



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη ευφούς πράκτορα για διαδικτυακές αγορές διαφημίσεων (Ad Exchanges)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΣΠΥΡΟΥ-ΧΡΙΣΤΟΦΟΡΟΥ ΘΕΟΔΩΡΟΠΟΥΛΟΥ

Επιβλέπων : Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Ιωάννης Βέτσικας
Ε.Κ.Ε.Φ.Ε. “Δημόκριτος”

Αθήνα, Ιούλιος 2016



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη ευφούς πράκτορα για διαδικτυακές αγορές διαφημίσεων (Ad Exchanges)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΣΠΥΡΟΥ-ΧΡΙΣΤΟΦΟΡΟΥ ΘΕΟΔΩΡΟΠΟΥΛΟΥ

Επιβλέπων : Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.

Ιωάννης Βέτσικας
Ε.Κ.Ε.Φ.Ε “Δημόκριτος”

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18/06/2016.

.....
Γεώργιος Στάμου
Επίκουρος Καθηγητής Ε.Μ.Π.

.....
Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

.....
Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2016

.....

ΣΠΥΡΟΣ-ΧΡΙΣΤΟΦΟΡΟΣ ΘΕΟΔΩΡΟΠΟΥΛΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2016 – All rights reserved

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη ενός ευφυούς πράκτορα λογισμικού στο ρόλο ενός διαφημιστικού δικτύου (Ad Network) σε μια κεντρική αγορά διαφημίσεων (Ad Exchange). Ο πράκτορας αναπτύχθηκε για το περιβάλλον προσομοίωσης του διαγωνισμού TAC AdX '16 στον οποίο και έλαβε μέρος κερδίζοντας την πρώτη θέση. Ο διαγωνισμός απαιτούσε την υλοποίηση τριών διαφορετικών στρατηγικών: στρατηγική για την ανάληψη διαφημιστικής εκστρατείας, για την αγορά διαφημιστικού χώρου (εντυώσεις) και για την αγορά δεδομένων για τους χρήστες του διαδικτύου. Οι στρατηγικές αυτές αφορούν δημοπρασίες δεύτερης τιμής με ανώτατη και κατώτατη τιμή, δεύτερης τιμής μόνο με κατώτατη τιμή και γενικευμένες δεύτερης τιμής αντίστοιχα. Τη σχεδιαστική πρόκληση αποτελεί η μεγιστοποίηση του κέρδους του πράκτορα ταυτόχρονα με την επαρκή ικανοποίηση των συμβολαίων.

Για την ανάληψη εκστρατειών προβλέπεται η πιθανότητα νίκης στη δημοπρασία με χαμηλό ποντάρισμα σε μορφή χρονοσειράς και γίνεται φιλτράρισμα των ζημιογόνων εκστρατειών με τη μέθοδο AdaBoostM1 πάνω σε δεδομένα από ελεγχόμενες προσομοιώσεις. Ανάλογα με αυτήν την πιθανότητα ο πράκτορας ποντάρει επιθετικά ή συντηρητικά. Για την αγορά εντυώσεων μεγιστοποιείται το συνολικό μακροπρόθεσμο κέρδος του πράκτορα με βάση ιστορικό ελεγχόμενων πειραμάτων και προσαρμόζεται το ποντάρισμα ανάλογα με την επιθετικότητα των ανταγωνιστών του με πρόβλεψη Naïve Bayes. Τέλος το ποντάρισμα για την πληροφορία αντιμετωπίζεται ως χρονοσειρά επιλέγοντας τυχαίες τιμές γύρω από ένα μεταβλητό σταθερό επίπεδο.

Οι προσομοιώσεις ελέγχου αλλά και ο διαγωνισμός έδειξαν ότι ο πράκτορας είναι ιδιαίτερα σταθερός και προσαρμοστικός, καταφέρνοντας να αναλύσει το περιβάλλον της αγοράς ώστε να πάρει καλές αποφάσεις.

Λέξεις Κλειδιά: Ευφρείς Πράκτορες, Διαδικτυακή Διαφήμιση, Ad Exchange, Trading Agent Competition (TAC), Δημοπρασίες, Βελτιστοποίηση, Μηχανική Μάθηση

Abstract

The purpose of this thesis was the development of an intelligent software agent to act as an Ad Network in an Ad Exchange market. The agent was developed for the TAC AdX '16 competition environment, in which it later took part achieving the first place. The competition required the implementation of three different strategies: campaign bidding, impression bidding and User Classification Service bidding. These regard second-price auctions with high and low reserve price, with only low reserve price and generalized second price auctions respectively. The challenge for the designer is the maximization of profit, while completing all contracts to a satisfactory degree.

For the campaign bidding, the probability of winning the auction with a small bid is predicted in the form of a time series while unprofitable campaigns are screened out using the AdaBoostM1 method on experiment-based data. Depending on this probability the agent bids aggressively or conservatively. For buying impressions the long-term total expected profit is maximized based again on a history of controlled experiments. The bid is then adjusted according to the aggressiveness of other competitors using the Naive Bayes method. Finally, the UCS bid is formed as a random value around a predicted base level.

The testing simulations as well as the competition show that the agent is both robust and adaptive as it manages to analyze the market environment and make good decisions.

Keywords: Intelligent Agents, Online Advertising, Ad Exchanges, Trading Agent Competition (TAC), Auctions, Optimization, Machine Learning

Ευχαριστίες

Θα ήθελα να ευχαριστήσω καταρχάς τον Δρ. Γιώργο Στάμου επίκουρο καθηγητή Ε.Μ.Π. που δέχτηκε να επιβλέψει την εκπόνηση αυτής της εργασίας. Ευχαριστώ ιδιαίτερα τον Δρ. Γιάννη Βέτσικα ερευνητή στο Ινστιτούτο Πληροφορικής και Τηλεπικοινωνιών του Ε.Κ.Ε.Φ.Ε. “Δημόκριτος” για την πολύτιμη καθοδήγησή του. Τέλος ευχαριστώ τον Γρηγόρη Δημόπουλο φοιτητή του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Ε.Κ.Π.Α. για τη γόνιμη συνεργασία του.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Η διαφήμιση στο διαδίκτυο.....	1
1.2	Η χρήση υπηρεσιών Ad Exchange στις διαδικτυακές διαφημίσεις.....	3
1.3	Ευφρείς πράκτορες για αγοραπωλησίες και οι διαγωνισμοί TAC.....	4
1.4	Ο διαγωνισμός TAC AdX.....	6
1.4.1	Γενική περιγραφή του παιχνιδιού.....	6
1.4.2	Η ροή του παιχνιδιού.....	7
1.4.3	Διαφημιστικές Εκστρατείες.....	8
1.4.4	Δημοπρασίες συμβολαίων.....	9
1.4.5	Δημοπρασίες εντυπώσεων.....	10
1.4.6	Υπηρεσία UCS.....	10
1.4.7	Διαφημιστικά Δίκτυα.....	11
2	Σχετικές εργασίες.....	12
2.1	Οι πράκτορες του TAC AdX 14'.....	12
2.1.1	Ο πράκτορας Agent00.....	13
2.1.2	Ο πράκτορας Agent2.....	14
2.1.3	Ο πράκτορας BCM.....	15
2.1.4	Ο πράκτορας giza.....	16
2.1.5	Ο πράκτορας ANL.....	17
2.2	Ιδέες και μέθοδοι από πράκτορες άλλων διαγωνισμών.....	18
2.2.1	Η πρόβλεψη τιμής στον διαγωνισμό classic TAC.....	18
3	Θεωρητικό υπόβαθρο.....	20
3.1	Δημοπρασίες δεύτερης τιμής και μηχανισμοί Vickrey-Clarke-Groves.....	20
3.2	Γενικευμένες δημοπρασίες δεύτερης τιμής.....	22
3.3	Συνδυαστικοί ταξινομητές και η μέθοδος boosting.....	22
3.3.1	Ταξινόμηση με δέντρα αποφάσεων.....	22
3.3.2	Bootstrap aggregation και bagging.....	24
3.3.3	Boosting και η μέθοδος AdaBoost.....	25
3.4	Η μέθοδος απλής εκθετικής εξομάλυνσης για την εξαγωγή προβλέψεων.....	27
3.5	Ταξινομητές Bayes.....	28

4	Η σχεδίαση του πράκτορα Hermes.....	30
4.1	Στρατηγική για τις δημοπρασίες συμβολαίων.....	30
4.1.1	Φιλτράρισμα ακατάλληλων εκστρατειών.....	31
4.1.1	Πρόβλεψη τιμής πριν τη δημοπρασία συμβολαίων.....	31
4.1.2	Πότε μια εκστρατεία είναι κατάλληλη.....	32
4.1.3	Εύρεση του κατάλληλου πονταρίσματος.....	32
4.1.4	Έλεγχος καταλληλότητας του πονταρίσματος βάσει προηγούμενων παιχνιδιών.....	33
4.2	Στρατηγική για τις δημοπρασίες εντυπώσεων.....	36
4.2.1	Εύρεση σημείου μέγιστου κέρδους για την τρέχουσα εκστρατεία.....	36
4.2.2	Εύρεση μέγιστου αναμενόμενου κέρδους για το υπόλοιπο του παιχνιδιού.....	38
4.2.2.1	Στατιστική προσέγγιση της συνάρτησης G.....	39
4.3	Προσαρμογή του πράκτορα στις αλλαγές των συνθηκών του παιχνιδιού.....	40
4.3.1	Προσαρμογή της ανταγωνιστικότητας c κατά τη διάρκεια ενός γύρου.....	40
4.3.1.1	Γιατί απλή εκθετική εξομάλυνση;.....	41
4.3.2	Προσαρμογή της τιμής αγοράς εντυπώσεων.....	42
4.4	Στρατηγική για την υπηρεσία κατηγοριοποίησης χρήστη (UCS).....	43
4.5	Ο πράκτορας dadNet.....	43
4.6	Ο τελικός πράκτορας DemokriTAC.....	44
5	Αξιολόγηση.....	46
5.1	Οργάνωση πειραμάτων.....	46
5.2	Παράμετροι αξιολόγησης.....	47
5.2.1	Παράμετροι για τη σύγκριση στρατηγικών για συμβόλαια.....	47
5.2.2	Παράμετροι για τη σύγκριση στρατηγικών αγοράς εντυπώσεων.....	48
5.2.3	Παράμετροι για τη σύγκριση στρατηγικών UCS.....	48
5.3	Αποτελέσματα.....	48
5.3.1	Επιλογή μεθόδου χαρακτηρισμού των υποψήφιων εργασιών.....	51
5.3.2	Αποτελέσματα σχετικά με τις στρατηγικές συμβολαίων.....	53
5.3.3	Επιλογή μεθόδου χαρακτηρισμού της επιθετικότητας.....	55
5.3.4	Αποτελέσματα σχετικά με τις στρατηγικές εντυπώσεων.....	56
5.3.5	Αποτελέσματα σχετικά με τις στρατηγικές UCS.....	59
5.4	Σύνοψη συμπερασμάτων αξιολόγησης.....	61
5.5	Αποτελέσματα του διαγωνισμού TAC AdX '16.....	62
5.5.1	Έλεγχος του τελικού πράκτορα.....	62
5.5.2	Αποτελέσματα διαγωνισμού.....	63

6	Τεχνικές λεπτομέρειες.....	66
6.1	Λεπτομέρειες υλοποίησης.....	67
6.2	Πλατφόρμες και προγραμματιστικά εργαλεία.....	82
7	Επίλογος.....	84
7.1	Σύνοψη και συμπεράσματα.....	84
7.2	Μελλοντικές επεκτάσεις.....	85
8	Βιβλιογραφία.....	86

1

Εισαγωγή

1.1 Η διαφήμιση στο διαδίκτυο

Μία από τις μεγαλύτερες και ταυτόχρονα πιο επικερδείς σύγχρονες τάσεις στον τομέα του ηλεκτρονικού εμπορίου είναι οι διαδικτυακές διαφημίσεις. Πλέον πολλές εταιρείες και οργανισμοί δίνουν ιδιαίτερη σημασία στην προσέλκυση καταναλωτών μέσω του διαδικτύου, γεγονός που έχει δημιουργήσει μια τεράστια αγορά με πολλές νέες υπηρεσίες.

Συγκεκριμένα, μπορούμε να χωρίσουμε τις διαδικτυακές διαφημίσεις σε πέντε τύπους, τις διαφημίσεις ηλεκτρονικού ταχυδρομείου, μηχανών αναζήτησης, κοινωνικών δικτύων, πλαισίου σε ιστοσελίδα (display – web banner) και φορητών συσκευών. Την παρούσα εργασία αφορούν κυρίως τα δύο τελευταία είδη.

Οι κύριες οντότητες στην αγορά που έχει δημιουργηθεί γύρω από τις διαδικτυακές διαφημίσεις είναι ο διαφημιστής (advertiser), που θέλει να προωθήσει κάποιο προϊόν ή υπηρεσία και ο εκδότης (publisher) που ενσωματώνει τη διαφήμιση στο περιεχόμενο της ιστοσελίδας του (π.χ. πλαίσια, βίντεο, αναδύομενο παράθυρο κ.α.). Άλλες βασικές οντότητες είναι τα διαφημιστικά πρακτορεία που παίζουν τον ρόλο του διαμεσολαβητή ανάμεσα στις δύο προηγούμενες και ειδικοί εξυπηρετητές που καταγράφουν στατιστικά και δημογραφικά στοιχεία. [7]

Ανάλογα με το ποιες οντότητες παίζουν τον ρόλο διαμεσολαβητή η διαδρομή που ακολουθείται από τον διαφημιστή μέχρι την προβολή της διαφήμισης από τον εκδότη ποικίλουν. Στην πιο απλή περίπτωση, ο εκδότης είναι εκείνος που διαλέγει ποιο διαφημιστικό περιεχόμενο θα προβάλλει, όπως συμβαίνει με πολυσύχναστες ιστοσελίδες που συχνά έχουν

δικά τους διαφημιστικά τμήματα. Άλλη συνήθης περίπτωση είναι ο εκδότης να διατηρεί συμβόλαιο με κάποιο διαφημιστικό πρακτορείο, το οποίο εξυπηρετεί διάφορους διαφημιστές. Υπάρχουν φορές όμως που η διαδρομή μπορεί να περιλαμβάνει περισσότερα στάδια, όπως σε περίπτωση που χρησιμοποιούνται υπηρεσίες Ad exchange. [25] Σε αυτές τις υπηρεσίες οι αγοροπωλησίες βασίζονται σε δημοπρασίες για τον χώρο προβολής των διαφημίσεων, όπως θα αναλυθεί με περισσότερη λεπτομέρεια παρακάτω.

Από οικονομικής άποψης ενδιαφέρον παρουσιάζουν και οι διαφορετικοί τρόποι πληρωμής των εκδοτών για τις διαφημίσεις που προβάλλουν. [33] Οι πιο διαδεδομένοι είναι:

- CPM (Cost Per Mille), όπου ο εκδότης πληρώνεται για κάθε χίλιες εντυπώσεις (impressions), δηλαδή φορές που φορτώθηκε η διαφήμιση από τον περιηγητή του χρήστη.
- CPC (Cost Per Click), όπου η πληρωμή γίνεται για κάθε φορά που ο χρήστης επιλέγει τον σύνδεσμο της διαφήμισης
- CPV (Cost Per View), CPE (Cost Per Engagement), όταν ο χρήστης απλά δει ή αλληλεπιδράσει με το περιεχόμενο της διαφήμισης αντίστοιχα.
- Πληρωμή βασισμένη στην επίδοση (Performance-based), όπου ο διαφημιστής μοιράζεται με τον εκδότη ένα μέρος των κερδών του που προήλθε από τη διαφήμιση, κάτι που μεταφέρει ένα μέρος της ευθύνης να πετύχει η διαφήμιση στον εκδότη.
- Σταθερό κόστος, που συνήθως κανονίζεται μέσω συμβολαίου κ.α.

Τα τελευταία χρόνια ο κλάδος της διαδικτυακής διαφήμισης αποτελεί έναν από τους πιο επικερδείς αλλά και αποτελεσματικούς οικονομικούς κλάδους φτάνοντας στις Η.Π.Α. το πρώτο μισό του 2013 τα 20.1 δις δολάρια. Το γεγονός αυτό δεν είναι καθόλου περίεργο αν αναλογιστεί κανείς, ότι οι διαφημίσεις προβάλλονται πλέον σε ένα πολύ ευρύ κοινό με πολύ χαμηλό κόστος. Επίσης μέσω της χρήσης ψηφιακών τεχνολογιών διευκολύνεται η συλλογή δεδομένων καθώς και η εξαγωγή συμπερασμάτων για την συμπεριφορά των χρηστών, με αποτέλεσμα οι διαφημίσεις να μπορούν να παρουσιάζονται στοχευμένα σε πολύ συγκεκριμένα τμήματα της αγοράς (market segments).

Φυσικά μέσα σε όλα αυτά εγείρονται πολλές αμφιβολίες σχετικά με την ιδιωτικότητα και την προστασία των προσωπικών δεδομένων των καταναλωτών καθώς και την έκθεση τους σε κακόβουλο περιεχόμενο μέσω των διαφημίσεων. Για αυτόν τον λόγο γίνονται πολλές προσπάθειες για τον νομικό έλεγχο του τομέα με στόχο την προστασία του χρήστη. [7]

1.2 Η χρήση υπηρεσιών Ad Exchange στις διαδικτυακές

διαφημίσεις

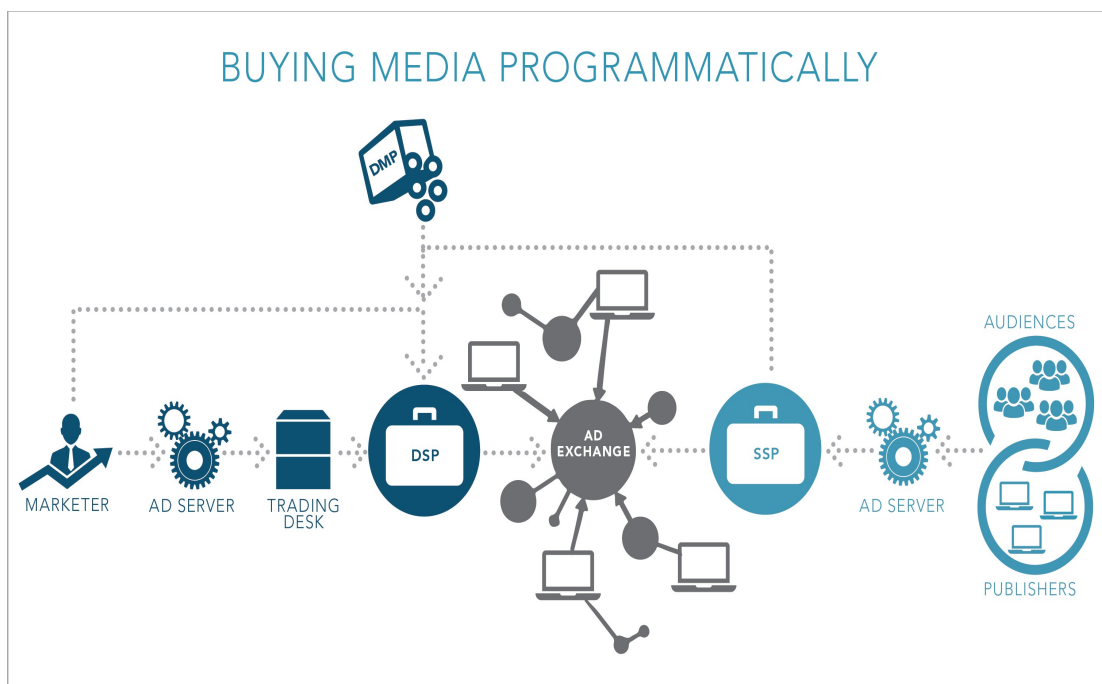
Όπως έχει αναφερθεί ο παραδοσιακός τρόπος αγοραπωλησίας διαφημιστικού χώρου βασιζόταν στη συνεννόηση ανάμεσα στον διαφημιστή (ή το πρακτορείο που τον εκπροσωπεί) και τον εκδότη. Λόγω της ραγδαίας ανάπτυξης του διαδικτύου τα τελευταία χρόνια, ο χώρος για προβολή διαφημίσεων σε ιστοσελίδες αυξήθηκε, με αποτέλεσμα να υπάρχουν πάρα πολλές σελίδες που να προσφέρουν τέτοιο χώρο ο οποίος όμως να μένει ανεκμετάλλευτος.

Το παραπάνω είχε ως επακόλουθο την εισαγωγή νέων οντοτήτων στην αγορά διαφημίσεων, τα διαφημιστικά δίκτυα (Ad Networks). Ένα διαφημιστικό δίκτυο έχει ως στόχο να συγκεντρώνει σελίδες με ανεκμετάλλευτο χώρο και να δίνει προσφορές στους διαφημιστές για την αξιοποίηση του.

Λόγω της ευκολίας εισόδου ενός διαφημιστικού δικτύου στην αγορά, εμφανίστηκαν γρήγορα πολλές τέτοιες επιχειρήσεις, με αποτέλεσμα η αγορά να γίνει χαώδης και να είναι δύσκολο να βρεθεί ο σωστός διαμεσολαβητής και από τις δύο πλευρές. Την ανάγκη αυτή ήρθαν να καλύψουν οι υπηρεσίες ανταλλαγής διαφημίσεων (Ad Exchanges). Στόχος μιας υπηρεσίας Ad Exchange είναι οι αγοραπωλησίες διαφημίσεων να γίνονται συγκεντρωτικά, δημοπρατώντας τον χώρο προβολής των ιστοσελίδων σε πραγματικό χρόνο. Δημοφιλείς τέτοιες υπηρεσίες είναι οι Doubleclick, AppNexus, AdECN, Microsoft AdExchange. [25]

Οι υπηρεσίες αυτές έδωσαν την ευκαιρία οι διαδικτυακές διαφημίσεις να λαμβάνουν χώρα πιο αποτελεσματικά και σε πιο ανταγωνιστικό κόστος. Επίσης λόγω του κεντροποιημένου τρόπου αγοραπωλησίας ήταν πλέον εφικτό να μαζευτούν μεγάλες και αξιόπιστες ποσότητες δεδομένων με στόχο τη βελτίωση του όλου συστήματος. [28]

Προσπαθώντας να εκμεταλλευτούν τα παραπάνω οι διαφημιστές δημιούργησαν τις δικές τους πλατφόρμες διαχείρισης των αγορών (DSP, Demand Side Platforms), οι οποίες θα διαχειρίζονταν τα διαθέσιμα δεδομένα (π.χ. πρότυπα συμπεριφοράς, δημογραφικά στοιχεία) για να βελτιστοποιήσουν τις αγορές. Το αντίστοιχο συνέβη και με τους εκδότες με τις SSP (Sell Side Platforms). Η τελική, σύνθετη μορφή της αγοράς φαίνεται στο παρακάτω σχήμα:



1.1 Η αγορά διαδικτυακών διαφημίσεων με ad exchanges

Πηγή: www.bnd.com

Για να αντεπεξέλθουν στα νέα δεδομένα τα διαφημιστικά δίκτυα επεκτάθηκαν, συμμετέχοντας πλέον και στις αγοραπωλησίες στους νέους κεντρικούς χώρους δημοπρασιών. Έτσι ερχόμαστε στο θέμα που πραγματεύεται αυτή η εργασία, που είναι η ανάπτυξη ενός ευφυούς πράκτορα που εκπροσωπεί ένα διαφημιστικό δίκτυο σε έναν χώρο δημοπρασιών AdExchange.

Παρότι η βιβλιογραφία, που θα εξετασθεί και παρακάτω (Ενότητα 2.1) σχετικά με την σχεδίαση τέτοιων πρακτόρων είναι λιγοστή, είναι ξεκάθαρη η ανάγκη της βέλτιστης και προσαρμοστικής διαχείρισής τους. Επίσης όπως αναφέρεται και στην επόμενη ενότητα, οι ευφυείς πράκτορες παρουσιάζουν ενδιαφέρον και εφαρμογές σε διάφορα συστήματα αγοραπωλησιών, και θα μπορούσαν σίγουρα να βρουν εφαρμογή και στις αγοραπωλησίες διαφημίσεων.

1.3 Ευφυείς πράκτορες για αγοραπωλησίες και οι διαγωνισμοί

TAC

Ένας ευφυής πράκτορας για αγοραπωλησίες (trading agent) είναι ένα αυτόνομο πρόγραμμα με σκοπό να διευκολύνει τον χρήστη του, ο οποίος μπορεί να στερείται της απαραίτητης

εξειδικευμένης γνώσης, σε συναλλαγές συγκεκριμένου είδους (π.χ. συναλλαγές που αφορούν ένα ταξίδι), με βάση τις προτιμήσεις του.

Οι πράκτορες αγοραπωλησιών έχουν συγκεντρώσει μεγάλο ερευνητικό ενδιαφέρον, κυρίως μέσω των διαγωνισμών TAC (Trading Agent Competitions) που διοργανώνονται από το 2001. Στον κλασικό διαγωνισμό TAC [16] για παράδειγμα οι διαγωνιζόμενοι υλοποιούν ευφυείς ταξιδιωτικούς πράκτορες, που προσπαθούν να συνθέσουν ταξιδιωτικά πακέτα ανάλογα με τις ανάγκες των πελατών τους. Ένας τέτοιος πράκτορας πρέπει να μεγιστοποιήσει την ικανοποίηση των πελατών με το μικρότερο δυνατό κόστος αγοράζοντας μια σειρά από συμπληρωματικά αγαθά σε δημοπρασίες: εισιτήρια για τις πτήσεις, διαμονή σε ξενοδοχείο, εισιτήρια για μουσεία, πάρκα ψυχαγωγίας κλπ.

Στο παραπάνω παιχνίδι υπάρχουν οχτώ πελάτες και οχτώ πράκτορες που ανταγωνίζονται μεταξύ τους. Η μεταξύ τους αλληλεπίδραση σε συνδυασμό με την συμπληρωματικότητα των αγαθών και τους δύο αντικρουόμενους στόχους, μεγιστοποίηση της ικανοποίησης του πελάτη από τη μία και ελαχιστοποίηση του κόστους από την άλλη κάνουν το παιχνίδι ιδιαίτερα πολύπλοκο.

Με αντίστοιχη λογική έχουν δομηθεί και οι παρακάτω διαγωνισμοί:

- *TAC Supply Chain Management*, όπου οι πράκτορες αναλαμβάνουν την οργάνωση της εφοδιαστικής αλυσίδας βιομηχανιών έτσι ώστε να ικανοποιούν τη ζήτηση με ελάχιστο κόστος. [35]
- *Power TAC*, όπου οι πράκτορες ως μεσάζοντες ανάμεσα στην παραγωγή και την κατανάλωση ηλεκτρικής ενέργειας προσπαθούν πάλι να ικανοποιήσουν οικονομικά τη ζήτηση της. [31]
- *TAC Ad Auctions*, ο οποίος είναι και ο πιο κοντινός στον διαγωνισμό TAC AdX που αναφέρεται η παρούσα εργασία και αφορά διαφημίσεις σε μηχανές αναζήτησης. Συγκεκριμένα, οι πράκτορες εκπροσωπούν την εταιρεία που θέλει να διαφημιστεί (advertiser) και συμμετέχουν σε δημοπρασίες έτσι ώστε οι διαφημίσεις τους να προβληθούν κάθε φορά που αναζητείται μία συγκεκριμένη λέξη κλειδί. [21] [22]

Σε όλους τους παραπάνω διαγωνισμούς όπως και στον TAC AdX η αγορά προσομοιώνεται με έναν κεντρικό εξυπηρετητή, περιλαμβάνοντας πάντοτε κάποιον παράγοντα τυχαιότητας και προσπαθώντας να αναπαρασταθούν όσο γίνεται με μεγαλύτερη ακρίβεια οι αντίστοιχες πραγματικές συνθήκες.

Τα παραπάνω περιβάλλοντα όπως και τα πραγματικά, δυσχεραίνουν την παιγνιοθεωρητική ανάλυση για επιλογή της καλύτερης στρατηγικής. Έτσι οι στρατηγικές που ακολουθούνται συνήθως βασίζονται αρκετά σε ευρετικές τεχνικές (heuristics) και μεθόδους μηχανικής ή/και στατιστικής εκμάθησης των δεδομένων. [15][20]

Επιπλέον στόχο των παραπάνω αποτελεί όχι μόνο η ανάπτυξη αποτελεσματικών πρακτόρων για τη διενέργεια των συναλλαγών, αλλά και η δοκιμή και βελτίωση των μηχανισμών που υπάρχουν σε αυτές τις αγορές έτσι ώστε ο ανταγωνισμός να είναι πιο δίκαιος.

1.4 Ο διαγωνισμός TAC AdX

Ο διαγωνισμός TAC AdX δημιουργήθηκε για να ενθαρρύνει την ανάπτυξη πρακτόρων λογισμικού που δρουν σε περιβάλλοντα Ad Exchange με βέλτιστο τρόπο. Ένας βέλτιστος πράκτορας σε αυτόν τον διαγωνισμό θα πρέπει να εκτελεί τα συμβόλαια που παίρνει από πελάτες με το μικρότερο δυνατό κόστος, φροντίζοντας παράλληλα να διατηρεί τους πελάτες ευχαριστημένους, εξασφαλίζοντας έναν ικανοποιητικό αριθμό εντυπώσεων. Καταφέροντας αυτό, ο πράκτορας μεγιστοποιεί το συνολικό του κέρδος. Όπως παρατηρούμε και ο TAC AdX ακολουθεί το ίδιο μοτίβο με τους υπόλοιπους διαγωνισμούς TAC, δηλαδή την ικανοποίηση δύο αντικρουόμενων στόχων. [13]

1.4.1 Βασικές οντότητες του παιχνιδιού

Στο παιχνίδι TAC AdX συμμετέχουν οχτώ πράκτορες, σε ρόλο διαφημιστικών δικτύων, που έχουν σαν στόχο τη μεγιστοποίηση του κέρδους τους μετά από 60 ημέρες. Οι πράκτορες συνδέονται σε έναν εξυπηρετητή ο οποίος προσομοιώνει τις εξής βασικές οντότητες του παιχνιδιού:

- Τους πελάτες, οι οποίοι αναζητούν κάποιο διαφημιστικό δίκτυο για να εκπληρώσει τα συμβόλαιά τους.
- Την υπηρεσία Ad Exchange, στην οποία δημοπρατείται ο χώρος προβολής των διαφημίσεων.
- Την υπηρεσία κατηγοριοποίησης των χρηστών (User Classification Service), η οποία παρέχει στους πράκτορες πληροφορίες σχετικά με τους χρήστες που επισκέπτονται τις σελίδες με τους δημοπρατούμενους χώρους προβολής.

