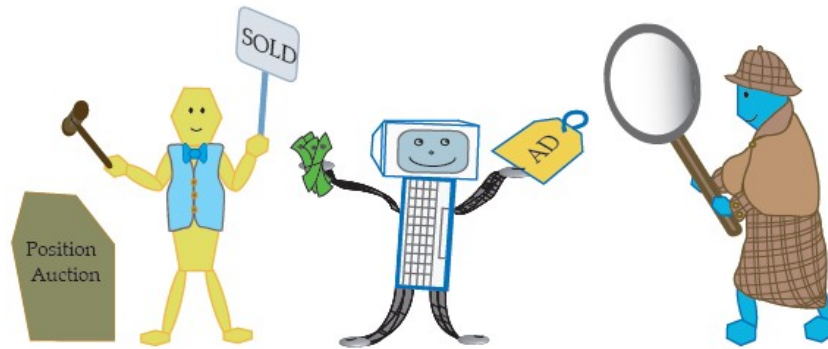


Προγραμματιστική Εργασία: Σχεδιασμός και
Υλοποίηση Πράκτορα για το Διαγωνισμό



**Trading
Agents Competition Ad Auctions
(TAC-AA)**

Σαπουντζής Νίκος 2007030007

Αγγελιδάκης Άγγελος 2007030032

ΠΛΗ 517, Άνοιξη 2012

Παράρτημα

Εισαγωγή-----	Σελ. 3-8
Αποκωδικοποίηση του Captain Sparrow-----	Σελ. 9
Υλοποιήσεις των συναρτήσεων-----	Σελ. 10-18
Τι πήγε καλά και τί όχι;-----	Σελ. 19
Μελλοντική δουλειά -----	Σελ. 19
Αναφορές-----	Σελ. 20-21

1. Εισαγωγή

Η κυριότερη οικονομική δραστηριότητα στο διαδίκτυο είναι η προβολή διαφημίσεων όταν γίνεται μία αναζήτηση. Για παράδειγμα, όταν κάποιος χρήστης χρησιμοποιεί τη μηχανή αναζήτησης της Google ή της Yahoo εμφανίζονται διαφημίσεις στην κορυφή και το πλάι των αποτελεσμάτων αναζήτησης, διαφημίσεις για τις οποίες οι διαφημιστές έχουν πληρώσει εκείνον που θα τις δημοσιεύσει:

The image shows two search engines, Yahoo! and Google, displaying results for the query "usb flash memory".

Yahoo! Search Results:

- Search bar: "usb flash memory" (77,700,000 results)
- Related Searches: "usb flash drive"
- Filter by Time: Anytime, Past day, Past week, Past month
- Advertisements (highlighted in red):
 - Bulk Custom USB Drives**: 100+ Styles/Top Quality/Great Service. Wholesale direct - 877.442.7465 www.printusb.com
 - usb flash memory**: Huge In-Stock Inventory. Order Online & Save Time! www.alltelect.com
 - Flash Memory**: Massive IC Selection at Digi-Key. Specs & Prices for Online Ordering. www.DigiKey.com
 - usb flash memory**: Your Logo Imprint on a USB Drive, Fast Turnaround, 64MB from \$3.25 www.usbmemorydirect.com/CustomUSB
 - Usb Flash Memory**: The Best Prices On The Online. Order Yours Today & Save! www.TigerDirect.com
 - Usb Flash Memory**: Avoid Data Disasters. Save on Usb Flash Memory! MemoryCards.Shopzilla.com
 - Usb Flash Memory**: Search. Compare. Conquer. Save on Usb Flash Memory! BizRate.com
 - USB Flash Drive Reviews**: Read the Reviews, Compare Prices, and Buy the Best USB Flash Drives. www.ConsumerSearch.com
 - Usb Flash Memory**: Explore Hot Deals at Low Prices. Shop For Usb Flash Memory! LowPriceShopper.com
 - Usb Flash Memory**: Shop Usb Flash Memory at Walmart & Update your Computer for Less!
- Shopping Results:
 - PNY HP 4GB USB Flash Drive \$208.42 CostCentral.com
 - SanDisk 32GB Cruzer USB Flash Drive \$29.99 Newegg.com

Google Search Results:

- Search bar: "usb flash memory" (About 69,600,000 results (0.42 seconds))
- Everything:
 - Ad related to usb flash memory:
 - USB Flash - Μεγάλη Ποικιλία σε Usb Flash** | getitnow.gr
www.getitnow.gr/752 Σημεία Δωρεάν Παραλαβής!
↳ Laptops - Tablets - Computers - Software
 - USB flash drive - Wikipedia, the free encyclopedia**
en.wikipedia.org/wiki/USB_flash_drive - Cached
A **USB flash drive** is a data storage device that includes flash memory with an integrated Universal Serial Bus (USB) interface. **USB flash drives** are typically ...
↳ ReadyBoost - Portable application - USB Flash Drive Alliance
 - Shopping results for usb flash memory**
 - SanDisk Cruzer USB flash drive - 32 GB**
★★★★★ 326 reviews - \$23 - 266 stores
 - SanDisk Cruzer Blade USB flash drive - 8 GB**
★★★★★ 34 reviews - \$7 - 153 stores
 - SanDisk Cruzer Blade USB flash drive - 4 GB**
★★★★★ 4,528 reviews - \$6 - 93 stores
 - Ads - Why these ads?
 - Διαφημιστικά USB Drives**
www.myusb.gr/
USB Φλας & Στικ από 128Mb έως 32Gb. Με το δικό σας λογότυπο από 3€ τμ.
 - Φθηνά USB Memory Stick**
www.ricardo.gr/
Φθηνά **USB** Sticks Kingston, SanDisk, Verbatim, PNY, HP στο ricardo.gr!
 - USB Software for Flash**
www.jungo.com/
Enable **USB** functionality to your **flash**/mass storage device
 - Banner Ads on Google**
adwords.google.com/
Show Banners on hundreds of Sites on the Google Content Network!
- All results: Sites with images, Related searches, More search tools

