



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

**A Mechanism Design and Deep Learning Approach for
Revenue Maximization on Cloud Dynamic Resource
Provisioning Systems**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αστέριος Τσιούρβας

Επιβλέπων:
Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019



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

A Mechanism Design and Deep Learning Approach for Revenue Maximization on Cloud Dynamic Resource Provisioning Systems

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Τσιούρβα Αστέριου

Επιβλέπων: Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή τη 16η Οκτωβρίου 2019.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Νεκτάριος Κοζύρης
Καθηγητής
Ε.Μ.Π.

.....
Ανδρέας Σταφυλοπάτης
Καθηγητής
Ε.Μ.Π.

.....
Δημήτριος Φωτάκης
Επίχ. Καθηγητής
Ε.Μ.Π.

Αθήνα, Οκτώβριος 2019

.....
ΑΣΤΕΡΙΟΣ ΤΣΙΟΥΡΒΑΣ

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

Copyright © Αστέριος Τσιούρβας, 2019.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί οι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Τα τελευταία χρόνια παρατηρείται σημαντική αύξηση στην ανάγκη για την επεξεργασία μεγάλου όγκου δεδομένων (Big Data). Για την επεξεργασία αυτών αναπτύχθηκαν νέες τεχνολογίες οι οποίες στις περισσότερες περιπτώσεις φιλοξενούνται σε διαμοιραζόμενους υπολογιστικούς πόρους, στα λεγόμενα υπολογιστικά νέφη.

Οι περισσότεροι οργανισμοί που διαχειρίζονται μεγάλο όγκο δεδομένων επιλέγουν να μεταφέρουν τις εγκαταστάσεις τους από τοπικούς υπολογιστές και servers του οργανισμού, στα υπολογιστικά νέφη. Με τον τρόπο αυτό εξοικονομούν κέρδη, λαμβάνουν δυνατότητα κλιμάκωσης, διατηρούν τα δεδομένα τους ασφαλή και έχουν κάθε στιγμή την απαραίτητη τεχνική υποστήριξη από τον πάροχο.

Μία υπηρεσία που παρουσίασε ραγδαία άνοδο τα τελευταία χρόνια είναι τα Spot Instances της Amazon. Με την υπηρεσία αυτή η Amazon θέτει σε χαμηλές τιμές αχρησιμοποίητες εικονικές μηχανές της. Οι χρήστες κάνουν τις προσφορές τους και λαμβάνουν τις μηχανές. Σε περίπτωση που προσφορά κάποιου νέου χρήστη είναι υψηλότερη από την τρέχουσα για τον ίδιο υπολογιστικό πόρο, τότε ο παλιός χρήστης ειδοποιείται και χάνει την εικονική μηχανή του. Μερικές εταιρείες που χρησιμοποιούν τη συγκεκριμένη υπηρεσία είναι: AOL, BloomReach, Mapbox, Moovit, Zillow, Novartis κ.α.

Στο πλαίσιο της παρούσας διπλωματικής εργασίας μελετάμε το σύστημα κατανομής υπολογιστικών πόρων και καθορισμού τιμών για το σύστημα Spot Instances. Κύριος στόχος μας είναι να δημιουργήσουμε ένα σύστημα δυναμικής τιμολόγησης που θα μεγιστοποιεί τα έσοδα του παρόχου, ενώ παράλληλα θα δίνει κάποιες παραπάνω δυνατότητες στο χρήστη.

Προκειμένου να λύσουμε το πρόβλημα βασιζόμαστε στην Αλγοριθμική Θεωρία Παιγνίων και πιο συγκεκριμένα στη Θεωρία του Σχεδιασμού Μηχανισμών και στις δημοπρασίες. Χρησιμοποιούμε προσεγγιστικούς αλγόριθμους, στοιχεία θεωρίας πιθανοτήτων για εκτιμήσεις παραμέτρων και νευρωνικά δίκτυα.

Ελέγχουμε τις προσεγγίσεις τόσο σε συνθετικά δεδομένα όσο και σε πραγματικά. Στο τέλος καταλήγουμε σε μία πρόταση, η οποία δείχνει να είναι η πλέον συμφέρουσα με βάση πάντα τα αποτελέσματα των προσομοιώσεων, τα οποία παρουσιάζουμε αναλυτικά.

Στο τέλος της εργασίας, πραγματοποιούμε μία σύνοψη των πεπραγμένων μας ενώ παρουσιάζουμε πιθανές μελλοντικές επεκτάσεις της δουλειάς μας.

Λέξεις Κλειδιά

Υπολογιστικό Νέφος, Spot Instances, Αλγοριθμική Θεωρία Παιγνίων, Σχεδιασμός Μηχανισμών, Δημοπρασίες, Προσεγγιστικοί Αλγόριθμοι, Εκτίμηση Παραμέτρων, Μηχανική Μάθηση, Νευρωνικά Δίκτυα

Abstract

Over the last years, we have witnessed a severe increment in the use and processing of Big Data. For the process of Big Data, several new technologies have emerged. The majority of these new technologies relies on shared computing resources, on the so called Cloud Services.

The vast majority of corporations that manipulate Bid Data rely on shared computing resources instead of their own local servers and personal devices. This approach benefits modern enterprises due to the diminish of operational costs, the scalability of cloud systems, the safety of their private and critical data and the consistent customer support that they receive.

One cloud service that has increased its popularity over the years is Amazon Spot Instances. In this cloud service, Amazon sells unused virtual machines at low prices. Users that bid enough for these machines occupy them. In the case where, a new user bids a larger amount of money for the virtual machine of an old user, then the old user is warned about an interruption and its virtual machine is granted to the new user. Examples of modern enterprises that use Spot Instances are: AOL, BloomReach, Mapbox, Moovit, Zillow, Novartis and others.

In this specific diploma thesis we study the allocation and pricing system for the system of Spot Instances. Our main goal is to create a new dynamic allocation and pricing system that maximizes profit for the cloud operator while it is fair and offers new capabilities to users.

In order to solve this problem, our approach is based on Algorithmic Game Theory and specifically to Mechanism Design and Auction Theory. We use approximation algorithms, elements of probability theory for estimation of parameters and neural networks in order to solve the problem.

Our approaches are tested both on synthetic and real data of cloud services. In the end of the thesis, we present our final proposition, which is the most efficient based on the simulation results.

Finally, we sum up our thesis and present future possible expansions of the our work.

Key Words

Cloud Computing, Spot Instances, Algorithmic Game Theory, Mechanism Design, Auction Theory, Approximation Algorithms, Estimation of Parameters, Machine Learning, Neural Networks

Ευχαριστίες

Ολοκληρώνοντας αυτή την εργασία θα ήθελα αρχικά να ευχαριστήσω τον καθηγητή κ. Νεκτάριο Κοζύρη, ο οποίος έδειξε εμπιστοσύνη προς το πρόσωπό μου δίνοντας μου την ευκαιρία να δουλέψω υπό την επίβλεψη του πάνω σε ένα ξεχωριστό αντικείμενο που συνδυάζει πολλαπλούς επιστημονικούς κλάδους υπό την καθοδήγηση εξαιρετικών συνεργατών.

Θα ήθελα να ευχαριστήσω τον μεταδιδακτορικό ερευνητή κ. Ιωάννη Κωνσταντίνου ο οποίος ήταν συνεχώς δίπλα μου για να με καθοδηγεί, να μου προτείνει νέες ιδέες και να με ωθεί στο να ξεπερνώ τα όριά μου και να εξελίσσομαι επιστημονικά.

Θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Κωνσταντίνο Μπισάκο ο οποίος υποστήριξε θερμά από την πρώτη στιγμή τις ιδέες μου και ήταν δίπλα μου αδιάκοπα συνεισφέροντας με κάθε τρόπο γεγονός για το οποίο αισθάνομαι ευγνώμων.

Θεωρώ πως η βοήθεια, η υποστήριξη και η συνεργασία που είχα κατά τη διάρκεια αυτής της εργασίας ήταν εξαιρετική από κάθε άποψη και πιστεύω πως δε θα μπορούσα να έχω καλύτερη.

Καθώς η συγκεκριμένη διπλωματική εργασία σηματοδοτεί το τέλος ενός σημαντικού κύκλου της ζωής μου και για το λόγο αυτό θα ήθελα να ευχαριστήσω την οικογένεια μου για τις αξίες που μου μετέδωσε και για την αδιάκοπη υποστήριξη που μου έχει προσφέρει καθ' όλη τη διάρκεια της ζωής μου.

Τέλος, θα ήθελα να ευχαριστήσω τον καλό μου φίλο Θανάση Λεραί για τις χρήσιμες συμβουλές του ως προς τη διάρθρωση του συγκεκριμένου τόμου.

Αστέριος Τσιούρβας
Αθήνα, 16^η Οκτωβρίου 2019

Περιεχόμενα

| | |
|--|-----------|
| Κατάλογος Σχημάτων | 7 |
| Κατάλογος Πινάκων | 10 |
| 1 Εισαγωγή | 12 |
| 1.1 Αντικείμενο της εργασίας | 13 |
| 1.2 Οργάνωση του τόμου | 14 |
| 2 Το Σύστημα Spot Instances | 15 |
| 2.1 Περιγραφή του Συστήματος | 16 |
| 2.1.1 Η Ιδέα | 16 |
| 2.1.2 Βασικές Έννοιες | 16 |
| 2.1.3 Διαφορές Μεταξύ Spot Instances και Κανονικών Εικονικών Μηχανών | 17 |
| 2.1.4 Στρατηγικές Χρήστης και Αρχιτεκτονική Συστήματος | 17 |
| 2.1.5 Παρόμοιες Υπηρεσίες | 18 |
| 2.2 Το Κίνητρό μας | 19 |
| 2.2.1 Πλεονεκτήματα και Μειονεκτήματα | 19 |
| 2.2.2 Η Ιδέα μας | 19 |
| 3 Σχεδιασμός Μηχανισμών και Δημοπρασίες | 21 |
| 3.1 Εισαγωγή στη Θεωρία Παιγνίων | 22 |
| 3.1.1 Βασικές Κατηγορίες Παιγνίων | 23 |
| 3.1.2 Ισοροπίες Nash | 23 |
| 3.2 Εισαγωγή στο Σχεδιασμό Μηχανισμών | 25 |
| 3.2.1 Κοινωνική Επιλογή | 25 |
| 3.3 Δημοπρασίες με Χρήματα - Ενός Αντικειμένου | 26 |
| 3.3.1 Μηχανισμοί με Χρήματα | 26 |
| 3.3.2 Δημοπρασίες Ενός Αντικειμένου | 26 |
| 3.3.3 Μοντέλο Σφραγισμένων Προσφορών | 27 |
| 3.4 Δημοπρασίες First και Second Price | 28 |
| 3.4.1 Φιλαλήθεια | 28 |
| 3.4.2 Δημοπρασίες First Price | 28 |
| 3.4.3 Δημοπρασίες Second Price | 28 |
| 3.4.4 Ιδανικές Δημοπρασίες | 29 |
| 3.5 Λήμμα του Myerson | 31 |
| 3.5.1 Περιβάλλον Μοναδικής Παραμέτρου | 31 |
| 3.5.2 Κατανομή και Κανόνες Πληρωμής | 31 |
| 3.5.3 Διατύπωση και Απόδειξη Λήμματος Myerson | 32 |
| 3.6 Δημοπρασίες Knapsack | 35 |
| 3.6.1 Ορισμός του Προβλήματος | 35 |
| 3.6.2 Μειστοποίηση Ωφέλειας σε DSIC Δημοπρασίες Knapsack | 35 |
| 3.7 Δημοπρασίες Μεγιστοποίησης Εσόδων | 36 |
| 3.7.1 Μπεϋζιανή Ανάλυση | 36 |

