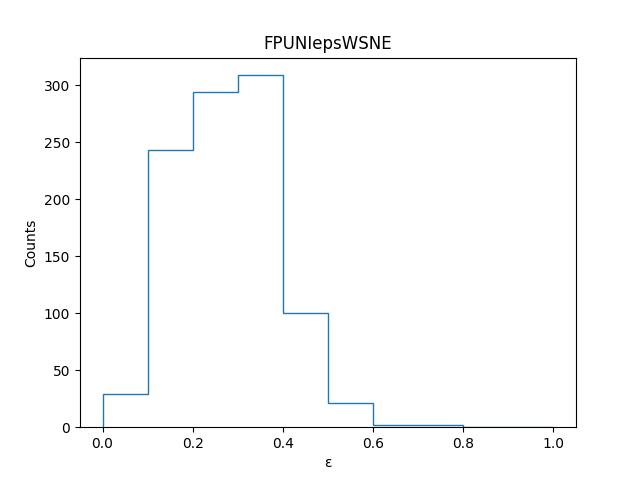
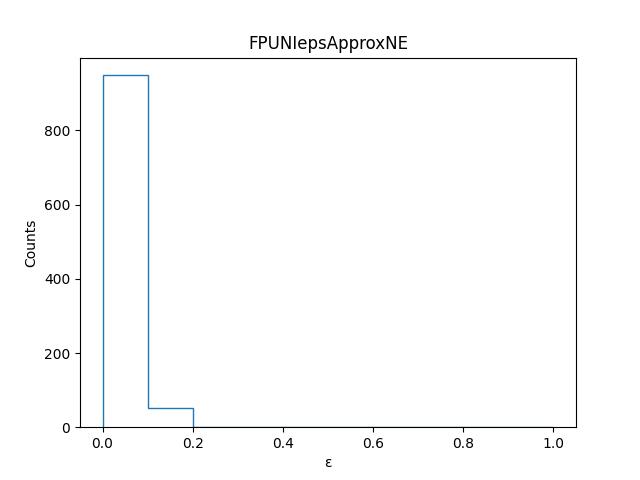
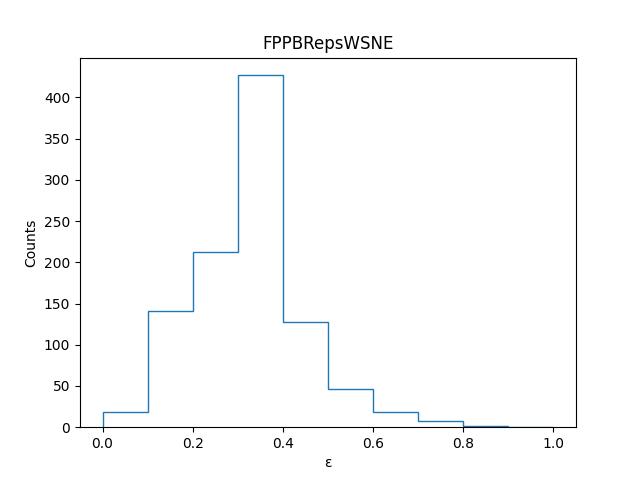
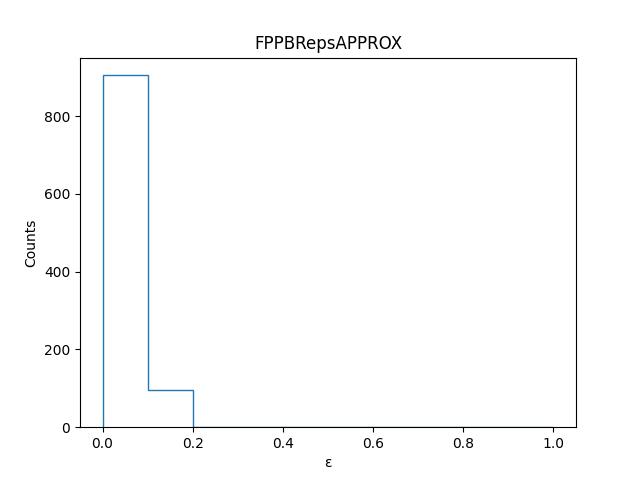
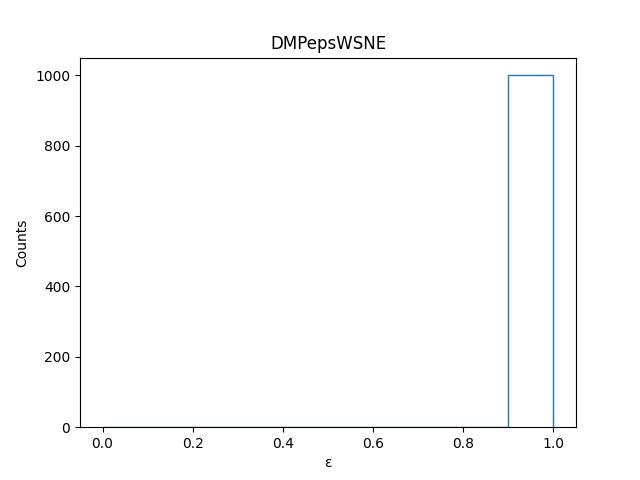