Αυτή η δραστηριότητα αποτελεί κύριο έσοδο των μηχανών αναζήτησης. Έτσι λοιπόν, οι εταιρίες που επιθυμούν να διαφημιστούν πρέπει να υποβάλλουν προσφορές σε δημοπρασίες για λέξεις κλειδιά (keywords) ώστε να εμφανιστούν σε πελάτες που ψάχνουν στην μηχανή αναζήτησης τα αντίστοιχα προϊόντα ενώ ταυτόχρονα να εμφανίζονται σε όσον το δυνατόν περισσότερες συναφείς αναζητήσεις. Ο διεθνής διαγωνισμός TAC Ad Auctions είναι το διεθνές benchmark για αυτό το ιδιαίτερα σημαντικό πρόβλημα. Σκοπός της εργασίας είναι ο σχεδιασμός ενός πράκτορα ο οποίος θα προσπαθήσει να ανταγωνιστεί άλλους παρόμοιους πράκτορες και μέσω κάποιων πακέτων προσφορών να κερδίσει μία καλή θέση στη διαφημιστική λίστα. Αυτό φυσικά προϋποθέτει να μην του κοστίζουν του πράκτορα τα κλικ που θα γίνουν, περισσότερο από ότι τα έσοδα των αγορών από τους χρήστες που χρησιμοποιούν την αντίστοιχη μηχανή αναζήτησης.

Οι πράκτορες που αλληλεπιδρούν μεταξύ τους κατά τη διεξαγωγή της δημοπρασίας είναι ο Διαφημιστής, ο Χρήστης και ο Δημοπράτης.

- Ο διαφημιστής διαθέτει τις εξής ενέργειες:
 - παραθέτει προσφορά για να τοποθετηθεί η διαφήμιση του στη λίστα
 - επιλέγει τη διαφήμιση που θέλει προβληθεί
 - λαμβάνει καθημερινά αναλυτική αναφορά (κλικς, αγορές, rank κλπ.)







- ο Χρήστης:
 - κλικ στις διαφημίσεις
 - αγοράζει προϊόντα

- ο Δημοπράτης:
 - τρέχει δημοπρασίες για κάθε query που υποβάλλει ο χρήστης στη βάση δεδομένων
 - επεξεργάζεται τα ερωτήματα των χρηστών και τα κλικς
 - διανέμει καθημερινά αναφορές των ερωτημάτων, που κάνουν οι χρήστες, στους διαφημιστές

Για κάθε προσομοιωμένη μέρα:

1. Οι χρήστες υποβάλουν ερωτήματα που βασίζονται στις μουστικές τους προτιμήσεις. Ο Δημοπράτης καθορίζει τη θέση του slot της διαφήμισης και επιστρέφει μια λίστα κατάταξης στους χρήστες, που ονομάζεται *impression list*.
2. Οι χρήστες μπορούν να κάνουν κλικ στις διαφημίσεις και, εάν ναι:
3. Οι διαφημιζόμενοι πληρώνουν το κόστος-ανά-κλικ (καθορίζεται από τη δημοπρασία)
4. Οι χρήστες μπορούν να αγοράσουν το προϊόν κάνοντας κάποια συνδιαλλαγή
5. Όταν οι χρήστες κάνουν τη συνδιαλλαγή, οι διαφημιστές κερδίζουν κάποια έσοδα
6. Ο Εκδότης ενημερώνει τους διαφημιστές με τις στατιστικές αναφορές

Στη συγκεκριμένη αγορά έχουμε 3 κατασκευάστριες εταιρείες, 3 τύπους προϊόντων για κάθε εταιρεία, δηλαδή 9 μεμονομένα προϊόντα:

Manufacturer			
	Lioneer	PG	Flat
Type			
	TV	Audio	DVD

Τα ερωτήματα που κάνουν οι χρήστες μπορεί να είναι μια συλλογή από λέξεις. Στην περίπτωση μας: {κατασκευάστρια εταιρεία, τύπος προϊόντος}. Ο κάθε χρήστης επομένως κάθε μέρα στέλνει μία από τις 9 δυνατές συλλογές:

- {LIONEER,TV},{LIONEER,Audio},{LIONEER,DVD}
- {PG,TV},{PG,Audio},{PG,DVD}
- {Flat,TV},{Flat,Audio},{Flat,DVD}

Οι χρήστες που κάνουν την αναζήτηση διακρίνονται σε 3 κατηγορίες:

1. Low Focus (F0 – 1 query, ie. {null, null})
2. Intermediate Focus (F1 – 6 queries, ie. {PG, null}, {null, DVD})
3. High Focus (F2 – 9 queries, ie. {Flat, TV}, {Lioneer , Audio})

Οι διαφημιζόμενοι επιλέγουν τις διαφημίσεις που θα εμφανίζονται για κάθε query. Επιλέγουν ανάμεσα σε γενικές και στοχευμένες διαφημίσεις και υποβάλλουν το ποσό προσφοράς για κάθε διαφήμιση, κάθε μέρα. Στην αρχή του παιχνιδιού, ο κατασκευαστής και το εξειδικευμένο προϊόν του κάθε ένα αναφέρονται στη δημοπρασία. Κάθε μέρα, για κάθε query επιλέγονται οι προς εμφάνιση διαφημίσεις (γενική ή στοχοθετημένη). Επιπλέον καθημερινά ενημερώνεται η τιμή προσφοράς, μπαίνουν όρια στις δαπάνες (προαιρετικά), ο Δημοπράτης στέλνει μία στατιστική αναφορά στους διαφημιστές, ενώ οι διαφημιστές ενημερώνονται παράλληλα για την κατάσταση του τραπεζικού τους λογαριασμού.

Για κάθε query ο Διαφημιστής στέλνει στον Δημοπράτη:

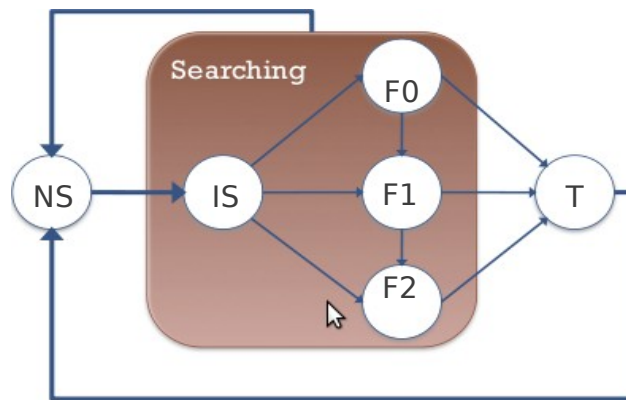
- τον τύπο της διαφήμισης, το CPC(cost-per-click) και
- το όριο στις δαπάνες.