1.4.2 Η ροή του παιχνιδιού

Την πρώτη ημέρα του παιχνιδιού ανατίθεται σε κάθε πράκτορα ένα τυχαίο συμβόλαιο, με όλα τα χαρακτηριστικά του επιλεγμένα τυχαία εκτός από τη διάρκεια που είναι πέντε ημέρες. Στις

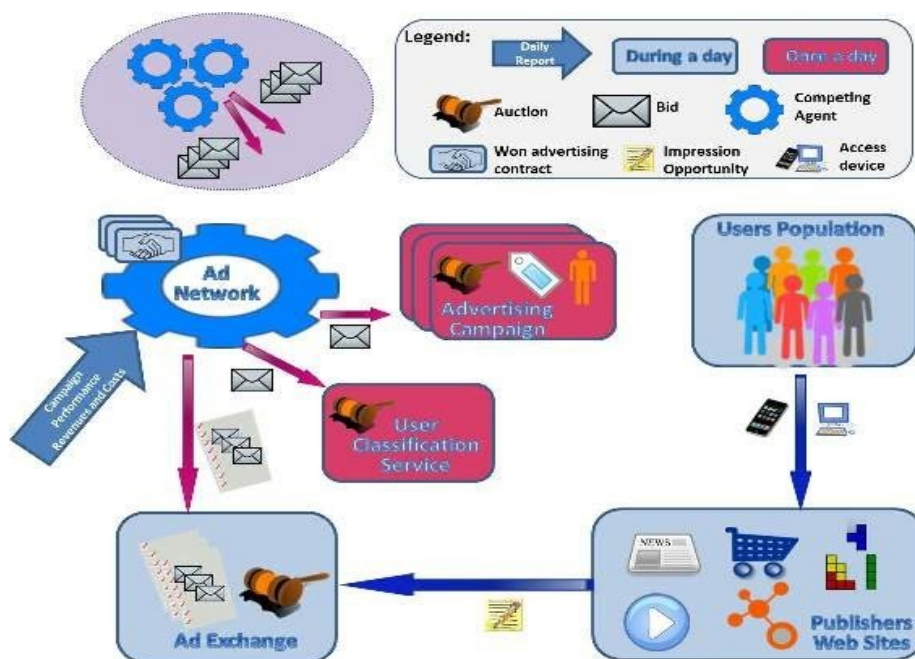
επόμενες ημέρες τα συμβόλαια δημοπρατούνται σε δημοπρασίες δεύτερης τιμής με κρυφό ποντάρισμα, όπου νικητής είναι ο πράκτορας με το χαμηλότερο ποντάρισμα.

Μαζί με το ποντάρισμά του για να αναλάβει μια εκστρατεία κάθε πράκτορας ποντάρει σε άλλη μια δημοπρασία δεύτερης τιμής με κρυφό ποντάρισμα, για τον βαθμό εξυπηρέτησης από την υπηρεσία UCS. Ανάλογα με το ύψος του πονταρίσματος ο πράκτορας λαμβάνει περισσότερη πληροφορία για τις εντυπώσεις που δημοπρατούνται στην Ad Exchange. Η καινούργια εκστρατεία ξεκινά πάντα δύο μέρες μετά από τη δημοπρασία, με μια μέρα να παρεμβάλλεται για να ανακοινωθεί ο νικητής.

Στη συνέχεια ο πράκτορας καλείται να ποντάρει για τις εντυπώσεις που διατίθενται από την υπηρεσία Ad Exchange. Και εδώ η δημοπρασία είναι δεύτερης τιμής με κρυφό ποντάρισμα, με το μεγαλύτερο ποντάρισμα να είναι και το νικητήριο.

Τέλος ο πράκτορας μαθαίνει πόσες εντυπώσεις κατάφερε να αγοράσει. Όταν μια διαφημιστική εκστρατεία τελειώνει μεταβάλλεται ο βαθμός ποιότητας του πράκτορα, ανάλογα με τον βαθμό συμπλήρωσής της.

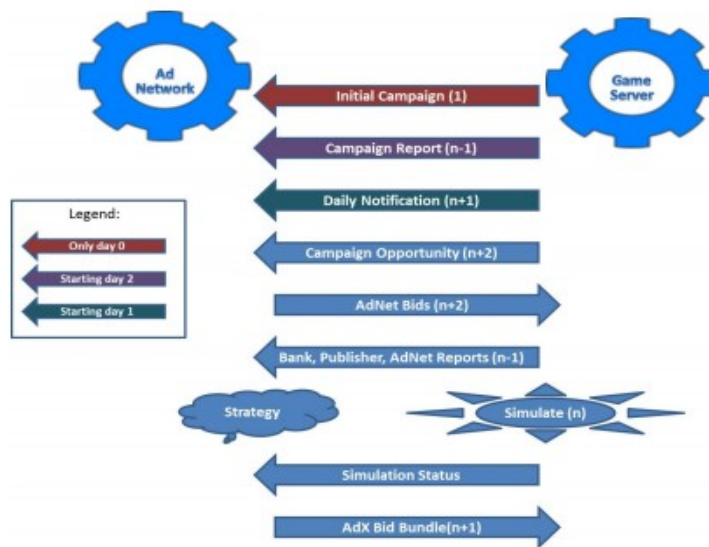
Η παραπάνω ροή φαίνεται πιο ξεκάθαρα στο παρακάτω σχήμα:



1.2 Σχέσεις μεταξύ των οντοτήτων του παιχνιδιού TAC-AdX

Πηγή: sites.google.com/site/gameadx/

Ενώ στο επόμενο σχήμα φαίνεται η ακριβής σειρά ανταλλαγής μηνυμάτων μεταξύ πράκτορα και εξυπηρετητή:



1.3 Η ανταλλαγή μηνυμάτων μεταξύ του πράκτορα και του server

Πηγή: sites.google.com/site/gameadx/

Στις επόμενες υποενότητες αναλύεται ο ρόλος που παίζουν οι βασικές οντότητες κατά της διάρκεια του παιχνιδιού καθώς και η λεπτομερής λειτουργία τους σύμφωνα με τις προδιαγραφές του διαγωνισμού.

1.4.3 Διαφημιστικές εκστρατείες

Οι διαφημιστικές εκστρατείες που διεκδικούν και στη συνέχεια καλούνται να εκπληρώσουν οι πράκτορες έχουν τα εξής βασικά χαρακτηριστικά:

1. Τμήμα αγοράς: κάθε εκστρατεία αφορά ένα συγκεκριμένο τμήμα του πληθυσμού. Κάθε μέρα οι 10000 χρήστες που επισκέπτονται τις ιστοσελίδες διαφήμισης χωρίζονται σε OLD/YOUNG , LOW INCOME/HIGH INCOME, MALE/ FEMALE με σταθερές πιθανότητες. Κάθε τμήμα αγοράς είναι ένας συνδυασμός των παραπάνω υποσυνόλων, π.χ [YOUNG, HIGH INCOME], [MALE], [YOUNG, LOW INCOME, FEMALE].
2. Απαιτούμενες εντυπώσεις (C_R): ο αριθμός των εντυπώσεων που πρέπει να αγοράσει ο πράκτορας. Πρέπει να είναι στοχευμένες στο τμήμα της αγοράς της εκστρατείας που έχει αναλάβει.

3. Προϋπολογισμός (C_B): ο προϋπολογισμός που είναι διαθέσιμος στον πράκτορα για να αγοράσει τις απαιτούμενες εντυπώσεις. Είναι ίσος με το ποντάρισμά του στις δημοπρασίες συμβολαίων.
4. Βαθμός ολοκλήρωσης: ο βαθμός ολοκλήρωσης (Effective Reach Ratio) υπολογίζεται από τον τύπο:

$$ERR_n(C) = \frac{2}{a} \left(\arctan \left(a \frac{I_n(C)}{C_R} - b \right) - \arctan(-b) \right) \quad (1)$$

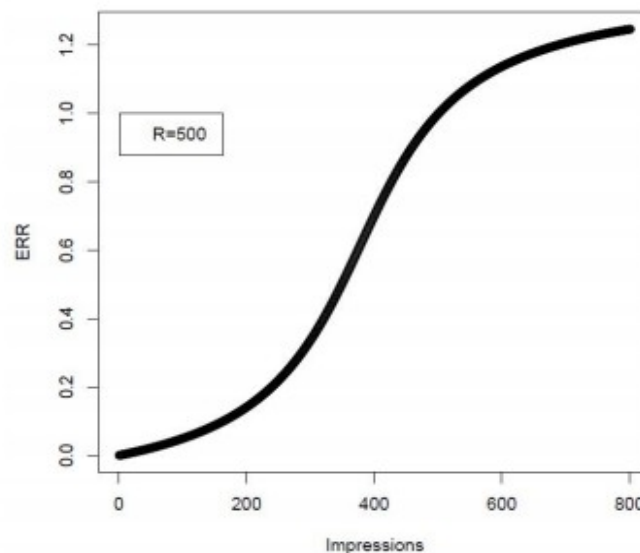
Όπου $I_n(C)$ οι εντυπώσεις που αγόρασε ο πράκτορας για την εκστρατεία C και $a = 4.08577$, $b = 3.08577$. Με βάση αυτόν υπολογίζονται τα έσοδα από την εκστρατεία:

$$E_n(C) = ERR_n(C) \cdot C_B \quad (2)$$

Καθώς και ο βαθμός ποιότητας:

$$Q_n^{new} = (1 - \eta) \cdot Q_{n-1} + \eta \cdot ERR_n(C) \quad (3)$$

Παρακάτω φαίνεται η γραφική παράσταση του ERR για $C_R = 500$.



1.4: Το ERR (Effective Reach Ratio) για μια εκστρατεία με ζητούμενες εντυπώσεις $R = 500$

Πηγή: sites.google.com/site/gameadx/

1.4.4 Δημοπρασίες συμβολαίων

Όπως έχει ήδη αναφερθεί, οι δημοπρασίες συμβολαίων είναι δημοπρασίες δεύτερης τιμής με κρυφό ποντάρισμα (second-price sealed bid auctions). Για τα επιτρεπτά πονταρίσματα πρέπει

να ισχύουν: $B_n Q_n > C_R R_{CampaignMin}$ (4) και $\frac{B_n}{Q_n} < C_R R_{CampaignMax}$ (5). Τα παραπάνω όρια

έχουν τεθεί, προφανώς για να μην είναι ανεξέλεγκτα χαμηλά η ψηλά τα πονταρίσματα αλλά και για να ευνοήσουν τους παίκτες που διατηρούν καλό Q, δίνοντάς τους μεγαλύτερο εύρος να ποντάρουν.

Επίσης έχει γίνει προσθήκη και ενός παράγοντα τυχαιότητας $P_{RandomCampaign}$ σύμφωνα με τον οποίον ανατίθεται μια εκστρατεία τυχαία σε κάποιον από τους παίκτες εκτός αυτού με το μικρότερο ποντάρισμα.

1.4.5 Δημοπρασίες εντυπώσεων

Ο εξυπηρετητής του παιχνιδιού προσομοιώνει την δημοπρασία που λαμβάνει χώρα στις υπηρεσίες Ad Exchange, όπου οι εκδότες (publishers) πωλούν εντυπώσεις για τις ιστοσελίδες τους.

Η συμπεριφορά των επισκεπτών των σελίδων προσομοιώνεται επίσης από τον εξυπηρετητή. Κάθε μέρα 10000 χρήστες επισκέπτονται τις σελίδες του παιχνιδιού. Κάθε σελίδα w έχει μια δημοτικότητα $P_W(w)$ και κάθε εντύπωση έχει πιθανότητα $P_{Adtype} \langle \cdot | w \rangle$ να είναι είτε τύπου βίντεο είτε τύπου κειμένου. Επίσης υπάρχει πιθανότητα $P_{Device} \langle \cdot | w \rangle$ ο χρήστης να χρησιμοποιεί Desktop ή κινητή συσκευή. Αν η εντύπωση προέρχεται από κινητή συσκευή ή/και είναι βίντεο, τότε προσμετράται σαν περισσότερες εντυπώσεις (βλ. Πίνακες στο τέλος της ενότητας). Έτσι η πιθανότητα ένας χρήστης με ηλικία a , φύλλο g και εισόδημα i να επισκεφτεί μια ιστοσελίδα είναι:

$$Pr([a,g,i]visitsw) = P_{Age} \langle a | w \rangle P_{Gender} \langle g | w \rangle P_{Income} \langle i | w \rangle P_W(w) \quad (6)$$

Επίσης κάθε χρήστης έχει μία πιθανότητα $P_{Continue}$ να συνεχίσει να επισκέπτεται άλλες σελίδες μετά την πρώτη του επίσκεψη για έναν σταθερό αριθμό φορών $N_{ContinueMax}$.

Τέλος οι δημοπρασίες εντυπώσεων έχουν συχνά και μια οριακή τιμή, κάτω από την οποία το ποντάρισμα θεωρείται άκυρο. Η τιμή αυτή τίθεται τυχαία με έναν από τρεις διαφορετικούς

τρόπους ανάλογα με τα προηγούμενα πονταρίσματα και τα χαρακτηριστικά προέλευσης της εντύπωσης (για λεπτομέρειες βλ. Κεφάλαιο 4).

1.4.6 Υπηρεσία UCS

Η υπηρεσία UCS καθορίζει την πιθανότητα με την οποία κάθε πράκτορας μαθαίνει σε ποιο τμήμα της αγοράς ανήκει μια εντύπωση που πιθανόν να θέλει να αγοράσει. Οι πράκτορες συμμετέχουν σε μια γενικευμένη δημοπρασία δεύτερης τιμής, όπου ο πράκτορας με το μεγαλύτερο ποντάρισμα βλέπει τα χαρακτηριστικά των χρηστών με πιθανότητα 1, ενώ ένας πράκτορας στη θέση $i > 1$ τα βλέπει με πιθανότητα $P_{UserRevelation}^{i-1}$.

Τα τελικά έξοδα ενός πράκτορα με ποντάρισμα c_i για την υπηρεσία είναι:

$$K_i = P_{UserRevelation}^{i-1} \cdot c_{i+1} \quad (7)$$

1.4.7 Διαφημιστικά δίκτυα

Σύμφωνα με τα προηγούμενα η συνάρτηση που πρέπει να μεγιστοποιήσει κάθε πράκτορας-διαφημιστικό δίκτυο είναι αυτή του συνολικού κέρδους:

$$N_n = \sum_C N_n(C) - \sum_d K_n(d) \quad (8)$$

Όπου:

$$N_n(C) = E_n(C) - \sum_{i \in I_n(C)} c_i \quad (9)$$

το κέρδος από κάθε εκστρατεία που έχει αναλάβει το διαφημιστικό δίκτυο και $K_n(d)$ το ημερήσιο κόστος για την υπηρεσία UCS.

2

Σχετικές εργασίες

Μιας και ο διαγωνισμός έχει διοργανωθεί μονάχα δύο φορές δεν υπάρχει μεγάλη βιβλιογραφία σχετικά με αυτόν. Ήταν όμως διαθέσιμες οι περιγραφές των πρακτόρων που είχαν συμμετάσχει στον 1ο διαγωνισμό το 2014. Επίσης δεδομένου ότι όλοι οι διαγωνισμοί TAC παρουσιάζουν ομοιότητα ως προς τη λογική τους, χρησιμοποιήθηκαν και διάφορες ιδέες από πράκτορες άλλων διαγωνισμών.

2.1 Οι πράκτορες του TAC AdX '14

Στον TAC AdX 14' που ήταν τριήμερος συμμετείχαν 8 πράκτορες οι οποίοι διαγωνίστηκαν σε 120 παιχνίδια (40 κάθε μέρα), με τα παρακάτω αποτελέσματα:

Table 1: Results (\$) of the TAC AdX'14

Agent	Day 1	Day 2	Day 3	Total
ANL	1098.4	704.0	1266.5	3068.9
giza	447.8	748.6	732.4	1928.8
Agent2	596.8	454.5	363.6	1414.8
tau	-93.8	488.0	727.7	1122.0
livadx	233.5	227.7	123.4	584.5
blue	60.1	169.8	82.3	312.2
WinnieTheBot	-0.8	119.5	80.6	199.4
Amunra	-54.0	-1.8	0.0	-55.9

2.1 Αποτελέσματα του TAC AdX '14 [4]

Στη συνέχεια γίνεται μια περιγραφή ορισμένων από τους παραπάνω πράκτορες που ξεχώρισαν, καθώς και ορισμένων ακόμα που υπάρχουν στη βιβλιογραφία του διαγωνισμού και προσέφεραν πολύτιμες ιδέες.

2.1.1 Ο πράκτορας Agent00

Ο πράκτορας έχει σαν στόχο τη μεγιστοποίηση της πιθανότητας να κερδίζει νέες διαφημιστικές εκστρατείες και στη συνέχεια να αγοράζει όσες περισσότερες εντυπώσεις μπορεί ώστε να μεγιστοποιεί το κέρδος και την ποιότητά του. [8]

Στρατηγική δημοπρασιών συμβολαίου:

Στις δημοπρασίες συμβολαίου ο πράκτορας δίνει προτεραιότητα στο να μην ανταγωνίζεται τον εαυτό του. Ο πράκτορας προσπαθεί αρχικά με τη βοήθεια ιστορικού να προβλέψει πόσες εντυπώσεις θα μπορέσει να αγοράσει στις δημοπρασίες εντυπώσεων και ανάλογα με τον αριθμό τους κρίνει αν μια εκστρατεία είναι εφικτή ή όχι. Σε περίπτωση που είναι εφικτή την ελέγχει με βάση τις εκστρατείες που έχει ήδη. Ελέγχει επίσης τα τμήματα της αγοράς που συμπίπτουν και αν έχει μεγάλη επικάλυψη την απορρίπτει.

Στρατηγική δημοπρασιών εντυπώσεων:

Στις δημοπρασίες εντυπώσεων ο Agent00 χρησιμοποιεί τις εξής δομές:

- **Users Class:** Κρατάει αποθηκευμένα στατιστικά για τον πληθυσμό και τις προτιμήσεις των χρηστών και τα χρησιμοποιεί για να προβλέψει γρήγορα τις εν δυνάμει εντυπώσεις για ένα τμήμα της αγοράς
- **Load Balancer:** Είναι ένα ραβδοειδές (bar-like) μοντέλο που κρατά πληροφορίες για όλες τις τρέχουσες εκστρατείες. Συγκεκριμένα κρατάει τα χαρακτηριστικά της εκστρατείας (διάρκεια, απαιτούμενες εντυπώσεις κλπ.), τον προϋπολογισμό και τον πράκτορα που την κέρδισε.
- **Neuro:** Ένα σύνολο από δισδιάστατους πίνακες για κάθε πιθανό τμήμα της αγοράς (6 για την ηλικία, 4 για το εισόδημα, 2 για το φύλλο κλπ.). Οι γραμμές και οι στήλες αντιπροσωπεύουν διαφορετικούς αντιπάλους, και τα κελιά ορίζουν τη συμπεριφορά του ενός αντιπάλου προς τον άλλον σε ένα τμήμα της αγοράς.
- **Bid bundle calculator:** Χρησιμοποιεί την πληροφορία από τις τρεις παραπάνω δομές για να υπολογίσει τα πονταρίσματα για τις εντυπώσεις.

- Querying mechanism: Μηχανισμός παρόμοιος με την SQL για γρήγορη ανάκτηση δεδομένων.

2.1.2 Ο πράκτορας Agent2

Η υλοποίηση του πράκτορα δίνει έμφαση σε διαφορετικές στρατηγικές, ανάλογα με την κατάσταση στην οποία βρισκόταν στο κοντινό παρελθόν, καθεμία από τις οποίες εκμεταλλεύεται διαφορετικές παραμέτρους του παιχνιδιού. Οι περισσότερες στρατηγικές που χρησιμοποιεί είναι προσαρμοστικές και τροποποιούν την συμπεριφορά τους ανάλογα με τη μέρα, την εκστρατεία, το παιχνίδι, την απόδοση του πράκτορα και των αντιπάλων του. Η αρχή του παιχνιδιού προσεγγίζεται επιθετικά, ενώ επίσης οι στρατηγικές του προλαμβάνουν και για απρόβλεπτα γεγονότα. Τέλος χρησιμοποιείται μηχανισμός που κάνει τις δυνατές στρατηγικές να επιβιώνουν και τις αδύναμες να μην χρησιμοποιούνται. [5]

Στρατηγική δημοπρασιών συμβολαίων:

- History winnings: Ο πράκτορας αποθηκεύει σε βάση δεδομένων πονταρίσματα από εκστρατείες που είχαν κερδηθεί σε παλαιότερα παιχνίδια και τις χρησιμοποιεί για να εξάγει το καλύτερο ποντάρισμα για κάθε τμήμα της αγοράς.
- Random luck: Ο πράκτορας προσπαθεί να εκμεταλλευτεί τον παράγοντα τυχαιότητας στις δημοπρασίες συμβολαίου, ποντάροντας το μέγιστο επιτρεπτό και ελπίζοντας να πάρει μια εκστρατεία με τεράστιο προϋπολογισμό.
- Aggressive (μέρες 5-7): Ποντάρεται το χαμηλότερο επιτρεπτό όριο ώστε να αυξηθεί η πιθανότητα λήψης εκστρατείας.

Στρατηγική δημοπρασιών εντυπώσεων:

- Champion Strategy: Η στρατηγική αυτή βασίζεται σε τρεις τυχαίες μεταβλητές: τις μέχρι τώρα αγορασμένες εντυπώσεις για κάθε εκστρατεία, το επίπεδο επιτυχίας της προηγούμενης ημέρας και το συνολικό επίπεδο επιτυχίας. Αυτές οι τρεις μεταβλητές αντιστοιχίζονται μέσω ενός πίνακα σε ένα συντελεστή που πολλαπλασιάζεται με κάποιες ευριστικές παραμέτρους και με τον λόγο προϋπολογισμού προς απαιτούμενες εντυπώσεις, για να δώσει το τελικό ποντάρισμα.
- First Campaign Strategy: Εδώ τα πονταρίσματα είναι υψηλότερα, αφού όλοι οι πράκτορες έχουν τουλάχιστον μια εκστρατεία και ο ανταγωνισμός για εντυπώσεις είναι υψηλός.

- **Unknown User Classification:** Χρησιμοποιείται μόνο για εντυπώσεις που δε φανερώνονται από το UCS και μόνο για τις σελίδες που είναι πιο πιθανό να βρει κανείς τους ζητούμενους χρήστες.

Στρατηγική UCS:

Γίνεται προσπάθεια να διατηρηθεί ένα επίπεδο ίσο με 0.8 έτσι ώστε να φανερώνονται τα χαρακτηριστικά των περισσότερων χρηστών αλλά και να μην γίνεται πόλεμος τιμών. Το ποντάρισμα γίνεται με τον μέσο όρο των διαμέσων στα προηγούμενα παιχνίδια. Αν ο πράκτορας μείνει πάνω από 5 ημέρες με χαμηλό UCS το ποντάρισμα αναπροσαρμόζεται.

2.1.3 Ο πράκτορας BCM (tau)

Η στρατηγική του BCM βασίζεται στη διατήρηση υψηλού βαθμού ποιότητας, ο οποίος εκτιμάται ότι θα φέρει περισσότερα κέρδη μακροπρόθεσμα. [9]

“Οι πρώτες 5 ημέρες”: Οι μέρες αυτές θεωρούνται επικίνδυνες από τον πράκτορα που αναμένει ότι θα έχει απώλειες τις μέρες αυτές, τις οποίες θα ανακτήσει χάρη στον υψηλότερο βαθμό ποιότητας σε σχέση με τους αντιπάλους του. Τα πονταρίσματα για εντυπώσεις είναι επίσης υψηλά για να διατηρηθεί αυτό το προβάδισμα.

Αμέσως μετά τη μέρα 5: Τα πονταρίσματα για εκστρατείες είναι χαμηλά έτσι ώστε ο πράκτορας να πάρει εύκολα εκστρατείες και να ανεβάσει γρήγορα τον βαθμό ποιότητας.

“Το μέσο του παιχνιδιού”: Αν όλα έχουν πάει καλά ο πράκτορας θα μπορεί να παίρνει εκστρατείες με μεγαλύτερους προϋπολογισμούς από ότι αρχικά. Επίσης το ποντάρισμα εντυπώσεων αυξάνεται ανάλογα με τον κορεσμό του τμήματος της αγοράς.

“Το τέλος του παιχνιδιού”: Ο πράκτορας συνεχίζει να χρησιμοποιεί τον βαθμό ποιότητας για να αυξήσει τα κέρδη του. Διαπιστώνεται ότι ένα μεγάλο μέρος των εσόδων του πράκτορα ξοδεύεται τελικά σε εντυπώσεις λόγω της “άπληστης” στρατηγικής του.

2.1.4 Ο πράκτορας giza

Ο πράκτορας αυτός χρησιμοποιεί πολλές στρατηγικές ανάλογα με την κατάσταση του παιχνιδιού και τις συμπεριφορές των αντιπάλων του.

Συγκεκριμένα χωρίζει το παιχνίδι σε τρία μέρη, την αρχή (7 πρώτες μέρες), τη μέση (8-45) και το τέλος (46-60). Όπως και ο BCM, ο giza ξεκινά επιθετικά ενώ προτιμά εκστρατείες που τελειώνουν μετά τις 5 πρώτες μέρες για να αποφύγει τα κορεσμένα τμήματα της αγοράς. Στο μέσο στόχος είναι η αποτελεσματικότητα όσον αφορά το κόστος και η διατήρηση ενός υψηλού βαθμού ποιότητας ο οποίος θα εξαργυρωθεί στο τέλος του παιχνιδιού. Στο τέλος του παιχνιδιού που ο συνωστισμός στα τμήματα της αγοράς είναι μικρότερος στόχος είναι η μεγιστοποίηση των εσόδων. Επίσης, εφόσον λίγοι πράκτορες επιβιώνουν μέχρι τότε, η πιθανότητα να είναι τυχερός και να κερδίσει μια εκστρατεία με μέγιστο προϋπολογισμό αυξάνεται, άρα κάνει μέγιστα πονταρίσματα στις δημοπρασίες συμβολαίου. Τέλος δεν ποντάρει σε εκστρατείες με μικρό budget που εκτιμά ότι θα είναι δύσκολο να τις ολοκληρώσει. [10]

Στρατηγικές δημοπρασιών συμβολαίων: Σύμφωνα με τα προηγούμενα χρησιμοποιούνται τρεις στρατηγικές. Η Cost Efficient Bid Strategy που εκτιμά τον προϋπολογισμό που θα έπρεπε να έχει μια υπνήφια εκστρατεία, έτσι ώστε να αποφέρει κέρδος οριακά. Οι Minimum και Maximum Bid στρατηγικές.

Στρατηγικές δημοπρασιών εντυπώσεων:

- Cost Efficient Bid Bundle Strategy: Τα πονταρίσματα μιας εκστρατείας ξεκινούν χαμηλά στις πρώτες μέρες της και αυξάνονται σταδιακά μόνο αν υπάρξει δυσκολία στην συμπλήρωσή της, μέχρι να φτάσουν ένα μέγιστο όριο.
- Aggressive Bid Bundle Strategy: Ίδια με την Cost Efficient αλλά χωρίς ανώτατο όριο.

Στρατηγικές UCS:

- Recommended Price: Υπολογίζεται η μέση τιμή των UCS πονταρισμάτων μέχρι την συγκεκριμένη μέρα και προστίθεται η διαφορά μεταξύ του επιθυμητού UCS και του προβλεπόμενου.
- Price from regression: Γίνεται γραμμική πρόβλεψη του πονταρίσματος για ένα επίπεδο-στόχο με βάση τη μέρα και τον αριθμό του παιχνιδιού.

2.1.5 Ο πράκτορας ANL

Η κύρια ιδέα πίσω από τον πράκτορα ANL [4], που ήταν και ο νικητής του TAC AdX 14', είναι η διατύπωση ενός προβλήματος μεγιστοποίησης. Συγκεκριμένα χρησιμοποιώντας δύο δείκτες, τον PI που εκφράζει το εκτιμώμενο κόστος εντύπωσης και τον CI που εκφράζει πόσο πολύ θέλει ο πράκτορας να πάρει μια νέα εκστρατεία, διατυπώνεται το εξής πρόβλημα:

$$E[U_i] = b_i \cdot q(\eta_i) - PI \cdot \eta_i \cdot R_i \quad (1)$$

Όπου U_i το κέρδος από μια εκστρατεία, b_i ο προϋπολογισμός-ποντάρισμα, η_i ο βαθμός ολοκλήρωσης ERR, $q(\eta_i)$ ο βαθμός ποιότητας του ERR, και R_i οι απαιτούμενες εντυπώσεις μιας εκστρατείας.

Με περιορισμούς:

$$\frac{R_i}{Q} < b_i < \frac{R_i \cdot Q}{10} \quad (2)$$

$$b_i \leq PI \cdot CI \cdot R_i \quad (3)$$

$$0.9 < \eta_i < 1.3 \quad (4)$$

Στρατηγικές δημοπρασιών συμβολαίων:

Στις δημοπρασίες συμβολαίων χρησιμοποιούνται τρεις στρατηγικές. Η κύρια είναι εκείνη που αντιστοιχεί στη μέγιστη τιμή του παραπάνω προβλήματος. Αν ο πράκτορας έχει πολύ χαμηλό Q (< 0.8) τότε προσπαθεί απεγνωσμένα να κερδίσει κάποιο συμβόλαιο ποντάροντας το ελάχιστο δυνατό, ενώ όταν μια εκστρατεία θεωρηθεί αδύνατη να πραγματοποιηθεί ποντάρει το μέγιστο. Η επιλογή ανάμεσα στις παραπάνω στρατηγικές γίνεται με βάση το Competing Index (CI) το οποίο μετβάλλεται ανάλογα με το αν ο πράκτορας κερδίζει συμβόλαια ή όχι.

Στρατηγικές δημοπρασιών εντυπώσεων:

Στις δημοπρασίες εντυπώσεων ο ANL προσπαθεί να κρατήσει υψηλό τον βαθμό ποιότητας Q , θεωρώντας την ενδεχόμενη έλλειψη κέρδους προσωρινή. Επίσης αυξάνει τα πονταρίσματά του όταν η τελευταία μέρα της εκστρατείας πλησιάζει για να διατηρήσει υψηλό Q .

Στρατηγική UCS:

Για να ποντάρει για το UCS ο πράκτορας υπολογίζει πόσα έσοδα θα έχανε αν το επίπεδό του έπεφτε στο αμέσως επόμενο ($\times 0.9$) και καταλήγει στην παρακάτω ανισότητα που άμα ισχύει αυξάνει το ποντάρισμά του:

$$\frac{r_0}{\beta} \geq \frac{20}{3} \frac{1+G_{UCS}}{E[p]} \quad (5)$$

Όπου β το τρέχον επίπεδο και G_{UCS} το βήμα αύξησης.