| | | |
|----------|---|-----------|
| 3.7.2 | Τυπικός Ορισμός | 37 |
| 3.7.3 | Virtual Valuations | 37 |
| 3.7.4 | Αναμενόμενα Έσοδα και Αναμενόμενη Εικονική Ωφέλεια | 38 |
| 3.7.5 | Μεγιστοποίηση Αναμενόμενης Εικονικής Ωφέλειας | 39 |
| 3.8 | Τυχαίες Δημοπρασίες | 41 |
| 3.8.1 | Ορισμός | 41 |
| 3.8.2 | Φιλαλήθεια στις Τυχαίες Δημοπρασίες | 41 |
| 4 | Τεχνικές Εκτίμησης Κατανομών Προσωπικής Αξίας | 43 |
| 4.1 | Κλασική Εκτίμηση Παραμέτρων | 44 |
| 4.1.1 | Εκτιμητρίες Συναρτήσεις | 44 |
| 4.1.2 | Εκτίμηση Μέγιστης Πιθανοφάνειας | 45 |
| 4.1.3 | Εκτίμηση της Μέσης Τιμής και Διακύμανσης μίας Τυχαίας Μεταβλητής | 46 |
| 4.2 | Lognormal Κατανομή | 47 |
| 4.2.1 | Ορισμός | 47 |
| 4.2.2 | Εκτιμητής Μέγιστης Πιθανοφάνειας | 48 |
| 4.3 | Αλγόριθμος Expectation-Maximization (EM) | 49 |
| 4.3.1 | Ορισμός | 49 |
| 4.3.2 | Αλγόριθμος Expectation-Maximization για Δημοπρασίες | 51 |
| 4.4 | Απόσταση Kullback–Leibler | 52 |
| 4.4.1 | Εντροπία | 52 |
| 4.4.2 | Ορισμός Απόστασης Kullback–Leibler | 53 |
| 4.5 | Νευρωνικά Δίκτυα | 54 |
| 4.5.1 | Το Perceptron | 54 |
| 4.5.2 | Νευρωνικά Δίκτυα Πολλαπλών Στρωμάτων | 56 |
| 4.5.3 | Εκπαίδευση Νευρωνικού Δικτύου Πολλαπλών Στρωμάτων | 57 |
| 4.6 | Περιγραφή Αλγορίθμων Βελτιστοποίησης Gradient Descent | 60 |
| 4.6.1 | Batch Gradient Descent | 60 |
| 4.6.2 | Stochastic Gradient Descent | 60 |
| 4.6.3 | Mini-Batch Gradient Descent | 61 |
| 4.6.4 | Adaptive Moment Estimation (Adam) | 61 |
| 4.7 | Νευρωνικό Δίκτυο για Δημοπρασίες | 63 |
| 4.7.1 | Η Δομή του Δικτύου | 63 |
| 5 | Προσεγγίσεις για την Επίλυση του Προβλήματος | 66 |
| 5.1 | Μεγιστοποίηση Κοινωνικής Ωφέλειας | 67 |
| 5.1.1 | Αλγόριθμοι Μεγιστοποίησης Κοινωνικής Ωφέλειας | 67 |
| 5.2 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών | 68 |
| 5.2.1 | Προσεγγιστικός Αλγόριθμος για Limited Supply | 68 |
| 5.2.2 | Τυχαίες Δημοπρασίες για Unimited Supply | 69 |
| 5.3 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας | 71 |
| 5.3.1 | Αλγόριθμοι Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπική Αξίας | 71 |
| 5.4 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων | 72 |
| 5.4.1 | Αλγόριθμοι Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων | 72 |
| 6 | Πειραματική Αξιολόγηση | 74 |
| 6.1 | Περιβάλλον Προσομοίωσης | 75 |
| 6.1.1 | Παραμετροποίηση | 75 |
| 6.1.2 | Κατανομές Προσφορών και Αιτημάτων Χρηστών | 75 |
| 6.1.3 | Σύνολο Δεδομένων Google Trace | 76 |
| 6.1.4 | Μετρικές Αξιολόγησης | 77 |
| 6.2 | Αποτελέσματα Μεγιστοποίησης Κοινωνικής Ωφέλειας | 78 |
| 6.2.1 | Πρώτη Προσέγγιση | 78 |
| 6.2.2 | Δεύτερη Προσέγγιση | 85 |

| | | |
|----------|---|------------|
| 6.3 | Αποτελέσματα Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών | 92 |
| 6.3.1 | Πρώτη Προσέγγιση | 92 |
| 6.3.2 | Δεύτερη Προσέγγιση | 98 |
| 6.4 | Αποτελέσματα Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας | 104 |
| 6.4.1 | Πρώτη Προσέγγιση | 104 |
| 6.4.2 | Δεύτερη Προσέγγιση | 111 |
| 6.5 | Αποτελέσματα Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων | 117 |
| 6.5.1 | Πρώτη Προσέγγιση | 117 |
| 6.5.2 | Δεύτερη Προσέγγιση | 124 |
| 7 | Επίλογος | 131 |
| 7.1 | Σύνοψη και Συμπεράσματα | 132 |
| 7.2 | Μελλοντική Εργασία | 133 |
| 8 | Βιβλιογραφία | 134 |

Κατάλογος Σχημάτων

| | | |
|------|--|----|
| 2.1 | Πρώτη Στρατηγική Χρήσης της Υπηρεσίας Spot Instances. | 17 |
| 2.2 | Σχηματικά η Αρχιτεκτονική της Υπηρεσίας Spot Instances. | 18 |
| 2.3 | Σχηματικά η Επιθυμητή για εμάς Αρχιτεκτονική της Υπηρεσίας Spot Instances. | 20 |
| 3.1 | Παράδειγμα καμπυλών κατανομής $x(\cdot)$. Πηγή: [20] | 33 |
| 3.2 | Επικουρικές εικόνες για περιγραφή απόδειξης του Λήμματος Myerson. | 34 |
| 4.1 | Συνάρτηση πυκνότητας πιθανότητας Lognormal | 47 |
| 4.2 | Συνάρτηση κατανομής πιθανότητας Lognormal | 48 |
| 4.3 | Σχηματική Αναπαράσταση Νευρικού Συστήματος | 54 |
| 4.4 | Perceptron | 54 |
| 4.5 | Παράδειγμα Multilayer Perceptron με 3 Κρυφά Επίπεδα και Ένα Κόμβο Εξόδου | 57 |
| 4.6 | Πορεία Αλγορίθμου Gradient Descent σε Τρισδιάστατη Κυρτή Συνάρτηση. | 58 |
| 4.7 | Νευρωνικό δίκτυο που εκτιμά και προσεγγίζει τη συνάρτηση virtual valuation $\bar{\phi}_i(b_i)$ για τον στρατηγικό παίχτη i . Πηγή: [8] | 64 |
| 4.8 | Υλοποίηση πιθανοτικού κανόνα κατανομής. Πηγή: [8] | 64 |
| 4.9 | Υλοποίηση κανόνα πληρωμής t . Πηγή: [8] | 65 |
| 4.10 | Συνολικό νευρωνικό δίκτυο. Πηγή: [8] | 65 |
| 6.1 | Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Binary Knapsack | 78 |
| 6.1 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Binary Knapsack | 79 |
| 6.2 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Έσοδα - Ημιτονικές Αιτήσεις - Binary Knapsack | 79 |
| 6.3 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Χρησιμοποίηση CPU - Ημιτονικές Αιτήσεις - Binary Knapsack | 80 |
| 6.4 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Λόγος Προσέγγισης - Ημιτονικές Αιτήσεις - Binary Knapsack | 81 |
| 6.5 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Fractional Knapsack | 82 |
| 6.6 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Έσοδα - Ημιτονικές Αιτήσεις - Fractional Knapsack | 83 |
| 6.7 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Έσοδα - Ημιτονικές Αιτήσεις - Fractional Knapsack | 84 |
| 6.8 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Binary Knapsack | 85 |
| 6.9 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Έσοδα - Google Trace Αιτήσεις - Binary Knapsack | 86 |
| 6.10 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Binary Knapsack | 87 |
| 6.11 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Λόγος Προσέγγισης - Google Trace Αιτήσεις - Binary Knapsack | 88 |
| 6.12 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Fractional Knapsack | 89 |
| 6.13 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Έσοδα - Google Trace Αιτήσεις - Fractional Knapsack | 90 |
| 6.14 | Μεγιστοποίηση Κοινωνικής Ωφέλειας - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Fractional Knapsack | 91 |

| | | |
|------|---|-----|
| 6.15 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Binary Knapsack | 92 |
| 6.15 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Binary Knapsack | 93 |
| 6.16 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Έσοδα - Ημιτονικές Αιτήσεις - Binary Knapsack | 93 |
| 6.17 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Χρησιμοποίηση CPU - Ημιτονικές Αιτήσεις - Binary Knapsack | 94 |
| 6.18 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Fractional Knapsack | 95 |
| 6.19 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Έσοδα - Ημιτονικές Αιτήσεις - Fractional Knapsack | 96 |
| 6.20 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Χρησιμοποίηση CPU - Ημιτονικές Αιτήσεις - Fractional Knapsack | 97 |
| 6.21 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Binary Knapsack | 98 |
| 6.22 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Έσοδα - Google Trace Αιτήσεις - Binary Knapsack | 99 |
| 6.23 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Binary Knapsack | 100 |
| 6.24 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Fractional Knapsack | 101 |
| 6.25 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Έσοδα - Google Trace Αιτήσεις - Fractional Knapsack | 102 |
| 6.26 | Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Fractional Knapsack | 103 |
| 6.27 | Εκτιμήσεις της κατανομής προσωπικής αξίας των χρηστών | 104 |
| 6.28 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Binary Knapsack | 105 |
| 6.29 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Έσοδα - Ημιτονικές Αιτήσεις - Binary Knapsack | 106 |
| 6.30 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Χρησιμοποίηση CPU - Ημιτονικές Αιτήσεις - Binary Knapsack | 107 |
| 6.31 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Fractional Knapsack | 108 |
| 6.32 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Έσοδα - Ημιτονικές Αιτήσεις - Fractional Knapsack | 109 |
| 6.33 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Χρησιμοποίηση CPU - Ημιτονικές Αιτήσεις - Fractional Knapsack | 110 |
| 6.34 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Binary Knapsack | 111 |
| 6.35 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Έσοδα - Google Trace Αιτήσεις - Binary Knapsack | 112 |
| 6.36 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Binary Knapsack | 113 |
| 6.37 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Fractional Knapsack | 114 |
| 6.38 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Έσοδα - Google Trace Αιτήσεις - Fractional Knapsack | 115 |
| 6.39 | Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Fractional Knapsack | 116 |
| 6.40 | Εκτιμήσεις των virtual valuations των πρώτων 20 χρηστών από το νευρωνικό δίκτυο | 117 |
| 6.41 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Binary Knapsack | 118 |

| | | |
|------|--|-----|
| 6.42 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Έσοδα - Ημιτονικές Αιτήσεις - Binary Knapsack | 119 |
| 6.43 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Χρησιμοποίηση CPU - Ημιτονικές Αιτήσεις - Binary Knapsack | 120 |
| 6.44 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Κοινωνική Ωφέλεια - Ημιτονικές Αιτήσεις - Fractional Knapsack | 121 |
| 6.45 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Έσοδα - Ημιτονικές Αιτήσεις - Fractional Knapsack | 122 |
| 6.46 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Χρησιμοποίηση CPU - Ημιτονικές Αιτήσεις - Fractional Knapsack | 123 |
| 6.47 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Binary Knapsack | 124 |
| 6.48 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Έσοδα - Google Trace Αιτήσεις - Binary Knapsack | 125 |
| 6.49 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Binary Knapsack | 126 |
| 6.50 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Κοινωνική Ωφέλεια - Google Trace Αιτήσεις - Fractional Knapsack | 127 |
| 6.51 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Έσοδα - Google Trace Αιτήσεις - Fractional Knapsack | 128 |
| 6.52 | Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων - Χρησιμοποίηση CPU - Google Trace Αιτήσεις - Fractional Knapsack | 129 |