Ενώ οι αναφορές που λαμβάνουν οι διαφημιστές περιέχουν:

- τον τύπο της διαφήμισης του κάθε διαφημιστή, η οποία προβήθηκε
- η κατά μέσο όρο θέση στην διαφημιστική λίστα

- η κατά μέσο όρο τιμή του CPC
- τον αριθμό των φορών που οι χρήστες είδαν την διαφήμιση κάποιου διαφημιστή κατά την αναζήτηση τους
- τον αριθμό των φορών που οι χρήστες έκαναν κλικ σε κάποια διαφήμιση

Οι πληθυσμοί των χρηστών που κάνουν αναζήτηση μπορούν να κατηγοριοποιηθούν γραφικά έτσι:



- Non-searching (NS)
- Searching
 - Informational search (IS)
 - Shopping, focus level 0 (F0)
 - Shopping, focus level 1 (F1)
 - Shopping, focus level 2 (F2)
- Transacted (T)

Όλοι οι πληθυσμοί αρχικοποιούνται στην κατάσταση NS, είναι η κατάσταση στην οποία δεν κάνουν καμία αναζήτηση. Στη συνέχεια μεταβαίνουν στην κατάσταση IS

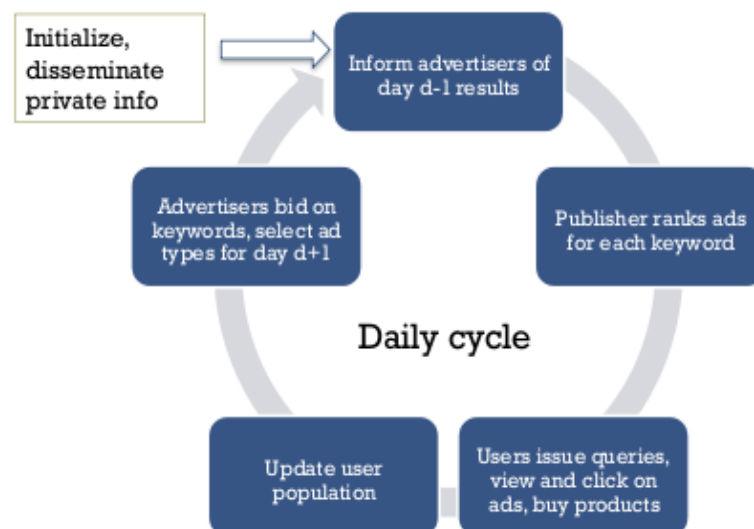
που είναι η κατάσταση από την οποία μπορούν να μεταβούν στο focus level ή να παραμείνουν στην ίδια. Το focus level είναι η κατάσταση όπου ο χρήστης αναζητά κάποια εταιρεία ή προϊόν(F1) ή και τα δύο (F2) ή σε διαφορετική περίπτωση κάνει κάποια μη σχετική αναζήτηση(F0). Ενώ αν κάποιος χρήστης επιχειρήσει συνδιαλλαγή μεταβαίνει στην κατάσταση T . Η πιθανότητα σταθερής συνδιαλλαγής δίνεται από τη παράμετρο: $Pr_{NS \rightarrow IS}^{standard}$ ενώ η εκρηκτική πιθανότητα μετάβασης από την παράμετρο:

$Pr_{NS \rightarrow IS}^{burst}$

Οι καθημερινές εργασίες που πρέπει να γίνουν είναι:

- Οι ημερήσιες αναφορές που συνοψίζουν d-1 ημέρες στέλνονται στους διαφημιστές
- Μία δημοπρασία διαφημίσεων για κάθε ερώτημα καθορίζει την κατάταξη της διαφήμισης και τα CPC από τον Δημοπράτη
- Οι Χρήστες θεματοποιούν τα queries, λαμβάνουν αποτελέσματα, εξετάζουν την αγορά και τα κλικς
- Ο Δημοπράτης παρακολουθεί τη δραστηριότητα και επαναλαμβάνει τις δημοπρασίες όταν χρειάζεται
- Ο πληθυσμός ενημερώνεται και αν χρειαστεί μέρος του πληθυσμού ή και όλο μεταβαίνει σε νέα κατάσταση
- Οι διαφημιστές υποβάλουν προσφορά και ενημερώνουν τον Δημοπράτη για την επιλογή της διαφήμισης την ημέρα d +1

Ακολουθεί ένα γράφημα που παρουσιάζει έναν ημερήσιο κύκλο στη ζωή ενός τέτοιου πράκτορα:

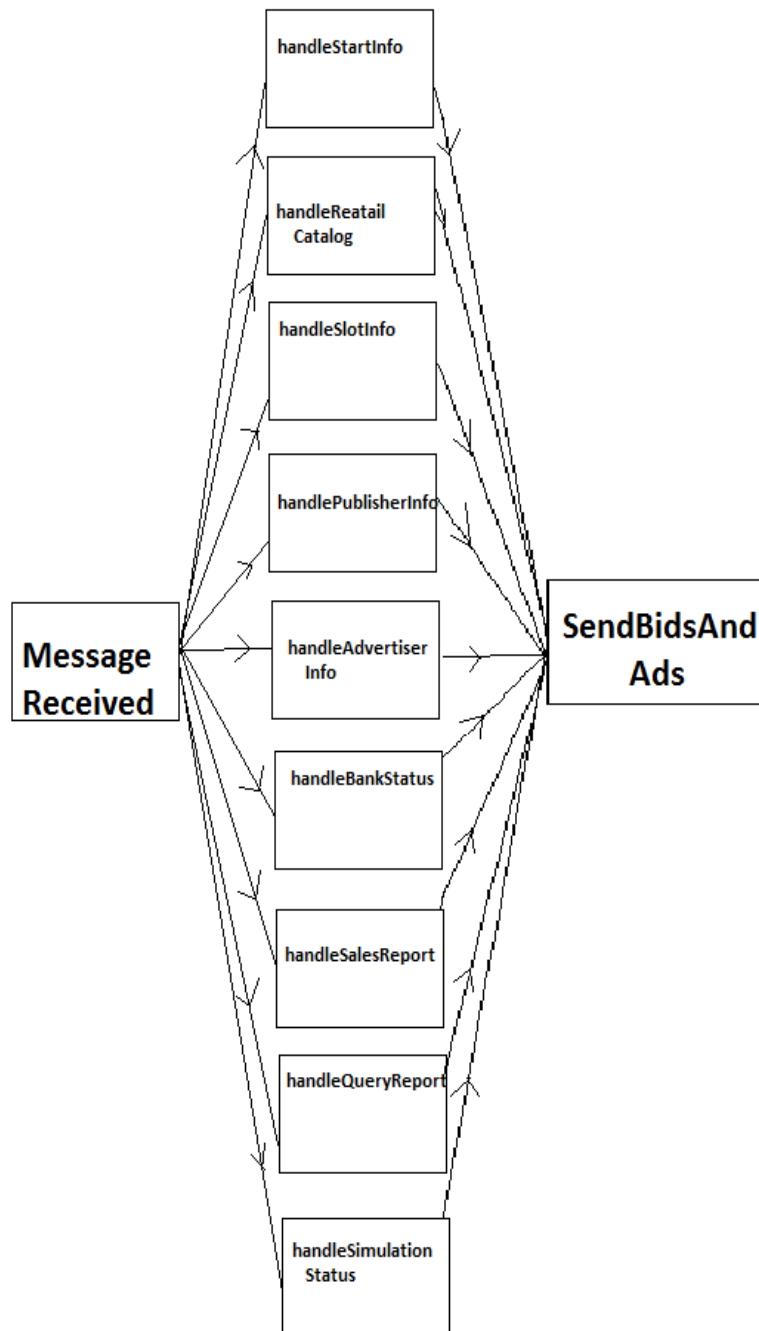


2. Αποκωδικοποίηση του Captain Sparrow

Αρχικά ο SERVER στέλνει μέσω ενός μηνύματος τύπου Message στον πράκτορα μας, μία αναφορά σχετικά με το έγινε την προηγούμενη ημέρα (δημοπρασίες, αποτελέσματα τους, transactions κ.α.). Το μήνυμα αυτό πρέπει να αποκωδικοποιηθεί από τον πράκτορα μας, περνώντας από διάφορες συναρτήσεις handle() οι οποίες το αποκωδικοποιούν. Η διαδικασία αυτή περιγράφεται από το διάγραμμα που παρατίθεται δεξιά.

Κάθε μία από τις συναρτήσεις handle(), εξάγει χρήσιμες πληροφορίες για τον πράκτορα μας, οι οποίες θα του χρειαστούν για την στρατηγική που θα ακολουθήσει στο τρέχων bid, όπως για παράδειγμα η handleQueryReport(), φέρει πληροφορίες σχετικά με τα πιθανά Queries (κλικς, CPC, impressions κ.α.). Η συνάρτηση SendBidsAndAds(), πραγματεύεται την στρατηγική του πράκτορα καθώς μέσα στο σώμα της αναπτύσσεται το «πως» θα κινηθεί ο πράκτορα μας στο

τρέχων bid, ανάλογα με το τι πληροφορίες εισέπραξε τις προηγούμενες μέρες από τις αναφορές που του έστειλε ο Server. Η συνάρτηση SendBidsAndAds() πρόκειται να αναλυθεί λεπτομερώς παρακάτω. Οι 9 συναρτήσεις handle(), αποθηκεύουν όλες τις πληροφορίες μας σε κατάλληλους πίνακες και μεταβλητές. Ενδιαφέρον παρουσιάζει να αναλύσουμε σε βάθος το εσωτερικό της handleQueryReport() από την οποία



παίρνουμε και τις χρησιμότερες πληροφορίες που θα χρειαστούμε για την στρατηγική μας.

3. Υλοποιήσεις των συναρτήσεων

HandleStartInfo()

Μας παρέχονται κάποιες γενικές πληροφορίες για το παιχνίδι: το ID του simulation, πόσες μέρες θα διαρκέσει η προσομοίωση και τυπώνεται στην κονσόλα σχετικά μηνύματα.

HandleRetailCatalog()

Αποθηκεύεται στην κλάση των Parameters η οποία κρατάει μία λίστα από σταθερές και μεταβλητές, παραμέτρους που θα χρησιμοποιηθούν αργότερα (USP, Manufacturers, Components) .

HandleSlotInfo()

Αποθηκεύεται στην κλάση των Parameters άλλες παράμετροι που θα χρησιμοποιήσει αργότερα και αφορούν τα slots(PSB,promoSlots,regSlots).

HandlePublisherInfo()

Αποθηκεύεται στην κλάση των Parameters άλλες παράμετροι που θα χρησιμοποιήσει αργότερα και σχετίζονται με τα Focus Users , και την σταθερά Squashing Parameter.

HandleAdvertiserInfo()

Αποθηκεύεται στην κλάση των Parameters άλλες παράμετροι οι οποίες έχουν σχέση με τους advertisers (CSB, component – manufacturer specialty, MSB, distribution capacity, lamda, length of window W, publisher ID, Target Effect, piF0, piF1, piF2)

HandleBankStatus()

Μας ενημερώνει για το τρέχων profit μας, δηλαδή το account balance μας.

HandleSalesReport()

Αρχικά αποθηκεύουμε την τρέχουσα ημέρα που βρισκόμαστε σε μία μεταβλητή dateIndex η οποία θα μας χρειαστεί στους υπολογισμούς μας. Έπειτα υπολογίζουμε μία παράμετρο η οποία παίρνει τιμές στο διάστημα [0,15], και δείχνει σε ποιο από τα 16 query βρισκόμαστε. Στην συνέχεια, αποθηκεύουμε σε έναν πίνακα registry στην τρέχουσα ημέρα, και για κάθε ένα από τα 16 δυνατά query που έχουμε πληροφορία, τα έσοδα που είχαμε και τα conversions που έγιναν.

HandleSimulationStatus()

Αυτή η μέθοδος απλά αναφέρει σε ποια από τις 60 ημέρες του παιχνιδιού βρισκόμαστε, την τυπώνει στην κονσόλα και έπειτα καλεί την SendBidsAndAds(). Είναι η τελευταία συνάρτηση handle() που καλείται.