2.2 Ιδέες και μέθοδοι από πράκτορες άλλων διαγωνισμών

Ένα θέμα που είναι κοινό μεταξύ των διαγωνισμών TAC και για το οποίο θα μπορούσαμε να μελετήσουμε ιδέες από πράκτορες άλλων διαγωνισμών είναι η πρόβλεψη της τιμής μίας δημοπρασίας. Παρακάτω θα δούμε ότι πληροφορίες όπως η οι τιμές σε προηγούμενα παιχνίδια, οι προτιμήσεις των πελατών και η συμπεριφορά των αντιπάλων μπορούν να χρησιμοποιηθούν αποτελεσματικά για αυτήν την πρόβλεψη.

2.2.1 Η πρόβλεψη τιμής στον classic TAC

Σύμφωνα με τους Wellman et. al. (2004) [16] στον 2ο διαγωνισμό TAC classic οι μέθοδοι που χρησιμοποιήθηκαν για τον προσδιορισμό των τιμών μπορούν να χωριστούν σε τρεις κατηγορίες:

1. Προβλέψεις εξαγωγής μέσου όρου από ιστορικό:

Αυτή ήταν η πιο δημοφιλής μέθοδος του διαγωνισμού. Η εξαγωγή μέσου όρου έγινε είτε από ένα μεγάλο ιστορικό παλιών παιχνιδιών, είτε ως κινούμενος μέσος όρος λίγων πρόσφατων παιχνιδιών. Μία άλλη προσέγγιση ήταν η οργάνωση των δεδομένων σε κλάσεις διαφορετικής ανταγωνιστικότητας και η εξαγωγή μέσου όρου για κάθε κλάση. Τέτοια μέθοδο χρησιμοποίησε και ο πράκτορας WhiteBear. [11]

2. Προβλέψεις με χρήση μηχανικής μάθησης:

Η προσέγγιση αυτή υποθέτει ότι διάφορα συγκεκριμένα στοιχεία του παιχνιδιού (π.χ. αρχικές τιμές πτήσεων, ώρες ξενοδοχείων, για τον classic TAC) συσχετίζονται με τις διακυμάνσεις της τιμής. Με τέτοια στοιχεία ο πράκτορας ATTAC [23] χρησιμοποίησε έναν αλγόριθμο βασισμένο στην τεχνική προώθησης (boosting) για την εκτίμηση της υπό συνθήκη κατανομής

της τιμής. Ο πράκτορας kanayaH χρησιμοποίησε νευρωνικά δίκτυα με είσοδο εκτός από τα παραπάνω στοιχεία και διαφορές τιμών ανάμεσα σε συνεχόμενες ημέρες

3. Προβλέψεις με ανταγωνιστική ανάλυση:

Ο πράκτορας Walverine [26] ήταν ο μόνος που χρησιμοποίησε την παραπάνω μέθοδο βασιζόμενος στην υπόθεση ότι το παιχνίδι μπορεί να μοντελοποιηθεί σαν μια ανταγωνιστική οικονομία. Έτσι προσπάθησε να υπολογίσει την τιμή ισορροπίας της αγοράς εκτιμώντας την ζήτηση των αντίπαλων πρακτόρων. (Λόγω της διαφοράς των χαρακτηριστικών του παιχνιδιού από αυτό της παρούσας εργασίας το πλήρες επιχειρήμα του δεν είναι ιδιαίτερα σχετικό και παραλείπεται)

Τελικά από τις παραπάνω μεθόδους αυτές των πρακτόρων Walverine και ATTAC κρίθηκαν ως οι καλύτερες.

2.2.2 Μεθοδολογίες πειραμάτων στον classic TAC

Ο πειραματικός έλεγχος καθώς και ο προσδιορισμός διαφόρων παραμέτρων ενός πράκτορα με βάση ελεγχόμενες προσομοιώσεις παίζουν καθοριστικό ρόλο κατά τον σχεδιασμό του. Παρακάτω παρατίθενται συστηματικές μέθοδοι διεξαγωγής πειραμάτων που αναπτύχθηκαν με αφορμή τον διαγωνισμό classic TAC:

- Οι D. Reeves, M.Wellman et al. [36] ερευνούν έναν μικρό χώρο στρατηγικών οι οποίες καθορίζονται από συγκεκριμένες παραμέτρους που εκφράζουν τις προτιμήσεις των πρακτόρων, καθώς και από τις πληροφορίες που είναι διαθέσιμες για τις δημοπρασίες. Χρησιμοποιώντας τη μέθοδο Monte Carlo υπολογίζουν την μήτρα των εκ των προτέρων (ex ante) αναμενόμενων κερδών για κάθε συνδυασμό στρατηγικών. Στη συνέχεια εξετάζουν τα αποτελέσματα τριών μεθόδων (εξελικτική, GAMBIT, Amoeba) για να υπολογίσουν πιθανά σημεία ισορροπίας Nash του παιχνιδιού.
- Οι I. Vetsikas και B.Selman [11] εξέτασαν συνδυασμούς από τρεις διαφορετικές παραμέτρους. Η λογική των πειραμάτων είναι η εύρεση της βέλτιστης τιμής μιας παραμέτρου κρατώντας τις δύο άλλες σταθερές. Στη συνέχεια χρησιμοποιώντας τη βέλτιστη πρώτη παράμετρο και σταθερή την τρίτη υπολογίζεται η βέλτιστη δεύτερη παράμετρος. Η παραπάνω μέθοδος συνεχίζεται κυκλικά μέχρι να βρεθεί το καλύτερο σετ παραμέτρων. Παρότι η διαδικασία ενδέχεται αν μη συγκλίνει στην πράξη έδωσε μια βέλτιστη στρατηγική.

3

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό γίνεται μια αναλυτική περιγραφή σε θεωρητικό επίπεδο των τεχνικών που δοκιμάστηκαν για την υλοποίηση του πράκτορα Hermes, αλλά και των χαρακτηριστικών του παιχνιδιού που τις επηρεάζουν.

3.1 Δημοπρασίες δεύτερης τιμής και μηχανισμοί Vickrey-

Clarke-Groves

Στην ενότητα αυτή γίνεται μια σύντομη επισκόπηση της θεωρίας πίσω από τις δημοπρασίες του παιχνιδιού οι οποίες εμπίπτουν στην ευρύτερη κατηγορία των μηχανισμών Vickrey-Clarke-Groves (VCG) [30].

Ένας τέτοιος μηχανισμός (X, p) έχει τη μορφή:

$$X(\hat{v}) \in \underset{x}{\operatorname{argmax}} \sum_i \hat{v}_i(x) \quad (1)$$

$$p_i(\hat{v}) = \max_x \left(\sum_{i \neq j} \hat{v}_j(x) - \sum_{i \neq j} \hat{v}_j(X(\hat{v})) \right) \quad (2)$$

Όπου \hat{v} το όφελος (utility) κάθε συμμετέχοντα στη δημοπρασία, X το κοινωνικό όφελος και p η τιμή που ποντάρει ένας παίκτης.

Η σημασία της δεύτερης εξίσωσης είναι ότι ο παίκτης πρέπει να πληρώσει τη διαφορά ανάμεσα στο συνολικό κόστος για τους υπόλοιπους παίκτες χωρίς τη συμμετοχή του μείον το συνολικό κόστος όταν συμμετέχει. Με λίγα λόγια πληρώνει το “κοινωνικό κόστος” του.

Έτσι σε περίπτωση που ο παίκτης i θέλει να μεγιστοποιήσει το προσωπικό του όφελος:

$$\max \left(v_i \left(X \left(\hat{v}_i, \hat{v}_{-i} \right) \right) - p_i \left(X \left(\hat{v}_i, \hat{v}_{-i} \right) \right) \right) \quad (3)$$

αντικαθιστώντας το p για έναν μηχανισμό VCG και αφαιρώντας τους όρους που δεν εξαρτώνται από τη στρατηγική που επιλέγει ο i έχουμε:

$$\max \left(v_i(x) + \sum_{j \neq i} \hat{v}_j(x) \right) \quad (4)$$

Κοιτώντας την παραπάνω και την πρώτη εξίσωση του VCG παρατηρούμε ότι ο μηχανισμός θα διαλέξει ένα x τέτοιο ώστε $v_i(x) = \hat{v}_i(x)$. Δηλαδή η βέλτιστη στρατηγική είναι το ποντάρισμα της πραγματικής του αξίας.

Οι δημοπρασίες δεύτερης τιμής με κρυφό ποντάρισμα είναι μια ειδική περίπτωση του παραπάνω μηχανισμού. Συγκεκριμένα:

-Σε περίπτωση που ο i συμμετάσχει στη δημοπρασία και δεν κερδίσει θα πληρώσει 0 και θα πάρει 0 όφελος.

-Σε περίπτωση που κερδίσει θα πάρει την αξία του και θα στερήσει από τον δεύτερο στη δημοπρασία το όφελος που θα έπαινε, άρα σύμφωνα με τον μηχανισμό VCG θα πρέπει να πληρώσει την τιμή του δεύτερου.

Άρα στις δημοπρασίες δεύτερης τιμής η βέλτιστη στρατηγική είναι πάλι το ποντάρισμα της πραγματικής ιδιωτικής αξίας του παίκτη. Το παραπάνω είναι και διαισθητικά λογικό αφού άμα ο παίκτης ποντάρει $x > v$ και κάποιος άλλος παίκτης ποντάρει $v < y < x$ τότε ο πρώτος θα πληρώσει παραπάνω από την αξία του. Αν ποντάρει $x < v$ ρισκάρει να χάσει την αξία του άμα κάποιος άλλος ποντάρει $x < y < v$.

Ο λόγος που χρησιμοποιήθηκαν τέτοιες δημοπρασίες για τις εκστρατείες και τις εντυπώσεις είναι προφανώς οι πράκτορες να προσπαθήσουν να εκτιμήσουν την πραγματική αξία που έχουν για αυτούς οι εκστρατείες/εντυπώσεις. Βέβαια περιβάλλον του παιχνιδιού adx όπου οι δημοπρασίες αλληλοεξαρτώνται καθώς και ο παράγοντας τυχαιότητας που έχει προστεθεί στη δημοπρασία συμβολαίων κάνουν το ποντάρισμα πιο περίπλοκο.

3.2 Γενικευμένες δημοπρασίες δεύτερης τιμής

Οι γενικευμένες δημοπρασίες δεύτερης τιμής χρησιμοποιούνται για την υπηρεσία UCS. Επίσης χρησιμοποιούνται ευρέως στην πραγματικότητα κυρίως σε δημοπρασίες διαφημίσεων για λέξεις κλειδιά σε μηχανές αναζήτησης π.χ. Google AdWords. [3]

Στις δημοπρασίες αυτές υπάρχουν k παίκτες που ανταγωνίζονται για k θέσεις όπου κάθε θέση συνδέεται με μια πιθανότητα (π.χ. πιθανότητα να χρησιμοποιήσει το keyword ένας χρήστης, ή η πιθανότητα φανέρωσης των χαρακτηριστικών ενός χρήστη στο UCS). Διατάσσοντας τα ποντάρισμα και τις πιθανότητες από τον μεγαλύτερο αριθμό στον μικρότερο, κάθε παίκτης παίρνει την αντίστοιχη με το ποντάρισμά του θέση, πληρώνοντας τιμή ίση με το αμέσως μικρότερο ποντάρισμα.

Σε αυτές τις δημοπρασίες υπάρχουν περιπτώσεις όπου το ποντάρισμα της αληθινής αξίας από όλους τους παίκτες δεν οδηγεί σε ισορροπία Nash. Παρόλα αυτά έχει αποδειχθεί ότι υπάρχει ισορροπία που να μεγιστοποιεί το κοινωνικό όφελος με αξία τουλάχιστον ίση της συνολικής αληθινής αξίας.

3.3 Συνδυαστικοί ταξινομητές και η μέθοδος boosting

Η μέθοδος boosting που ανήκει στην κατηγορία των συνδυαστικών ταξινομητών, χρησιμοποιήθηκε για να κριθεί αν μια εκστρατεία που πρόκειται να αναλάβει ο πράκτορας θα του αποφέρει κάποιο πλεονέκτημα ή όχι. Η βασική ιδέα πίσω από τους συνδυαστικούς ταξινομητές είναι η χρήση πολλών απλών ταξινομητών για την εξαγωγή ενός τελικού συμπεράσματος που θα είναι ένας σταθμισμένος μέσος όρος των συμπερασμάτων από κάθε ταξινομητή.

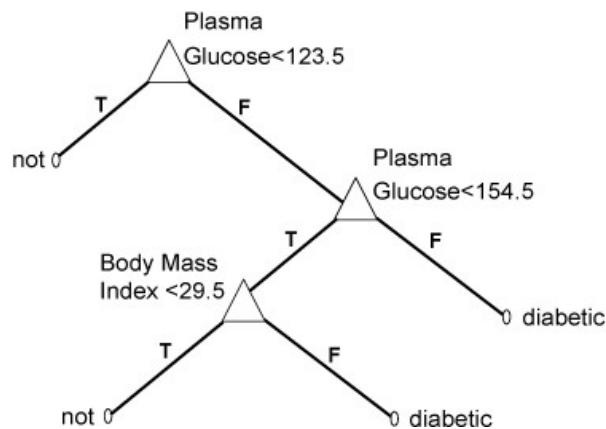
Η μέθοδος boosting αποτελεί μια τέτοιου τύπου μέθοδο, όπου κάθε ταξινομητής επικεντρώνεται στο να “καλύψει” το σφάλμα του προηγούμενου. Η εκπαίδευση των ταξινομητών γίνεται έτσι σειριακά. Χρησιμοποιώντας στο τέλος τα αποτελέσματα όλων αυτών των ταξινομητών (π.χ. με σταθμισμένο μέσο όρο, με πλειοψηφία κλπ.) μετατρέπουμε ένα σύνολο πολλών “αδύναμων” ταξινομητών σε έναν “δυνατό”. [24]

3.3.1 Ταξινόμηση με δέντρα αποφάσεων

Ένα τυπικό παράδειγμα απλού ταξινομητή που χρησιμοποιείται στη μέθοδο boosting είναι τα δέντρα αποφάσεων. [12] Τα δέντρα αποφάσεων είναι μια δενδρική δομή κόμβων σύγκρισης. Ένα σημείο εισόδου ξεκινά από τη ρίζα του δέντρου και κατεβαίνει προς τα κάτω ανάλογα με

την τιμή μιας συγκεκριμένης διάστασής του (π.χ. με χρήση κατωφλιού). Αυτό γίνεται μέχρι να φτάσει σε ένα φύλλο (αφότου εξετασθούν όλες οι συντεταγμένες του), το οποίο αντιπροσωπεύει μία από τις κλάσεις ταξινόμησης, στην οποία και το κατατάσσει το δέντρο.

Παρακάτω φαίνεται ένα απλό δέντρο που αποφασίζει αν κάποιος είναι διαβητικός ή όχι ανάλογα με το ποσοστό γλυκόζης στο πλάσμα και τον δείκτη μάζας σώματος.



3.1 Απλό παράδειγμα δέντρου απόφασης

Πηγή: resenv.media.mit.edu

Η διαδικασία μάθησης σε ένα τέτοιο δέντρο αφορά στην σωστή επιλογή των τιμών των κατωφλιών. Συνήθως χρησιμοποιείται εξαντλητική αναζήτηση και επιλέγεται η τιμή που ελαχιστοποιεί κάποια συνάρτηση ολικού σφάλματος. Εδώ πρέπει να σημειωθεί ότι η εξαντλητική αναζήτηση δεν πρέπει να εξετάσει όλα τα σημεία του χώρου όπου ανήκει η είσοδος. Αντίθετα για κάθε ζεύγος σημείων της εισόδου πρέπει να εξεταστεί μόνο ένα από τα

ενδιάμεσα σημεία, δηλαδή συνολικά $\binom{n}{2}$ κατώφλια.

Η συνάρτηση συνολικού κόστους που χρησιμοποιείται συχνότερα είναι το κέρδος πληροφορίας για μια διαμέριση \tilde{S} του S:

$$IG(\tilde{S}) = \sum_i |S_i| \cdot (H(p_S) - H(p_{S_i})) \quad (5)$$

Όπου:

$$H(p_S) = - \sum_y p_S(y) \log_2(p_S(y)) \quad (6)$$

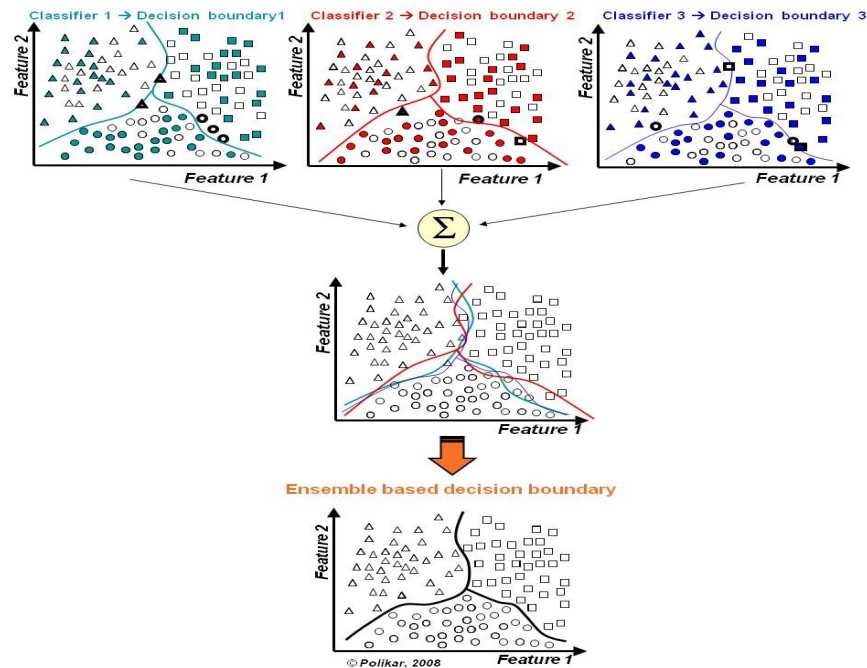
η εντροπία του συνόλου S με βάση την κατανομή της εξόδου y.

Για να μειωθεί η πολυπλοκότητα της παραπάνω μεθόδου πολλές φορές οριοθετείται το βάθος μέχρι το οποίο μπορεί να φτάσει το δέντρο. Επίσης για να είναι η πρόβλεψη πιο γρήγορη για τα νέα δεδομένα γίνεται πολλές φορές κλάδεμα των κόμβων που συνεισφέρουν λιγότερη νέα πληροφορία. Αυτό γίνεται εξετάζοντας αν η διαφορά ανάμεσα στο IG ενός γονέα και αυτό των παιδιών του είναι μεγαλύτερη από ένα ορισμένο κατώφλι. [12]

3.3.2 Bootstrap Aggregation και bagging

Η Bootstrap Aggregation (ή bagging) είναι μια στατιστική τεχνική που χρησιμοποιείται για να αποφεύγεται η υπερπροσαρμογή ενός μοντέλου στα δεδομένα (over-fitting).

Συγκεκριμένα η μέθοδος αυτή δημιουργεί διαφορετικά υποσύνολα πλήθους $N' < N$, μικρότερου από το μέγεθος των συνόλων των δεδομένων εκπαίδευσης επιτρέποντας την επανάληψη κάποιων σημείων ανάμεσά τους (δειγματοληψία με επανατοποθέτηση). Σε κάθε τέτοιο υποσύνολο εκπαιδεύεται ένας -όταν το κάνουμε για λόγους αποφυγής του over-fitting, ο ίδιος-ταξινομητής και στο τέλος βγαίνει συγκεντρωτικά ένα συμπέρασμα με βάση όλους τους ταξινομητές. Έτσι ο συνδυασμός τους θα έχει καλύτερο αποτέλεσμα στα δεδομένα από έναν απλό ταξινομητή εκπαιδευμένο σε όλο το σύνολο εκπαίδευσης. Παρακάτω φαίνεται ένα απλό παράδειγμα όπου η έξοδος είναι ο μέσος όρος τριών ταξινομητών εκπαιδευμένων σε διαφορετικά υποσύνολα του συνόλου εκπαίδευσης.[24]



3.2 Βελτίωση της επίδοσης ενός ταξινομητή με χρήση bootstrap aggregation

3.3.3 Boosting και η μέθοδος AdaBoost

Όπως αναφέρθηκε η μέθοδος boosting βασίζεται στην εν σειρά εκπαίδευση απλών ταξινομητών, όπου κάθε ταξινομητής προσπαθεί να διορθώσει τα λάθη που έκανε ο προηγούμενος.

Για να υλοποιηθεί αυτό ο βασικός ταξινομητής που θα χρησιμοποιήσουμε θα πρέπει να εκπαιδεύεται στοχεύοντας στην ελαχιστοποίηση ενός σταθμισμένου σφάλματος. Έτσι σε κάθε νέο ταξινομητή τα σημεία όπου έκανε λάθος ο προηγούμενος (ή περισσότεροι προηγούμενοι) θα έχουν μεγαλύτερο βάρος στη συνάρτηση ολικού σφάλματός του, ενώ εκείνα όπου η πρόβλεψη του προηγούμενου ήταν σωστή θα έχουν μικρότερο. Τελικά το αποτέλεσμα είναι μια μορφή μέσου όρου (κανονικός, σταθμισμένος, πλειοψηφία κλπ.) από τα αποτελέσματα των εν σειρά ταξινομητών.

Ένας από τους πιο δημοφιλείς και καλούς αλγόριθμους που χρησιμοποιεί την παραπάνω μέθοδο είναι ο AdaBoost. Ο αλγόριθμος αυτός ελαχιστοποιεί την παρακάτω συνάρτηση ολικού σφάλματος για N παρατηρήσεις, T ταξινομητές και K κλάσεις ταξινόμησης:

$$\varepsilon_t = \frac{1}{2} \sum_{n=1}^N \sum_{k \neq y_n} d_{n,k}^{(t)} \cdot \left(1 - h_t(x_n, y_n) - h_t(x_n, k) \right), \quad (7)$$

όπου $h_t(x_n, k) \in [0, 1]$ η εμπιστοσύνη που έχει ο ταξινομητής στην πρόβλεψή του, και $d_{n,k}^{(t)}$ το βάρος του σημείου εισόδο τη στιγμή t για την κλάση k .

Για κάθε νέο σημείο δεδομένων η πρόβλεψη της μεθόδου είναι:

$$f(x) = \sum_{t=1}^T a_t \cdot f_t(x) = \sum_{t=1}^T a_t \cdot h(x_n, k) \quad (8)$$

Με $a_t = \frac{1}{2} \cdot \log \frac{1 - \varepsilon_t}{\varepsilon_t}$ (9) τα βάρη των ταξινομητών.

Τέλος η ανανέωση των βαρών γίνεται ως εξής (μετά ακολουθεί και κανονικοποίησή τους):

$$w_t = w_{t-1} \cdot e^{-a_{t-1} \cdot y_n \cdot f_{t-1}(x_n)} \quad (10)$$

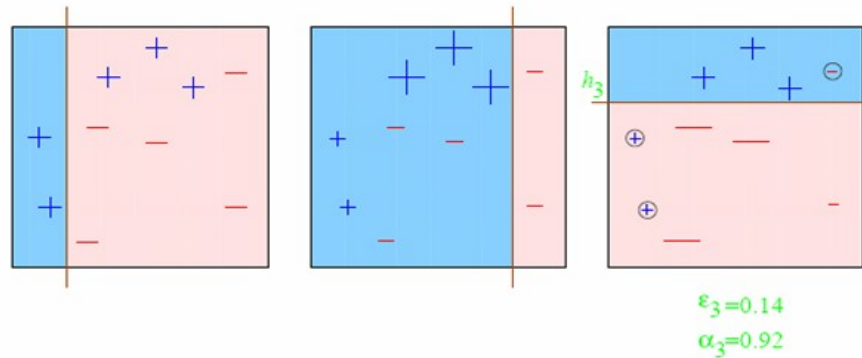
Και στοχεύει στην ελαχιστοποίηση της ολικής εκθετικής απώλειας:

$$C_{\text{exp}}(x,y) = \sum_{n=1}^N e^{-y_n f(x_n)} \quad (11)$$

Η παραπάνω συνάρτηση έχει τα πλεονεκτήματα ότι είναι κυρτή και ότι αποτελεί ένα άνω φράγμα για την συνάρτηση ολικού σφάλματος, γεγονός που διευκολύνουν πολύ την ελαχιστοποίησή της.

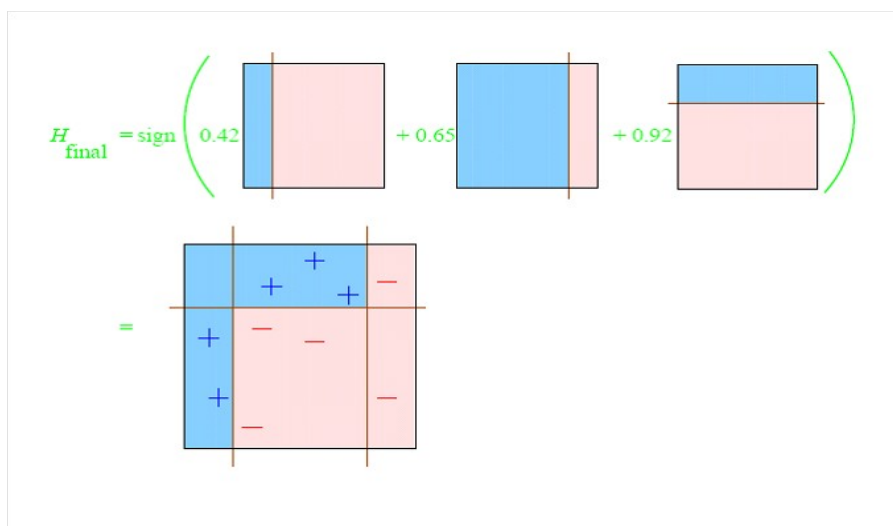
Στην απλούστερη περίπτωση που έχουμε δύο κλάσεις η συνάρτηση εμπιστοσύνης h παίρνει τιμές 0 ή 1.

Η μέθοδος φαίνεται ξεκάθαρα στο παρακάτω παράδειγμα όπου χρησιμοποιούνται decisions stumps δηλαδή δέντρα απόφασης με έναν μόνο κόμβο σύγκρισης:



3.3 Η διαδικασία σειριακής εκπαίδευσης των ταξινομητών στην AdaBoost

Όπου παρατηρούμε ότι η μεγέθυνση των "+" και των "-" σύμφωνα με τον κανόνα ανανέωσης βαρών, ανάλογα αν τα παραδείγματα είχαν κατηγοριοποιηθεί σωστά ή όχι από τον προηγούμενο ταξινομητή. Παρακάτω φαίνεται και η τελική εξαγωγή των αποτελεσμάτων η οποία οδηγεί σε μια σαφώς πιο πολύπλοκη διαμέριση η οποία θα ήταν αδύνατη για τους απλούς ταξινομητές.



3.4 Ο τελικός συμψηφισμός των "αδύναμων" ταξινομητών

3.4 Η μέθοδος απλής εκθετικής εξομάλυνσης για την εξαγωγή προβλέψεων

Κατά τη διάρκεια του παιχνιδιού ο πράκτορας καλείται να προβλέψει διάφορα μεγέθη έτσι ώστε να αποφασίσει πως θα κινηθεί. Πολλά από αυτά εξαρτώνται από τους ανταγωνιστές του αλλά και από τυχαίους παράγοντες γεγονός που καθιστά δύσκολη την πρόβλεψή τους με μηχανική μάθηση. Όπως θα παρουσιαστεί και στο κεφάλαιο της σχεδίασης, ο πράκτορας βλέπει τέτοια μεγέθη σαν χρονοσειρές και για αυτόν τον λόγο δανειστήκαμε την παραπάνω μέθοδο ως την πιο κατάλληλη.

Η μέθοδος απλής εκθετικής εξομάλυνσης (Simple Exponential Smoothing) χαρακτηρίζεται από τις παρακάτω εξισώσεις:

$$e_t = Y_t - F_t \quad (11)$$

$$S_t = S_{t-1} + \alpha \cdot e_t \quad (12)$$

$$F_{t+1} = S_t \quad (13)$$

Όπου e το σφάλμα, Y η πραγματική τιμή της χρονοσειράς, F η πρόβλεψη, και S το “σταθερό επίπεδο” γύρω από το οποίο υποθέτουμε ότι κινείται η χρονοσειρά. Βασική υπόθεση λοιπόν, που πρέπει να γίνει πριν χρησιμοποιήσουμε τη μέθοδο SES είναι ότι η χρονοσειρά μας διατηρεί τον μέσο όρο της περίπου σταθερό.

Πρακτικά η SES χρησιμοποιείται όταν θέλουμε να κάνουμε μόνο μια πρόβλεψη (βραχυπρόθεσμη). Επίσης είναι πολύ χρήσιμη όταν τα δεδομένα είναι πολύ θορυβώδη και παρουσιάζουν μεγάλη τυχαιότητα κάτι που όπως θα δούμε στη συνέχεια συμβαίνει στις προβλέψεις που θέλουμε να κάνουμε. [33]

Σημαντικό ρόλο στην παραπάνω μέθοδο παίζει η παράμετρος α η οποία αναγκάζει τη χρονοσειρά να ακολουθήσει την κατεύθυνση του σφάλματος της προηγούμενης πρόβλεψης. Σε πολύ θορυβώδη δεδομένα με σταθερό μέσο όρο το α πρέπει να παραμένει μικρό (<0.2) για να μην υπάρχει μεγάλη αντίδραση στο θόρυβο, ενώ σε περίπτωση που ο μέσος όρος αλλάζει επίπεδο τότε πρέπει να χρησιμοποιείται μεγαλύτερο α (π.χ. κοντά στο 0.5).

Πρέπει να σημειωθεί επίσης ότι η επιλογή μιας αρχικής πρόβλεψης επηρεάζει τις επόμενες προβλέψεις ιδιαίτερα όταν τα δεδομένα είναι λίγα. Αυτό φαίνεται αν απαλείψουμε το S και προσπαθήσουμε να εκφράσουμε αναδρομικά την καινούργια πρόβλεψη:

$$F_{t+1} = \alpha \cdot Y_t + (1-\alpha) \cdot Ft = \alpha Y_t + a(1-a)Y_{t-1} + a(1-a)^2 Y_{t-2} + \dots + a(1-a)^{t-1} Y_1 + (1-a)^t F_1$$

(14)

Συνήθως μια καλή επιλογή είναι ο μέσος όρος των ήδη υπαρχόντων δεδομένων ή ο υπολογισμός του σταθερού όρου από την εφαρμογή γραμμικής παλινδρόμησης.