Κατάλογος Πινάκων

| | | |
|------|---|-----|
| 2.1 | Διαφορές Μεταξύ Spot Instances και Κανονικών Εικονικών Μηχανών | 17 |
| 3.1 | Το δίλημμα του φυλακισμένου | 22 |
| 3.2 | Πίνακας κόστους για αυτοκίνητα που διασταυρώνονται | 24 |
| 4.1 | Ιδιότητες Lognormal Κατανομής | 47 |
| 4.2 | Βασικές Συναρτήσεις Ενεργοποίησης | 55 |
| 6.1 | Βασικές Παράμετροι Προσομοίωσης | 75 |
| 6.2 | Κατανομές Προσφορών και Αιτημάτων Χρηστών | 76 |
| 6.3 | Χαρακτηριστικά Amazon EC2 Instance Type a1.medium | 77 |
| 6.4 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας | 81 |
| 6.5 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας | 84 |
| 6.6 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας | 88 |
| 6.7 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας | 91 |
| 6.8 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών | 94 |
| 6.9 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών | 97 |
| 6.10 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών | 100 |
| 6.11 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών | 103 |
| 6.12 | Αποτελέσματα Εκτίμησης της Κατανομής Προσωπικής Αξίας των Χρηστών | 104 |
| 6.13 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας. | 107 |
| 6.14 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας. | 110 |
| 6.15 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας. | 113 |
| 6.16 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας. | 116 |
| 6.17 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων | 120 |
| 6.18 | Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων | 123 |
| 6.19 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων | 126 |

| | | |
|------|--|-----|
| 6.20 | Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων | 129 |
|------|--|-----|

Κεφάλαιο 1

Εισαγωγή

Στο κεφάλαιο αυτό πραγματοποιείται μία σύντομη εισαγωγή στην παρούσα διπλωματική εργασία. Παρουσιάζεται το αντικείμενο της εργασίας, οι απαραίτητες γνώσεις για την κατανόησή της καθώς και η συνολική δομή της.

Στην ενότητα [1.1](#) παρουσιάζεται το αντικείμενο της εργασίας καθώς και οι βασικές γνώσεις που απαιτούνται για την κατανόηση αυτής.

Στην ενότητα [1.2](#) παρουσιάζεται η συνολική δομή της εργασίας κάνοντας αναφορά σε κάθε ένα κεφάλαιο ξεχωριστά.

1.1 Αντικείμενο της εργασίας

Η συγκεκριμένη διπλωματική εργασία εστιάζει στην πειραματική αξιολόγηση τεσσάρων διαφορετικών μοντέλων κατανομής υπολογιστικών πόρων και καθορισμού τιμών πώλησης για την υπηρεσία Spot Instances της Amazon. Για την κατανόηση των προσεγγίσεων απαιτούνται γνώσεις από διαφορετικούς επιστημονικούς κλάδους.

Αρχικά, θα πρέπει να γίνει μία πλήρης εισαγωγή στο σύστημα Spot Instances, στις βασικές έννοιες του και στην αρχιτεκτονική του. Στη συνέχεια, θα πρέπει να κατανοηθούν βασικά πλεονεκτήματα και μειονεκτήματα του συστήματος.

Έπειτα, μελετώνται ξεχωριστά διάφορα είδη δημοπρασιών με έμφαση σε φιλαλήθεις δημοπρασίες. Ειδικότερα γίνεται μία μελέτη στις δημοπρασίες τύπου Knapsack, στο Λήμμα του Myerson, στις δημοπρασίες μεγιστοποίησης εσόδων καθώς και στις τυχαίες δημοπρασίες. Ιδιαίτερα οι δημοπρασίες τύπου Knapsack συνδέονται άρρηκτα με το ζήτημα, καθώς μπορούν να μοντελοποιήσουν το πρόβλημα θεωρώντας πως χρησιμοποιείται ένας τύπος εικονικής μηχανής.

Για την εκτίμηση των κατανομών προσωπικής αξίας είναι απαραίτητη η κατανόηση βασικών εννοιών από τη θεωρία πιθανοτήτων που αφορά την κλασική εκτίμηση παραμέτρων καθώς και του αλγορίθμου Expectation - Maximization. Επιπλέον, ιδιαίτερα χρήσιμη είναι η γνώση της εντροπίας από τη Θεωρία Πληροφορίας.

Για την ανάλυση των δεδομένων εξαιρετικά χρήσιμες είναι οι γνώσεις μηχανικής μάθησης. Για το λόγο αυτό, γίνεται εισαγωγή στα νευρωνικά δίκτυα και στην εκπαίδευσή τους με τη βοήθεια του αλγορίθμου back-propagation και της μεθόδου gradient descent.

Η ανάλυση δεδομένων και οι προσομοιώσεις πραγματοποιήθηκαν σε γλώσσα προγραμματισμού Python. Οι παράμετροι που επιλέχθηκαν καθώς και το σύνολο δεδομένων που χρησιμοποιήθηκε είναι επιλεγμένα με βάση διεθνείς εργασίες με σκοπό τις πλέον ρεαλιστικές προσομοιώσεις. Τα αποτελέσματα συγκρίθηκαν μεταξύ τους και με βάση ορισμένες υποθέσεις επιλέχθηκε το καλύτερο κατά την άποψή μας μοντέλο για την επίλυση του προβλήματος.

1.2 Οργάνωση του τόμου

Η παρούσα διπλωματική εργασία χωρίζεται σε 6 κεφάλαια.

Στο **Κεφάλαιο 1** πραγματοποιούμε μία εισαγωγή στο αντικείμενο της εργασίας και παρουσιάζεται η συνολική δομή της διπλωματικής.

Στο **Κεφάλαιο 2** παρουσιάζουμε το σύστημα Spot Instances της Amazon το οποίο αποτέλεσε έναυσμα για την παρούσα διπλωματική εργασία. Πιο συγκεκριμένα, περιγράφουμε την ιδέα πίσω από το σύστημα, τη δομή του, βασικές στρατηγικές χρήσης του και παρόμοιες υπηρεσίες. Επιπλέον, αναφέρουμε τα μειονεκτήματα και τα πλεονεκτήματα του από τη δική μας σκοπιά και παρουσιάζουμε τις ιδέες μας για την αντιμετώπιση κάποιων μειονεκτημάτων.

Στο **Κεφάλαιο 3** παρουσιάζουμε το θεωρητικό υπόβαθρο του σχεδιασμού μηχανισμών που απαιτείται για την κατανόηση της διπλωματικής εργασίας. Πιο συγκεκριμένα, γίνεται εισαγωγή στη θεωρία παιγνίων, στο σχεδιασμό μηχανισμών και στη συνέχεια πραγματοποιείται παρουσίαση των βασικών δημοπρασιών με χρήματα, εστιάζοντας στο λήμμα του Myerson και αναλύοντας τις δημοπρασίες τύπου Knapsack, τις δημοπρασίες Μεγιστοποίησης Κερδών και τις Τυχαίες δημοπρασίες.

Στο **Κεφάλαιο 4** παρουσιάζουμε το θεωρητικό υπόβαθρο πάνω στο οποίο βασιστήκαμε για την εκτίμηση των κατανομών προσωπικής αξίας των στρατηγικών χρηστών που ποντάρουν στην αγορά των εικονικών μηχανών. Πιο συγκεκριμένα, κάναμε μία εισαγωγή στην κλασική εκτίμηση παραμέτρων, παρουσιάσαμε τη βασική κατανομή lognormal, μία τροποποίηση του αλγορίθμου Expectation-Maximization για δημοπρασίες, την απόσταση Kullback-Leibler ενώ εστίασαμε στα νευρωνικά δίκτυα και στους αλγορίθμους βελτιστοποίησής τους, παρουσιάζοντας ένα νευρωνικό δίκτυο το οποίο με κατάλληλη τροποποίηση πραγματοποιεί μεγιστοποίηση κερδών για δημοπρασίες τύπου Knapsack που μοντελοποιούν την αγορά των εικονικών μηχανών.

Στο **Κεφάλαιο 5** παρουσιάζουμε τις προσεγγίσεις μας για την επίλυση του προβλήματος για την κατανομή των εικονικών μηχανών στους χρήστες με απώτερο στόχο τη μεγιστοποίηση των εσόδων για τον πωλητή/πάροχο, αναφέροντας τέσσερις διαφορετικές προσεγγίσεις. Πιο συγκεκριμένα, η πρώτη μας προσέγγιση είναι μη ρεαλιστική και γίνεται για λόγους πληρότητας καθώς μεγιστοποιεί την κοινωνική ωφέλεια του συστήματος. Η δεύτερη προσέγγιση μεγιστοποιεί τα έσοδα του παρόχου με χρήση ενός προσεγγιστικού αλγορίθμου και με τη βοήθεια τυχαίων δημοπρασιών. Η τρίτη προσέγγιση μεγιστοποιεί τα έσοδα του πωλητή εκτιμώντας την πιθανοτική κατανομή της προσωπικής αξίας του χρήστη. Τέλος, η τέταρτη προσέγγιση αποτελεί ένα υβριδικό σύστημα το οποίο μεγιστοποιεί τα έσοδα του πωλητή χρησιμοποιώντας νευρωνικό δίκτυο και τυχαίες δημοπρασίες.

Στο **Κεφάλαιο 6** παρουσιάζουμε τα πειραματικά αποτελέσματά μας για τις προσεγγίσεις μας για την επίλυση του προβλήματος για την κατανομή των εικονικών μηχανών στους χρήστες με απώτερο στόχο τη μεγιστοποίηση των εσόδων για τον πωλητή/πάροχο. Πιο συγκεκριμένα, αρχικά περιγράφουμε το περιβάλλον προσομοίωσης, ορίζοντας παραμέτρους και κατανομές προσφορών/αιτημάτων χρηστών, παρουσιάζουμε το σύνολο δεδομένων της προσομοίωσης και αναφέρουμε τις μετρικές που χρησιμοποιούνται για την πειραματική αξιολόγηση. Στη συνέχεια, παραθέτουμε τα πειραματικά αποτελέσματα και για τις τέσσερις προσεγγίσεις. Αυτά αφορούν τα έσοδα, την κοινωνική ωφέλεια, τη χρησιμοποίηση CPU και το λόγο προσέγγισης σε ορισμένους αλγορίθμους. Τέλος, παρουσιάζεται η δική μας ιδανική προσέγγιση για την επίλυση του προβλήματος.

Στο **Κεφάλαιο 7** πραγματοποιείται ο επίλογος της παρούσας διπλωματικής εργασίας. Γίνεται μία σύνοψη των αποτελεσμάτων και παραθέτονται κάποια βασικά συμπεράσματα. Τέλος, παρουσιάζονται πιθανές μελλοντικές κατευθύνσεις που θα έχουν ως έναυσμα την παρούσα εργασία.

Κεφάλαιο 2

Το Σύστημα Spot Instances

Στο κεφάλαιο αυτό πραγματοποιείται μία εισαγωγή στο σύστημα Spot Instances το οποίο αποτέλεσε το έναυσμα για την παρούσα διπλωματική εργασία. Παρουσιάζεται το σύστημα και οι βασικές έννοιες του, οι κυρίαρχες στρατηγικές χρήσης του, παρόμοιες υπηρεσίες καθώς και περιγράφεται η δική μας ανάλυση για το σύστημα και οι ιδέες μας για τη βελτίωσή του.

Στην ενότητα [2.1](#) περιγράφεται η βασική ιδέα πίσω από το σύστημα, αναλύεται η λειτουργία, παρουσιάζονται οι διαφορές μεταξύ των κανονικών εικονικών μηχανών και των Spot Instances, αναφέρονται κυρίαρχες στρατηγικές χρήσης της υπηρεσίας και τέλος γίνεται μία αναφορά σε παρόμοιες υπηρεσίες.

Στην ενότητα [2.2](#) περιγράφονται σύμφωνα με τη δική μας σκοπιά τα πλεονεκτήματα και τα μειονεκτήματα της υπηρεσίας καθώς και οι ιδέες μας για την τροποποίηση της και την επίλυση κάποιων μειονεκτημάτων της.