HandleQueryReport()

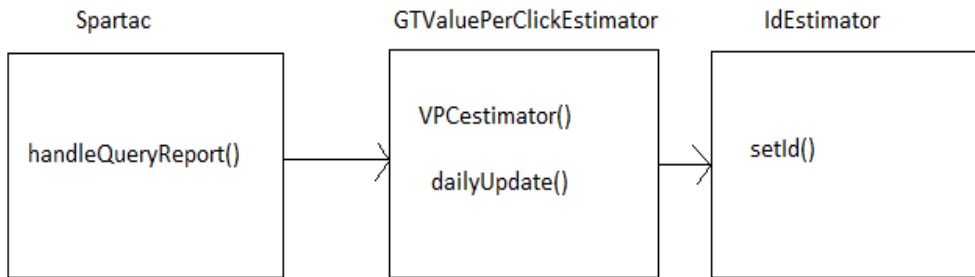
Αυτή η μέθοδος είναι από τις σημαντικότερες του προγράμματος μας, γιατί «τραβάμε» από την αναφορά του σέρβερ, σημαντικές πληροφορίες όπως πόσα κλικ έγιναν για κάθε ένα από διαφορετικά query της προηγούμενης ημέρας, το CPC, το rank στην λίστα και τα Impressions. Οι τιμές αυτές έπειτα αποθηκεύονται σε μία λίστα registry, η οποία στο υπόλοιπο του προγράμματος καλείτε κάθε φορά που θέλουμε να πάρουμε κάποια τιμή από τις παραπάνω (π.χ. πόσα impressions έγιναν για το query “Lioneer TV” την 16^η μέρα;) και να κάνουμε κάποιον υπολογισμό.

Ακολουθεί το κομμάτι του προγράμματος στο οποίο προχωράμε στο 1^ο βήμα για το *estimate* το VPC δεδομένου των conversions που έγιναν για ΟΛΑ τα queries τις τελευταίες 5 ημέρες (ή λιγότερων αν η τρέχων ημέρα < 5), και του distribution capacity για την τρέχουσα ημέρα. Το άθροισμα των conversions που έγιναν για ΟΛΑ τα queries τις τελευταίες 5 ημέρες υπολογίζεται στην μέθοδο HandleSalesReport() με ένα απλό βρόγχο while{ }.

Για να ολοκληρωθεί το 1^ο στάδιο της διαδικασίας της εκτίμησης του VPC ανατρέχουμε σε 2 κλάσεις τις οποίες καλούμε σειριακά, και εν τέλει καταλήγουμε στον υπολογισμό του I_d ο οποίος και μας ενδιαφέρει. Έτσι, αρχικά από την κύρια κλάση που βρισκόμαστε (Spartac) μεταφερόμαστε στην κλάση GTValuePerClickEstimator και συγκεκριμένα στην μέθοδο dailyUpdate(). Από εκεί καλούμε την συνάρτηση getLastConversions() της ίδιας κλάσης στην οποία αθροίζουμε όλα τα conversions των τελευταίων 5 ημερών, και μεταφερόμαστε στην κλάση IdEstimator στον οποίο υπολογίζουμε και κάνουμε setId με ανάλογη μέθοδο, το I_d από την κατάλληλη σχέση:

$$I_d = \lambda \left(\left(\sum_{t=d}^{d-(W-1)} c_t \right) - C^{cap} \right)^+$$

Το «λ» βρίσκεται αποθηκευμένο στην κλάση Parameters, και έχει σταθερή τιμή από την αρχή του προγράμματος. Η διαδικασία αυτή φαίνεται συγκεντρωτικά στο παρακάτω διάγραμμα:



Έτσι επιστρέφόμενοι στην κλάση Spartac και συγκεκριμένα στην μέθοδο handleQueryReport() έχει ολοκληρωθεί η εκτίμηση του Id, με την οποία και ολοκληρώνεται η υλοποίηση της handleQueryReport().

SendBidsAndAds()

Αυτή η μέθοδος είναι η κυριότερη του συστήματος μας, γιατί εδώ μέσα υλοποιείται η κύρια στρατηγική του πράκτορα μας CaptainSparrow, και γίνονται οι περισσότεροι υπολογισμοί που θα βοηθήσουν στον υπολογισμό του τελικού Bid που θα στείλουμε για την τρέχουσα ημέρα!

Όλοι οι υπολογισμοί που ακολουθούν γίνονται 16 φορές, μία για κάθε ένα από τα 16 δυνατά διαφορετικά Queries. Αρχικοποιούμε όλες τις μεταβλητές που θα χρησιμοποιήσουμε στο 1 .

ΠΡΟΣΟΧΗ: Δεν αρχικοποιούμε τις μεταβλητές μας στο 0, γιατί θα απειρίζονταν μερικές μεταβλητές παρακάτω στην περίπτωση που δεν έχουμε «θετική» πληροφορία για αυτές – όμως στο τέλος των υπολογισμών πριν προχωρήσουμε στους υπολογισμούς μας το λαμβάνουμε υπ’ όψιν μας αυτήν την αρχικοποίηση και αφαιρούμε την μία μονάδα από την οποία ξεκινήσαμε.

Για να γίνει πιο απτό το γιατί αρχικοποιούμε στο 1 και όχι στο 0 αρκετές μεταβλητές, παίρνουμε ως παράδειγμα την μεταβλητή $V_{dynamic}$ (θα εξηγήσουμε παρακάτω τι

κάνει , πως χρησιμοποιείται κτλ – εδώ απλώς το αναφέρουμε ως παράδειγμα) η οποία ορίζεται ως εξής :

$$V_{dynamic} = \frac{\sum_{i=d-1}^{d-W} revenue_i - \sum_{i=d-1}^{d-W} cost_i}{\sum_{i=d-1}^{d-W} click_i}$$

Παρατηρούμε πως στον παρανομαστή υπάρχει ένα άθροισμα (των clicks). Αν λοιπόν για κάποιο από τα 16 queries, για 5 συνεχόμενες ημέρες δεν είχαμε κανένα click από τους χρήστες (μικρή πιθανότητα αλλά όχι 0 σίγουρα με πιθανότητα “1”) το $V_{dynamic}$ θα απειρίζονταν γεγονός που θα μας έφερνε σε αδιέξοδο. Το $V_{dynamic}$ πρέπει να είναι πεπερασμένος αριθμός. Για να αποφύγουμε αυτήν την αδιέξοδη κατάσταση λοιπόν αρχικοποιούμε τα πάντα στο 1. Έτσι αν δεν έχουμε άλλα clicks, το revenue και το cost θα παραμείνουν σταθερά και επειδή είναι αρχικοποιημένα στο “1” θα δημιουργούσαμε μια διαφορά “0” και το $V_{dynamic}$ θα έβγαине “0”.