3.5 Ταξινομητές Bayes

Στο παιχνίδι TAC AdX, κάθε πράκτορας καλείται να παίρνει αποφάσεις σε ένα περιβάλλον αβεβαιότητας, τόσο λόγω της αβεβαιότητας που προσομοιώνεται από το ίδιο το παιχνίδι όσο και λόγω της απρόβλεπτης πολλές φορές συμπεριφοράς των αντιπάλων. Για αυτούς τους λόγους οι ταξινομητές Bayes είναι ένα ιδιαίτερα χρήσιμο εργαλείο για την μοντελοποίηση προβλημάτων απόφασης που καλείται να λύσει ο πράκτορας.

Ο νόμος του Bayes φαίνεται χρήσιμος όταν έχω μία παρατήρηση x για την τυχαία μεταβλητή X και θέλω να υπολογίσω ποια είναι η πιθανότητα να βρίσκομαι στην κλάση c:

$$P\langle C=c|X=x \rangle = \frac{P[C=c]P\langle X=x|C=c \rangle}{P[X=x]} = \frac{P[C=c]P\langle X=x|C=c \rangle}{\sum_i P\langle X=x|C=c_i \rangle}$$

(15)

Όπου c_i όλες οι υπόλοιπες κλάσεις και $P[C=c]$ είναι η εκ των προτέρων πιθανότητα να βρισκόμαστε στην κλάση c.

Ένας ταξινομητής bayes έχει σαν στόχο να μας δώσει την κλάση c η οποία ελαχιστοποιεί την πιθανότητα η πρόβλεψή μας $\hat{c}(x)$ να είναι διαφορετική από την πραγματική κλάση στην οποία βρισκόμαστε όταν παρατηρούμε την x:

$$P[\hat{c} \neq c] = \sum_{c' \neq c} P\langle X=x|C=c' \rangle = 1 - P\langle X=x|C=c \rangle$$

(16)

Για να μπορέσουμε να αξιοποιήσουμε την μέθοδο αυτή, πρέπει να “μάθουμε” τις κατανομές των κλάσεων C καθώς και την κατανομή των παρατηρήσεων X. Ξαναγράφοντας τον νόμο του Bayes με την παρακάτω μορφή:

$$P\langle C=c|X=x\rangle = \frac{P[X=x, C=c]}{P[X=x]} \quad (17)$$

παρατηρούμε ότι αρκεί να μάθουμε από τα δεδομένα εκπαίδευσης την κοινή κατανομή των X και C και την κατανομή της X . Το παραπάνω όμως είναι ιδιαίτερα δύσκολο όταν τα στοιχεία εκπαίδευσης που επηρεάζουν την επιλογή της κλάσης είναι πάρα πολλά, αφού για n με m τιμές το καθένα στοιχεία χρειάζονται $O(m^{n+1})$ διαφορετικές περιπτώσεις.

Μια λύση σε αυτό το πρόβλημα, που αν και απλή λειτουργεί πολύ καλά στην πράξη είναι η μέθοδος Naive Bayes. Σε αυτήν τη μέθοδο υποθέτουμε ότι το πόσο κάθε συνιστώσα ενός διανύσματος εισόδου επηρεάζει την επιλογή της κλάσης είναι ανεξάρτητο από τις υπόλοιπες συνιστώσες, δηλαδή:

$$P\langle x|c\rangle = \prod_i P\langle x_i|c\rangle \quad (18)$$

Το παραπάνω κάνει προφανώς τον όγκο των υπολογισμών και των δεδομένων πολύ μικρότερο, όμως αποτυγχάνει στο να εντοπίσει τυχόν συσχετίσει μεταξύ των χαρακτηριστικών εκπαίδευσης. Παρόλο που το τελικό αποτέλεσμα δεν αποτελεί ένα ακριβές μοντέλο για την κατανομή των x , έχει συνήθως καλή προβλεπτική ικανότητα σαν ταξινομητής. [6]

4

Η σχεδίαση του

πράκτορα Hermes

Όπως ήδη έχει αναφερθεί ένας ευφυής πράκτορας στον διαγωνισμό TAC AdX πρέπει να υλοποιεί 3 βασικές στρατηγικές: στρατηγική για τις δημοπρασίες διαφημιστικών εκστρατειών (Campaigns), στρατηγική για την απόκτηση πληροφορίας σχετικής με την κατηγοριοποίηση των χρηστών, που πρόκειται να δουν μία διαφήμιση (User Classification Service) και μια στρατηγική για τις δημοπρασίες “εντυπώσεων”, δηλαδή της δυνατότητας προβολής μιας διαφήμισης σε μια ιστοσελίδα (Impression Bidding).

Παρακάτω περιγράφονται οι τρεις αυτές στρατηγικές για τον πράκτορα Hermes.

4.1 Στρατηγική για τις δημοπρασίες συμβολαίων

Στις δημοπρασίες διαφημίσεων, νικητής θεωρείται ο πράκτορας που προσφέρεται να εκτελέσει μια εκστρατεία με τον χαμηλότερο προϋπολογισμό, ο οποίος αναλαμβάνει και την εκστρατεία, με προϋπολογισμό ίσο με τον δεύτερο μικρότερο που έχει προσφερθεί (Second Price Sealed Bid Auction). Επίσης υπάρχει μια πιθανότητα $P_{randBid}=0.36$ η εκστρατεία να ανατεθεί τυχαία σε κάποιον από τους υπόλοιπους πράκτορες. Τέλος υπάρχουν άνω και κάτω φράγματα στις επιτρεπόμενες προσφορές $RCampaignMax=0.001 \cdot C_R \cdot Q_r$ και $RCampaignMin=0.0001 \cdot C_R / Q_r$ αντίστοιχα.

Με αυτά τα δεδομένα και παρατηρώντας πειραματικά ότι σχεδόν πάντα υπάρχει κάποιος πράκτορας που θα προσφέρει τον χαμηλότερο επιτρεπόμενο προϋπολογισμό, συμπεραίνουμε ότι ο πράκτορας συμφέρει (κατά προσέγγιση) να έχει μία από τις εξής τρεις συμπεριφορές: είτε να προσφέρει τον μέγιστο είτε πολύ χαμηλό (κοντά στον ελάχιστο επιτρεπόμενο) προϋπολογισμό, είτε να μην συμμετάσχει στη δημοπρασία.

4.1.1 Φιλτράρισμα των ακατάλληλων εκστρατειών

Για να διατηρήσει ο πράκτορας την ανταγωνιστικότητά του στο παιχνίδι AdX, είναι σημαντικό να μπορεί να καθορίσει εκ των προτέρων ποια συμβόλαια θα είναι ζημιογόνα για αυτόν. Ένα συμβόλαιο μπορεί να είναι ζημιογόνο για έναν ή περισσότερους παρακάτω λόγους:

- Να ζητάει την αγορά περισσότερων εντυπώσεων από αυτές που διατίθενται κατά τη διάρκεια της εκστρατείας.

- Να αναφέρεται σε ένα τμήμα της αγοράς το οποίο να είναι κορεσμένο με αποτέλεσμα κάθε εντύπωση να έχει υπερβολικά υψηλό κόστος.

- Να αναφέρεται σε ένα τμήμα της αγοράς στο οποίο ο πράκτορας έχει ήδη μία ή περισσότερες εκστρατείες με αποτέλεσμα ο πράκτορας να ανταγωνίζεται τον εαυτό του στις δημοπρασίες εντυπώσεων.

Στόχος λοιπόν του πράκτορα Hermes είναι να εντοπίσει την ύπαρξη κάποιας/ων από τις παραπάνω συνθήκες. Αυτό υλοποιείται με την εκ των προτέρων πρόβλεψη της αναμενόμενης τιμής μίας εντύπωσης, στην οποία ενσωματώνονται και οι δύο τελευταίοι παράγοντες, ενώ ο πρώτος είναι αρκετά εύκολο να ελεγχθεί με μια εντολή if.

4.1.1.1 Πρόβλεψη τιμής εντυπώσεων πριν τη δημοπρασία

συμβολαίων

Έστω M το τμήμα της αγοράς στο οποίο αναφέρεται το συμβόλαιο C_i που ξεκινά τη μέρα i και $S_{M,i}$ το σύνολο των εκστρατειών σε εξέλιξη του πράκτορα. Έστω επίσης $C_{R,M,i}$ οι απαιτούμενες εντυπώσεις από το M των εκστρατειών όλων των παικτών που είναι σε ακόμη σε εξέλιξη τη μέρα i , I_M οι συνολικές διαθέσιμες εντυπώσεις του M , και $B(s)$ προϋπολογισμός της εκστρατείας s .

Η πρόβλεψη της τιμής για μία εντύπωση γίνεται με τον παρακάτω τύπο:

$$p = \frac{\sum_M C_{R,M,i}}{I_M} \cdot \frac{\sum_{s \in S} B(s) + B(c)}{\sum_S C_{R,S,i}} \quad (1)$$

Παρατηρούμε λοιπόν ότι η τιμή αυξάνεται αν έχουμε εκστρατείες με μεγάλο προϋπολογισμό στο ίδιο τμήμα της αγοράς με την υποψήφια εκστρατεία, όπως επίσης και όταν υπάρχει μεγάλη συσσώρευση παικτών σε αυτό το τμήμα, όπου η συσσώρευση σε ένα τμήμα της αγοράς (market segment) θα αντιπροσωπεύεται από εδώ και πέρα από τον όρο :

$$\text{congestion}(M) = \frac{\sum_M C_{R,M,i}}{I_M} \quad (2)$$

4.1.2 Πότε μια εκστρατεία είναι ακατάλληλη

Μία εκστρατεία θεωρείται ακατάλληλη όταν δεν μπορεί να προσφέρει στον πράκτορα ούτε κέρδος, ούτε αύξηση του βαθμού ποιότητας Q_r . Εφόσον και οι δύο αυτοί στόχοι απαιτούν σχεδόν πάντα, ο πράκτορας να ολοκληρώσει την εκστρατεία, θεωρούμε ένα καλό πρώτο κριτήριο καταλληλότητας την εξής συνθήκη:

$$p \cdot C_r < 2 \cdot B(s) \quad (3)$$

Δηλαδή το αναμενόμενο κόστος για την απόκτηση ακριβώς όλων των εντυπώσεων που απαιτεί το συμβόλαιο να μην υπερβαίνει τον προϋπολογισμό και την τελική πληρωμή μας μαζί (που εκφράζεται ως το ποσοστό ολοκλήρωσης επί τον προϋπολογισμό), κάτι που θα οδηγούσε τον πράκτορά σε ζημία.

4.1.3 Επιλογή του κατάλληλου πονταρίσματος

Μετά το φιλτράρισμα των ακατάλληλων εκστρατειών και την απόφαση του πράκτορα να συμμετάσχει στην δημοπρασία για το συγκεκριμένο συμβόλαιο, πρέπει να υπολογιστεί ο προϋπολογισμός που θα ζητηθεί για την εκπλήρωση του συμβολαίου.

Όπως αναφέρθηκε και παραπάνω υπάρχει μια πιθανότητα $P_{randBid} = 0.36$ η εκστρατεία να δωθεί σε κάποιον πράκτορα εκτός του νικητή τυχαία, κάτι που παίζει καθοριστικό ρόλο στην

επιλογή της προσφοράς αφού είτε ο πράκτορας είναι σε μια ενδιάμεση θέση της δημοπρασίας (π.χ. τρίτος), είτε προσφέρει τον μέγιστο δυνατό προϋπολογισμό (τελευταία θέση), η πιθανότητα να του δοθεί η εκστρατεία είναι ίδια. Το γεγονός αυτό καθιστά ασύμφορες όλες τις προσφορές εκτός της ελάχιστης και της μέγιστης επιτρεπτής. Έτσι το πρόβλημα απόφασης εκφυλίζεται στην σύγκριση της αναμενόμενης τιμής για προσφορά του μέγιστου έναντι του ελάχιστου προϋπολογισμού.

Για τον υπολογισμό των αναμενόμενων τιμών χρειάζεται να βρεθεί πρώτα η πιθανότητα να αναλάβει τελικά ο πράκτορας την εκστρατεία. Στην περίπτωση μέγιστης προσφοράς αυτή

είναι $\frac{P_{randBid}}{n-1}$ όπου n είναι ο συνολικός αριθμός των παικτών.

Για τον υπολογισμό της ίδιας πιθανότητας για ελάχιστο bid κρατιέται κατά τη διάρκεια του παιχνιδιού ιστορικό με τις συχνότητες ανάληψης εκστρατείας για κάθε πράκτορα. Μετρώντας τους πράκτορες που είναι επιθετικοί, δηλαδή παίρνουν εκστρατείες περισσότερες από το 1/8 των συνολικών που τους αντιστοιχεί υπολογίζουμε το μέτρο ανταγωνιστικότητας:

$$competitiveness \in [1, 2, \dots, 8]$$

Το αντίστροφό του είναι μία καλή προσέγγιση της ζητούμενης πιθανότητας, δεδομένου ότι δε γίνεται τυχαία ανάθεση (γεγονός με πιθανότητα $1 - P_{randBid}$).

Συνεπώς όταν η ακόλουθη συνθήκη (<αριθμός εξίσωσης>) αληθεύει τότε προσφέρουμε τον ελάχιστο προϋπολογισμό αλλιώς τον μέγιστο.

$$\frac{P_{randBid}}{n-1} \cdot B_{max}(c) < \frac{(1 - P_{randBid})}{competitiveness} \cdot B_{min}(c) \quad (5)$$

Όπου $B_{max}(c), B_{min}(c)$ ο μέγιστος και ελάχιστος επιτρεπτός προϋπολογισμός αντίστοιχα.

4.1.4 Έλεγχος καταλληλότητας του πονταρίσματος με βάση

προηγούμενα παιχνίδια

Πριν σταλεί το ποντάρισμα γίνεται ένας επιπλέον έλεγχος με ιστορικό που έχει συλλεχθεί από ελεγχόμενα πειράματα, ο οποίος αποτελεί σημαντικό κομμάτι της στρατηγικής δημοπρασιών συμβολαίων.

Το ιστορικό περιλαμβάνει τα εξής χαρακτηριστικά:

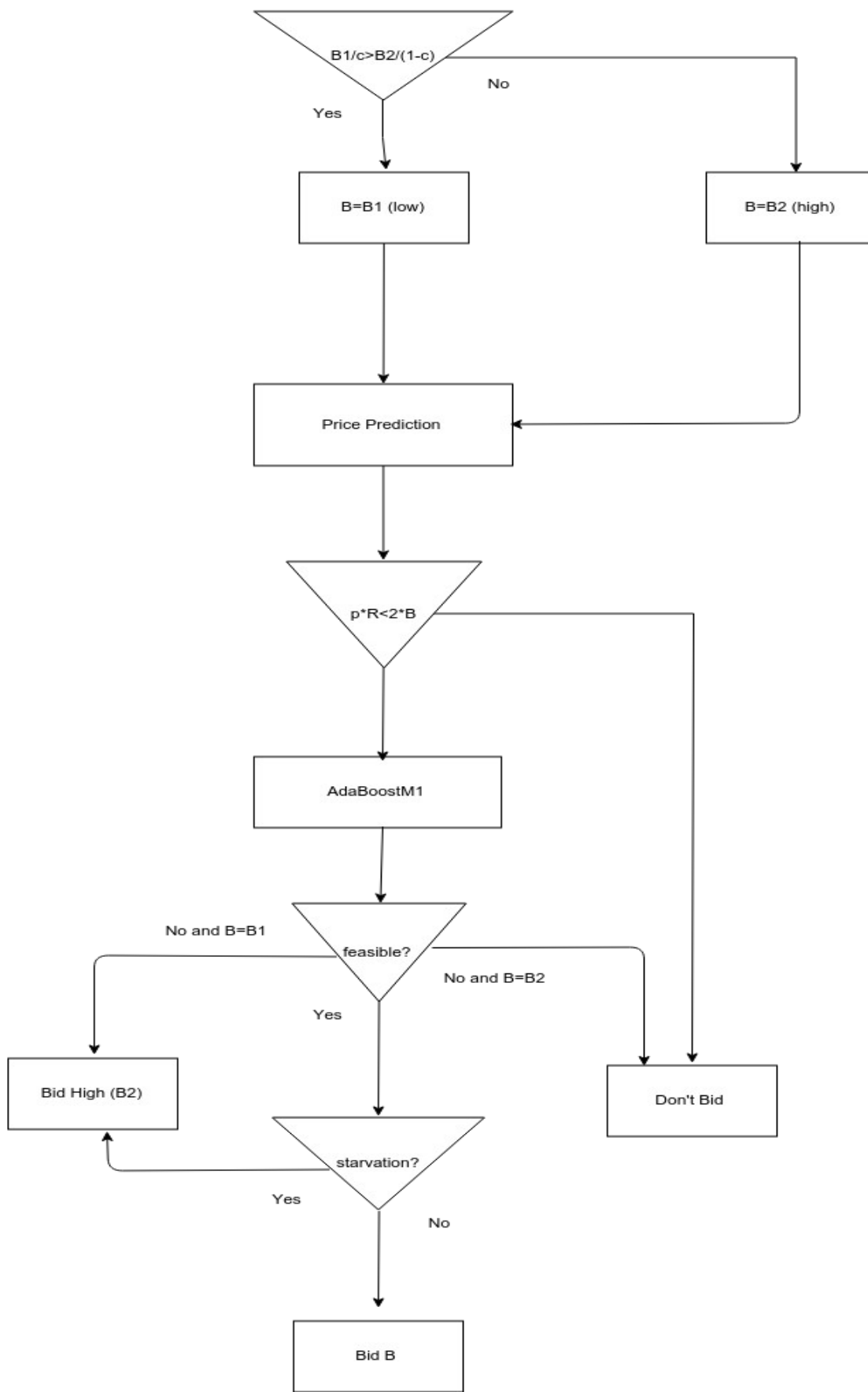
- Τον προϋπολογισμό της εκστρατείας

- Τις απαιτούμενες εντυπώσεις
- Τη μέρα
- Το βαθμό συμπλήρωσής της
- Το κέρδος του πράκτορα από αυτήν
- Το ποσοστό ζητούμενων εντυπώσεων από τον ίδιο τον πράκτορα σε σχέση με τις συνολικές διαθέσιμες εντυπώσεις ανά μέρα
- Τον κορεσμό του τμήματος αγοράς (congestion)
- Μια τιμή, +1 ή -1 που αντιπροσωπεύει αν η εκστρατεία πήγε καλά, αν δεν επηρέασε ιδιαίτερα την πορεία του πράκτορα ή αν ήταν ζημιογόνα.

Ο χαρακτηρισμός της πορείας της εκστρατείας γίνεται με τους παρακάτω κανόνες:

1. Αν η εκστρατεία είχε βαθμό ολοκλήρωσης ανάμεσα στο 0.9 και το 1.2 με οριακά θετικό κέρδος (<1) ή κέρδος πάνω από 10 με βαθμό ολοκλήρωσης πάνω από 0.4, τότε βάζουμε 1
2. Αν η εκστρατεία είχε βαθμό ολοκλήρωσης πάνω από 1.2 με κέρδος (>1), τότε βάζουμε 1
3. Διαφορετικά η εκστρατεία κρίνεται ακατάλληλη, -1

Το παραπάνω ιστορικό είναι το σύνολο δεδομένων εκπαίδευσης της μεθόδου AdaBoost με συμψηφισμό 250 δένδρων αποφάσεων. Αν η νέα εκστρατεία κατηγοριοποιηθεί στην τιμή 1 τότε προχωράμε κανονικά ποντάροντας. Αν έχει -1 και ο προϋπολογισμός είναι υψηλός δεν συμμετέχουμε στη δημοπρασία. Αν είναι χαμηλός τότε αντί για αυτόν ποντάρουμε τον μέγιστο δυνατό.



ροής στρατηγικής συμβολαίων

4.1 Διάγραμμα

4.2 Στρατηγική για τις δημοπρασίες εντυπώσεων

Στις δημοπρασίες εντυπώσεων ο πράκτορας έχει δύο στόχους:

- Να μεγιστοποιήσει το κέρδος του από την εκπλήρωση των συμβολαίων του
- Να διατηρήσει έναν αρκετά καλό βαθμό ποιότητας έτσι ώστε να μπορεί να είναι ανταγωνιστικός στις δημοπρασίες για διαφημιστικές εκστρατείες. Εδώ πρέπει να σημειωθεί ότι βάσει του τύπου:

$$0.0001 \cdot C_R / Q_r < B(c) < 0.001 \cdot C_R \cdot Q_r \quad (6)$$

αύξηση του Q_r σημαίνει όχι μόνο δυνατότητα χαμηλότερου πονταρίσματος από άλλους παίκτες και άρα μεγαλύτερη ανταγωνιστικότητα του πράκτορα, αλλά και δυνατότητα λήψης μεγαλύτερου προϋπολογισμού σε περίπτωση μέγιστου πονταρίσματος και άρα αυξημένου κέρδους.

Οι δύο παραπάνω στόχοι είναι προφανώς αντικρουόμενοι, αφού σε πολλές περιπτώσεις η αύξηση του Q_r απομακρύνει τον πράκτορα από το σημείο βέλτιστου κέρδους.

Για να αντιμετωπιστεί το παραπάνω πρόβλημα έγινε διατύπωση μιας αντικειμενικής συνάρτησης για το αναμενόμενο κέρδος του πράκτορα σε δύο φάσεις, αρχικά για μία τρέχουσα εκστρατεία και στη συνέχεια για όλο το υπόλοιπο παιχνίδι.

4.2.1 Εύρεση σημείου μέγιστου κέρδους για την τρέχουσα

εκστρατεία

Η πληρωμή του πράκτορα για τις υπηρεσίες του γίνεται ως ποσοστό επί του προϋπολογισμού που του δόθηκε ανάλογα με το ποσοστό ολοκλήρωσης της εκστρατείας (Effective Reach Ratio) που πέτυχε:

$$ERR = \frac{2}{a} \cdot \left(\arctan \left(a \cdot \frac{I}{C_r} - b \right) - \arctan(-b) \right) \quad (6)$$

όπου I ο αριθμός των συνολικών εντυπώσεων που αγοράστηκαν για αυτήν την εκστρατεία.

Τα συνολικά έσοδα είναι λοιπόν:

$$Revenue = ERR \cdot B(c) \quad (7)$$

Και το συνολικό κέρδος από την εκστρατεία:

$$Profit = ERR \cdot B(c) - p \cdot I \quad (8)$$

όπου p η προβλεπόμενη τιμή ανά εντύπωση για το τμήμα της αγοράς που αφορά στην τρέχουσα εκστρατεία.

Για το μέγιστο κέρδος πρέπει να ισχύει:

$$\frac{\partial Profit}{\partial I} = \frac{\partial ERR}{\partial I} \cdot B(c) - p = 0 \Leftrightarrow \quad (9)$$

$$\frac{\partial ERR}{\partial I} = \frac{p}{B(c)} \quad (10)$$

Το παραπάνω στέκει και διαισθητικά αφού ψάχνουμε τον αριθμό εντυπώσεων στον οποίο έστω και μια εντύπωση παραπάνω μας είναι ασύμφορη.

$$\frac{\partial ERR}{\partial I} = \frac{2}{C_r \cdot \left(1 + \left(a \cdot \frac{I}{C_r} - b\right)^2\right)} \quad (11)$$

Εξισώνοντας τις (10) και (11) και μετασχηματίζοντας καταλήγουμε στην παρακάτω εξίσωση δευτέρου βαθμού:

$$\left(\frac{a}{C_r}\right)^2 \cdot I^2 - \frac{2 \cdot a \cdot b}{C_r} \cdot I + 1 + b - \frac{2 \cdot B(c)}{C_r \cdot p} = 0 \quad (12)$$

Με λύσεις τις:

$$I_{1/2} = \frac{2 \cdot a \cdot b}{C_r} \pm \frac{2 \cdot a}{C_r} \cdot \sqrt{b^2 - 1 - b + \frac{2 \cdot B(c)}{C_r \cdot p}}$$

από τις οποίες κρατάμε αυτήν με το “+” όπου $\frac{\partial^2 ERR}{\partial^2 I} < 0$

άρα:

$$I^{max} = \frac{2 \cdot a \cdot b}{C_r} + \frac{2 \cdot a}{C_r} \cdot \sqrt{b^2 - 1 - b + \frac{2 \cdot B(c)}{C_r \cdot p}} \quad (13)$$

Η παραπάνω ανάλυση φυσικά δε λαμβάνει καθόλου υπ' όψιν τα μελλοντικά κέρδη λόγω του βαθμού ποιότητας. Ο παράγοντας αυτός προσεγγίστηκε στην αρχή με ευρετικό τρόπο,

χρησιμοποιώντας το παραπάνω μέγιστο σε εκστρατείες με μεγάλο προϋπολογισμό και χρησιμοποιώντας τις μικρότερες για να αυξηθεί στο μέγιστο δυνατό ο βαθμός ποιότητας.

Έχοντας ως στόχο τον παραπάνω αριθμό εντυπώσεων ορίζουμε την τιμή ανα εντύπωση, με ανάλογο τρόπο με την ενότητα 4.1 ως:

$$p = \frac{\sum_M C_{R,M,i}}{I_M} \cdot \frac{B(c)}{I^{max}} \quad (14)$$

Με βάση αυτήν την τιμή καθορίζεται ο μέγιστος αριθμός εντυπώσεων.

4.2.2 Εύρεση μέγιστου αναμενόμενου κέρδους για το υπόλοιπο του παιχνιδιού

Για την μοντελοποίηση του ισοζυγίου μεταξύ βαθμού ποιότητας και κέρδους γίνεται εισαγωγή μιας συνάρτησης $G(d, Q_r, c)$, όπου d η τρέχουσα μέρα και c ο βαθμός ανταγωνιστικότητας της τρέχουσας εκστρατείας. Δεδομένου ότι ο βαθμός απόδοσης είναι συνάρτηση του αριθμού των εντυπώσεων η G μπορεί να γραφτεί $G(d, I, c)$.

Τώρα στόχος είναι να μεγιστοποιήσουμε την συνάρτηση συνολικού αναμενόμενου κέρδους:

$$TotalProfit = ERR \cdot B(c) - p \cdot I + G(d, I, c) \quad (15)$$

Για να γίνει αυτό είναι εμφανές ότι πρέπει να υπολογιστεί η παράγωγος της G . Λόγω των πολλών αλληλοεξαρτήσεων και της τυχαιότητας που υπάρχουν στο παιχνίδι, ο αναλυτικός υπολογισμός της είναι ιδιαίτερα δύσκολος. Έτσι ο πράκτορας Hermes προσπαθεί να μάθει όσες τιμές της τον ενδιαφέρουν στατιστικά μέσα από ελεγχόμενα πειράματα.

Αν χειριστούμε λοιπόν την παράγωγο της G σαν άγνωστη κατά διαστήματα γραμμική συνάρτηση του I , φτάνουμε με τον ίδιο τρόπο όπως προηγουμένως στην:

$$I^{max} = \frac{2 \cdot a \cdot b}{C_r} + \frac{2 \cdot a}{C_r} \cdot \sqrt{b^2 - 1 - b + \frac{2 \cdot B(c)}{C_r \cdot p} + \frac{\partial G}{\partial I}} \quad (16)$$

Για την προσέγγιση της G που χρησιμοποιήσαμε, γίνεται εξαντλητικός έλεγχος σε περιορισμένες περιπτώσεις που μας ενδιαφέρουν έτσι ώστε να βρεθεί ο αριθμός εντυπώσεων που δίνει το μέγιστο συνολικό αναμενόμενο κέρδος.

Το τελικό ποντάρισμα (τιμή) είναι η πραγματική αξία που έχει κάθε εντύπωση για τον πράκτορα, άρα τελικά:

$$p = \frac{\text{Profit}(I^{max})}{I^{max}} \quad (17)$$

4.2.2.1 Στατιστική προσέγγιση της συνάρτησης G

Η στατιστική προσέγγιση της G βασίζεται σε δεδομένα που συλλέχθηκαν μέσα από ελεγχόμενα πειράματα (βλ. Κεφάλαιο 5).

Συγκεκριμένα βρέθηκε η μέση τιμή της G για διαστήματα (και όχι σημεία) του πεδίου ορισμού της. Λόγω της απρόβλεπτης φύσης του παιχνιδιού, θεωρήθηκε ότι μια ακριβής αναπαράσταση της G δεν θα είχε ιδιαίτερο νόημα.

Επίσης το γεγονός ότι θέλουμε να χρησιμοποιήσουμε την παράγωγο της g μας ενδιαφέρει απλά να έχουμε μια καλή εικόνα των αυξομειώσεών της και όχι ακριβείς τιμές.

Συγκεκριμένα το πεδίο ορισμού της G χωρίστηκε αρχικά ως προς την ανταγωνιστικότητα c σε n, ίσο με το πλήθος των αντιπάλων, διαστήματα αφού άλλωστε η c παίρνει n διακριτές τιμές.

Στη συνέχεια κάθε ένα από αυτά τα υποδιαστήματα χωρίστηκε στα έντεκα, για έντεκα διαφορετικά εύρη τιμών του Q_r όπως φαίνεται παρακάτω.

Αριθμός διαστήματος	Εύρος Q_r
1	0.0 - 0.4
2	0.4 - 0.8
3	0.8 - 0.9
4	0.9 - 1.0
5	1.0 - 1.05
6	1.05 - 1.1
7	1.1 - 1.15
8	1.15 - 1.2
9	1.2 - 1.25
10	1.25 - 1.3
11	1.3 +

Τα παραπάνω διαστήματα για το Q_r καθορίστηκαν εμπειρικά, ανάλογα με το ποιες περιοχές τιμών χρησιμοποιούνται πιο συχνά και χρειάζονται μεγαλύτερη ακρίβεια.

Τέλος τα 8X11 διαστήματα χωρίστηκαν στα 60, για κάθε ημέρα ενός γύρου του παιχνιδιού.

Η παράγωγος της G εκτιμάται λοιπόν για το διάστημα i του πίνακα με δεδομένα τα c και d ως εξής:

$$\frac{\partial G}{\partial I} \approx \frac{G_{i+1} - G_i}{I_{max,i+1} - I_{max,i}} \quad (18)$$

4.3 Προσαρμογή του πράκτορα στις αλλαγές των συνθηκών του παιχνιδιού

Λόγω της πολυπλοκότητας του παιχνιδιού, πρέπει να προνοήσουμε και στην περίπτωση που κάποιο από τα παραπάνω στοιχεία του μοντέλου δεν έχει προβλεφθεί με ακρίβεια. Έτσι τόσο στις δημοπρασίες συμβολαίων όσο και στις δημοπρασίες εντυπώσεων πρέπει ο πράκτορας να καταλαβαίνει πότε αποκλίνει από τους στόχους του και να αναπροσαρμόζεται με τέτοιο τρόπο ώστε η συμπεριφορά του να παραμένει βέλτιστη.