2.1 Περιγραφή του Συστήματος

Ξεκινάμε την ενότητα αυτή περιγράφοντας το σύστημα Spot Instances [1] της Amazon, το οποίο αποτέλεσε την αφορμή για την παρούσα διπλωματική εργασία.

2.1.1 Η Ιδέα

Η ιδέα των Spot Instances προέκυψε από την ανάγκη για χρησιμοποίηση των εικονικών μηχανών της Amazon οι οποίες δεν έχουν δοθεί σε κάποιο χρήστη. Ένα Spot-Instance είναι μια αχρησιμοποίητη εικονική μηχανή. Για το λόγο αυτό, η Amazon αποφάσισε να διαθέσει αυτές τις εικονικές μηχανές σε χαμηλότερες τιμές από ότι οι κανονικές (On-Demand) εικονικές μηχανές. Η τιμή κάθε Spot Instance αλλάζει ανά μία ώρα, διαφέρει ανά γεωγραφική ζώνη και παραμένει σταθερή για το διάστημα αυτό. Για να μπορέσει κάποιος να παραλάβει Spot Instance θα πρέπει να προσφέρει τιμή μεγαλύτερη ή ίση από αυτή την τιμή. Σε περίπτωση που η προσφορά του κάθε χρήστη γίνει μικρότερη από την τιμή του Spot Instance καθώς για παράδειγμα υπάρχουν μεγαλύτερες προσφορές ή έχει μειωθεί το πλήθος των διαθέσιμων Spot Instances, τότε η Amazon έχει το δικαίωμα να πάρει πίσω μέσα σε πολύ μικρό χρονικό διάστημα το Spot Instance από το χρήστη και να το διαθέσει αλλού. Η τιμή αυτή υπολογίζεται σταδιακά βασισμένη στη μακροπρόθεσμη ζήτηση και προσφορά για Spot Instances.

Σε γενικές γραμμές τα Spot Instances είναι μία οικονομική προσέγγιση για χρήστες οι οποίοι τρέχουν εφαρμογές οι οποίες δεν επεξεργάζονται κρίσιμο φορτίο και μπορούν να διακοπούν. Παραδείγματα τέτοιων εφαρμογών είναι οι εργασίες ανάλυσης δεδομένων, batch jobs, background processing, optional tasks κ.λ.π.

2.1.2 Βασικές Έννοιες

Παρακάτω παρουσιάζονται οι βασικές έννοιες των Spot Instances.

- *Spot Instance Pool*: Ένα σύνολο αχρησιμοποίητων εικονικών μηχανών της Amazon που είναι του ίδιου τύπου, έχουν ίδιο λογισμικό, ανήκουν στην ίδια γεωγραφική ζώνη και χρησιμοποιούν την ίδια πλατφόρμα δικτύου.
- *Spot Price*: Η τιμή ενός Spot Instance για την τρέχουσα ώρα.
- *Spot Instance Request*: Παρέχει τη μέγιστη τιμή που κάποιος επιθυμεί να δώσει ανά ώρα για ένα Spot Instance. Αν κάποιος χρήστης δεν προσδιορίσει την τιμή αυτή, τότε η Amazon ορίζει ως μέγιστη τιμή την τρέχουσα τιμή των κανονικών εικονικών μηχανών που διαθέτει. Όταν η μέγιστη τιμή ξεπεράσει την Spot Price, τότε αν υπάρχει διαθέσιμη χωρητικότητα η Amazon καταχωρεί το Spot Instance στο χρήστη. Ένα Spot Instance Request μπορεί να είναι μίας φορές (one-time) ή διαρκές (persistent). Αν είναι διαρκές, τότε μόλις σταματήσει το Spot Instance το οποίο είναι συνδεδεμένο με το Request, η Amazon αυτόματα το ξανά πραγματοποιεί. Ένα Spot Instance Request μπορεί να περιέχει διάρκεια για τη χρήση των Spot Instances.
- *Spot Fleet*: Πρόκειται για ένα σύνολο από Spot Instances το οποίο ξεκινάει για το χρήστη και πληρεί τα κριτήρια που ο χρήστης έχει ορίσει. Το Spot Fleet επιλέγει τα Spot Instance Pools τα οποία ικανοποιούν τις ανάγκες του χρήστη και τα ξεκινάει. Τα Spot Fleets ορίζονται προκειμένου να διατηρούν τις απαιτήσεις του χρήστη σε υπολογιστικούς πόρους αντικαθιστώντας τα Spot Instances τα οποία τερματίζονται με άλλα ισοδύναμα. Ένα Spot Fleet μπορεί να ζητηθεί σε one-time Request.
- *Spot Instance Interruption*: Η Amazon διακόπτει (terminates), σταματά (stops) και παύει (hibernates) όταν το Spot Price ξεπερνά τη μέγιστη προσφορά του χρήστη ή όταν δεν υπάρχει διαθέσιμη εικονική μηχανή. Η Amazon προσφέρει ένα μήνυμα διακοπής 2 λεπτά πριν τερματιστεί το Spot Instance.

2.1.3 Διαφορές Μεταξύ Spot Instances και Κανονικών Εικονικών Μηχανών

Στον παρακάτω πίνακα παρουσιάζουμε τις κύριες διαφορές μεταξύ των Spot Instances και των κανονικών εικονικών μηχανών (On-Demand) που διαθέτει η Amazon.

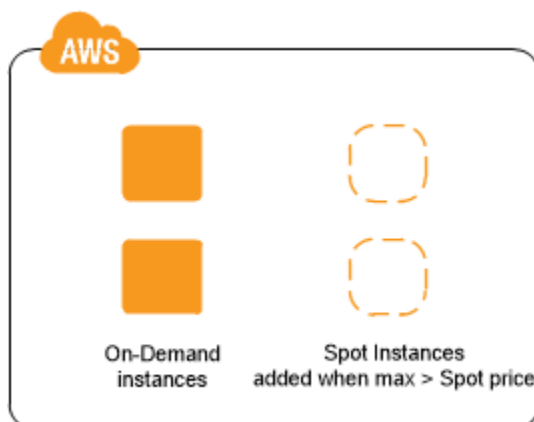
| | Spot Instances | On-Demand Instances |
|------------------------------|--|---|
| Launch Time | Μπορούν να ξεκινήσουν άμεσα μόνο αν το Spot Request είναι ενεργό και υπάρχει διαθεσιμότητα. | Μπορούν να ξεκινήσουν άμεσα αν πραγματοποιηθεί manual αίτημα εκκίνησης και υπάρχει διαθεσιμότητα. |
| Available Capacity | Αν δεν υπάρχει διαθεσιμότητα το Spot Request συνεχίζει αυτόματα να πραγματοποιεί launch request μέχρι να υπάρξει διαθέσιμη χωρητικότητα. | Αν δεν υπάρχει διαθεσιμότητα όταν γίνεται launch request, τότε ο χρήστης λαμβάνει error. |
| Hourly Price | Η τιμή για ένα Spot Instance αλλάζει ανάλογα με τη ζήτηση. | Η τιμή για ένα On-Demand Instance είναι στατική. |
| Instance Interruption | Ο χρήστης δε μπορεί να σταματήσει και να ξεκινήσει ένα Spot Instance. Αυτό μπορεί να το κάνει μόνο η Amazon αν δεν υπάρχει διαθεσιμότητα, αν η Spot Price ξεπερνά τη μέγιστη προσφορά του χρήστη ή αν αυξηθεί η ζήτηση για Spot Instances. | Ο χρήστης καθορίζει πότε θα διακόψει ένα On-Demand Instance. |

Πίνακας 2.1: Διαφορές Μεταξύ Spot Instances και Κανονικών Εικονικών Μηχανών

2.1.4 Στρατηγικές Χρήστης και Αρχιτεκτονική Συστήματος

Παρακάτω παρουσιάζουμε δύο βασικές στρατηγικές χρήσης της υπηρεσίας Spot Instances.

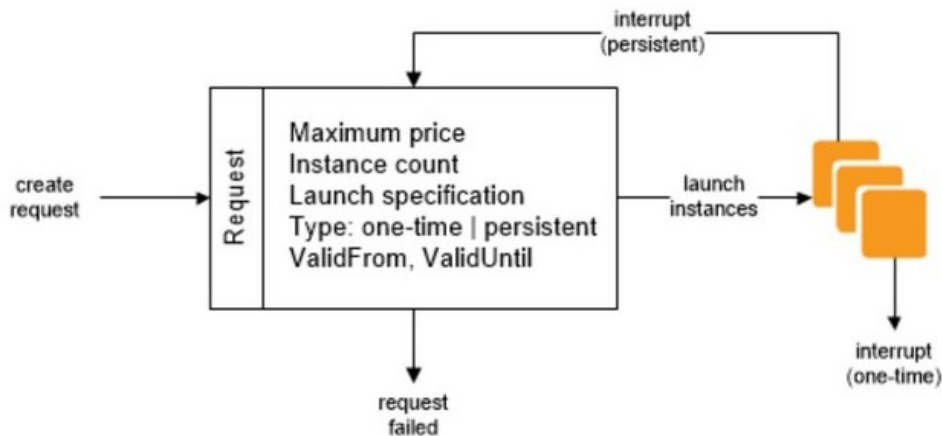
Η πρώτη στρατηγική αφορά τη διατήρηση ενός ελαχίστου επιπέδου υπολογιστικών πόρων από On-Demand Instances προκειμένου να τρέχουν οι εφαρμογές του χρήστη και τον εμπλουτισμό του με Spot Instances κάθε φορά όπου η μέγιστη προσφορά του χρήστη είναι μεγαλύτερη από το Spot Price. Σχηματικά, η τεχνική απεικονίζεται παρακάτω.



Εικόνα 2.1: Πρώτη Στρατηγική Χρήσης της Υπηρεσίας Spot Instances.

Η δεύτερη στρατηγική αφορά τη χρήση των Spot Instances για ένα συγκεκριμένο χρονικό διάστημα (Spot Block) τα οποία έχουν σχεδιαστεί έτσι ώστε να μη διακόπτονται και θα τρέξουν συνεχόμενα για τη διάρκεια που έχει επιλέξει ο χρήστης. Πολύ σπάνια, ένα Spot Block μπορεί να διακοπεί λόγω έλλειψης διαθεσιμότητας. Σε τέτοια περίπτωση ο χρήστης ενημερώνεται με μήνυμα δύο λεπτά πριν τον τερματισμό και δε χρεώνεται ακόμα και αν χρησιμοποίησε τα Spot Instances.

Με βάση όλα τα παραπάνω παρουσιάζουμε σχηματικά τη συνολική αρχιτεκτονική του συστήματος.



Εικόνα 2.2: Σχηματικά η Αρχιτεκτονική της Υπηρεσίας Spot Instances.

Με βάση αυτή, στην επόμενη ενότητα θα παρουσιάσουμε το σχέδιο μας για μία συγκεκριμένη μετατροπή στο σύστημα προκειμένου να λυθούν κάποια από τα μειονεκτήματα που παρουσιάζει. Τέλος, κάνουμε μία σύντομη αναφορά σε παρόμοιες υπηρεσίες.

2.1.5 Παρόμοιες Υπηρεσίες

Κλείνουμε την ενότητα παρουσιάζοντας παρόμοιες υπηρεσίες με την υπηρεσία Spot Instances της Amazon.

Μία παρόμοια υπηρεσία είναι η υπηρεσία Preemptible VM (PVM) της Google. Στη συγκεκριμένη υπηρεσία, ο χρήστης μπορεί να αποκτήσει μία εικονική μηχανή σε χαμηλή τιμή η οποία όμως θα σταματήσει να λειτουργεί σε μόλις 24 ώρες μετά την απόκτησή της από το χρήστη. Η Google έχει τη δυνατότητα να αποσύρει την εικονική μηχανή για λόγους διαθεσιμότητας. Τα PVMs χρησιμοποιούνται για λόγους testing, για σύντομες batch jobs και για fault tolerant εφαρμογές. Ένα PVM εξοικονομεί μέχρι και 80% για το χρήστη σε σχέση με μία κανονική εικονική μηχανή της Google. Μία διαφορά σε σχέση με την υπηρεσία Spot Instances, είναι πως τα PVMs έχουν σταθερή τιμή, ενώ τα Spot Instances δυναμική. Μια ακόμα διαφορά είναι πως τα PVMs πάντα διακόπτονται και πρέπει να επανεκκινήθούν manually ενώ τα Spot Instances απλά σταματούν και μπορούν να επανεκκινήθούν αυτόματα όταν υπάρξει διαθεσιμότητα ή όταν η προσφορά του χρήστη είναι αρκετή για να τα αποκτήσει.