Μετά από την παραπάνω παρατήρηση είμαστε σε θέση να μπούμε σε πιο πρακτικά κομμάτια της υλοποιηθείσας στρατηγικής μας. Υπό την προϋπόθεση ότι η τρέχουσα ημέρα είναι μεγαλύτερη του 5 (απαραίτητη υπόθεση για την επιτυχή ολοκλήρωση των υπολογισμών μας), ανατρέχουμε με ένα for loop στις 5 προηγούμενες μέρες από την σημερινή, και αθροίζουμε για το συγκεκριμένο query όλα τα Revenues, costs, sum of clicks, sum of conversions, sum of conversions (for a specialty component/manufacturer) ανατρέχοντας στους κατάλληλους πίνακες registry οι οποίοι έχουν γεμίσει σε προηγούμενες μεθόδους από τις reports μας. Τρέχουμε ξεχωριστό for loop έπειτα, για ΟΛΕΣ τις ημέρες από την 1^η έως την τρέχουσα στο οποίο αθροίζουμε τα clicks και impressions, το οποίο και θα μας χρειαστεί παρακάτω.

Έπειτα ανατρέχουμε ξανά στην κλάση GTValuePerClickEstimator, και συγκεκριμένα στην μέθοδο getValuePerClick(), δεδομένου του query, και του sum of conversions που υπολογίστηκαν. Εκεί υπολογίζεται το VPC από το γινόμενο

$$VPC = (\text{expected Revenue}) * (\text{id estimator})$$

(1)

• Από την `getValuePerClick()`, ανατρέχουμε στην μέθοδο `getExpRevenueAfterClick()` από την οποία υπολογίζονται τα expected Revenues από την σχέση:

$$v_{con} = \begin{cases} USP(1 + MSB) & \text{if } q_m = MS \\ \frac{2}{3}USP + \frac{1}{3}USP(1 + MSB) & \text{if } q_m = null \\ USP & \text{otherwise} \end{cases}$$

Οι τιμές USP, MSB έχουν πάρει τιμή στην αρχή του προγράμματος από την αποκωδικοποίηση των reports. Το MSB είναι το MS bonus rate και έχει τιμή συνήθως 50%. Το q_m συμβολίζει το manufacturer part του query που τρέχουμε αυτήν την στιγμή.

1. Ο πρώτος κλάδος της παραπάνω εξίσωσης σημαίνει για όσα queries ισχύει $q_m=MS$ να επιδέχονται 50% επιπλέον conversion value από το Normal USP. Κάτι απόλυτα λογικό γιατί για τα Specialty manufacturers θέλουμε υψηλότερο conversion value, που καθορίζεται από το MSB!
2. Ο δεύτερος κλάδος ότι για τα queries που ισχύει $q_m=null$ έχουν 1/3 πιθανότητα να επιδεχτούν 50% επιπλέον conversion value, και 2/3 normal USP.
3. Ο τρίτος βρόγχος σημαίνει ότι για τα queries που ισχύει για τα διαφορετικά του null $q_m =$ (διαφορετικό manufacturer του MS) το conversion value τους να είναι ακριβώς USP.

• Από την κλάση `Idestimator` και συγκεκριμένα από την μέθοδο `getConversionProb()`, υπολογίζεται το expected conversion probability από την σχέση:

$$P_{conversion} = \begin{cases} \frac{(1+CSB)\pi_l}{1+CSB\pi_l} & \text{if } q_c = CS \\ \frac{2}{3}\pi_l + \frac{1}{3}\frac{(1+CSB)\pi_l}{1+CSB\pi_l} & \text{if } q_c = null \\ \pi_l & \text{otherwise} \end{cases}$$

Όπου π_l είναι το conversion rate για κάθε ένα από τα 3 διαφορετικά focus users ($l \in F0, F1, F2$). Ισχύει ότι $\pi_{F2} > \pi_{F1} > \pi_{F0}$. Έτσι αφού βρούμε το τρέχων query index (0 – 16) υπολογίζουμε το focus user με τον εξής απλό αλγόριθμο :

```

if(focus<9)
  pl=Parameters.piF2;
else if(focus<15)
  pl=Parameters.piF1;
else
  pl=Parameters.piF0;

```

Εφόσον έχουμε υπολογίσει το π προχωράμε στον υπολογισμό του conversion probability, από την σχέση που δίδεται παραπάνω. Έτσι επιστρέφουμε στο σημείο όπου υπολογίζουμε τελικά το VPC από την σχέση (1) και επιστρέφουμε στην κύρια κλάση και μέθοδο του προγράμματός μας (SendBidsAndAds()).

Εδώ έχουμε τον πρώτο σημαντικό υπολογισμό για την επίτευξη της στρατηγικής μας :

$$v_static = VPC = (\text{expected Revenue}) * (\text{id estimator})$$

Έχουμε λοιπόν υπολογίσει το v_static το οποίο προκύπτει από την παραπάνω διαδικασία με τον υπολογισμό των expected Revenue , id estimator. Queries με μεγάλο v_static είναι αναμενόμενο να προσφέρουν περισσότερο κέρδος Profit.

Προχωράμε στον υπολογισμό ενός δυναμικού value per click, το οποίο θα υπολογίζεται δυναμικά ανάλογα του το τι συνέβη τις προηγούμενες ημέρες. Ο τύπος για τον υπολογισμό του είναι:

$$v_{dynamic} = \frac{\sum_{i=d-1}^{d-W} revenue_i - \sum_{i=d-1}^{d-W} cost_i}{\sum_{i=d-1}^{d-W} click_i}$$

Το άθροισμα των revenues, costs, clicks των 5 τελευταίων ημερών έχουν όλα υπολογιστεί παραπάνω, από το for loop που έχουμε περιγράψει. Άρα σε αυτό το σημείο έχουμε υπολογίσει και το $v_dynamic$. Το bid που θα γίνει βάσει του $v_dynamic$ θα υπολογίζεται από την σχέση :

$$bid = v_dynamic * U * R$$

Έχουμε υπολογίσει από την παραπάνω σχέση το $v_dynamic$ οπότε μένει ο υπολογισμός του U, R που θα εξηγήσουμε στην συνέχεια τι είναι κάθε μία από αυτές τις μεταβλητές.