4.3.1 Προσαρμογή της ανταγωνιστικότητας (c) και κατά τη διάρκεια ενός γύρου του παιχνιδιού

Η ανταγωνιστικότητα (c), μέγεθος πολύ χρήσιμο για την επιλογή στρατηγικής στις δημοπρασίες συμβολαίων είναι μέγεθος που προβλέπεται από παιχνίδι σε παιχνίδι με τη μέθοδο απλής εκθετικής εξομάλυνσης.

Δυστυχώς αυτό σημαίνει ότι σε περίπτωση που κάποιος ή κάποιοι αντίπαλοι αλλάξουν στρατηγική σε επόμενο γύρο, ο πράκτορας Hermes θα αργήσει έναν ή ίσως και δύο γύρους να προσαρμοστεί, ανάλογα με το πόσο διαφορετικό θα γίνει το νέο περιβάλλον του.

Για αυτόν τον λόγο η ανταγωνιστικότητα πρέπει να προσαρμόζεται και κατά τη διάρκεια ενός γύρου ως εξής:

$$c = \left(1 - \frac{d}{60}\right) \cdot c_{pred} + \frac{d}{60} \cdot c_{ingame} \quad (19)$$

όπου d η τρέχουσα μέρα.

Ο παραπάνω τύπος μπαίνει σε ισχύ από τον 15ο γύρο, όταν έχουμε ήδη αρκετοί πληροφορία (10 γύρους δημοπρασιών συμβολαίου) για το πόσο ανταγωνιστικός είναι ο γύρος.

Παρατηρούμε ότι όσο ο γύρος εξελίσσεται και η εντός γύρου πρόβλεψη γίνεται πιο ακριβής, τόσο μεγαλύτερο βάρος δίνεται σε αυτήν.

4.3.1.1 Γιατί απλή εκθετική εξομάλυνση;

Η απλή εκθετική εξομάλυνση, όπως έχει αναφερθεί (βλ. Ενότητα 3.4) είναι μια μέθοδος πρόβλεψης για χρονοσειρές που παραμένουν σχετικά σταθερές. Επίσης είναι ιδανική όταν χρειάζεται να γίνει μόνο μια μελλοντική πρόβλεψη όπως στην περίπτωση μας.

Από τα πειράματα που έγιναν παρατηρήθηκε ότι η ανταγωνιστικότητα παραμένει σχετικά σταθερή, με τιμές ανάμεσα στο 2 και το 4, χωρίς ιδιαίτερα απότομες μεταβολές, και για αυτόν τον λόγο χρησιμοποιήθηκε απλή εκθετική εξομάλυνση με παράμετρο $\alpha = 0.3$.

Επειδή η SES με $\alpha = 0.3$ παρουσιάζει δυσκολία στο να ακολουθήσει μεγάλες αλλαγές του επιπέδου, σε περίπτωση που έχουμε 4 συνεχόμενες προβλέψεις (10% των γύρων του TAC AdX) κάτω ή πάνω από τις αντίστοιχες παρατηρήσεις, τότε αλλάζουμε το α σε 0.7 για τους επόμενους 4 γύρους.

4.3.2 Προσαρμογή της τιμής αγοράς εντυπώσεων στις

δημοπρασίες εντυπώσεων

Και στις δημοπρασίες εντυπώσεων υπάρχουν πολλά γεγονότα που θα μπορούσαν να βγάλουν τον πράκτορά από την προβλεπόμενη πορεία του ανεβάζοντας την τιμή, όπως:

-Η είσοδος ενός επιθετικού πράκτορα με τεράστιο προϋπολογισμό στο ίδιο τμήμα αγοράς με την εκστρατεία του

-Η απρόβλεπτη υπερσυσσώρευση ζητούμενων εντυπώσεων, επειδή εκδόθηκαν πολλές νέες εκστρατείες για το τρέχον τμήμα της αγοράς

-Κάποιος κακόβουλος πράκτορας που στοχεύει στο να σαμποτάρει τους αντιπάλους του

Το προφανές που μπορεί να γίνει για να λυθούν τα παραπάνω είναι η αύξηση της τιμής αγοράς των εντυπώσεων. Το ερώτημα που τίθεται όμως είναι πότε αξίζει και πότε όχι να συνεχίσουμε να αυξάνουμε την τιμή. Αυτό σκοπεύει να απαντήσει η παρακάτω ανάλυση.

Έστω P_{pred}, P_{new} το παλιό προβλεπόμενο και το νέο προβλεπόμενο κέρδος αντίστοιχα. Τότε το αναμενόμενο ποσό ανά εντύπωση που θα χάσουμε αν δεν ανεβάσουμε την τιμή είναι:

$$L = \frac{P_{pred} - P_{new}}{I_{rest}}$$

Και συνεπώς προσαρμόζουμε τη νέα τιμή στο να είναι:

$$p' = (1 + L) \cdot p \quad (20)$$

Επειδή ο παραπάνω τρόπος είναι αρκετά ομοιόμορφος και εύκολος να μοντελοποιηθεί ενδεχομένως από έναν αντίπαλο πράκτορα, καθώς επίσης και ότι η σημασία αύξησης της τιμής γίνεται μεγαλύτερη όσο η εκστρατεία πλησιάζει προς το τέλος της, τραβάμε τη νέα προσφερόμενη τιμή από μια φραγμένη εκθετική συνάρτηση με μέγιστη τιμή την $2.7p'$.

$$p'' = e^{\frac{(d_{end} - d)}{d_{end} - d_{start} + 1}} \cdot p' \quad (21)$$

αν $p'' < p'$ τότε $p'' = p'$.

4.3.3 Προσαρμογή της τιμής αγοράς εντυπώσεων με βάση

την επιθετικότητα των άλλων πρακτόρων

Επίσης γίνεται επιπλέον προσαρμογή της αγοράς εντυπώσεων με βάση τη συμπεριφορά των άλλων πρακτόρων στις δημοπρασίες εντυπώσεων. Αρχικά συλλέγονται τα εξής δεδομένα:

- Ανταγωνιστικότητα (c)
- Η μεταβλητότητα της συχνότητας ανάληψης συμβολαίων όλων των παικτών
- Η μέση τιμή ανά εντύπωση του πράκτορα
- Ο μέσος βαθμός ποιότητας
- Το ποσοστό του κέρδους προς τα συνολικά έσοδα

τα οποία ταξινομούνται στις κλάσεις 1 (λίγο επιθετικοί) μέχρι 7 (πολύ επιθετικοί) με τη βοήθεια ενός ταξινομητή Bayes. Η τελική τιμή διαμορφώνεται ως:

$$p_{final} = (1 + 0.05 \cdot pred_{Bayes}) \cdot p'' \quad (22)$$

4.4 Στρατηγική UCS

Μια καλή θέση στη δημοπρασία για την κατηγοριοποίηση χρηστών σημαίνει πρακτικά ότι ο πράκτορας μπορεί να “δει” περισσότερες στοχευμένες εντυπώσεις από τους αντιπάλους του. Σε αυτές τις εντυπώσεις ο πράκτορας θα έχει μειωμένο ανταγωνισμό και άρα θα μπορεί να τις αγοράσει με χαμηλότερη τιμή. Παρόλα αυτά είναι δύσκολος ο καθορισμός του κέρδους από τη βελτίωση (ή χειροτέρευση) του επιπέδου UCS, καθώς δεν είναι γνωστό το επίπεδο των αντιπάλων ενώ επίσης η επιλογή των εντυπώσεων που δεν θα δουν οι αντίπαλοι είναι τυχαία. Συνεπώς ο πράκτορας συνήθως δεν μπορεί να καθορίσει και με ποιους άλλους πράκτορες ανταγωνίζεται σε κάθε εντύπωση, άρα δεν μπορεί να υπολογίσει επακριβώς και την τιμή.

Για αυτούς τους λόγους η στρατηγική που χρησιμοποιείται είναι να προσαρμόζεται ο πράκτορας ανάλογα με την επίδοσή του στις τρεις τελευταίες μέρες, έτσι ώστε το επίπεδό του να μην πέφτει ποτέ πολύ χαμηλά.

Συγκεκριμένα χρησιμοποιείται ένα σταθερό επίπεδο UCS_{base} που ξεκινά με τιμή 0.1, ενώ το bid καθορίζεται με την προσθήκη ενός παράγοντα τυχαιότητας:

$$UCS_{Bid} = UCS_{base} + r, \quad (23)$$

με $r \in [0, 0.1]$.

Αν ο μέσος όρος των επιπέδων UCS των τριών τελευταίων γύρων είναι μικρότερος από 0.72, την τέταρτη δηλαδή θέση της δημοπρασίας, τότε αυξάνουμε το σταθερό επίπεδο κατά 0.01.

4.5 Ο πράκτορας dadNet

Ο πράκτορας dadNet αναπτύχθηκε από τον Γρηγόρη Δημόπουλο φοιτητή του τμήματος Πληροφορικής στο ΕΚΠΑ, ο οποίος ασχολήθηκε με το ίδιο θέμα στα πλαίσια της διπλωματικής του και με τον οποίο συνεργαστήκαμε για τη διαμόρφωση του τελικού πράκτορα. Οι βασικές στρατηγικές του dadNet είναι συνοπτικά οι εξής:

Campaign Bidding – Experienced Campaign Strategy: Ο πράκτορας κρατά σε μορφή ευρετηρίου ένα ιστορικό από πονταρίσματα για παλιότερες εκστρατείες μαζί με το περιβάλλον τους και στοιχεία για την έκβασή τους. Συγκεκριμένα: το τμήμα αγοράς, τη διάρκεια, τις

απαιτούμενες εντυπώσεις, τις εντυπώσεις και τον αριθμό των αντιπάλων που έχουν επικάλυψη, το κόστος της εκστρατείας και το βαθμό συμπλήρωσής της αφότου έχει ολοκληρωθεί. Ψάχνοντας στο ευρετήριο την πιο κοντινή εκστρατεία, ακολουθώντας τα κλειδιά με προτεραιότητα όπως αναφέρθηκαν, καταλήγει σε μια εκτίμηση για το κόστος της εκστρατείας και άρα τον χαμηλότερο προϋπολογισμό που θα ζητήσει. Επίσης σε ότι αφορά τις πέντε πρώτες μέρες, όπου ο ανταγωνισμός είναι μεγαλύτερος ποντάρει χαμηλά για εκστρατείες που η διάρκειά τους υπερβαίνει το διάστημα αυτό και μέγιστα για μικρές εκστρατείες που δεν το υπερβαίνουν.

Impression Bidding – Increased Quality Strategy: Η βάση του τελικού πονταρίσματος για κάθε εντύπωση είναι ένας μέσος όρος των παρακάτω: του μέσου μελλοντικού κόστους ανά εντύπωση, του αναμενόμενου κέρδους ανά εντύπωση, του μέσου κόστους ανά εντύπωση που έχει ήδη αγοραστεί. Οι τιμές των παραπάνω εξαρτώνται από την παράμετρο targetERR που καθορίζει τον βαθμό ολοκλήρωσης που στοχεύει η εκστρατεία. Στη συνέχεια για διακριτά ποσοστά αυτού του βασικού πονταρίσματος υπολογίζεται με βάση ιστορικό η πιθανότητα νίκης στη δημοπρασία καταλήγοντας σε ένα ποσοστό με αρκετά μεγάλη ($>1 - 1/c$) πιθανότητα όπου c ο αριθμός των επικαλυπτόμενων εκστρατειών.

UCS Bidding: Σε αυτή τη στρατηγική το περιβάλλον χωρίζεται σε τέσσερις κατηγορίες [Many, Few] Campaigns και [Many, Few] Segments ανάλογα με τις εκστρατείες του πράκτορα και τα τμήματα αγοράς τους. Για κάθε τέτοια κατηγορία, συνυπολογίζοντας και τον βαθμό ποιότητας, υπολογίζεται η επιδιωκόμενη θέση στη δημοπρασία UCS. Με τη βοήθεια ενός ιστορικού παλιότερων πονταρισμάτων βρίσκεται το κατάλληλο ποντάρισμα για αυτήν τη θέση.

4.6 Ο τελικός πράκτορας DemokriTAC

Για τον τελικό πράκτορα χρησιμοποιήθηκαν οι παρακάτω στρατηγικές που βασίζονται σε όσες αναφέρθηκαν:

- Για τις δημοπρασίες συμβολαίων η στρατηγική του Hermes αντικαθιστώντας το ελάχιστο επιτρεπτό ποντάρισμα με ποντάρισμα που καθορίζεται από την Experienced Campaign Strategy του dadNet, η οποία είχε πολύ καλά αποτελέσματα σαν επιθετική στρατηγική.
- Για τις δημοπρασίες εντυπώσεων χρησιμοποιήθηκε η στρατηγική του Hermes η οποία είχε τα καλύτερα αποτελέσματα.

- Για την UCS η στρατηγική του dadNet η οποία πετύχαινε παρόμοια επίπεδα UCS, όμως με αρκετά χαμηλότερο κόστος.

5

Αξιολόγηση

Η αξιολόγηση του πράκτορα αφορά την επίδοσή του σε διαφορετικά περιβάλλοντα, όπως περιβάλλοντα που οι αντίπαλοι είναι πολύ ανταγωνιστικοί ή που βασίζονται πολύ στα τυχαία στοιχεία του παιχνιδιού.

Επίσης παρουσιάζονται και τα αποτελέσματα των πειραμάτων παραμετροποίησης για τις τεχνικές μηχανικής μάθησης και πρόβλεψης που χρησιμοποιήθηκαν.

5.1 Οργάνωση πειραμάτων

Για την αξιολόγηση του πράκτορα χρησιμοποιείται ένα σύνολο από 7 διαφορετικά σενάρια (σύνολα αντιπάλων) αυξανόμενης επιθετικότητας με 40 παιχνίδια στο καθένα (όσα και στον διαγωνισμό TAC AdX). Ως benchmark χρησιμοποιείται μια παλιότερη έκδοση του πράκτορα μας η οποία δε χρησιμοποιεί καμία τεχνική μάθησης από παλιότερα δεδομένα παρά μόνο πρόβλεψη της ανταγωνιστικότητας με απλή εκθετική εξομάλυνση. Επίσης σαν παράδειγμα κατεξοχήν επιθετικού πράκτορα χρησιμοποιείται ο πράκτορας *sample*, ο οποίος χρησιμοποιείται επίσης σαν benchmark, που είχε αναπτυχθεί κατά την αρχή εκπόνησης της εργασίας μέσα από εμπειρικές παρατηρήσεις. Η κύρια ιδέα πίσω από αυτόν τον πράκτορα είναι να πάρει όσες περισσότερες εκστρατείες μπορεί στοχεύοντας να έχει πάντα τον υψηλότερο βαθμό ποιότητας.

Τα 7 σενάρια είναι τα εξής:

Σενάριο 1: Hermes, Hermes_old, Dummy Adnetworks 1-6 (Πράκτορες που χρησιμοποιούνται αυτόματα από τον adx-server για να συμπληρώσουν τις 8 θέσεις)

Σενάριο 2: Hermes, Hermes_old, giza , Agent2, agent00, BCM, ibm, OOS

Σενάριο 3: Hermes, Hermes_old, giza , Agent2, agent00, BCM, ibm, sample

Σενάριο 4: Hermes, Hermes_old, giza(x2), Agent2, BCM, ibm, sample

Σενάριο 5: Hermes, Hermes_old, giza(x2), Agent2, ibm, sample(x2)

Σενάριο 6: Hermes, Hermes_old, giza(x2) , ibm, sample(x2)

Σενάριο 7: Hermes, Hermes_old, giza(x2), ibm, sample(x3)

Οι παραπάνω πράκτορες χωρίζονται εμπειρικά σε επιθετικούς και συντηρητικούς ως εξής:

- *Επιθετικοί:* sample, giza
- *Αμυντικοί:* Agent2, agent00, BCM, ibm, OOS, Dummy Adnetworks 1-6

Ο πράκτορας giza είναι επίσης επιθετικός οπότε η επιθετικότητα αυξάνεται προσθέτοντας sample και giza εναλλάξ στο σενάριο 2. Κάθε φορά που προστίθεται ένας νέος ανταγωνιστικός πράκτορας αφαιρείται ο χειρότερος παλιός.

5.2 Παράμετροι αξιολόγησης

Αρχικά οι σημαντικότερες παράμετροι που εξετάζονται είναι το συνολικό κέρδος του πράκτορα στα 7 παραπάνω σενάρια καθώς και το μέσο κέρδος ανά σενάριο και ο μέσος βαθμός ποιότητας. Επιπλέον παράμετροι εξετάζονται για την αξιολόγηση των μεθόδων που χρησιμοποιήθηκαν οι οποίες ταξινομούνται ανά στρατηγική.

5.2.1 Παράμετροι για την σύγκριση στρατηγικών συμβολαίων

Για τη σύγκριση στρατηγικών συμβολαίων χρησιμοποιούνται επιπλέον οι παρακάτω παράμετροι:

- Ποσοστό αποτυχημένων εκστρατειών ($ERR < 1.0$ και $\text{κέρδος} < 2$)
- Μέσος αριθμός εκστρατειών ανά παιχνίδι

5.2.2 Παράμετροι για την σύγκριση στρατηγικών εντυπώσεων

Για τις στρατηγικές εντυπώσεων χρησιμοποιούνται επιπλέον:

- Ποσοστό εκστρατειών με μεγάλο βαθμό συμπλήρωσης ($ERR > 1.2$)
- Μέση τιμή ανά εντύπωση
- Μέσο ποντάρισμα ανά εντύπωση

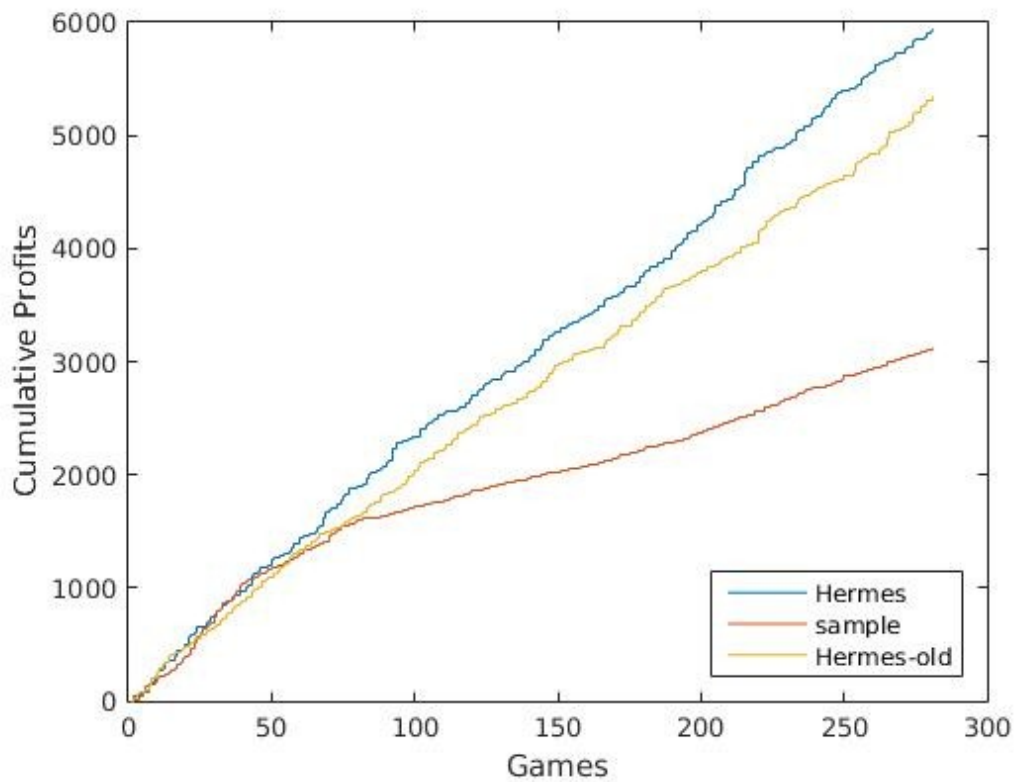
5.2.3 Παράμετροι για την σύγκριση στρατηγικών UCS

Για τις στρατηγικές εντυπώσεων χρησιμοποιούνται επιπλέον:

- Μέσο επίπεδο UCS
- Μέσο ποντάρισμα UCS

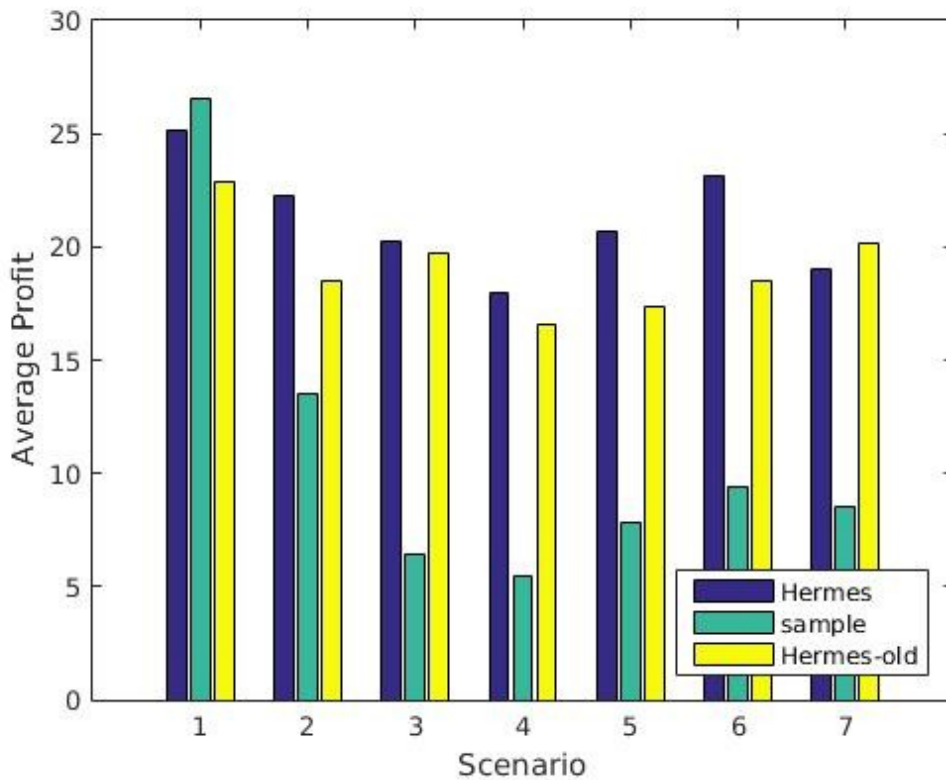
5.3 Αποτελέσματα

Παρακάτω βλέπουμε τα τελικά αποτελέσματα του κυρίως πειράματος που αποτελείται από 7 σενάρια με 40 παιχνίδια στο καθένα (σύνολο: 280)



5.1 Εξέλιξη του συνολικού κέρδους

Εδώ βλέπουμε ότι η έκδοση του πράκτορα που “μαθαίνει” το περιβάλλον της έχει σταθερά μεγαλύτερα κέρδη από την παλιά έκδοση που δεν χρησιμοποιεί μηχανική μάθηση. Επίσης ο επιθετικός πράκτορας sample ο οποίος δεν προσαρμόζει καθόλου την στρατηγική του, ενώ αρχίζει πολύ καλά στα λιγότερο ανταγωνιστικά σενάρια, μένει γρήγορα πίσω.



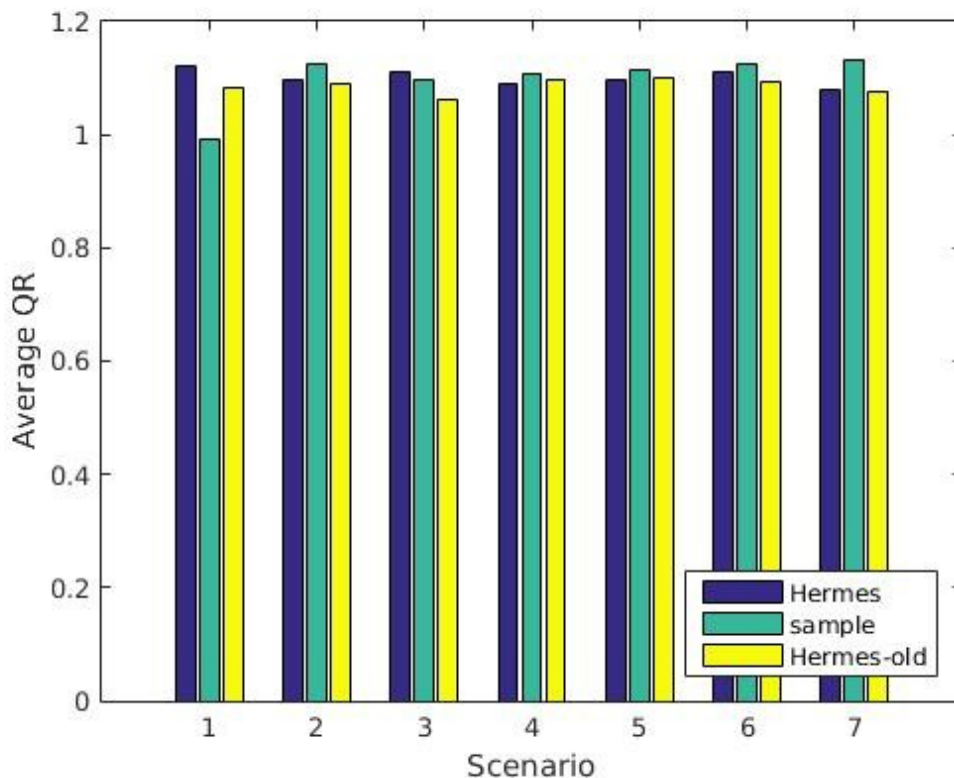
5.2 Μέσο κέρδος ανά σενάριο. Η επιθετικότητα κάθε σεναρίου αυξάνεται από αριστερά προς τα δεξιά

Εδώ παρατηρούμε πάλι ότι η τελευταία έκδοση του πράκτορα τα πηγαίνει καλύτερα σε όλα τα σενάρια εκτός από το τελευταίο. Εδώ πρέπει να σημειωθεί ότι στα τελευταία δύο σενάρια, ο τυχαίος παράγοντας παίζει μεγάλο ρόλο αφού λόγω του μεγάλου ανταγωνισμού, ο πράκτορας κρίνει πιο συμφέρον να βασιστεί σε εκστρατείες με πολύ μεγάλο προϋπολογισμό που του ανατίθενται τυχαία.

Η τάση στο παιχνίδι είναι τα σκορ να είναι χαμηλότερα όσο η ανταγωνιστικότητα αυξάνει. Το χαμηλό σκορ στο σενάριο 4 οφείλεται ότι ο πράκτορας αναπροσαρμόζει πολύ συχνά την συμπεριφορά του. Πιθανότατα μια πιο γρήγορη μέθοδος προσαρμογής να έφερνε και καλύτερο σκορ.

Επίσης στο πρώτο σενάριο ο πράκτορας sample έχει την καλύτερη επίδοση. Αυτό συμβαίνει γιατί οι άλλοι δύο πράκτορες κάνουν μερικά μέτρια παιχνίδια στην αρχή μέχρι η μέθοδος της απλής εκθετικής εξομάλυνσης να προσεγγίσει την κατάλληλη τιμή ανταγωνιστικότητας.

Σε σχέση τέλος με τους υπόλοιπους πράκτορες οι hermes και hermes_old έχουν μεγαλύτερα σκορ σχεδόν σε όλα τα παιχνίδια.



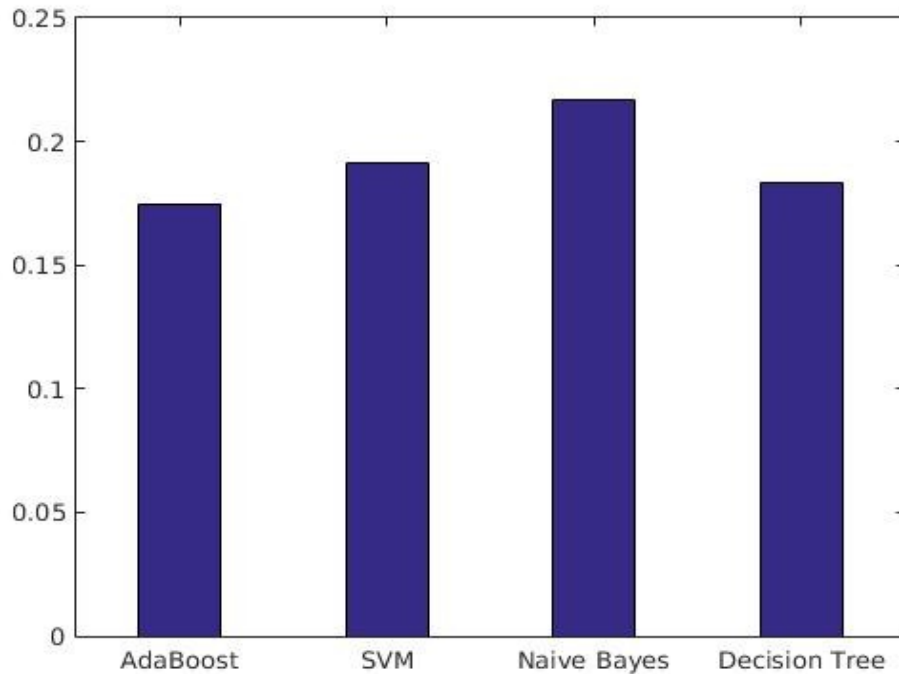
5.3 Μέσος βαθμός ποιότητας (Quality Rating) ανά σενάριο. Η επιθετικότητα κάθε σεναρίου αυξάνεται από αριστερά προς τα δεξιά

Εδώ βλέπουμε ότι ο πράκτορας κρατά σε όλα τα σενάρια έναν πολύ καλό βαθμό ποιότητας, κάτι που σημαίνει ότι συμπληρώνει όλες του σχεδόν τις εκστρατείες (και με το παραπάνω). Ο sample προφανώς έχει τα υψηλότερα QR αφού η στρατηγική του βασίζεται περισσότερο σε αυτόν τον παράγοντα.