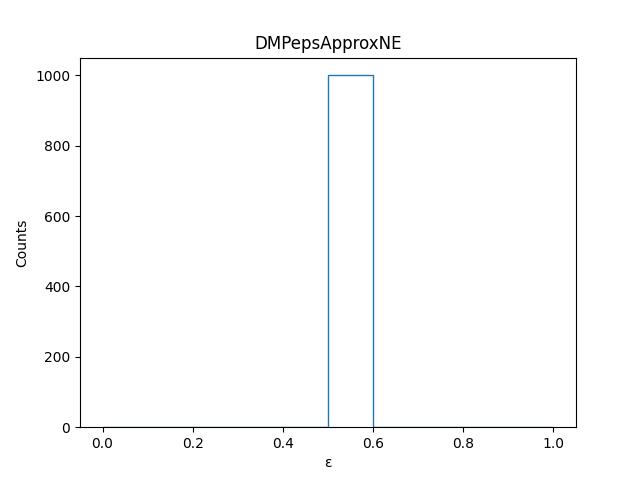
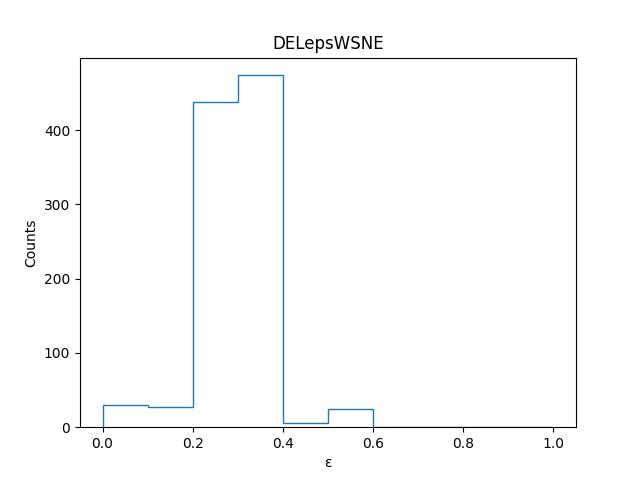
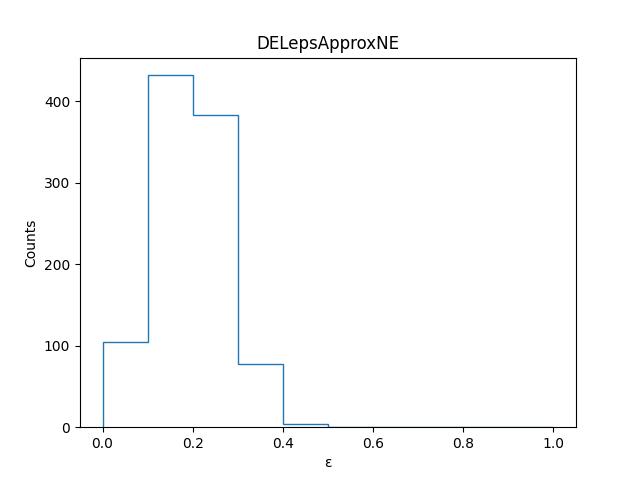
**Πείραμα Π1**



**Πείραμα Π2**



**Πείραμα Π3**



**Πείραμα Π4**