Το U είναι ένας κατάλληλος δείκτης ο οποίος ρυθμίζει το bid από το $v_dynamic$, ανάλογα ενός ratio ανάμεσα στο capacity (300 , 450 ,600) και ενός 5-day aggregate conversion. Για την συνέχεια της στρατηγικής ορίζουμε μία νέα έννοια μεταβλητής ($Cagg,d$) η οποία συνδέει το capacity και τα conversions των 5 προηγούμενων ημερών ζυγισμένα με w_i βάρη (τα conversions των πιο πρόσφατων ημερών έχουν μεγαλύτερο βάρος από των πιο μακρινών ημερών). Το w παίρνει τιμή στο διάστημα (0,1), και το ορίζουμε στην κλάση Parameters ανάλογα του πως θέλουμε να κινηθούμε στα βάρη των προηγούμενων ημερών. Εμείς στην προσέγγιση μας, το ορίζουμε $w=0,7$ καθώς μετά από αρκετά πειράγματα παρατηρήσαμε πως είναι το βέλτιστο για πραγματικούς αντίπαλους (για dummies ο Captain Sparrow παίζει καλύτερα για χαμηλότερη τιμή του w)

$$c_{agg,d} = \frac{\sum_{t=d-1}^0 (w_t \sum_{i=t}^{t-(W-1)} c_i)}{\sum_{t=d-1}^0 w_t}$$

Έτσι έχουμε προχωρήσει στον υπολογισμό του $Cagg,d$ (το d συμβολίζει την μέρα d), και προχωράμε στον υπολογισμό του δείκτη U .

Το Backorder για την μέρα $i-1$ προκύπτει από την σχέση:

$$B_{d-1}(x) = \sum_{i=d-W}^{d-2} c_i(x) - C(x) \quad (2)$$

Έτσι, έχουμε τις περιπτώσεις:

- $B_{d-1} > 0$ δεν έχουμε περιθώριο για άλλες συναλλαγές ($c_i(x) > C(x)$) ,
- $B_{d-1} < 0$ έχουμε περιθώριο για επιπλέον συναλλαγές

ΣΗΜΑΝΤΙΚΗ ΠΑΡΑΤΗΡΗΣΗ : Να παρατηρήσουμε εδώ πως χρειάστηκε μεγάλη προσοχή όταν φτιάχναμε τους δείκτες (όρια) για τα for loops που υπολογίζουν τα παραπάνω γιατί η report για την μέρα $d-1$ γίνεται διαθέσιμη ΕΦΟΣΟΝ έχουμε ολοκληρώσει τα bid για την μέρα d .

Αν λύσουμε την σχέση (2) με την συνθήκη $B_{d-1}(x) < 0$ (κάτι το οποίο επιδιώκουμε για να μην φτάσουμε σε κατάσταση backorder) έχουμε ότι θέλουμε διαρκώς

$$\frac{Ccap}{\sum ci} > 1$$

Προχωράμε στον υπολογισμό του U . Αν αντί το $\sum ci$, αντικαταστήσουμε το $Cagg,d$ το οποίο είναι και πάλι τα conversions που έγιναν «ζυγισμένα» κατάλληλα, καταλήγουμε στην σχέση :

$$U = \frac{Ccap}{Cagg,d}$$

Αν $U > 1$ τότε το $Ccap$ είναι underfilled στην συγκεκριμένη ημέρα μας, και το bid αναμένεται να αυξηθεί κατά δ . Όμοια αν $U < 1$ το bid αναμένεται να μειωθεί κατά δ ! Άρα ο δείκτης « δ » είναι πολύ χρήσιμος και σημαντικός γιατί μας δείχνει αν υπάρχει Backorder, αν είμαστε κοντά ή μακριά από την ανεπιθύμητη κατάσταση backorder και τον χρησιμοποιούμε για να κινηθούμε ανάλογα στο bid.

Το R ισούται με τον λόγο:

$$(1 + e'_q)^{-x} = \left(1 + \frac{\sum_{i=0}^{d-1} Click_q}{\sum_{i=0}^{d-1} Impression_q} \right)^{-x}$$

Το R χρησιμοποιείται για να προσαρμόσουμε τα bid ανάλογα με το e_q και την squashing parameter (ranking mechanism adapter). Το e_q είναι το «advertiser effect» ή click through rate (CTR) για την query q . Προφανώς, όσο πιο κοντά στο 0 είναι το x , τόσο το rank βασίζεται σε «rank by bid» ενώ όσο πιο κοντά στο 1 «rank by revenue», εξού και η αιτιολόγηση της δύναμης x . Λόγω του ότι το R είναι αντίστροφα ανάλογο του $1 + e_q$ πρέπει να είναι στον παρανομαστή η ποσότητα $(1 + e_q)$ ή η δύναμη x , να είναι αρνητική ($-x$).

Προχωράμε στην «καρδιά» του υπολογισμού του bid. Διακρίνουμε τις εξής περιπτώσεις :

- Αν η τρέχουσα ημέρα είναι μικρότερη του 5 , τότε κάνουμε bid βάση του v_static αφού το $v_dynamic$ δεν έχει νόημα μιας και οι διαφορές μεταβλητές μας για το παράθυρο $W=5$ δεν μπορούν να υπολογισθούν, με την υβριδική σχέση:

$$bid = (\alpha * v_static) * (Ccap - \Sigma ci) / Ccap;$$

Όπου το α είναι σταθερά στο διάστημα (0,1). Όσο μειώνεται ο πράκτορας μας τόσο πιο συντηρητικά παίζει, και εμείς το έχουμε ορίσει στο 0,3 καθώς είναι η βέλτιστη τιμή του για πραγματικούς αντίπαλους,.