Μία άλλη υπηρεσία που παρέχει οικονομικές εικονικές μηχανές είναι η υπηρεσία Droplets της DigitalOcean η οποία λειτουργεί με τη λογική των On-Demands Instances της Amazon. Παρόλα αυτά, οι τιμές της είναι αρκετά χαμηλές, γεγονός που την κάνει να ξεχωρίζει σε σχέση με τους ανταγωνιστές. Μία διαφορά σε σχέση με την υπηρεσία Spot Instances, είναι πως τα Droplets έχουν σταθερή τιμή, ενώ τα Spot Instances δυναμική.

2.2 Το Κίνητρό μας

Στην ενότητα αυτή θα περιγράψουμε τη μελέτη που κάναμε για το σύστημα και τις ιδέες μας για τη βελτίωση του. Αρχικά θα αναφερθούμε σε πλεονεκτήματα και μειονεκτήματα του συστήματος.

2.2.1 Πλεονεκτήματα και Μειονεκτήματα

Στην ενότητα αυτή περιγράψουμε τα πλεονεκτήματα και τα μειονεκτήματα του συστήματος Spot Instances.

Τα **πλεονεκτήματα** που εντοπίσαμε είναι:

1. Το σύστημα Spot Instances είναι οικονομικό από την άποψη γλιτώνει στο χρήστη μέχρι και 90% σε σχέση με On-Demand cloud computing costs.
2. Το σύστημα Spot Instances είναι κατάλληλο για noncritical jobs (stateless, fault-tolerant ή flexible εφαρμογές όπως εφαρμογές σχετικές με big data, containerized workloads web servers, high-performance computing (HPC), batch jobs και άλλα test & development workloads).
3. Το σύστημα Spot Instance δίνει τη δυνατότητα στο χρήστη να επιλέξει από μόνος το είδος του interruption.
4. Το σύστημα Spot Instance παρέχει ειδοποίηση για το Interruption 2 λεπτά πριν αυτό προκύψει ειδοποιώντας έγκαιρα το χρήστη.

Αντίθετα, τα **μειονεκτήματα** που εντοπίσαμε είναι:

1. Το σύστημα Spot Instances δεν έχει εγγυημένη διαθεσιμότητα. Μπορεί κάποιο Spot Instance να διακοπεί οποιαδήποτε στιγμή.
2. Το σύστημα Spot Instances λόγω του παραπάνω δεν είναι κατάλληλο για κρίσιμα jobs και "ευαίσθητα" δεδομένα.
3. Στο σύστημα Spot Instances από το 2018 το pricing μοντέλο έχει αλλάξει και κάνει πιο εύκολη την πρόβλεψη για το πως θα κινηθούν οι τιμές. Το πρόβλημα σε αυτό είναι πως σε περίπτωση που οι αιτήσεις των χρηστών ξεπεράσουν την υπάρχουσα διαθεσιμότητα η τιμή παραμένει σχεδόν η ίδια (με μικρή αύξηση) γεγονός που i) τερματίζει κάποια instances όχι λόγω τιμής αλλά λόγω του ότι η υπηρεσία έχει δεχτεί παραπάνω αιτήματα από όσα μπορεί να εξυπηρετήσει ii) δε μεγιστοποιεί τα κέρδη για την υπηρεσία καθώς δε λαμβάνονται υπ' όψη τόσο οι προσφορές. Άρα, αν κάποιος χρήστης κάνει προσφορά πάνω από την Spot Price δε θα λάβει απαραίτητα ένα Spot Instance. Επιπλέον, ένα υπάρχον Instance μπορεί να χαθεί ακόμα και αν η προσφορά είναι καλύτερη από την υπάρχουσα τιμή (οι συνθήκες interruption δεν είναι ξεκάθαρες). Συνεπώς, έχουμε εικονικές τιμές.
4. Το σύστημα Spot Instances διαθέτει default περιορισμό 20 Spot Instances ανά γεωγραφική ζώνη. Παραπάνω αύξηση σε Spot Instances γίνεται μόνο μετά από ειδική αίτηση.
5. Το σύστημα Spot Instances δε μπορεί να διαθέσει partial αριθμό απο Spot Instances. Για παράδειγμα, ένα χρήστης θα μπορούσε να ζητήσει 10 και άμα η προσφορά του δεν είναι επαρκής να λάβει μόνο 7. Κάτι τέτοιο θα ήταν χρήσιμο για batch jobs όπως Machine-Learning Algorithms.
6. Επειδή δεν είναι 100% σαφές το pricing μοντέλο, δε γνωρίζουμε αν οι χρήστες προσφέρουν την πραγματική τους αξία για να αποκτήσουν εικονικές μηχανές ή όχι.

2.2.2 Η Ιδέα μας

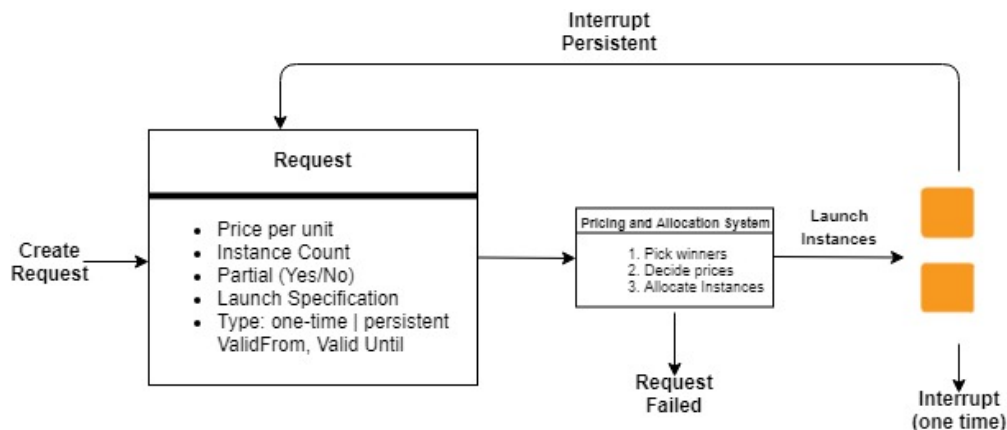
Παρατηρώντας όλα τα παραπάνω και λαμβάνοντας υπόψιν τη ραγδαία άνοδο της υπηρεσίας παγκοσμίως θεωρήσαμε πως μέσα από τη συγκεκριμένη εργασία θα μπορούσαμε να βελτιστοποιήσουμε την υπηρεσία επιλύοντας κάποια από τα μειονεκτήματα που παρουσιάζει.

Πιο συγκεκριμένα, στοχεύσαμε στα μειονεκτήματα 3, 4, 5 και 6. Στόχος μας είναι να δημιουργήσουμε ένα εναλλακτικό σύστημα κατανομής των Spot Instances και τιμολόγησης για το σύστημα το οποίο 1) θα μεγιστοποιεί κατά το δυνατό τα έσοδα του πωλητή λαμβάνοντας υπόψιν αποκλειστικά τις προσφορές των χρηστών 2) δε θα διαθέτει περιορισμούς στα spot instances ανά γεωγραφική ζώνη που θα μπορεί να ζητήσει ο χρήστης και 3) θα μπορεί να δίνει partial αριθμό instances άμα το επιθυμεί ο χρήστης καθώς πολλές non-critical batch jobs μπορούν να παραλληλοποιηθούν σε μεγάλο βαθμό και επομένως δεν απαιτούν standard αριθμό υπολογιστικών πόρων για να ολοκληρωθούν (όσο περισσότερους workers θα κατέχει ο χρήστης τόσο, πιο γρήγορα θα ολοκληρωθεί ο αλγόριθμος), ενώ παράλληλα 4) θα οδηγεί έμμεσα τους χρήστες να φανερώσουν την πραγματική τους αξία για μία εικονική μηχανή.

Για να επιτύχουμε το στόχο μας καταφεύγουμε στην Αλγοριθμική Θεωρία Παιγνίων, στο Σχεδιασμό Μηχανισμών και πιο συγκεκριμένα στη Θεωρία Δημοπρασιών και στην κατηγορία Δημοπρασιών Knapsack. Θα πραγματοποιήσουμε διαφορετικές προσεγγίσεις οι οποίες θα κατέχουν μία κοινή πολύ σημαντική ιδιότητα, αυτή της φιλαλήθειας, η οποία θα αναλυθεί στο κεφάλαιο 3.

Θα επικεντρωθούμε σε δημοπρασίες τύπου Knapsack. Η πρώτη μας προσέγγιση θα αφορά τη μεγιστοποίηση της κοινωνικής ωφέλειας, ενώ οι υπόλοιπες τρεις θα αφορούν τη μεγιστοποίηση των εσόδων για τον πωλητή. Για την πραγματοποίηση της μεγιστοποίησης των εσόδων θα δοκιμάσουμε προσεγγιστικούς αλγόριθμους και θα προσπαθήσουμε να εκτιμήσουμε την πιθανοτική κατανομή των προσφορών των χρηστών με χρήση της θεωρίας εκτίμησης παραμέτρων και με τη βοήθεια νευρωνικών δικτύων. Όλες οι προσεγγίσεις παρουσιάζονται αναλυτικά στο κεφάλαιο 5 ενώ το θεωρητικό υπόβαθρο αυτών στα κεφάλαια 3 και 4.

Κλείνουμε το κεφάλαιο παρουσιάζοντας σχηματικά την επιθυμητή μορφή του συστήματος που θέλουμε να δημιουργήσουμε. Η δική μας παρέμβαση αφορά το Pricing and Allocation System της εικόνας το οποίο θα υλοποιείται σύμφωνα με τις προαναφερθείσες προσεγγίσεις.



Εικόνα 2.3: Σχηματικά η Επιθυμητή για εμάς Αρχιτεκτονική της Υπηρεσίας Spot Instances.

Κεφάλαιο 3

Σχεδιασμός Μηχανισμών και Δημοπρασίες

Στο κεφάλαιο αυτό πραγματοποιείται μία εισαγωγή στην τυπική μελέτη του σχεδιασμού μηχανισμών και στις δημοπρασίες. Παρουσιάζονται βασικές κατηγορίες δημοπρασιών και θεωρητικές τους ιδιότητες εστιάζοντας κυρίως στις στο Λήμμα του Myerson, στις δημοπρασίες Knapsack, στις δημοπρασίες μεγιστοποίησης εσόδων και στις τυχαίες δημοπρασίες.

Στην ενότητα 3.1 γίνεται μια εισαγωγή στη θεωρία παιγνίων στις βασικές κατηγορίες αυτών και στις κατηγορίες των ισοροπιών Nash.

Στην ενότητα 3.2 γίνεται μια εισαγωγή στο σχεδιασμό μηχανισμών.

Στην ενότητα 3.3 παρουσιάζονται οι δημοπρασίες με χρήματα εστιάζοντας στις δημοπρασίες ενός αντικειμένου, ορίζοντας την ωφέλεια χρήστη, την κοινωνική ωφέλεια, τα έσοδα πωλητή και το μοντέλο σφραγισμένων προσφορών.

Στην ενότητα 3.4 παρουσιάζονται οι δημοπρασίες first και second price ορίζοντας την έννοια της φιλαλήθειας ως κυρίαρχη στρατηγική, τις δημοπρασίες DSIC και τις ιδανικές δημοπρασίες.