5.3.1 Επιλογή μεθόδου χαρακτηρισμού των υποψήφιων εκστρατειών

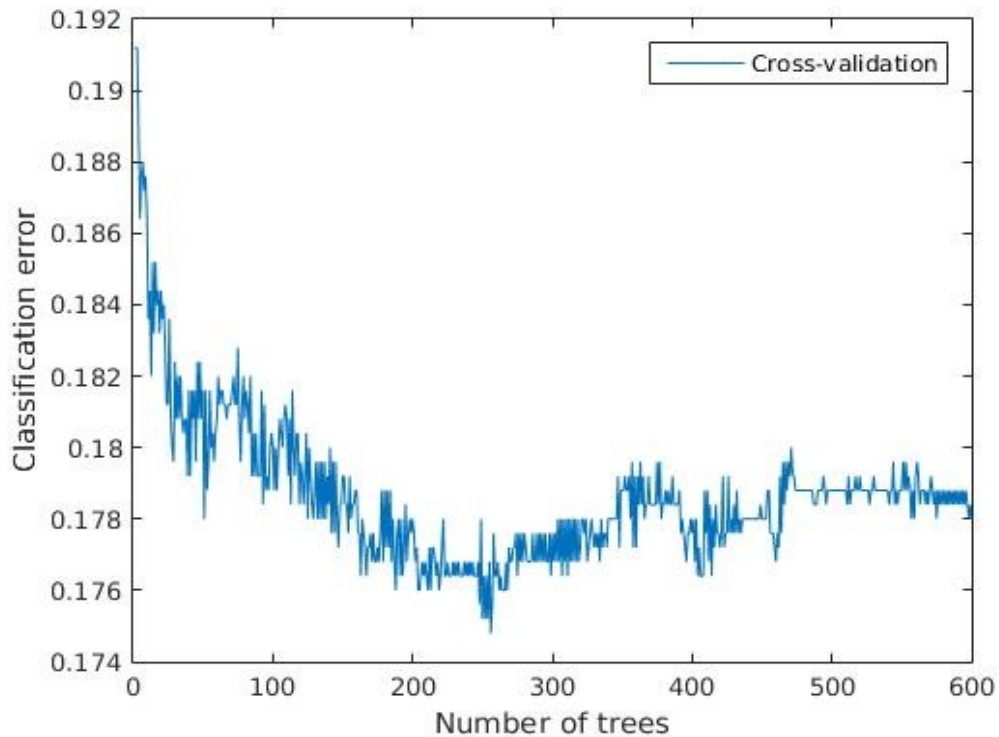
Για την επιλογή του κατάλληλου ταξινομητή εκστρατειών σε κατάλληλες και ακατάλληλες έγινε σύγκριση μεταξύ τεσσάρων ειδών ταξινομητών: AdaBoostM1, SVM, Naïve Bayes, Decision Tree. Οι ταξινομητές αυτοί επιλέχθηκαν κυρίως λόγω της απλότητας και της ταχύτητάς τους, για να μπορούν να ανταποκριθούν στους χρονικούς περιορισμούς του διαγωνισμού AdX. Εκπαιδεύτηκαν σε ένα σετ 2500 παραδειγμάτων και ως μέτρο σύγκρισης

χρησιμοποιήθηκε το ποσοστό λανθασμένων προβλέψεων σε 10-fold cross-validation. Για κάθε ταξινομητή έγινε επιλογή ενός όσο γίνεται καλύτερου σει παραμέτρων.



5.4 Σύγκριση σφάλματος μεθόδων για την εκμάθηση της καταλληλότητας μιας εκστρατείας

Το ελάχιστο σφάλμα για την AdaBoost προκύπτει όταν έχουμε περίπου 250 βασικούς ταξινομητές τύπου δέντρου:

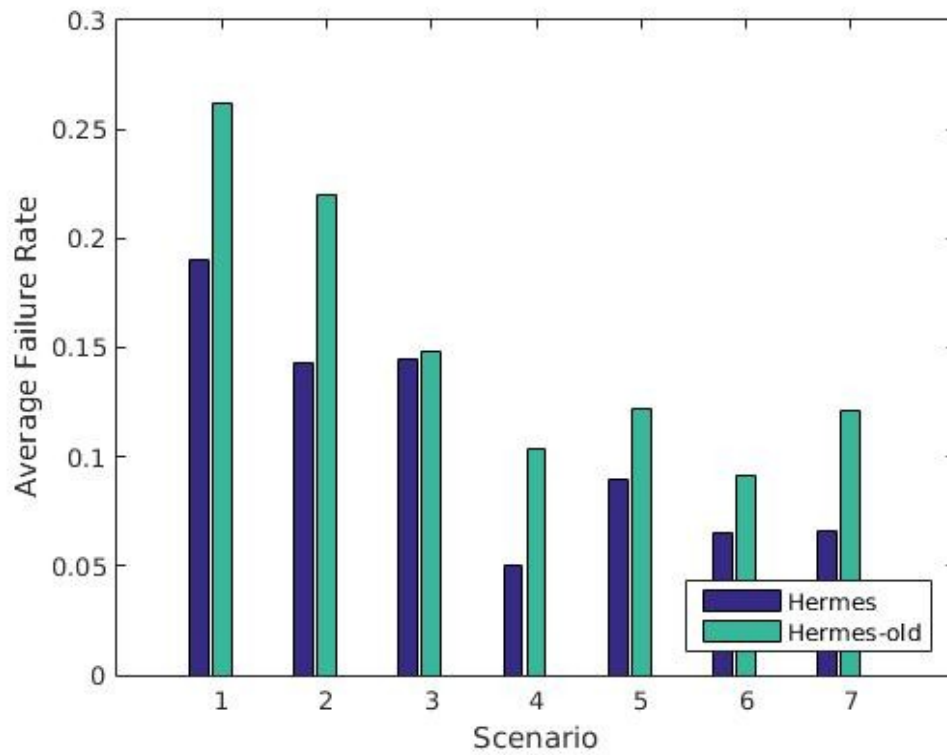


5.5 Σφάλμα μεθόδου AdaBoost συναρτήσει του αριθμού βασικών ταξινομητών (δέντρα)

Για τον ταξινομητή SVM το μικρότερο σφάλμα είχε το SVM με γκαουσιανή συνάρτηση πυρήνα, ενώ για τα δέντρα αποφάσεων επιλέχθηκαν 60 παρατηρήσεις ανά φύλλο.

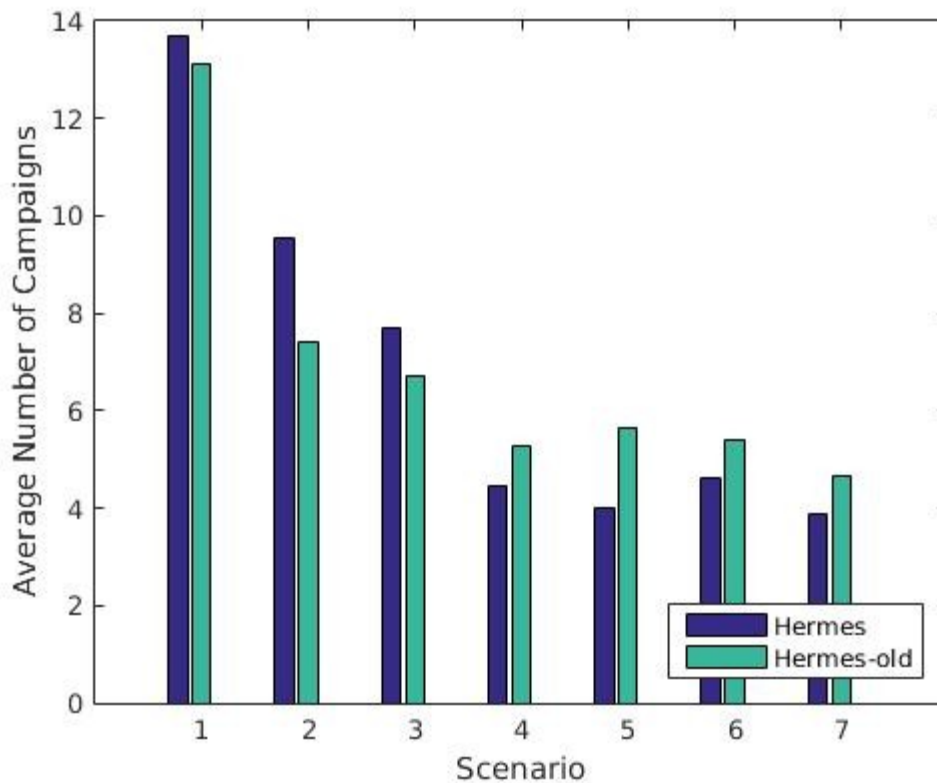
5.3.2 Αποτελέσματα εκστρατειών συμβολαίων

Για την αξιολόγηση της στρατηγικής συμβολαίων χρησιμοποιούμε τον μέσο αριθμό αποτυχημένων εκστρατειών ανά σενάριο καθώς και τον μέσο αριθμό εκστρατειών ανά σενάριο.



5.6 Μέσο ποσοστό αποτυχημένων εκστρατειών ανά σενάριο

Στο παραπάνω διάγραμμα βλέπουμε μεγάλη διαφορά ανάμεσα στην παλιά και τη νέα έκδοση του πράκτορα, κάτι που κάνει εμφανές το πλεονέκτημα χρήσης της AdaBoost.



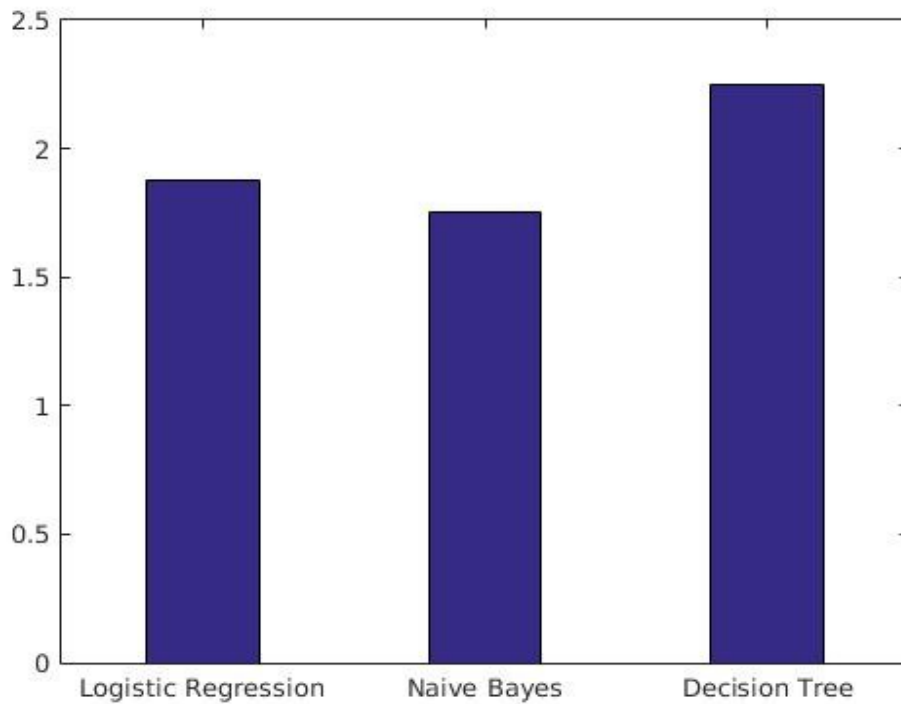
5.7 Μέσος αριθμός εκστρατειών ανά σενάριο

Εδώ βλέπουμε ότι στα λιγότερο ανταγωνιστικά σενάρια ο νέος πράκτορας αναλαμβάνει περισσότερες εκστρατείες ενώ στα πολύ ανταγωνιστικά λιγότερες. Αυτό οφείλεται στο ότι όσο αυξάνεται η ανταγωνιστικότητα όλο και περισσότερες εκστρατείες κρίνονται ακατάλληλες ή κατάλληλες μόνο με μεγάλο budget και άρα η συχνότητα των εκστρατειών μειώνεται.

5.3.3 Επιλογή μεθόδου χαρακτηρισμού της επιθετικότητας

στις δημοπρασίες εντυπώσεων

Και εδώ για την επιλογή του κατάλληλου ταξινομητή έγινε σύγκριση μεταξύ τριών ειδών ταξινομητών: K-Nearest Neighbours, Naive Bayes, Decision Tree. Οι ταξινομητές αυτοί επιλέχθηκαν κυρίως λόγω της απλότητας και της ταχύτητάς τους, για να μπορούν να ανταποκριθούν στους χρονικούς περιορισμούς του διαγωνισμού AdX, όπως επίσης και λόγω των περιορισμένων διαθέσιμων δεδομένων (300 παραδείγματα για 8 κλάσεις). Ως μέτρο σύγκρισης χρησιμοποιήθηκε το MSE (μέσο τετραγωνικό σφάλμα – mean absolute error) σε ένα test set 40 παραδειγμάτων. Δεν έγινε cross-validation αφού με τόσα λίγα παραδείγματα δεν υπάρχει περίπτωση overfitting.



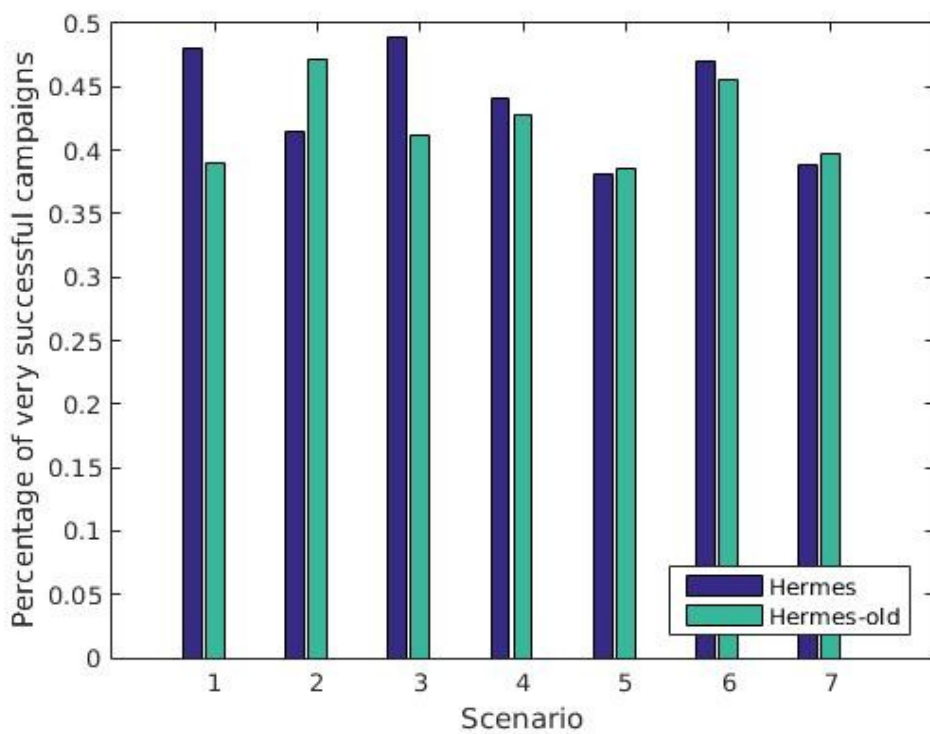
5.8 Σύγκριση μεθόδων εκμάθησης επιθετικότητας στις δημοπρασίες εντυπώσεων

Τελικά επιλέχθηκε η μέθοδος naïve bayes που είχε το μικρότερο μέσο τετραγωνικό σφάλμα.

5.3.4 Αποτελέσματα στρατηγικών για τις δημοπρασίες

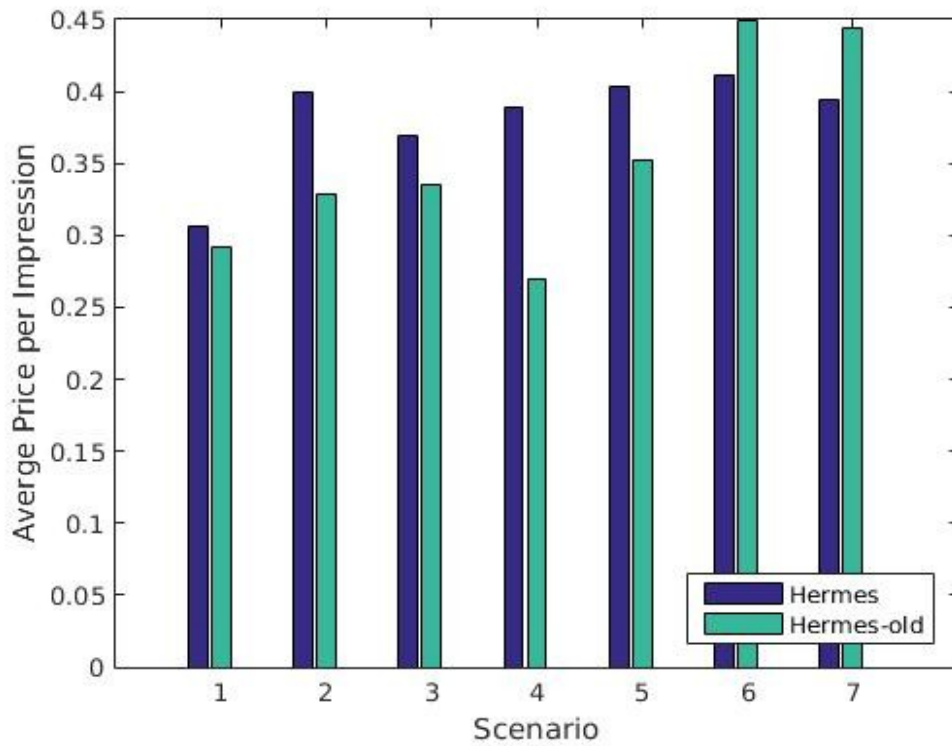
εντυπώσεων

Για την αξιολόγηση της στρατηγικής δημοπρασιών εντυπώσεων χρησιμοποιούμε τον μέσο αριθμό πολύ επιτυχημένων εκστρατειών ανά σενάριο τη μέση τιμή ανά εντύπωση, ανά σενάριο και το μέσο ποντάρισμα ανά εντύπωση ανά σενάριο.



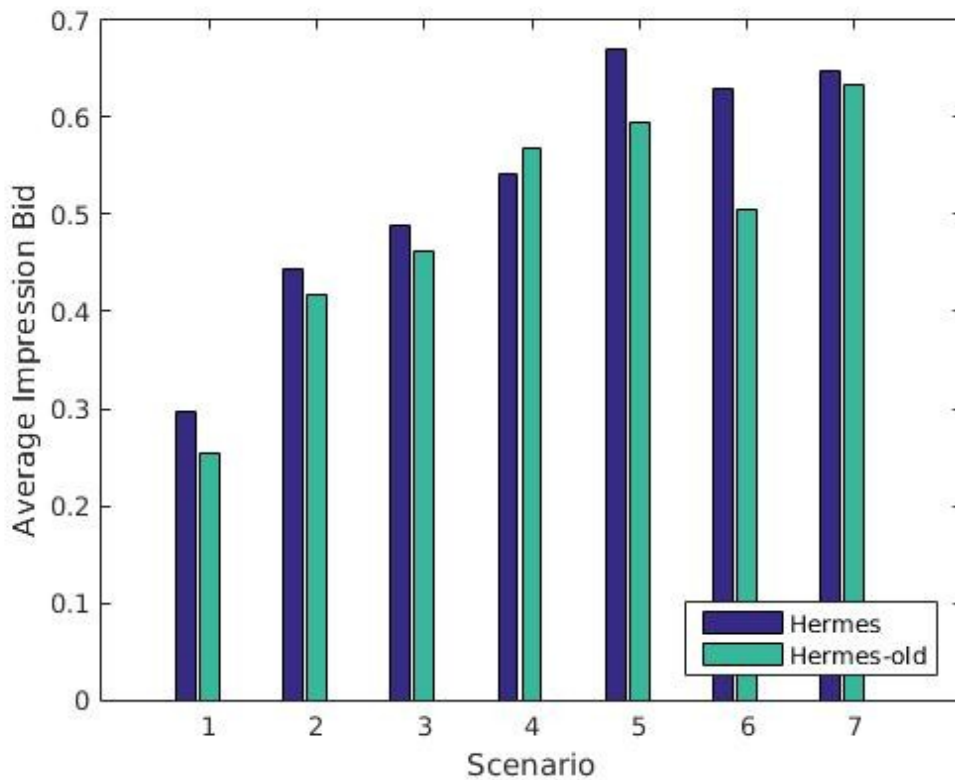
5.9 Ποσοστό πολύ πετυχημένων εκστρατειών ανά σενάριο

Παρατηρούμε ότι ο πράκτορας καταφέρνει και συμπληρώνει σχεδόν τις μισές του εκστρατείες με $ERR > 1.2$ κάτι που τον βοηθά πολύ στη διατήρηση καλού βαθμού ποιότητας. Επίσης τα ποσοστά του φαίνονται ελαφρώς καλύτερα από αυτά της παλιάς έκδοσης.



5.10 Μέση τιμή ανά εντύπωση ανά σενάριο

Εδώ παρατηρούμε ότι η μέση τιμή ανά εντύπωση (σε χιλιοστά - millis) που πληρώνει ο πράκτορας παραμένει σχετικά σταθερή και στα 7 σενάρια.



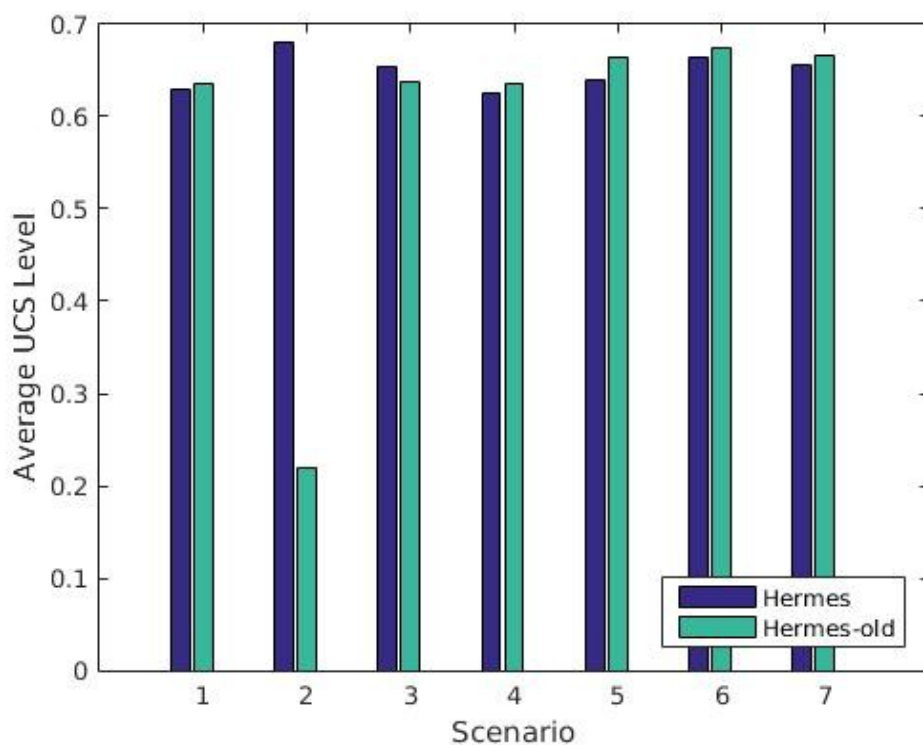
5.11 Μέσο ποντάρισμα ανά εντύπωση ανά σενάριο

Παρατηρούμε ότι όσο αυξάνεται η ανταγωνιστικότητα και ο πράκτορας στοχεύει σε εκστρατείες με μεγάλο budget τόσο αυξάνεται η αξία που έχει κάθε εντύπωση για αυτόν άρα και το ποντάρισμά του. Επίσης τα πονταρίσματα είναι πάλι ελαφρώς υψηλότερα από την παλιά έκδοση.

5.3.4 Αποτελέσματα στρατηγικών για την υπηρεσία

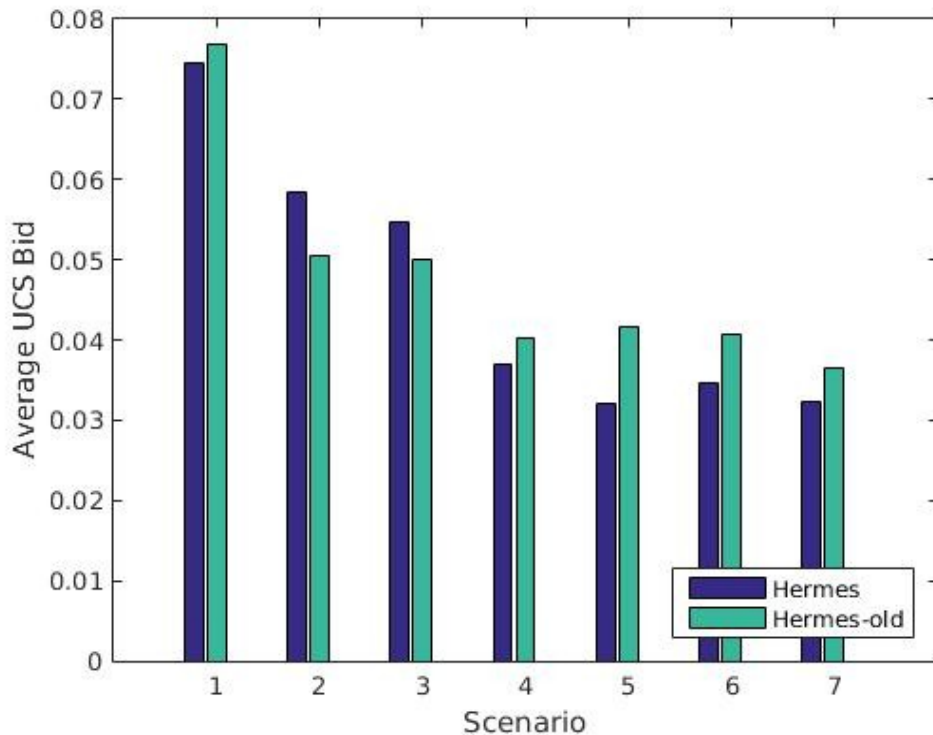
κατηγοριοποίησης χρηστών UCS

Σε αυτήν την ενότητα αξιολογούνται το μέσο επίπεδο UCS καθώς και το μέσο ποντάρισμα στις γενικευμένες δημοπρασίες δεύτερης τιμής.



5.12 Μέσο επίπεδο UCS ανά σενάριο

Στο παραπάνω διάγραμμα ο πράκτορας φαίνεται να κρατά ένα ικανοποιητικό επίπεδο UCS το οποίο παραμένει σταθερό σε όλα τα σενάρια αντίθετα με την παλιά έκδοση που δεν περιλάμβανε αλλαγή του σταθερού επιπέδου. Εδώ έχουν ληφθεί υπόψιν και τα μηδενικά πονταρίσματα (όταν ο πράκτορας δεν έχει εκστρατείες) και άρα είναι ασφαλές να συμπεράνουμε ότι συνήθως βρίσκεται πάνω από την τρίτη θέση της δημοπρασίας.



5.13 Μέσο ποντάρισμα UCS ανά σενάριο

Στο διάγραμμα μέσου κόστους παρατηρούμε ότι ο πράκτορας ξοδεύει περισσότερα για την υπηρεσία UCS στα λιγότερο ανταγωνιστικά σενάρια που έχει και περισσότερες εκστρατείες.

5.4 Σύνοψη συμπερασμάτων αξιολόγησης

Συνοψίζοντας τα παραπάνω αποτελέσματα οδηγούμαστε στα εξής συμπεράσματα:

- Ο πράκτορας καταφέρνει και είναι προσαρμοστικός κρατώντας σχεδόν πάντα το υψηλότερο σκορ υιοθετώντας την κατάλληλη συμπεριφορά σε κάθε περιβάλλον.
- Η χρήση των τεχνικών μηχανικής μάθησης με συλλογή παλιότερων δεδομένων παρότι δεν παίζει κεντρικό ρόλο (ο πράκτορας βασίζεται κυρίως στο παρόν για τις σημαντικότερες αποφάσεις του) βελτιστοποιεί σημαντικά την επίδοσή του.
- Η ταχύτητα προσαρμογής του πράκτορα στις δημοπρασίες συμβολαίων εξαρτάται κυρίως από τη σύγκλιση της απλής εκθετικής εξομάλυνσης σε μια χαρακτηριστική για το σενάριο τιμή της ανταγωνιστικότητας. Όπως φαίνεται στο Σενάριο 1 σημαντικό ρόλο παίζει η επιλογή της αρχικής πρόβλεψης που εδώ τίθεται ίση με 2 και οδηγεί στον πράκτορα να είναι πιο αμυντικός από ότι θα έπρεπε στους πρώτους

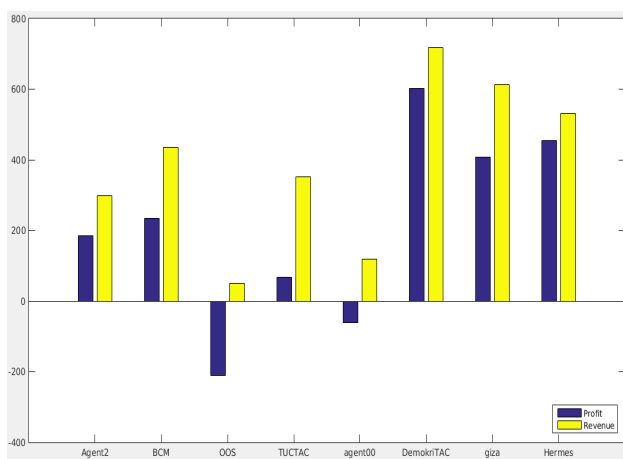
γύρους κάτι που δεν του επιτρέπει να ξεπεράσει τον sample μέσα σε 40 παιχνίδια. Η διαφορά αυτή μειώνεται όσο περισσότεροι γύροι παιχτούν.

- Η στρατηγική που έχει η μεγαλύτερη επίπτωση στην επίδοση του πράκτορα φαίνεται να είναι η στρατηγική συμβολαίων, ενώ μετά έρχεται η στρατηγική αγοράς εντυπώσεων και τέλος η UCS που δεν είναι τόσο σημαντική όσο οι άλλες δύο (παρόλα αυτά μια αξιοπρεπής επίδοση είναι αναγκαία).
- Για μια καλή στρατηγική συμβολαίων είναι σημαντικότερη η ανάλυση της συμπεριφοράς και των άλλων παικτών, ενώ για την αγορά εντυπώσεων πρωτεύοντα ρόλο φαίνεται να παίζει η όσο γίνεται βέλτιστη αξιοποίηση των εκστρατειών που ο πράκτορας ήδη έχει.