- Αν $v_dynamic \leq 0$ που σημαίνει πως αναμένουμε να έχουμε αρνητικά κέρδη για το συγκεκριμένο query, δηλαδή έσοδα – έξοδα < 0 , τότε κάνουμε bid ξανά βάση της υβριδικής σχέσης

$$bid = (\alpha * v_static) * (Ccap - \Sigma ci) / Ccap;$$

- Διαφορετικά, κάνουμε bid βάση του $v_dynamic$, U , R και έχουμε

$$bid = v_dynamic * U * R;$$

Άρα γενικά κινούμαστε γύρω από εκτιμήσεις του VPC, είτε στατικά όπως περιγράφονται στο v_static , είτε δυναμικά βασιζόμενοι στο τι έγινε τις προηγούμενες ημέρες όπως αναφέρεται στο $v_dynamic$.

Τι πήγε καλά και τι όχι;

Η χρήση ενός στατικού και ενός δυναμικού “V” είναι κάτι που βελτίωσε τα κέρδη του πράκτορα. Το πρώτο για τις 5 πρώτες μέρες (προσοχή στη χρήση του, καλό θα ήταν να χρησιμοποιηθεί για λίγες μέρες, όχι πάνω από 10). Το δυναμικό “V” είναι πιθανό να χρειάζεται κάποια διόρθωση στις παραμέτρους, ίσως μέσα από κάποια ακόμη πειράγματα, ώστε να δουλέψει λίγο καλύτερα. Η απουσία των spend limits σίγουρα δεν είχε θετικά αποτελέσματα, όταν ο πράκτορας διατηρεί μια πολιτική χωρίς να κοιτάει το budget του οδηγούμαστε σε λάθος εκτιμήσεις. Όταν ο πράκτορας ξεπερνάει κάποια όριο εξόδων είναι καλό να κάνουμε τον πράκτορα περισσότερο συντηρητικό. Όσον αφορά τα βάρη που χρησιμοποιούμε για να συνδέσουμε το capacity και τα conversions των 5 προηγούμενων ημερών θα ήταν καλύτερο να χρησιμοποιήσουμε $w \leq 0.4$. Αν και για dummies ο Captain Sparrow παίζει καλύτερα για χαμηλότερη τιμή του w ενώ σε μάχη ένας προς έναν το $w = 0.7$ δουλεύει καλά, όταν βρεθεί ο πράκτορας σας σε μία μάχη που δεν είναι όλοι μόνο dummies ή μάχη ένας προς έναν το βάρος αυτό δεν είναι καθόλου καλό. Τα πειράγματα στα οποία αναφερόμαστε στη σελίδα 16 δεξήχθησαν ως εξής. Ο πράκτορας μας αντιμετώπιζε μόνο dummies ή μόνο έναν αντίπαλο δύσκολου επιπέδου. Στο διαγωνισμό όμως το περιβάλλον θα είναι ανταγωνιστικό αλλά και εκούσια συνεργαστικό (σκεφτείτε πως όταν κάποιος από τους πράκτορες μάχονται ενάντια σε έναν πράκτορα που τα πάει πολύ καλά αυτό είναι θετικό για εσάς, αρκεί να μην είστε εσείς στη θέση αυτού που τα πάει καλά!)

FUTURE WORK

Ως εξέλιξη της παρούσας εργασίας, θα μπορούσαμε να προσθέσουμε κάποια τεχνική πρόβλεψης του τι θα κάνουν οι αντίπαλοι μελετώντας τις κινήσεις τους παρελθοντικά. Μία τεχνική που θα μπορούσε να είναι η ενσωμάτωση κάποιου Particle Filter. Επίσης η χρήση spend limits στα κατάλληλα query. Ίσως να να χρειαστούν κάποια πειράγματα για την επιλογή των σωστών spend limits. Επιπλέον μία αποδοτική λύση είναι η χρήση ενισχυτικής μάθησης, όχι τόσο στα πλαίσια εσωτερικού διαγωνισμού όσο για τον διαγωνισμό tac ad auction. Δυστυχώς δεν μπορέσαμε να υλοποιήσουμε τα παραπάνω λόγω έλλειψης χρόνου.

References

- [Born to Trade : a genetically evolved keyword bidder for sponsored search](#)
 - [Designing a Successful Adaptive Agent for TAC Ad Auction](#)
 - [TAC-AA software](#)
- [TacTex09: A champion bidding agent for ad auctions. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems \(AAMAS 2010\), May 2010](#)
- http://www.aamas-conference.org/Proceedings/aamas2011/papers/D6_G69.pdf
 - <http://www.cs.utexas.edu/~dpardoe/papers/TacTexTR.pdf>
- [Yevgeniy Vorobeychik. Simulation-based game theoretic analysis of keyword auctions with low-dimensional bidding strategies. In Proceedings of Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, 2009](#)
- [Yevgeniy Vorobeychik. A Game Theoretic Bidding Agent for the Ad Auction Game. In Proceedings of the Third International Conference on Agents and Artificial Intelligence \(ICAART 2011\), January 2011](#)
 - <http://vorobeychik.com/2011/qt.pdf>
- [Jordan Berg, Amy Greenwald, Victor Naroditskiy, and Eric Sodomka. A knapsack-based approach to bidding in ad auctions](#)
- [Jordan Berg, Amy Greenwald, Victor Naroditskiy, and Eric Sodomka. A First Approach to Autonomous Bidding in Ad Auctions. In Proceedings of TADA 2010](#)
- [Meng Chang, Minghua He, and Xudong Luo. Designing a successful adaptive agent for tac ad auction. In Proceedings of the 19th European Conference on Artificial Intelligence \(ECAI 2010\)](#)

- <http://users.ecs.soton.ac.uk/nrj/download-files/TOIT-minghua.pdf>
- http://vivliothmmy.ee.auth.gr/1072/1/Diploma_Thesis_Skantza_Tzamtzi.pdf
- [Keyword Bidding in Sponsored Search Using an Estimation of Distribution Algorithm](#)
- [M. Munsey, J. Veilleux, S. Bikkani, A. Teredesai, and M. De Cock. Born to trade: A genetically evolved keyword bidder for sponsored search. In Proceedings of the 2010 IEEE Congress on Evolutionary Computation \(CEC\), pages 1–8, 2010](#)