Στην ενότητα 3.5 παρουσιάζεται και αποδεικνύεται το Λήμμα του Myerson ενώ περιγράφονται οι μηχανισμοί άμεσης αποκάλυψης.

Στην ενότητα 3.6 παρουσιάζονται οι δημοπρασίες Knapsack και περιγράφεται η μεγιστοποίηση της ωφέλειας σε αυτές.

Στην ενότητα 3.7 παρουσιάζονται δημοπρασίες που μεγιστοποιούν τα αναμενόμενα έσοδα. Οι δημοπρασίες αυτές προσεγγίζονται με τη βοήθεια Μπεϋζιανής Ανάλυσης, ορίζοντας την έννοια του virtual valuation και παρουσιάζοντας μηχανισμούς μεγιστοποίησης της εικονικής ωφέλειας.

Στην ενότητα 3.8 πραγματοποιείται μία εισαγωγή στις τυχαίες δημοπρασίες και ορίζεται η έννοια της φιλαλήθειας για αυτές.

3.1 Εισαγωγή στη Θεωρία Παιγνίων

Η Θεωρία Παιγνίων αφορά τη μελέτη σεναρίων σύμφωνα με την οποία πολλαπλοί λογικοί παίχτες, ανταγωνιστικοί συνήθως μεταξύ τους, καλούνται να λάβουν τη καλύτερη δυνατή απόφαση σύμφωνα πάντα με το δικό τους προσωπικό συμφέρον. Η μελέτη αυτή πραγματοποιείται με τη χρήση κατάλληλων μαθηματικών μοντέλων και εφαρμόζεται σε πληθώρα επιστημονικών πεδίων όπως τα οικονομικά, η επιστήμη των υπολογιστών και οι κοινωνικές επιστήμες.

Κάνουμε μία εισαγωγή στη Θεωρία Παιγνίων κάνοντας μία ιδιαίτερη αναφορά στο πιο χαρακτηριστικό παράδειγμα αυτής, αυτό του διλήμματος του φυλακισμένου.

Έστω δύο άτομα A και B που συλλαμβάνονται από την αστυνομία για ένα έγκλημα. Η αστυνομία δε διαθέτει όλα τα απαιτούμενα στοιχεία για να τους καταδικάσει και επομένως τους τοποθετεί σε ξεχωριστά δωμάτια και τους κάνει μία πρόταση:

- Αν ο A καταθέσει εναντίον του B για το έγκλημα και ο B δεν ομολογήσει τότε ο A θα μείνει ελεύθερος και ο B θα πάει φυλακή για 12 χρόνια. Το ίδιο και στην αντίθετη περίπτωση.
- Αν δεν ομολογήσει κανένας θα τιμωρηθούν και οι 2 μόνο με ένα χρόνο φυλάκισης.
- Αν και οι δυο καταθέσουν ενάντια στον άλλο, τότε θα τιμωρηθούν και οι δύο με 4 χρόνια.

Το παρακάτω φαίνεται σχηματικά στον παρακάτω πίνακα.

| | | Φυλακισμένος B | |
|----------------|----------|----------------|-------|
| | | Ομολογία | Σιωπή |
| Φυλακισμένος A | Ομολογία | 4,4 | 0,12 |
| | Σιωπή | 12,0 | 1,1 |

Πίνακας 3.1: Το δίλημμα του φυλακισμένου

Θεωρώντας πως οι δύο φυλακισμένοι είναι ορθολογικοί τότε σε κάθε περίπτωση βλέπουμε πως αυτό που τους συμφέρει είναι να ομολογήσουν και οι δύο. Αν ο A ομολογήσει και ο B δεν ομολογήσει, τότε ο A θα αφεθεί ελεύθερος και ο B θα μπει φυλακή για 12 χρόνια. Αν ομολογήσουν και οι 2 τότε θα μπου φυλακή για 4 χρόνια έκαστος. Το παράδοξο του προβλήματος είναι πως παρόλο που οι δύο παίχτες έχουν την επιλογή να φυλακιστούν για ένα χρόνο μόνο, η κυρίαρχη στρατηγική είναι να φυλακιστούν για 4 χρόνια έκαστος. Παρακάτω δίνεται ο ορισμός ενός παιγνίου.

Ορισμός 1: Παιγνίο

Προκειμένου να οριστεί ένα παίγνιο πρέπει να διαθέτουμε:

- Σύνολο N που αποτελείται από λογικούς παίχτες. Ισχύει $|N| = n$.
- Για κάθε λογικό παίχτη i υπάρχει ένα σύνολο S_i από στρατηγικές ανάλογα με τις αποφάσεις των υπολοίπων $n - 1$ λογικών παιχτών.
- Για κάθε λογικό παίχτη i υπάρχει μία συνάρτηση κόστους $C_i : S \rightarrow \mathbb{R}$, ή πιο απλά $C_i(s)$ με $s \in S = S_1 \times \dots \times S_n$ το οποίο αποτελεί σύνολο στρατηγικών ή έκβαση αυτών, η οποία αντιστοιχίζει την έκβαση των αποφάσεων των n λογικών παιχτών σε ένα βαθμωτό μέγεθος το οποίο εκφράζει το κέρδος ή το κόστος του λογικού παίχτη i για την έκβαση αυτή.

3.1.1 Βασικές Κατηγορίες Παιγνίων

Παρακάτω παρουσιάζονται οι βασικές κατηγορίες παιγνίων με βάση συγκεκριμένα χαρακτηριστικά τους.

Ορισμός 2: Κατηγορίες Παιγνίων

- *Συνεργατικά και Μη-συνεργατικά.* Στα συνεργατικά παίγνια η τελική στρατηγική προκύπτει από συνεργασία μεταξύ των παιχτών ενώ στα μη-συνεργατικά οι παίχτες λαμβάνουν μόνοι τους αποφάσεις. Το δίλημμα του φυλακισμένου αποτελεί ένα μη συνεργατικό παίγνιο.
- *Συμμετρικά και Μη-συμμετρικά.* Στα συμμετρικά παίγνια η έκβαση του αποτελέσματος για έναν παίχτη εξαρτάται από τις στρατηγικές που επέλεξαν οι αντίπαλοι και όχι από το ποιος παίχτης επέλεξε ποια στρατηγική. Στη μη-συμμετρικά το ποιος παίχτης επέλεξε ποια στρατηγική καθορίζει το αποτέλεσμα. Το δίλημμα του φυλακισμένου αποτελεί ένα συμμετρικό παίγνιο.
- *Πλήρους και Μερικής Πληροφόρησης.* Στα παίγνια πλήρους πληροφόρησης οι παίχτες γνωρίζουν τις στρατηγικές και της συναρτήσεις κόστους των αντιπάλων παιχτών. Στα παίγνια μερικής δε τις γνωρίζουν. Το σκάκι είναι παίγνιο πλήρους πληροφόρησης ενώ το πόκερ μερικής.
- *Μηδενικού ή Μη-Μηδενικού Αθροίσματος.* Στα παίγνια μηδενικού αθροίσματος το κέρδος του ενός παίχτη προκύπτει από το κόστος του αντιπάλου. Το άθροισμα κέρδους και κόστους είναι μηδέν. Αντίθετα στα παίγνια μη-μηδενικού αθροίσματος το άθροισμα κέρδους και κόστους δεν είναι μηδέν. Ένα παίγνιο μηδενικού αθροίσματος είναι το πόκερ.
- *Ακολουθιακά ή Παράλληλα.* Στα ακολουθιακά παίγνια κάθε παίχτης αποφασίζει τη στρατηγική του σε σειρά, ενώ οι υπόλοιποι παίχτες περιμένουν γνωρίζοντας τις προηγούμενες στρατηγικές του αντιπάλου. Στα παράλληλα οι παίχτες αποφασίζουν την ίδια χρονική στιγμή. Ένα παράδειγμα ακολουθιακού παιγνίου είναι το σκάκι, ενώ ένα παράδειγμα παράλληλου παιγνίου είναι οι δημοπρασίες.

3.1.2 Ισορροπίες Nash

Θεωρώντας ένα παίγνιο και n στο πλήθος λογικούς παίχτες που έχουν ως στόχο να ελαχιστοποιήσουν το κόστος τους, οι παίχτες επιθυμούν να επιλέξουν τη βέλτιστη στρατηγική. Οι ισορροπίες Nash αφορούν αυτή τη βέλτιστη επιλογή στρατηγικής. Θα αναφέρουμε τρία είδη ισορροπιών Nash.

Ορισμός 3: Pure Nash Equilibria

Μια αμιγής ισορροπία Nash (pure Nash equilibria) είναι η κατάσταση στην οποία κάθε παίχτης που συμμετέχει στο παίγνιο έχει βρει τη βέλτιστη στρατηγική και δεν έχει κίνητρο να την αλλάξει. Σε μία τέτοια περίπτωση για κάθε λογικό παίχτη i ισχύει πως:

$$C_i(\mathbf{s}) \leq C_i(s'_i, \mathbf{s}_{-i})$$

όπου \mathbf{s}_{-i} το διάνυσμα όλων των στρατηγικών \mathbf{s} έχοντας αφαιρέσει την i στρατηγική.

Ορισμός 4: Mixed Nash Equilibrium

Σε μία μικτή στρατηγική ισορροπία Nash (mixed Nash equilibrium) η καλύτερη στρατηγική είναι κάθε λογικός παίχτης να εναλλάσσεται μεταξύ πολλαπλών στρατηγικών. Πιο συγκεκριμένα, θεωρώντας κατανομές στρατηγικών $\sigma_1, \dots, \sigma_n$ στο σύνολο των στρατηγικών S_1, \dots, S_n λαμβάνουμε μικτή στρατηγική ισορροπία Nash αν για κάθε λογικό παίχτη i και στρατηγική s'_i ισχύει πως:

$$\mathbb{E}_{\mathbf{s} \sim \sigma}[C_i(\mathbf{s})] \leq \mathbb{E}_{\mathbf{s} \sim \sigma}[C_i(s'_i, \mathbf{s}_{-i})]$$

Αποδεικνύεται πως για κάθε παιχνίδι με πεπερασμένοι αριθμό παιχτών, υπάρχει μία μικτή στρατηγική ισορροπία Nash.

Ένα παιχνίδι με μικτή στρατηγική ισορροπία Nash είναι το γνωστό παιχνίδι πέτρα-μολύβι-χαρτί το οποίο αποδεικνύεται πως η καλύτερη δυνατή στρατηγική είναι κάθε παίχτης να παίζει με πιθανότητα $\frac{1}{3}$ ένα από τα 3 δυνατή ενδεχόμενα.

Ορισμός 5: Correlated Nash Equilibria

Σε μία συσχετιστική στρατηγική ισορροπία Nash (correlated Nash equilibrium) η ισορροπία επιτυγχάνεται μόνο εάν εμπλακεί κάποιος έμπιστος συντονιστής. Πιο συγκεκριμένα, θεωρώντας κατανομές στρατηγικών $\sigma_1, \dots, \sigma_n$ στο σύνολο των στρατηγικών S_1, \dots, S_n λαμβάνουμε συσχετιστική στρατηγική ισορροπία Nash αν για κάθε λογικό παίχτη i , στρατηγική $s_i \in S_i$ και απόκλιση $s'_i \in S_i$ ισχύει πως:

$$\mathbb{E}_{\mathbf{s} \sim \sigma}[C_i(\mathbf{s})|s_i] \leq \mathbb{E}_{\mathbf{s} \sim \sigma}[C_i(s'_i, \mathbf{s}_{-i})|s_i]$$

Θεωρούμε πως κάθε στρατηγικός παίχτης επιλέγει μία στρατηγική. Ο έμπιστος συντονιστής βλέπει το σύνολο των στρατηγικών \mathbf{s} και προτείνει ιδιωτικά σε κάθε λογικό παίχτη i να ακολουθήσει τη στρατηγική s_i . Ο λογικός παίχτης αποφασίζει αν θα το δεχτεί ή όχι.