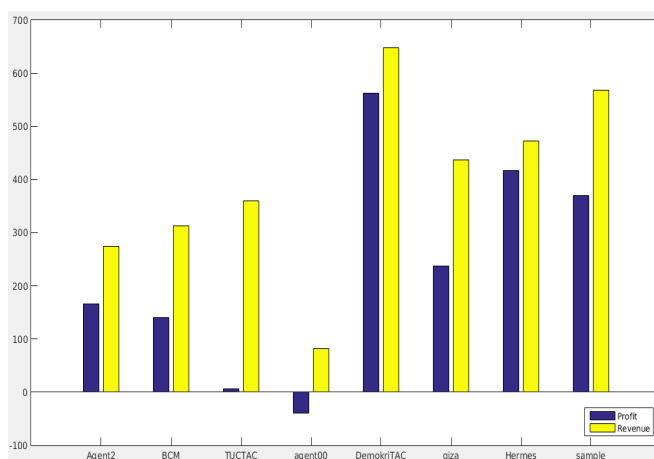
5.5 Αποτελέσματα του διαγωνισμού TAC AdX '16

Στα τελικά του διαγωνισμού συμμετείχε ο συνδυασμένος πράκτορας DemokriTAC που περιγράφεται στο προηγούμενο κεφάλαιο. Αρχικά παρουσιάζονται τα αποτελέσματά του σε σύγκριση με τον Hermes (στα διαγράμματα αναφέρεται ως NewAgent) και τους υπόλοιπους παλιούς πράκτορες και τέλος τα αποτελέσματα του τελικού. Για τα τελικά πειράματα χρησιμοποιήθηκαν πέντε σενάρια αυξανόμενης ανταγωνιστικότητας.

5.5.1 Αποτελέσματα ελέγχου του πράκτορα

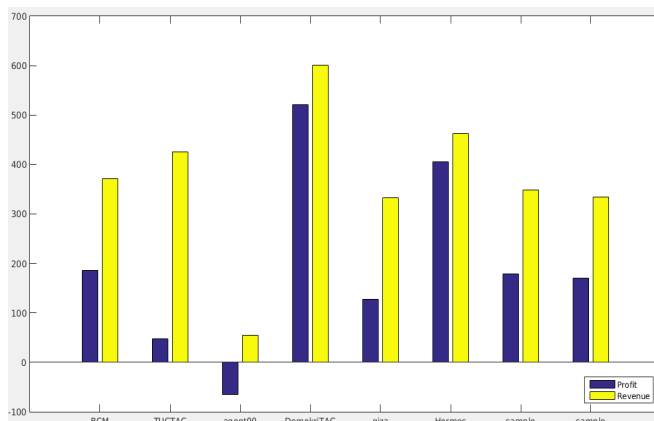


Σενάριο 1

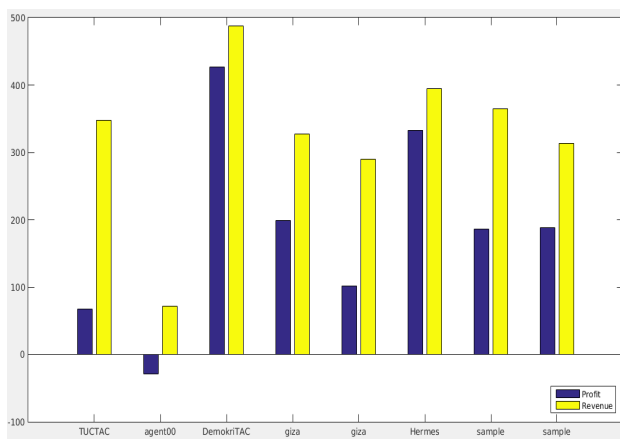


Σενάριο 2

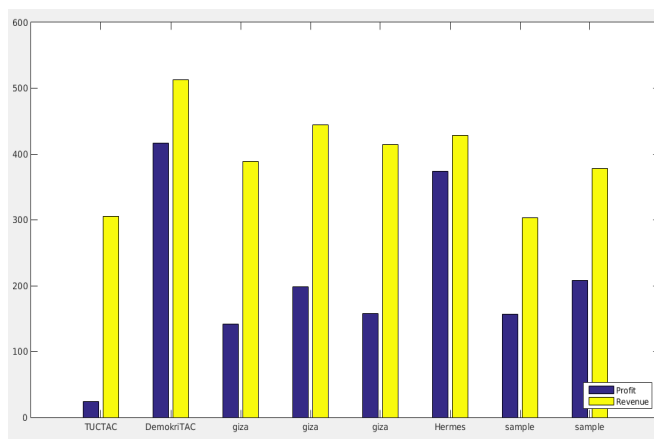
Σενάρια 1-5 από αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Το υψηλότερο σκορ το έχει ο συνδυασμένος πράκτορας και το δεύτερο υψηλότερο ο Hermes



Σενάριο 3



Σενάριο 4



Σενάριο 5

Παρατηρούμε ότι ο συνδυασμένος πράκτορας έχει πάντα καλύτερο σκορ αφού συνδυάζει σωστά την επιθετικότητα της ExperiencedCampaignStrategy με την προσαρμοστικότητα της στρατηγικής του Hermes στις δημοπρασίες συμβολαίων.

5.5.2 Αποτελέσματα του διαγωνισμού

Στα τελικά του διαγωνισμού συμμετείχαν επτά πράκτορες από άλλες ερευνητικές ομάδες ή ομάδες φοιτητών. Ο πράκτορας DemokriTAC κατάφερε να πετύχει το υψηλότερο σκορ με εμφανή διαφορά από τους υπόλοιπους όπως φαίνεται στον παρακάτω τελικό πίνακα:

Scores for finals2016 (game 132 - 171) at localhost

Competition started at 2016-07-10 12:00:00 and ended at 2016-07-10 21:55:00.

Position	Agent	Average Score	Games Played	Zero Games
1	DemokriTAC	22	40	1
2	LosCaparos	17	40	0
3	bob	15	40	0
4	AdXpertsAgent	13	40	0
5	edart	12	40	0
6	giza	11	40	0
7	baloothebot	4	40	12

Zero Games is the number of games that resulted in a score of zero (probably due to inactivity).

Table last updated 10 Jul 21:55:11

5.14

Αποτελέσματα TAC AdX '16

Πηγή: adxgame.org/localhost/history/competition/1/

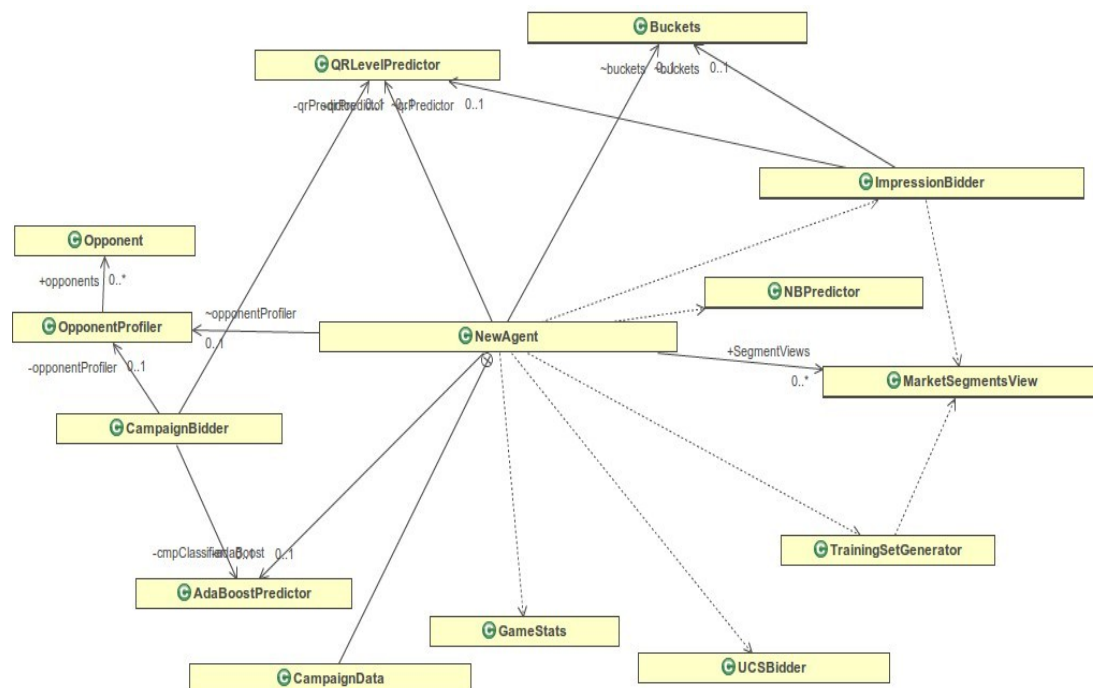
Παρατηρούμε ότι ο πράκτορας κατάφερε να αντεπεξέλθει επιτυχώς στις απαιτήσεις του διαγωνισμού κάτι που οφείλεται σε μεγάλο βαθμό στην ικανότητα ανάλυσης της ανταγωνιστικότητας του περιβάλλοντος καθώς και στη γρήγορη προσαρμογή του. Σε ότι αφορά το περιβάλλον του, παρότι λίγες πληροφορίες είναι προς το παρόν διαθέσιμες για τους άλλους πράκτορες, φάνηκε να είναι παρόμοιο των σεναρίων 4 και 5, καθώς οι περισσότεροι πράκτορες φάνηκαν να έχουν επιθετικές στρατηγικές, ενώ ο DemokriTAC φάνηκε να είναι πιο αμυντικός με αποτέλεσμα το μεγαλύτερο μέρος των κερδών του να προέρχεται από λίγες εκστρατείες με μεγάλο προϋπολογισμό. Εφαρμόζοντας t-test στα μη μηδενικά σκορ του παιχνιδιού (τα μηδενικά οφείλονται σε αποσύνδεση από τον server) βλέπουμε ότι η διαφορά του DemokriTAC από τους υπόλοιπους πράκτορες είναι στατιστικά σημαντική ($p < 0.05$):

Agents	p – vs. DemokriTAC
LosCaparos	0.042
bob	0.010
AdXpertsAgent	0.001
edart	0.004
giza	0.001
baloothebot	0.000

6

Τεχνικές λεπτομέρειες

Η γενική αρχιτεκτονική του συστήματος φαίνεται στο παρακάτω διάγραμμα κλάσης:



6.1 Διάγραμμα κλάσης του πράκτορα Hermes

Κεντρικά βρίσκεται η κλάση `NewAgent` που επεκτείνει την κλάση `Agent` του παιχνιδιού `adx` και είναι υπεύθυνη για την ανταλλαγή μηνυμάτων με τον `adx-server`. Οι τρεις βασικές κλάσεις που υλοποιούν τις τρεις βασικές στρατηγικές του πράκτορα είναι οι `CampaignBidder`, `ImpressionBidder` και `UCSBidder`. Στη συνέχεια υπάρχουν δύο κλάσεις συλλογής και διαχείρισης γενικών δεδομένων από το παιχνίδι, η `MarketSegmentsView`, η `CampaignData` και η `QRLevelPredictor`, ενώ υπάρχουν και πιο εξειδικευμένες τέτοιες κλάσεις όπως η `OpponentProfiler` για την `CampaignBidder` και η `Buckets` για την `ImpressionBidder`. Οι κλάσεις που αφορούν τις μεθόδους μηχανικής μάθησης είναι δύο ειδών: κλάσεις συλλογής δεδομένων εκπαίδευσης (`TrainingSetGenerator`, `GameStats`) και κλάσεις για την εξαγωγή προβλέψεων (`AdaBoostPredictor` και `NBPredictor`).

6.1 Λεπτομέρειες υλοποίησης

Σε αυτήν την ενότητα αναλύεται η υλοποίηση των βασικών κλάσεων και μεθόδων του πράκτορα `Hermes`.

Η κλάση `CampaignDataWrapper`:

Η κλάση αυτή είναι η πιο βασική και συχνά χρησιμοποιούμενη κλάση του πράκτορα και κρατάει δεδομένα για μία εκστρατεία όπως:

- οι απαιτούμενες εντυπώσεις
- οι εντυπώσεις του τμήματος αγοράς της
- οι μέρες έναρξης και λήξης
- το τμήμα αγοράς της
- ο συντελεστής για εντυπώσεις σε βίντεο και κινητές συσκευές
- ο προϋπολογισμός
- τον γεννήτορα των σετ δεδομένων για παλιές εκστρατείες

Η κλάση αυτή επεκτείνει την κλάση `CampaignData` του `adx-server`. Επίσης στο τέλος κάθε εκστρατείας το αντικείμενο `TrainingSetGenerator` που περιέχει καταγράφει κάποια στατιστικά της στο ιστορικό της μεθόδου `AdaBoost` (`TrainingSet.arff`).

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

`getMarketSegmentsView`: Η μέθοδος αυτή βρίσκει ποίο στιγμιότυπο `MarketSegmentsView` από αυτά που αντιστοιχούν στα τμήματα αγοράς που αφορά η εκστρατεία έχει τη μεγαλύτερη συσσώρευση και το επιστρέφει.

targetImpsToGo: Επιστρέφει πόσες εντυπώσεις απομένουν μέχρι τον επιθυμητό αριθμό εντυπώσεων που έχει καθοριστεί για την εκστρατεία.

getERR: Αν η εκστρατεία έχει ολοκληρωθεί υπολογίζει και επιστρέφει τον βαθμό συμπλήρωσής της.

Η κλάση MarketSegmentsView:

Υπάρχει μία τέτοια κλάση για κάθε τμήμα της αγοράς: MALE, FEMALE, YOUNG, OLD, LOW INCOME, HIGH INCOME. Κάθε φορά που μία νέα εκστρατεία ανατίθεται σε κάποιον πράκτορα η κλάση αυτή ενημερώνεται ώστε να υπάρχει μια ακριβής εικόνα των ζητούμενων εντυπώσεων σε κάθε τμήμα της αγοράς. Αν για παράδειγμα ανατεθεί μια εκστρατεία για MALE τότε ενημερώνεται μόνο το στιγμιότυπο που αφορά τους αρσενικούς χρήστες. Αν όμως έρθει μια για OLD, MALE, LOW INCOME τότε ενημερώνονται και τα τρία αντίστοιχα στιγμιότυπα.

Οι εντυπώσεις ανά ημέρα κρατούνται σε μια δομή ArrayList μεγέθους 60, όπου ο δείκτης της αντιστοιχεί στις ζητούμενες εντυπώσεις για εκείνη τη μέρα. Επίσης υπάρχει άλλος ένας τέτοιος πίνακας που κρατά όσες εντυπώσεις ζητούνται από τον ίδιο τον πράκτορα τη μέρα εκείνη.

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

updateMarketSegmentsView: Η μέθοδος αυτή παίρνει τέσσερα ορίσματα: την ημέρα έναρξης της νέας εκστρατείας, τη διάρκεια, τις ζητούμενες εντυπώσεις και το αν η εκστρατεία έχει κερδηθεί από τον ίδιο τον πράκτορα ή όχι. Στη συνέχεια ανανεώνει τα κελιά των πινάκων που αντιστοιχούν στις μέρες που διαρκεί η εκστρατεία προσθέτοντας στην ήδη υπάρχουσα τιμή κάθε κελιού τις μέσες ζητούμενες εντυπώσεις ανά ημέρα.

getCongestion: Η μέθοδος αυτή παίρνει δύο ορίσματα: την τρέχουσα ημέρα και την ημέρα λήξης μιας εκστρατείας. Στη συνέχεια για κάθε ενδιάμεση μέρα υπολογίζει τον λόγο των ζητούμενων εντυπώσεων προς τις συνολικά διαθέσιμες εντυπώσεις για το τμήμα της αγοράς τη συγκεκριμένη μέρα. Τέλος επιστρέφει τον μέσο όρο τους, ο οποίος αποτελεί και ένα μέτρο της δυσκολίας ολοκλήρωσης μιας εκστρατείας.

getAvgMyCampaigns: Η μέθοδος αυτή παίρνει δύο ορίσματα: την τρέχουσα ημέρα και την ημέρα λήξης μιας εκστρατείας και επιστρέφει τον μέσο αριθμό εντυπώσεων που διεκδικεί ο ίδιος ο πράκτορας τις ενδιάμεσες μέρες.

Η κλάση Opponent:

Δημιουργούνται συνολικά 8 τέτοιες κλάσεις μια για κάθε αντίπαλο (7) και μια για τον πράκτορά μας, οι οποίες περιλαμβάνουν τις εκστρατείες που του αντιστοιχούν.

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

addCampaign: Προσθήκη μιας νέας εκστρατείας που έχει κερδίσει ο συγκεκριμένος αντίπαλος στη λίστα με τις εκστρατείες του.

getCampaignFrequency: Η μέθοδος παίρνει ως όρισμα τη μέρα και επιστρέφει τη συχνότητα ανάληψη εκστρατειών του πράκτορα μέχρι εκείνη τη μέρα διαιρώντας τον αριθμό των εκστρατειών του με τον συνολικό αριθμό των εκστρατειών που έχουν εκδοθεί μέχρι εκείνη τη μέρα.

Η κλάση OpponentProfiler:

Η κλάση αυτή κρατά όλα τα στιγμιότυπα της κλάσης Opponent σε μια λίστα και είναι υπεύθυνη για την ανάλυση της ανταγωνιστικότητας στο τρέχον παιχνίδι.

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

calculateCompetitiveness: Η μέθοδος αυτή βρίσκει πόσοι από τους πράκτορες έχουν πάρει πάνω από το 1/8 των συνολικών εκστρατειών, δηλαδή πάνω από τις εκστρατείες που θα έπαιρναν άμα ήταν όλοι ίσοι. Το πλήθος αυτών των πρακτόρων επιστρέφεται σαν μέτρο της ανταγωνιστικότητας.

getFreqVar: Εδώ υπολογίζεται η συχνότητα ανάληψης εκστρατειών για όλους τους παίκτες και στη συνέχεια υπολογίζεται η διασπορά της η οποία χρησιμοποιείται ως χαρακτηριστικό του συγκεκριμένου παιχνιδιού.

Η κλάση QRLevelPredictor

Η QRLevelPredictor είναι υπεύθυνη για τη συλλογή δεδομένων από προηγούμενα παιχνίδια και για τη διεξαγωγή προβλέψεων σχετικών με αυτά. Τα δεδομένα που κρατάει είναι:

- Ο μέσος βαθμός ποιότητας και ο προβλεπόμενος βαθμός ποιότητας για αυτό το προηγούμενο παιχνίδι
- Η ανταγωνιστικότητα και η προβλεπόμενη ανταγωνιστικότητα
- Το συνολικό κέρδος του πράκτορα για όλα τα προηγούμενα παιχνίδια
- Τα ποσοστά υπέρ - επιτυχημένων και αποτυχημένων εκστρατειών
- Ο αριθμός των εκστρατειών για ένα παιχνίδι
- Ο μέσος όρος πονταρισμάτων για εκστρατείες για ένα παιχνίδι
- Ο μέσος όρος των πονταρισμάτων UCS, των επιπέδων UCS και ο λόγος μεταξύ των μη στοχευμένων προς τις συνολικές αποκτηθείσες εντυπώσεις

Τα παραπάνω δεδομένα αποθηκεύονται στο αρχείο Qrhistory.xml, η μορφή του οποίου είναι η εξής:

```
<?xml version="1.0" encoding="UTF-8"?>
```

<qrLevels>

```
<qr qrLevel="1.1891939578437243"
  predictedQrLevel="1.2250570422071547"
  competitiveness="2.0"
  predCompetitiveness="1.01"
  totalProfit="37.7804"
  failureRate="0.2"
  successRate="0.6"
  impressionBids="0.1190165265385793"
  ucsBids="0.1315622845124092"
  ucsLevels="0.7214707187499999"
  ucsRatio="0.0"
  numberOfCampaigns="30" />
```

</qrLevels>

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

writeXML: Η μέθοδος που είναι υπεύθυνη για την εγγραφή στο QRhistory.xml. Η μέθοδος ελέγχει αν το xml υπάρχει και αν όχι το δημιουργεί προσθέτοντάς του το ζητούμενο στοιχείο, αλλιώς προσθέτει μόνο το στοιχείο.

predictCompetitiveness: Η μέθοδος αυτή μαζεύει όλα τα attributes με όνομα “competitiveness” σε έναν πίνακα και όλα τα attributes “predCompetitiveness” σε έναν άλλον και εφαρμόζει στους δύο πίνακες τη μέθοδο απλής εκθετικής εξομάλυνσης επιστρέφοντας τη νέα πρόβλεψη για την ανταγωνιστικότητα. Το ίδιο κάνει και η μέθοδος MakePrediction για τον βαθμό ποιότητας (Attributes: “qrLevel” και “predictedQrLevel”)

updateBudget: Η μέθοδος αυτή εξάγει το τελευταίο attribute με όνομα “TotalProfit” και του προσθέτει το τρέχον κέρδος επιστρέφοντας το νέο συνολικό κέρδος του πράκτορα.

Η κλάση TrainingSetGenerator:

Η κλάση αυτή δημιουργεί το training set που χρησιμοποιεί η μέθοδος AdaBoost αποθηκεύοντας τα δεδομένα σε ένα αρχείο ARFF (TrainingSet.arff) με την παρακάτω μορφή:

@relation MyRelation

@attribute congestion numeric

@attribute budget numeric

@attribute reqImps numeric

@attribute dayStart numeric

@attribute duration numeric

@attribute qr numeric

@attribute mycmps numeric

@attribute comp numeric

@attribute class {-1,1}

@data

0.0,2.099,20980,4,9,1.0,0.0,1.744028,1

0.0,1.334,1159,13,2,1.151542,0.0,1.744028,1

1.0,2.969,2970,2,2,1.0,0.0,1.083089,-1

Τα παραπάνω δεδομένα ανανεώνονται με κάθε νέα εκστρατεία που αναλαμβάνει ο πράκτορας.

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

printToLog: Η μέθοδος αυτή ορίζει το αρχείο TrainingSet.arff σαν log file. Σε περίπτωση που αυτό δεν υπάρχει το δημιουργεί και τυπώνει σε αυτό το arff header με τις κατάλληλες μεθόδους της βιβλιοθήκης weka. Και στις δύο περιπτώσεις τυπώνει στο αρχείο τα ζητούμενα δεδομένα.

calcLabel: Η μέθοδος παίρνει ως είσοδο ένα σημείο δεδομένων που χαρακτηρίζει μια ολοκληρωμένη εκστρατεία και επιστρέφει το class attribute που θα έχει ως δεδομένο εκπαίδευσης. Αν η εκστρατεία έχει κέρδος < 2 και βαθμό συμπλήρωσης < 1 τότε χαρακτηρίζεται ως αποτυχημένη (class = -1) ενώ αλλιώς ως επιτυχημένη (class = 1).

Η κλάση AdaBoostPredictor:

Η κλάση έχει σαν στόχο να προβλέψει αν μια εκστρατεία για την οποία πρόκειται να ποντάρει ο πράκτορας (έχοντας επιλέξει ήδη μια αρχική τιμή πονταρίσματος) θα έχει επιτυχία ή αν ο πράκτορας θα πρέπει να αλλάξει το ποντάρισμά του.

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

makePrediction: Η μέθοδος αυτή δημιουργεί ένα μοντέλο με τη μέθοδο AdaBoostM1 της βιβλιοθήκης weka και με βάση τα δεδομένα εκπαίδευσης στο TrainingSet.arff και με είσοδο

το data-point που παίρνει σαν όρισμα επιστρέφει μια πρόβλεψη αν η εκστρατεία είναι κατάλληλη ή όχι.

Η κλάση GameStats:

Η κλάση αυτή δημιουργεί το training set που χρησιμοποιεί η μέθοδος Naive Bayes για την κατηγοριοποίηση του περιβάλλοντος στις δημοπρασίες εντυπώσεων, αποθηκεύοντας τα δεδομένα σε ένα αρχείο ARFF (GameStats.arff) με την παρακάτω μορφή:

```
@relation MyRelation2
```

```
@attribute competitiveness numeric
```

```
@attribute cmpFreqVariance numeric
```

```
@attribute avgPricePerImp numeric
```

```
@attribute avgQuality numeric
```

```
@attribute revenue numeric
```

```
@attribute profitMargin numeric
```

```
@attribute class {1,2,3,4,5,6,7}
```

```
@data
```

```
1.0,0.355932,5.0E-6,1.111581,5.549831,5.549831,1
```

```
2.0,0.288136,3.0E-6,1.161475,32.9122,32.9122,1
```

```
0.0,0.186441,1.0E-5,0.803769,2.260701,2.260701,1
```

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

printNewStats: Η μέθοδος αυτή ορίζει το αρχείο GameStats.arff σαν log file. Σε περίπτωση που αυτό δεν υπάρχει το δημιουργεί και τυπώνει σε αυτό το arff header με τις κατάλληλες μεθόδους της βιβλιοθήκης weka. Και στις δύο περιπτώσεις τυπώνει στο αρχείο τα ζητούμενα δεδομένα.

Η κλάση NBPredictor:

Η κλάση έχει σαν στόχο να χαρακτηρίσει το περιβάλλον στις δημοπρασίες εντυπώσεων με βαθμό αυξανόμενης επιθετικότητας από 1 έως 8, ο οποίος θα χρησιμοποιηθεί στη διαμόρφωση των πονταρισμάτων του πράκτορα.

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

makePrediction: Η μέθοδος αυτή δημιουργεί ένα μοντέλο με τη μέθοδο Naive Bayes της βιβλιοθήκης weka και με βάση τα δεδομένα εκπαίδευσης στο GameStats.arff και με είσοδο το data-point που παίρνει σαν όρισμα επιστρέφει μια πρόβλεψη για την επιθετικότητα του περιβάλλοντος που βρίσκεται ο πράκτορας.

Η κλάση Buckets:

Η κλάση αυτή βοηθά τον πράκτορα να συσχετίσει το κέρδος που θα του αποφέρει η επίτευξη κάποιου βαθμού ποιότητας σε συγκεκριμένη μέρα, σε περιβάλλον συγκεκριμένης ανταγωνιστικότητας. Υπολογίζοντας το μέσο κέρδος για πολλά διαφορετικά παιχνίδια, η μέθοδος αυτή δημιουργεί μια προσέγγιση της συνάρτησης που ικανοποιεί την παραπάνω σχέση. Η τιμές της συνάρτησης αυτής αποθηκεύονται στο αρχείο Buckets.xml με την παρακάτω μορφή:

```
<buckets>
<comp-bucket0>
</comp-bucket0>
<comp-bucket1>
<qr-bucket0>
.....
</qr-bucket0>
.....
<qr-bucket6>
  <day0 profit="0.0" games="0" />
  .....
  <day7 profit="42.3152" games="285" />
  .....
  <day26 profit="64.633" games="210" />
  <day27 profit="64.633" games="203" />
  .....
</qr-bucket6>
.....
<qr-bucket10>
.....
</qr-bucket10>
```

.....

```
<comp-bucket8>  
</comp-bucket8>  
</buckets>
```

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

writeXML: Η μέθοδος έχει ως είσοδο τις τιμές των τριών ορισμάτων της παραπάνω συνάρτησης: ανταγωνιστικότητα, και βαθμοί ποιότητας για κάθε μέρα καθώς και την τιμή του συνολικού κέρδους του πράκτορα, την οποία τελικά γράφει στο κατάλληλο bucket του xml.

getExpectedTotalProfit: Όταν ο πράκτορας θέλει να συμβουλευτεί την συνάρτηση συνολικού αναμενόμενου κέρδους, περνάει σαν ορίσματα στην συνάρτηση αυτή τα χαρακτηριστικά της τρέχουσας εκστρατείας μαζί με τις εντυπώσεις που στοχεύει να πάρει και του επιστρέφεται η τιμή που εξάγεται από το αρχείο με βάση αυτά τα χαρακτηριστικά.

getDerivativeValue: Εδώ υπολογίζεται προσεγγιστικά η τιμή της παραγώγου της συνάρτησης συνολικού αναμενόμενου κέρδους ως προς τις εντυπώσεις. Σαν ορίσματα περνιούνται τα χαρακτηριστικά της τρέχουσας εκστρατείας. Με βάση αυτά εξάγεται από το αρχείο η τιμή του συνολικού αναμενόμενου κέρδους με τη βοήθεια της *getExpectedTotalProfit*. Στη συνέχεια με χρήση της ίδιας συνάρτησης υπολογίζεται το συνολικό αναμενόμενο κέρδος για το επόμενο qf-bucket (που αντιστοιχεί σε μεγαλύτερο qf) διατηρώντας την ανταγωνιστικότητα και τη μέρα σταθερές. Αυτά τα δύο αφαιρούνται και στη συνέχεια διαιρούνται με τις εντυπώσεις που χρειάζονται για να βρεθούμε από το αρχικό qfBucket στο επόμενο.

Η κλάση UCSBidder:

Η κλάση αυτή είναι υπεύθυνη για τους υπολογισμούς που περιγράφονται στην ενότητα 4.4. Τον σημαντικότερο ρόλο παίζει η μέθοδος *calcBid* η οποία εκτελεί τους παραπάνω υπολογισμούς με βάση τα δεδομένα που περνιούνται στην κλάση από την κεντρική κλάση *NewAgent* και επιστρέφει το ποντάρισμα που θα στείλει ο πράκτορας για τη δημοπρασία UCS.

Η κλάση ImpressionBidder:

Εδώ γίνεται ο καθορισμός της τιμής του πονταρίσματος μέσα από την κλάση *calcBid*. Λογώ της σημασίας αυτής της κλάσης παρατίθεται παρακάτω σχολιασμένος ο κώδικάς της:

```
double calcBid(){
```

```

//οι απαιτούμενες εντυπώσεις
long reqimps = cCampaign.reachImps;
//το ελάχιστο επιτρεπτό qr
double custom_qr = 0.9;
//οι εντυπώσεις που στοχεύουμε να πάρουμε
double targimps;
//οι υποψήφιες απαιτούμενες εντυπώσεις μία για κάθε qr-bucket
double[] cand_targimps = new double[12];
//τα αναμενόμενα κέρδη για κάθε qr-bucket
double[] cand_profits = new double[12];
//τα μέγιστα αναμενόμενα κέρδη από τα cand_profits και ο
//δείκτης και το qr που τους αντιστοιχούν
double max_profits = 0.0;
int max_profits_idx = 1;
double acc_qr = 1.2;
//η ανταγωνιστικότητα του τρέχοντος παιχνιδιού
double competitiveness = qrPredictor.predictCompetitiveness();
//οι σταθερές για τον υπολογισμό του ERR
double a = 4.08577;
double b = 3.08577;
//το πιο κορεσμένο τμήμα της αγοράς για τη εκστρατεία
MarketSegmentsView maxCmpSeg =
    cCampaign.getMarketSegmentsView();
//πρόβλεψη τιμής με βάση τον τύπο της ενότητας 4.2
price = maxCmpSeg.congestion[Math.min(day+1,59)] *
    (cCampaign.budget)/(reqimps);
//ο αριθμός εντυπώσεων που μεγιστοποιεί το κέρδος
targimps = (2*a/reqimps)*(b + Math.sqrt(Math.pow(b,2) - 1- b
+ (2*cCampaign.budget)/(reqimps*price)));
// αν entered==false σημαίνει ότι έγινε λάθος κατά την ανάγνωση
//από το Buckets.xml
boolean entered = true;

//εύρεση του αριθμού εντυπώσεων που να μεγιστοποιεί το συνολικό
//αναμενόμενο κέρδος με τη βοήθεια της συνάρτησης G (Buckets.xml)

```

```

for (int i=1; i<=10; i++){
    try {
        //για όλα τα qr-buckets υπολογίζω τις εντυπώσεις για
        //μέγιστο κέρδος με βάση τον τύπο στην 4.2
        cand_targimps[i] = targimps +
        buckets.getDerivativeValue(price,
        dayBiddingFor, competitiveness, targimps, reqimps);
        //υπολογίζω το qr που θα είχα αν έπαιρνα τις targimps
        acc_qr = 0.6*((2/a)*(Math.atan(a*targimps/reqimps - b)
        - Math.atan(-b))) + 0.4*curr_qr;
        //υπολογίζω τα αναμενόμενα κέρδη για το τρέχον qr-bucket
        cand_profits[i] = buckets.getExpectedTotalProfit(acc_qr,
        dayBiddingFor, competitiveness);
        //επιλογή των μέγιστων αναμενόμενων κερδών για όλα τα
        //qr-buckets
        if (cand_profits[i]>max_profits){
            max_profits=cand_profits[i];
            max_profits_idx = i;
        }
        } catch (DataConversionException e) {
            e.printStackTrace();
            entered = false;
        }
    }

    //ανανέωση των targimps ώστε τώρα να μεγιστοποιείται το
    //ΣΥΝΟΛΙΚΟ αναμενόμενο κέρδος
    if (entered && cand_targimps[max_profits_idx]>targimps)
        targimps = cand_targimps[max_profits_idx];

    //σε περίπτωση που γίνει κάποια εξαίρεση διαλέγω έναν
    //αριθμό εντυπώσεων που μου εγγυάται ότι δε θα πέσω
    //κάτω από το επιθυμητό qr

```



```

if ((2/a)*(Math.atan(a*targimps/reqimps - b)
        - Math.atan(-b)) < 1.0)
    targimps = Math.round(((Math.tan((custom_qr - 0.4*curr_qr +
0.6*(3/4.09)*Math.atan(-3.09))/(0.6*2/4.09)) + 3.09)/4.09)*reqimps);

cCampaign.targetImps = targimps;

double generalExpectedProfit =
(2/a)*(Math.atan(a*targimps/reqimps - b) -
        Math.atan(-b))*cCampaign.budget;

//η βασική τιμή καθορίζεται με βάση την αξία κάθε εντύπωσης για
//τον πράκτορα
double newprice = generalExpectedProfit/targimps;

//τα παρακάτω μεγέθη είναι χρήσιμα σε περίπτωση που
//είμαι “εκτός προγράμματος”
double expectedTargImps = cCampaign.stats.getTargetedImps()*
        (cCampaign.dayEnd-cCampaign.dayStart)/(day -
        cCampaign.dayStart+1);

double currentExpectedProfit =
(2/a)*(Math.atan(a*(cCampaign.stats.getTargetedImps() +
        expectedTargImps)/reqimps - b) -
        Math.atan(-b))*cCampaign.budget;

//σε περίπτωση που είμαι “εκτός προγράμματος” προσαρμόζω
//την τιμή στο με βάση τον τύπο της 4.3

double deltaProfit = generalExpectedProfit -
        currentExpectedProfit;

double expCoeff= Math.exp((day - cCampaign.dayStart)/
        (cCampaign.dayEnd-cCampaign.dayStart));

```

```

if (expCoeff<1) expCoeff = 1.0;

if (cCampaign.targetImpsTogo() > targimps *
1.0*(cCampaign.dayEnd-day)/(cCampaign.dayEnd-
cCampaign.dayStart))
    newprice = expCoeff*(newprice +
        deltaProfit/cCampaign.targetImpsTogo());

double rbid;
rbid= newprice*1000;

//το price_idx καθορίζεται από την κλάση NewAgent που
//καλεί τον NBPredictor

return price_idx*rbid;
}

```

Η κλάση CampaignBidder:

Εδώ γίνεται ο καθορισμός της τιμής του πονταρίσματος μέσα από την κλάση calcBid. Λογώ της σημασίας αυτής της κλάσης παρατίθεται παρακάτω σχολιασμένος ο κώδικάς της:

```

long calcBid(){
    //οι απαιτούμενες εντυπώσεις της υποψήφιας εκστρατείας
    long cmpimps = pendingCampaign.reachImps;
    //η πιθανότητα ένας χρήστης να προέρχεται από το τμήμα
    //αγοράς της εκστρατείας
    double segProb = 0;
    Set<MarketSegment> m_seg = pendingCampaign.targetSegment;
    //υπολογισμός του αναμενόμενου αριθμού χρηστών για αυτό
    //το τμήμα αγοράς
    for(MarketSegment ms : m_seg){
        for(MarketSegment currSeg:
            SegmentProbabilities.keySet()){
            if((segProb == 0) && (ms == currSeg)) segProb
            = SegmentProbabilities.get(currSeg);
        }
    }
}

```

```

        if((segProb != 0) && (ms == currSeg)) segProb
            *= SegmentProbabilities.get(currSeg);
    }
}
pendingCampaign.setSegmentImps(10000*segProb);

//εύρεση του πιο κορεσμένου τμήματος της αγοράς
MarketSegmentsView maxCmpSeg =
    pendingCampaign.getMarketSegmentsView();

//εύρεση καταλληλότερου bid (ενότητα 3.2)
double budget;
double maxBudget = 1.0*cmpimps*curr_qr - 1;
double minBudget = 1.0000001*cmpimps*0.1/curr_qr;
double comp = qrPredictor.predictCompetitiveness();
if (day>=15) comp =
(1.0*day/60)*opponentProfiler.calculateCompetitiveness(day,
true) + (60-1.0*day)/60;

if(comp<2) budget = minBudget;
else if ((0.36/7)*maxBudget > (1/comp)*minBudget)
    budget = maxBudget;
else budget = minBudget;

//εύρεση αναμενόμενης τιμής ανά εντύπωση με βάση την 3.2

double totalBudget = 0.0;
double totalImps = 0;

for(CampaignDataWrapper c: myCampaigns.values()){
    if(c.targetSegmentViews.contains(maxCmpSeg)){
        totalBudget += c.budget;
        totalImps += c.impsTogo();
    }
}

```

```

}

    double expectedPrice = maxCmpSeg.congestion[day+2] *
                            (budget + totalBudget)/
                            (pendingCampaign.reachImps + totalImps);

    //είναι η εκστρατεία κατάλληλη?
    boolean feasibility = true;

    if (expectedPrice*pendingCampaign.reachImps >= 2*budget
    || (pendingCampaign.dayStart<=5 && pendingCampaign.dayEnd<=6) )
        feasibility = false;

    //φιλτράρισμα με τη μέθοδο AdaBoostM1
    double classifierPrediction = 1;
    try {
        classifierPrediction =
cmpClassifier.makePrediction(// data-instance );
    } catch (Exception e) {
        e.printStackTrace();
    }
    if (classifierPrediction<1) feasibility = false;

    //το τελικό ποντάρισμα

    long cmpBidMillis = 0;
    //αν δεν έχω πάρει εκστρατεία εδώ και 5 μέρες
    //ποντάρω τον μέγιστο προϋπολογισμό
    if ((day - lastCmp>=5) && (curr_qr<=1.3) && (day>10))
        cmpBidMillis= (long) Math.ceil(maxBudget);
    else if (feasibility)
        cmpBidMillis= (long) Math.ceil(budget);
    else cmpBidMillis= (long) Math.ceil(maxBudget);

    return cmpBidMillis;
}

```

Η κλάση NewAgent:

Η κλάση αυτή αποτελεί την κεντρική κλάση της υλοποίησης και είναι υπεύθυνη για τον χειρισμό των εισερχόμενων μηνυμάτων από τον server καθώς και για την αποστολή των πονταρισμάτων του πράκτορα.

Οι σημαντικότερες μέθοδοι αυτής της κλάσης είναι:

handleCampaignOpportunityMessage: Εδώ ο πράκτορας χειρίζεται το `campaignOpportunityMessage` το οποίο περιέχει πληροφορίες για μια επικείμενη εκστρατεία την οποία ο πράκτορας θα διεκδικήσει. Οι πληροφορίες για την εκστρατεία μεταβιβάζονται στη συνέχεια σε ένα στιγμιότυπο της `CampaignBidder` και καλείται η `calcBid` η οποία επιστρέφει το ποντάρισμα και αυτό στέλνεται στον server. Μαζί με αυτό το μήνυμα ο πράκτορας στέλνει και ένα ακόμη (*riggy-backed*) το οποίο περιέχει το ποντάρισμά του για τη δημοπρασία UCS. Το ποντάρισμα αυτό καθορίζεται από την `calcBid` ενός στιγμιότυπου της `UCSBidder`.

handleCampaignInitialOpportunityMessage: Εδώ γίνεται η ανάθεση της πρώτης εκστρατείας του πράκτορα η οποία είναι υποχρεωτική και δε χρειάζεται να κερδιθεί σε δημοπρασία Ένα στιγμιότυπο `CampaignDataWrapper` με τις πληροφορίες της αποθηκεύεται στο `HashMap MyCampaigns` όπου ο πράκτορας αποθηκεύει όλες τις εκστρατείες που έχει κερδίσει.

handleAdnetworkDailyNotification: Αυτή η μέθοδος χειρίζεται το μήνυμα που πληροφορεί τους πράκτορες για το ποιος κέρδισε την εκστρατεία που ήταν σε δημοπρασία την προηγούμενη μέρα. Ανάλογα με το ποιος την κέρδισε η εκστρατεία προστίθεται στη λίστα του αντίστοιχου `Opponent` στην κλάση `OpponentProfiler`. Αν κερδιθεί από τον ίδιο τον πράκτορα προστίθεται επίσης στην `MyCampaigns`. Επίσης σε αυτό μήνυμα, ο πράκτορας μαθαίνει τον τρέχοντα βαθμό ποιότητας του. Τέλος η ενημερώνεται και το στιγμιότυπο της `marketSegmentsView` που παρακολουθεί τα τμήματα της αγοράς με τις νέες εντυπώσεις που ζητούνται.

sendBidAndAds: Η `sendBidAndAds` στέλνει το μήνυμα που περιέχει τα πονταρίσματα του πράκτορα για τη δημοπρασία εντυπώσεων. Οι οδηγίες για το ποντάρισμα δίνονται στον server μέσω της δομής `BidBundle` που είναι ένα σύνολο από `adxQueries` των τρέχοντων εκστρατειών. Κάθε εκστρατεία περιέχει στη δομή της (`CampaignDataWrapper`) όλες τις δυνατές `Queries` της μορφής (`PublisherName`, `MarketSegment`, `DeviceType`, `MediaType`) και σε κάθε μία τέτοια αντιστοιχίζεται μία τιμή πονταρίσματος. Ο πράκτορας χειρίζεται ομοιόμορφα όλες τις `Queries` (ποντάρει το ίδιο) εκτός από αυτές που αντιστοιχούν σε `video` ή/και `mobile` οι οποίες πολλαπλασιάζονται από τους αντίστοιχους συντελεστές (`mobileCoef` & `videoCoef`). Η τιμή του πονταρίσματος καθορίζεται από την `calcBid` της `ImpressionBidder` και

τη βοήθεια του `price_idx` που καθορίζεται με βάση την επιθετικότητα των άλλων πρακτόρων, όπως αυτή έχει κατηγοριοποιηθεί από την μέθοδο `Naive Bayes` της `NBPredictor`. Επίσης τίθενται όρια στα πονταρίσματα σε ότι αφορά το κόστος για τον πράκτορα και τον συνολικό αριθμό των εντυπώσεων. Τέλος, μία ακόμη λειτουργία αυτής της κλάσης είναι η αποστολή στατιστικών για τις εκστρατείες που έχουν τελειώσει στην κλάση `trainingSetGenerator`.

`handleBankStatus`: Η μέθοδος αυτή ενημερώνει τον πράκτορα για το σκορ. Επειδή το μήνυμα που την αφορά είναι το τελευταίο που στέλνεται σε μια μέρα, στην μέθοδος αυτή γίνεται και η ενημέρωση των κλάσεων και εν συνεχεία των αρχείων που κρατούν δεδομένα για το παιχνίδι που μόλις τελείωσε συνολικά: `GameStats`, `QrLevelPredictor`, `Buckets`.

6.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Η γλώσσα στην οποία προγραμματίστηκε ο πράκτορας είναι η `Java SE 8 (32-bit)`, και για την ανάπτυξή του έγινε χρήση του περιβάλλοντος `Eclipse`. Χρησιμοποιήθηκαν επίσης η βιβλιοθήκη μηχανικής μάθησης `weka [14][32]` και η βιβλιοθήκη `jdom2` για την διαχείριση των αρχείων `xml`. Η επεξεργασία και ανάλυση των πειραματικών αποτελεσμάτων έγινε στο εργαλείο `MATLAB r2015a`.

Η ανάπτυξη και ο έλεγχος αυτής της εργασίας έγινε σε περιβάλλον `unix` με χρήση `bash scripting` και οι οδηγίες εγκατάστασης αφορούν περιβάλλοντα που υποστηρίζουν την εκτέλεση τέτοιων `scripts`. Σε περιβάλλον `windows` μπορούν να χρησιμοποιηθούν τα αντίστοιχα βήματα με τη βοήθεια του προγράμματος `cygwin`.

Βασικό ρόλο παίζουν οι φάκελοι `adx-server` και `adx-agnet`. Ο πρώτος περιέχει τη νεότερη έκδοση του `server` του παιχνιδιού `adx` για χρήση στην τοποθεσία `localhost`. Ο `server` τρέχει με την εκτέλεση του `runServer.sh`. Αποκτώντας πρόσβαση στον `server` μέσω `localhost` από κάποιον `browser` μπορείτε να κάνετε εγγραφή του πράκτορα με ένα συγκεκριμένο όνομα και συγκεκριμένο `password`.

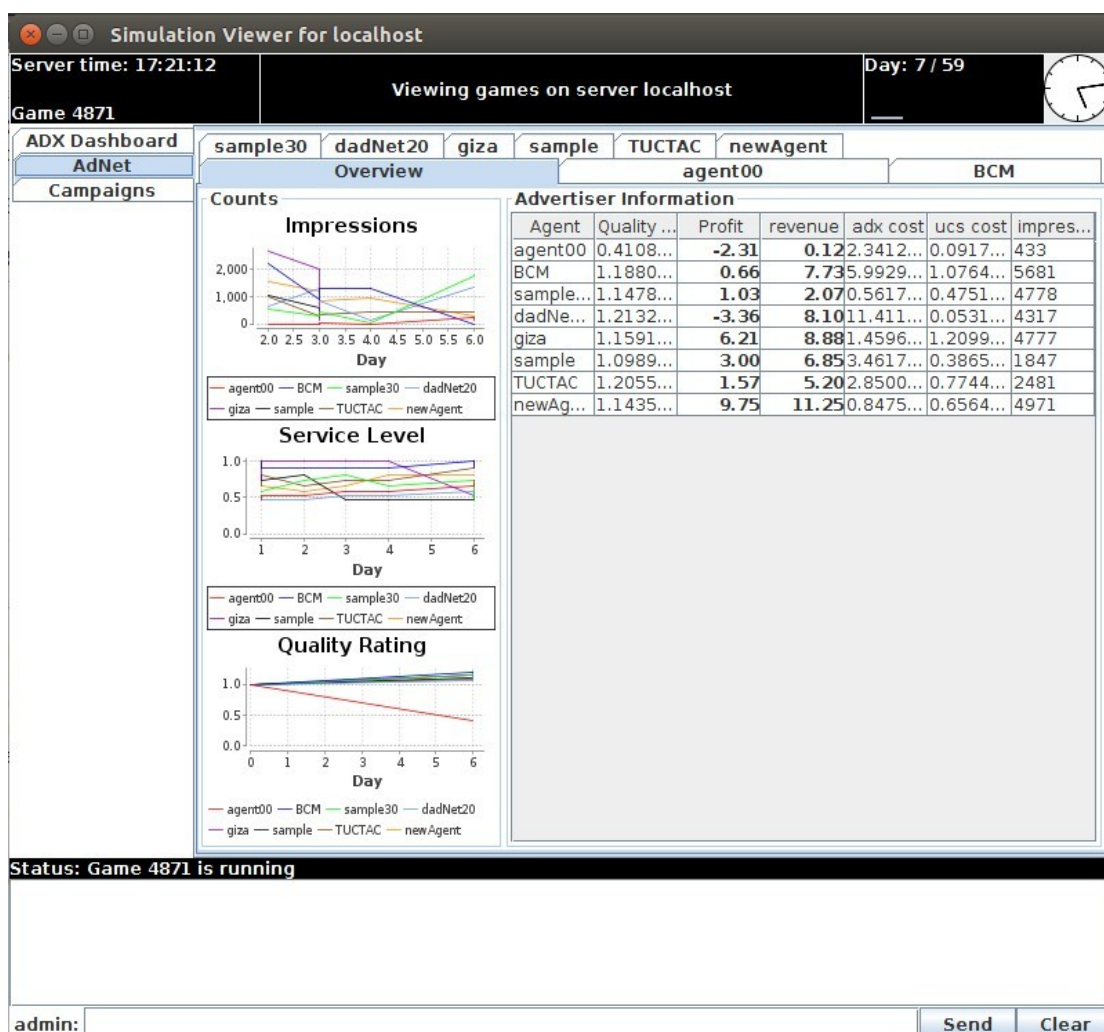
Ο `adx-agnet` περιέχει έναν φάκελο που λέγεται `lib`. Εκεί πρέπει να τοποθετηθούν τα αρχεία `jar` των πρακτόρων που θα συμμετάσχουν στην προσομοίωση. Επίσης για κάθε πράκτορα πρέπει να υπάρχει ένα `configuration file` που να περιέχει

- Τα `agentName` και `agentPassword` που χρησιμοποιήθηκαν για την εγγραφή του πράκτορα στον `server`
- Την τοποθεσία σύνδεσης (πρέπει να είναι `localhost`).
- Το όνομα της κύριας κλάσης του πράκτορα: πεδίο `agentImpl`

Για την εκτέλεση ελεγχόμενων προσομοιώσεων με διαφορετικούς πράκτορες έχει δημιουργηθεί το `runSimulation.sh`. Η εντολή που τρέχει έναν πράκτορα είναι της μορφής:

```
timeout 800s xterm -title "App 1" -e "echo $TACAA_HOME; echo $CLASSPATH; sleep 30;
java -cp \"lib/*\" tau.tac.adx.agentware.Main -config path/to/configuration/file.conf &
```

Το “&” υπάρχει επειδή οι πράκτορες πρέπει να τρέχουν παράλληλα. Προσθέτοντας την παραπάνω γραμμή σε ένα for-loop του script γίνεται να τρέξετε από έναν έως οχτώ πράκτορες σε μία προσομοίωση, ρυθμίζοντας τον αριθμό των επαναλήψεων στον επιθυμητό. Το runSimulation.sh τρέχει παράλληλα και τον server οπότε η εκτέλεση του runServer.sh δεν χρειάζεται να γίνει ξεχωριστά.



6.2 Η κύρια οθόνη του adx-server

7

Επίλογος

Παρακάτω παρουσιάζονται τα σημαντικότερα συμπεράσματα που εξάγονται από αυτήν την εργασία και δίνονται κάποιες πιθανές επεκτάσεις της.

7.1 Σύνοψη και συμπεράσματα

Το αποτέλεσμα αυτής της εργασίας ήταν ένας προσαρμοστικός πράκτορας που καταφέρνει να αναλύσει τις σημαντικότερες παραμέτρους του περιβάλλοντος στο οποίο δρα και να προσαρμόζει κατάλληλα τη δράση του. Αυτό φαίνεται τόσο στα πειραματικά αποτελέσματα όπου ο πράκτορας πετυχαίνει υψηλά σκορ ανεξαρτήτως της ατομικής ή συλλογικής συμπεριφοράς των αντιπάλων του, όσο και στα αποτελέσματα του διαγωνισμού όπου το περιβάλλον ήταν άγνωστο και πιθανότατα διαφορετικό από τα διάφορα πειραματικά σενάρια. Η σχεδίαση του πράκτορα περιείχε αρκετά καινοτόμα για τον διαγωνισμό στοιχεία, τα κυριότερα από τα οποία είναι:

- Η ανάλυση ανταγωνιστικότητας για την εκτίμηση της πιθανότητας που έχει ο πράκτορας να πάρει μια εκστρατεία με χαμηλό ποντάρισμα. Όπως επίσης και η χρήση αυτής της πιθανότητας για τον υπολογισμό της αναμενόμενης αξίας μιας εκστρατείας.
- Η ενσωμάτωση του τυχαίου παράγοντα στις δημοπρασίες συμβολαίων σαν κυρίαρχη εναλλακτική στρατηγική και όχι μόνο η εκμετάλλευση του σε εκστρατείες που θεωρούνται αδύνατες, όπως έκαναν παλιότεροι πράκτορες.

- Η προσπάθεια προσέγγισης του μακροπρόθεσμου συνολικού κέρδους και η προσαρμογή του πονταρίσματος εντυπώσεων με βάση μακροπρόθεσμους στόχους.

Είναι πολύ πιθανό τα παραπάνω στοιχεία να συνετέλεσαν σημαντικά στη νίκη του τελικού πράκτορα στον διαγωνισμό. Επίσης μπορούν να αποτελέσουν μια βάση για τον συστηματικό σχεδιασμό πρακτόρων σε μια αγορά παρόμοια με αυτή του διαγωνισμού TAC AdX.

7.2 Μελλοντικές επεκτάσεις

Η πολυπλοκότητα του διαγωνισμού TAC AdX μπορεί να δώσει την ευκαιρία σε πολύ διαφορετικές στρατηγικές να αναπτυχθούν καθώς και σε ακόμα περισσότερα περιβάλλοντα. Στόχος αυτής της εργασίας ήταν η σχεδίαση ενός προσαρμοστικού και ανθεκτικού πράκτορα και γι αυτό οι ακόλουθες προτάσεις περιορίζονται πάνω σε αυτό το υπόβαθρο:

- Χρήση μηχανισμού υψηλότερου επιπέδου για την πρόβλεψη της ανταγωνιστικότητας (π.χ. θα μπορούσαν να συγκρίνονται τα αποτελέσματα διαφορετικών τεχνικών πρόβλεψης και να χρησιμοποιείται η καλύτερη). Αυτό θα χρησίμευε σε περιβάλλοντα που αλλάζουν ραγδαία, μιας και πίσω από την απλή εκθετική εξομάλυνση γίνεται η υπόθεση ότι η ανταγωνιστικότητα θα συγκλίνει σε συγκεκριμένη τιμή. Επίσης σε πιο ομοιόμορφα περιβάλλοντα θα βελτιωνόταν η ταχύτητα σύγκλισης.
- Εκτίμηση της αξίας της πληροφορίας UCS με βάση σχετικές παραμέτρους του περιβάλλοντος και ενσωμάτωση του επιπέδου UCS στους τύπους για τη δημοπρασία εντυπώσεων.
- Πιο λεπτομερής στατιστική εκτίμηση της αναμενόμενης αξίας μιας εκστρατείας για περισσότερα επίπεδα πονταρισμάτων και μεγιστοποίηση της.
- Χρήση προβλεπτικού αλγόριθμου μηχανικής μάθησης για πρόβλεψη τιμής (τόσο στις δημοπρασίες εντυπώσεων όσο και συμβολαίων) ή χρήση ανάλυσης ισορροπίας της αγοράς όπως ο Walverine στον classic TAC.

Οι παραπάνω προτάσεις δεν είναι εγγυημένο ότι θα παράγουν καλύτερα αποτελέσματα, αφού η αβεβαιότητα του παιχνιδιού κάνει πολλές φορές το απλό να είναι καλύτερο. Παρόλα αυτά είναι ιδέες που θα έπρεπε να δοκιμαστούν σε μια πιθανή νεότερη έκδοση του πράκτορα.

8

Βιβλιογραφία

- [1] Alvin Roth and Ido Erev. Learning in extensive form-games: Experimental data and simple dynamic models in the intermediate term. *Games and Economic Behaviour*, 8(1):164–212, 1995.
- [2] Amy Greenwald. The 2002 trading agent competition: An overview of agent strategies. *AI Magazine*, 24(1):83, 2003.
- [3] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords. *American Economic Review* 97(1) p.242-259, 2007.
- [4] Bingyang Tao, Fan Wu and Guihai Chen. TAC AdX 14': Autonomous bidding agents for realtime Ad Exchange. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1111-1119, AAMAS, 2015.
- [5] Chen Doar, Roei Mischori, Nizan Kramf. Agent2: Ad exchange agent – project report. <https://sites.google.com/site/gameadx/>, 2014.
- [6] Daniel Lowd, Pedro Domingos. Naive Bayes Models for Probability Estimation. *ICML '05 Proceedings of the 22nd international conference on Machine learning* Pages 529 – 536, 2005.
- [7] David S. Evans. The online advertising industry: economics, evolution and privacy. *Journal of Economic Perspectives*, 2009.
- [8] Elad Haviv, Moshe Krush, Guy Shaanan. Adnetwork Project Report: Agent00. <https://sites.google.com/site/gameadx/>, 2014.

- [9] Gal Rotem, Lior Abadi, Tal Brender. BCM project report. <https://sites.google.com/site/gameadx/>, 2014.
- [10] Guy Gomberg, Itay Waxman, Zohar Paskar, Amit Zitsman. Giza Adnetwork Agent project report. <https://sites.google.com/site/gameadx/>, 2014.
- [11] Ioannis A. Vetsikas , Bart Selman. WhiteBear: An empirical study of design tradeoffs for autonomous trading agents. AAAI Workshop on Game Theoretic and Decision Theoretic Agents (GTDT), Edmonton, Canada, 81-88, 2002
- [12] J.R. Quinlan. Induction of Decision Trees. Kluwer Academic Publishers, Boston, 1986.
- [13] Mariano Schain, Yishay Mansour. The ad exchange game - specification version 1.4. <https://sites.google.com/site/gameadx/>, 2015.
- [14] Mark Hall, Ian Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems, 3rd Edition, 2011.
- [15] Michael Wellman, Amy Greenwald, and Peter Stone. Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition. MIT Press, 2007.
- [16] Michael Wellman, Daniel Reeves, Kevin Lochner, and Yevgeniy Vorobeychik. Price prediction in a trading agent competition. J. Artif. Intell. Res.(JAIR), 21:19– 36, 2004.
- [17] Michael Wellman and Peter Wurman. A trading agent competition for the research community. In IJCAI-99 Workshop on Agent-Mediated Electronic Trading. Citeseer, 1999.
- [18] Michael Wellman, Peter Wurman, Kevin O’Malley, Roshan Bangera, Shou de Lin, Daniel Reeves, William Walsh. Designing the market game for a trading agent competition. Internet Computing, IEEE, 5(2):43–51, 2001.
- [19] Or Sagi, Omer Rotem, Selly Grossman. OOS AdX Agent project report. <https://sites.google.com/site/gameadx/>, 2014.
- [20] Patrick Jordan, Michael Wellman and Guha Balakrishnan. Strategy and mechanism lessons from the first ad auctions trading agent competition. In Proceedings of the 11th ACM conference on Electronic Commerce, pages 287–296. ACM, 2010.
- [21] Patrick Jordan, Ben Cassell, Lee Callender, and Michael Wellman. The ad auctions game for the 2009 trading agent competition - specification version 0.9.6. 2009.
- [22] Patrick Jordan, Akshat Kaul, and Michael Wellman. The ad auctions game for the 2010 trading agent competition - specification for server version 10.1.0.0. 2010.
- [23] Peter Stone, Michael Littman, Satinder Singh, and Michael Kearns. Attac-2000: An adaptive autonomous bidding agent. In Proceedings of the fifth international conference on autonomous agents, pages 238–245. ACM, 2001.
- [24] Polikar, R. “Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine 6 (3): 21–45, 2006

- [25] S. Muthukrishnan. Ad Exchanges: Research Issues. WINE '09 Proceedings of the 5th International Workshop on Internet and Network Economics Pages 1-12, 2009.
- [26] Shih-Feh Cheng, Evan Leung, Kevin Lochner, Kevin O'Malley, Daniel Reeves, Julian Schwartzman, and Michael Wellman. Walverine: A walrasian trading agent. *Decision Support Systems*, 39(2):169–184, 2005.
- [27] Shuai Yuan, Ahmad Zainal Abidin, Marc Sloan, Jun Wang. Internet Advertising: An Interplay among Advertisers, Online Publishers, Ad Exchanges and Web Users. *Arxiv*, 2012.
- [28] Steven Gjerstad and John Dickhaut. Price formation in double auctions. *Games and Economic Behaviour*, 22(1):1–29, 1998.
- [29] Stuart Russel, Peter Norvig. *Artificial Intelligence: A Modern Approach* (3rd Edition). Prentice Hall, 2009.
- [30] Varian Hal R., Harris Christopher. The VCG Auction in Theory and Practice. *The American Economic Review* 104.5 : 442-445, 2014.
- [31] W. Ketter, J. Collins, P. Reddy, C. Flath, M. Weerdt. The power trading agent competition. ERIM Report Series Reference No. ERS-2011-027-LIS, 2011.
- [32] WEKA 3: Datamining Software in Java. <http://www.cs.waikato.ac.nz/ml/weka/>.
- [33] Yu J. Hu, Jiwoong Shin, Zhulei Tang. Performance based pricing models in online advertising: Cost Per Click vs. Cost Per Action.
- [34] Φώτιος Πετρόπουλος, Βασίλειος Ασημακόπουλος. *Επιχειρησιακές Προβλέψεις*. Εκδόσεις Συμμετρία, 2013.
- [35] J. Collins, R. Arunachalam et al. The Supply Chain Management Game for the 2007 Trading Agent Competition. CMU-ISRI-07-100, 2006.
- [36] D. Reeves, M. Wellman et al. Exploring bidding strategies for market-based scheduling. *Decision Support Systems* 39 67 – 85, 2005