Το παραπάνω φαίνεται με το εξής παράδειγμα στο οποίο εμπλέκονται δύο αυτοκίνητα τα οποία διασταυρώνονται, έστω A, B . Κάθε αυτοκίνητο έχει δύο επιλογές, να προχωρήσει ή να σταματήσει. Τα κόστη κάθε απόφασης φαίνονται στον παρακάτω πίνακα.

| | | | |
|---------------------|------|---------------------|-----|
| | | Αυτοκίνητο B | |
| | | Stop | Go |
| Αυτοκίνητο A | Stop | 1,1 | 1,0 |
| | Go | 0,1 | 5,5 |

Πίνακας 3.2: Πίνακας κόστους για αυτοκίνητα που διασταυρώνονται

Αν οι παίχτες αποφασίζουν μόνοι τους, το παίγνιο έχει δύο διαφορετικές αμιγείς ισορροπίες, οι οποίες είναι (Go,Stop) και (Stop,Go). Τώρα αν θεωρήσουμε έναν τροχονόμο ως συντονιστή, τότε αυτός θα αποφασίζει ποια είναι η βέλτιστη στρατηγική και ποιο από τα δύο αυτοκίνητα θα περάσει. Επομένως, το ένα αυτοκίνητο θα περάσει με κόστος 0, ενώ το δεύτερο θα περιμένει με κόστος 1, γιατί άμα περάσει θα έχει κόστος 5 και δε τον συμφέρει.

Κλείνουμε την εισαγωγή στη Θεωρία παιγνίων τονίζοντας τη σχέση μεταξύ των στρατηγικών αυτών $Pure \subset Mixed \subset Correlated$.

3.2 Εισαγωγή στο Σχεδιασμό Μηχανισμών

Ο τομέας του Σχεδιασμού Μηχανισμών συνιστά την επιστήμη της λήψης αποφάσεων. Ο στόχος του συγκεκριμένου τομέα είναι ο σχεδιασμός κανόνων για μία δομή, έτσι ώστε η στρατηγική συμπεριφορά των συμμετεχόντων στη δομή αυτή να είναι η επιθυμητή κατά το σχεδιαστή. Υπάρχουν πολλά παραδείγματα συστημάτων με στρατηγικούς συμμετέχοντες όπως αθλητικές διοργανώσεις, δημοπρασίες για διαδικτυακές διαφημίσεις κ.τ.λ.

3.2.1 Κοινωνική Επιλογή

Ο τομέας του Σχεδιασμού Αποφάσεων και της κοινωνικής επιλογής αφορά τη δημιουργία συστημάτων με συγκεκριμένους κανόνες έχοντας ως στόχο να επιτύχουν ιδιότητες όπως η μεγιστοποίηση του κοινωνικού οφέλους, η μεγιστοποίηση των εσόδων του σχεδιαστή ή τη φιλαλήθεια των συμμετεχόντων. Ωστόσο, υπάρχει περίπτωση ορισμένα συστήματα να μην έχουν σχεδιαστεί κατάλληλα έτσι ώστε να παρουσιάζεται η επιθυμητή συμπεριφορά κατά το σχεδιαστή.

Για παράδειγμα, σε ένα αθλητικό τουρνουά που οι ομάδες χωρίζονται σε ομίλους ο κύριος στόχος του σχεδιαστή είναι οι ομάδες που συμμετέχουν να διαγωνίζονται όσο το δυνατόν καλύτερα μπορούν, έτσι ώστε να κερδίσουν και να προσφέρουν το καλύτερο δυνατό θέαμα. Παρόλα αυτά, υπάρχει περίπτωση ορισμένες ομάδες να χάσουν εσκεμμένα έτσι ώστε να έχουν πιο ευνοϊκή διασταύρωση στα παιχνίδια νοκ-αουτ μετά τους ομίλους, γεγονός που έρχεται ενάντια στις επιθυμίες του σχεδιαστή. Γίνεται σαφές, πως οι κανόνες του συστήματος παίζουν τον κύριο λόγο στη διαμόρφωση του αποτελέσματος για ένα σύστημα. Ορίζουμε το νικητή Condorcet και παρουσιάζουμε ένα παράδειγμα κοινωνικής επιλογής στο οποίο παρουσιάζεται το πως διαφορετικοί μηχανισμοί, δίνουν διαφορετικά αποτελέσματα για ίδιους συμμετέχοντες.

Ορισμός 6: Νικητής Condorcet

Ο νικητής Condorcet σε μία εκλογική διαμάχη είναι αυτός ο οποίος βγαίνει νικητής κατά πλειοψηφία από όλους τους υπόλοιπους συνυποψηφίους αν η σύγκριση γίνει σε ζευγάρια.

Παράδειγμα 3.1: Κοινωνική Επιλογή - Εκλογές

Έστω ότι πραγματοποιούνται εκλογές με 3 υποψηφίους, έστω A , B και C . Θεωρούμε 1000 ψηφοφόρους με τις εξής προτιμήσεις:

- 499 ψηφοφόροι: $A > B > C$
- 498 ψηφοφόροι: $C > B > A$
- 3 ψηφοφόροι: $B > C > A$

Εξήγηση

- Αν χρησιμοποιήσουμε τον μηχανισμό της απόλυτης πλειοψηφίας νικητής των εκλογών θα είναι ο A .
- Αν λάβουμε υπόψη όλες τις προτιμήσεις και δώσουμε 2 πόντους στην πρώτη προτίμηση, 1 στη δεύτερη και 0 στην τρίτη, τότε νικητής είναι ο C .
- 3 Αν η σύγκριση γίνει σε ζευγάρια (Condorcet Criterion) νικητής είναι ο B , καθώς ο B προτιμάται κατά $498 + 3 = 501$ ψήφους από τον A και $499 + 3 = 502$ από τον C .

Από το παραπάνω παράδειγμα παρατηρούμε πως 3 διαφορετικοί μηχανισμοί οδηγούν σε 3 διαφορετικούς νικητές χωρίς να είναι ξεκάθαρο αν κάποιος μηχανισμός είναι καλύτερος από κάποιον άλλον, ενώ παράλληλα δεν εξασφαλίζεται πως οι ψηφοφόροι θα ψηφίσουν με βάση την πραγματική τους προτίμηση καθώς αν για παράδειγμα χρησιμοποιούταν ο πρώτος μηχανισμός, η ομάδα των δεύτερων ψηφοφόρων θα είχε κίνητρο και να επιλέξει ως πρώτη προτίμηση τον υποψήφιο C .

3.3 Δημοπρασίες με Χρήματα - Ενός Αντικειμένου

Στην ενότητα αυτή θα επικεντρωθούμε σε δημοπρασίες με χρήματα εστιάζοντας στις δημοπρασίες ενός αντικειμένου, ορίζοντας την ωφέλεια στρατηγικού παίχτη και το μοντέλο σφραγισμένων προσφορών.

3.3.1 Μηχανισμοί με Χρήματα

Παρακάτω παρουσιάζουμε φορμαλιστικά τους μηχανισμούς με χρήματα.

Ορισμός 7: Μηχανισμοί με Χρήματα

Για να περιγραφούν οι μηχανισμοί που εμπλέκουν χρήματα ορίζουμε τα εξής:

- Σύνολο N που αποτελείται από παίχτες. Ισχύει $|N| = n$.
- Σύνολο M που αποτελείται από εκβάσεις αποτελέσματος. Ισχύει $|M| = m$.
- Σύνολο συναρτήσεων V , το οποίο για κάθε παίχτη i , περιλαμβάνει μία συνάρτηση $V = \{u_i : M \rightarrow \mathbb{R}\}$ που εκφράζει την εκτίμηση για το παίχτη την αξία της κάθε έκβασης. Η συνάρτηση αυτή συνήθως αποτελεί ιδιωτική πληροφορία του παίχτη.
- Συνάρτηση έκβασης αποτελέσματος $f : V^n \rightarrow M$.
- Συνάρτηση πληρωμής, για κάθε παίχτη i , $p_i : V^n \rightarrow \mathbb{R}$, η οποία καθορίζει το ποσό που θα πληρώσει ο παίχτης.

3.3.2 Δημοπρασίες Ενός Αντικειμένου

Αφού ορίσαμε φορμαλιστικά τους μηχανισμούς με χρήματα πριν αναφερθούμε σε δημοπρασίες ενός αντικειμένου, ορίζουμε την ωφέλεια στρατηγικού παίχτη.

Ορισμός 8: Ωφέλεια Στρατηγικού Παίχτη

Η ωφέλεια u_i ενός στρατηγικού παίχτη i που κερδίζει το αντικείμενο σε μία δημοπρασία, ορίζεται ως η διαφορά της αξίας που έχει για αυτόν το αντικείμενο προς πώληση v_i , δηλαδή το μέγιστο ποσό που είναι διατεθειμένος να πληρώσει για να το αποκτήσει, μείον την τιμή που πληρώνει για να το αποκτήσει p_i . Σε κάθε άλλη περίπτωση η ωφέλεια του είναι 0. Συμβολικά έχουμε:

$$u_i = \begin{cases} v_i - p_i, & \text{for } i \text{ winner} \\ 0, & \text{for } i \text{ loser} \end{cases}$$

Κατά τη μελέτη σχεδιασμού μηχανισμών για την παρούσα διπλωματική εργασία θα ασχοληθούμε με τη μεγιστοποίηση της συνολικής ωφέλειας των στρατηγικών παιχτών η οποία ονομάζεται κοινωνική ωφέλεια καθώς και τη μεγιστοποίηση των εσόδων του πωλητή. Παρακάτω ορίζονται τα δύο αυτά μεγέθη.

Ορισμός 9: Κοινωνική Ωφέλεια

Σε μία δημοπρασία με n στρατηγικούς παίχτες, η κοινωνική ωφέλεια SW ορίζεται ως το άθροισμα της ωφέλειας όλων των στρατηγικών παιχτών. Συμβολικά έχουμε:

$$SW = \sum_{i=1}^n u_i$$

Ορισμός 10: Έσοδα Πωλητή

Σε μία δημοπρασία με n στρατηγικούς παίχτες, τα έσοδα του πωλητή ορίζονται ως το άθροισμα των τιμών που πληρώνουν οι στρατηγικοί παίχτες σε αυτόν. Συμβολικά έχουμε:

$$R = \sum_{i=1}^n p_i$$

Στο σημείο αυτό παρουσιάζουμε τις δημοπρασίες ενός αντικειμένου. Θεωρούμε έναν πωλητή που διαθέτει προς πώληση ένα αντικείμενο και n στρατηγικούς παίχτες που ενδιαφέρονται να το αγοράσουν. Κάθε στρατηγικός παίχτης θέλει να αγοράσει όσο το δυνατό φθηνότερα το αντικείμενο, με το μέγιστο ποσό που μπορεί να διαθέσει να είναι ίσο με την προσωπική αξία που έχει για αυτόν το αντικείμενο. Στο σημείο αυτό να τονιστεί πως η προσωπική αξία κάθε στρατηγικού παίχτη είναι ιδιωτική.

3.3.3 Μοντέλο Σφραγισμένων Προσφορών

Ολοκληρώνουμε αυτή την ενότητα παρουσιάζοντας μία απλή κλάση διαδικασιών δημοπρασίας, αυτή των σφραγισμένων προσφορών.

Ορισμός 11: Μοντέλο Σφραγισμένων Προσφορών

Τα βήματα που ακολουθούνται στο μοντέλο σφραγισμένων προσφορών είναι τα παρακάτω:

1. Κάθε στρατηγικός παίχτης i ιδιωτικά μεταφέρει την προσφορά του στον πωλητή, χωρίς να τη γνωρίζει κανένας άλλος στρατηγικός παίχτης (σφραγισμένη προσφορά).
2. Ο πωλητής αποφασίζει αν κάποιος στρατηγικός παίχτης θα λάβει το αντικείμενο και ποιος τελικά θα είναι αυτός.
3. Ο πωλητής αποφασίζει την τιμή πώλησης.

Ο πιο προφανής τρόπος για να υλοποιηθεί το βήμα 2 είναι να δοθεί το αντικείμενο στον στρατηγικό παίχτη με την υψηλότερη προσφορά. Σε ότι αφορά το βήμα 3, υπάρχουν πολλοί τρόποι για να υλοποιηθεί και κάθε επιλογή αλλάζει τη στρατηγική του κάθε παίχτη. Θα αναφερθούμε πιο αναλυτικά σε αυτό στις επόμενες ενότητες.

3.4 Δημοπρασίες First και Second Price

Στην ενότητα αυτή αναφερόμαστε σε δημοπρασίες με χρήματα που ακολουθούν μοντέλο σφραγισμένων προσφορών. Περιγράφουμε τις δημοπρασίες First και Second Price. Πριν περιγράψουμε τα δύο είδη δημοπρασιών θα ορίσουμε την έννοια της φιλαλήθειας.

3.4.1 Φιλαλήθεια

Ορισμός 12: Φιλαλήθεια

Ένας μηχανισμός λέμε ότι είναι φιλαλήθης όταν για κάθε στρατηγικό παίχτη i , η κυρίαρχη στρατηγική του είναι να δηλώσει την πραγματική του αξία v_i ως την προσφορά του για το αντικείμενο που τον ενδιαφέρει ανεξάρτητα από τις προσφορές των αντιπάλων του.

3.4.2 Δημοπρασίες First Price

Πρόκειται για τον πλέον παραδοσιακό μηχανισμό δημοπρασιών. Οι στρατηγικοί παίχτες πραγματοποιούν τις προσφορές τους και ο παίχτης με τη μεγαλύτερη κερδίζει το ζητούμενο αντικείμενο πληρώνοντας το ποσό το οποίο προσέφερε. Για παράδειγμα,

Ο μηχανισμός αυτός δεν είναι φιλαλήθης. Έστω μια δημοπρασία ενός αντικειμένου με δύο στρατηγικούς παίχτες A και B. Οι παίχτες σε κάθε περίπτωση προσπαθούν να μεγιστοποιήσουν την ωφέλειά τους. Έστω $v_A = 2, v_B = 1$ και $\varepsilon > 0$ πολύ μικρό. Αν ο B πόνταρε 1 και ο A ήξερε την προσφορά του B, τότε δε θα έκανε προσφορά 2, αλλά θα είχε κίνητρο να προσφέρει $1 + \varepsilon$ προκειμένου να μεγιστοποιήσει το όφελός του.

Παρατηρούμε πως ο μηχανισμός αυτός δημιουργεί πρόβλημα τόσο για το στρατηγικό παίχτη που δε γνωρίζει ποια πρέπει να είναι η στρατηγική του για να μεγιστοποιήσει την ωφέλεια του, όσο και για το σχεδιαστή του μηχανισμού που δε μπορεί να προβλέψει τη συμπεριφορά του συστήματος.

3.4.3 Δημοπρασίες Second Price

Θα μελετήσουμε τώρα ένα διαφορετικό μηχανισμό στον οποίο ο νικητής του αντικειμένου είναι αυτός με την υψηλότερη προσφορά και πληρώνει για το αντικείμενο τιμή ίση με τη δεύτερη υψηλότερη προσφορά. Για παράδειγμα, στην προηγούμενη περίπτωση δημοπρασίας με τους στρατηγικούς παίχτες A και B, αν ο B πόνταρε 1 και ο A πόνταρε 2, νικητής θα ήταν ο A καθώς $2 > 1$ και θα πλήρωνε τιμή ίση με την προσφορά του B, η οποία είναι ίση με 1. Ο συγκεκριμένος μηχανισμός είναι φιλαλήθης. Η απόδειξη παρουσιάζεται παρακάτω.

Πρόταση 1: Φιλαλήθεια ως κυρίαρχη στρατηγική σε δημοπρασίες second price

Σε μία δημοπρασία second price, κάθε στρατηγικός παίχτης i έχει ως κυρίαρχη στρατηγική να θέτει την προσφορά του b_i ίση με την πραγματική του αξία v_i .

ΑΠΟΔΕΙΞΗ: Έστω τυχαίος στρατηγικός παίχτης i , με προσωπική αξία v_i και \mathbf{b}_{-i} το διάνυσμα των προσφορών των στρατηγικών παιχτών δίχως την προσφορά του στρατηγικού παίχτη i . Θα δείξουμε πως η ωφέλεια του στρατηγικού παίχτη i μεγιστοποιείται όταν $b_i = v_i$. Έστω $B = \max_{j \neq i} \{b_j\}$ η μέγιστη προσφορά από κάποιον άλλο στρατηγικό παίχτη εκτός του i .

- Αν $b_i < B$, τότε ο i χάνει και η ωφέλεια του είναι ίση με 0.
- Αν $b_i \geq B$, τότε ο i κερδίζει και η ωφέλεια του είναι ίση με $v_i - B$.

Ο λόγος που οι παραπάνω τιμές είναι βέλτιστες είναι ο εξής:

- Αν $v_i < B$, η μέγιστη ωφέλεια που μπορεί να αποκτήσει ο στρατηγικός παίχτης είναι ίση με $\max\{0, v_i - B\} = 0$, δηλαδή όταν η προσφορά του είναι ίση με την προσωπική του αξία, καθώς αν $b_i < v_i$ τότε η ωφέλεια του θα ήταν πάλι 0, ενώ αν $b_i \geq v_i$ τότε ο στρατηγικός παίχτης μπορεί να πλήρωνε παραπάνω από την προσωπική του αξία, οπότε και η ωφέλεια του μπορεί να γινόταν αρνητική και δεν έχει όφελος να κερδίσει τη δημοπρασία.
- Αν $v_i \geq B$, τότε είναι προφανές πως η μέγιστη ωφέλεια επιτυγχάνεται όταν η προσφορά είναι ίση με την προσωπική του αξία και είναι ίση με $\max\{0, v_i - B\} = v_i - B$.

Συνολικά λοιπόν βλέπουμε πως σε μία δημοπρασία second price, κάθε στρατηγικός παίχτης i έχει ως κυρίαρχη στρατηγική να θέτει την προσφορά του b_i ίση με την πραγματική του αξία v_i . ■

Άλλη μία σημαντική ιδιότητα των δημοπρασιών second price είναι ότι ένας φιλαλήθης στρατηγικός παίχτης ποτέ δε αποκτά αρνητική ωφέλεια. Κοινώς δε μετανιώνει για τη συμμετοχή του στη δημοπρασία.

Πρόταση 2: Μη Αρνητική Ωφέλεια

Σε μία δημοπρασία second price, κάθε φιλαλήθης στρατηγικός παίχτης έχει μη αρνητική ωφέλεια.

ΑΠΟΔΕΙΞΗ: Οι ηττημένοι της δημοπρασίας έχουν ωφέλεια ίση με 0.

Έστω τυχαίος στρατηγικός παίχτης i , με προσωπική αξία v_i .

Αν ο στρατηγικός παίχτης i είναι ο νικητής, τότε η ωφέλεια του είναι ίση με $v_i - p$, όπου p η δεύτερη υψηλότερη προσφορά. Αφού ο i είναι ο νικητής, δηλαδή έχει την υψηλότερη προσφορά ίση με την πραγματική του αξία, ισχύει πως $v_i \geq p \implies v_i - p \geq 0$. ■

Στο σημείο αυτό, πριν προχωρήσουμε στον ορισμό των ιδανικών δημοπρασιών, κάνουμε μία σύντομη αναφορά στην έννοια του reserve price την οποία θα χρησιμοποιήσουμε στις προσεγγίσεις μας για την επίλυση του προβλήματος.

Ορισμός 13: Reserve Price

Πρόκειται για την ελάχιστη σταθερή τιμή στην οποία προσφέρεται ένα αντικείμενο σε μία δημοπρασία. Αν η προσφορά του στρατηγικού παίχτη είναι μικρότερη από το reserve price τότε δε λαμβάνει το αντικείμενο, αλλιώς το λαμβάνει σε τιμή ίση με reserve price.

3.4.4 Ιδανικές Δημοπρασίες

Παρατηρήσαμε πως οι δημοπρασίες second price προσφέρουν τρεις διαφορετικές και επιθυμητές ιδιότητες. Λαμβάνοντας αφορμή από αυτές κλείνουμε την ενότητα, παρουσιάζοντας τις λεγόμενες ιδανικές δημοπρασίες που πληρούν αυτές τις τρεις ιδιότητες. Παρακάτω παρουσιάζονται τρεις χρήσιμοι ορισμοί.

Ορισμός 14: Dominant-Strategy Incentive Compatible (DSIC)

Μία δημοπρασία είναι DSIC αν η φιλαλήθης προσφορά είναι πάντα κυρίαρχη στρατηγική για κάθε στρατηγικό παίχτη και αν οι στρατηγικοί παίχτες έχουν πάντα μη αρνητική ωφέλεια.

Ορισμός 15: Κοινωνική ωφέλεια

Ορίζουμε ξανά πιο φORMALISΤΙΚά την κοινωνική ωφέλεια ως εξής:

$$SW = \sum_{i=1}^n v_i x_i$$

όπου $x_i = 1$ αν ο στρατηγικός παίχτης i είναι νικητής και 0 αν δεν είναι. Επειδή έχουμε μόνο ένα αντικείμενο ισχύει πως $\sum_{i=1}^n x_i \leq 1$.

Ορισμός 16: Δημοπρασίες που λαμβάνουν μέγιστη κοινωνική ωφέλεια

Μία δημοπρασία λαμβάνει μέγιστη κοινωνική ωφέλεια, αν για φιλαλήθεις προσφορές, το αποτέλεσμα της έχει τη μέγιστη δυνατή κοινωνική ωφέλεια.

Μετά τους παραπάνω ορισμούς ορίζονται οι προϋποθέσεις για να είναι μια δημοπρασία ιδανική.

Ορισμός 17: Ιδανικές Δημοπρασίες

Μία δημοπρασία θεωρείται ιδανική αν ισχύουν οι παρακάτω τρεις ιδιότητες:

- Η δημοπρασία είναι DSIC.
- Η δημοπρασία μεγιστοποιεί την κοινωνική ωφέλεια.
- Η δημοπρασία μπορεί να υλοποιηθεί σε πολυωνυμικό χρόνο ως προς το μέγεθος της εισόδου, εννοώντας το πλήθος των bits τα οποία είναι απαραίτητα για να αναπαραστήσουν τα v_1, v_2, \dots, v_n .

Σημείωση: Οι δημοπρασίες second-price είναι ιδανικές.

3.5 Λήμμα του Myerson

Στην ενότητα αυτή θα παρουσιάσουμε το Λήμμα του Myerson [14], το οποίο αποτελεί ένα γενικό και ισχυρό εργαλείο για να καθοριστούν οι τιμές πωλήσεις, προκειμένου κυρίαρχη στρατηγική της δημοπρασίας για τους στρατηγικούς παίχτες να είναι η φιλαλήθεια.

3.5.1 Περιβάλλον Μοναδικής Παραμέτρου

Θα ορίσουμε το Λήμμα του Myerson σε περιβάλλοντα μοναδικής παραμέτρου (single-parameter environments). Συνεχίζουμε με έναν ορισμό και δύο παραδείγματα.

Ορισμός 18: Περιβάλλον Μοναδικής Παραμέτρου

Ένα περιβάλλον μοναδικής παραμέτρου περιλαμβάνει:

- n στο πλήθος στρατηγικούς παίχτες.
- Κάθε στρατηγικός παίχτης i διαθέτει μη αρνητική προσωπική αξία v_i , η οποία εκφράζει την αξία για το "ανά μονάδα" αντικείμενο που επιθυμεί να αποκτήσει.
- Ένα εφικτό σύνολο (feasible set) X , στο οποίο κάθε στοιχείο του είναι ένα μη αρνητικό διάνυσμα (x_1, x_2, \dots, x_n) , όπου το x_i δηλώνει το ποσό από το αντικείμενο το οποίο λαμβάνει ο στρατηγικός παίχτης i .

Παρακάτω παρουσιάζονται δύο χαρακτηριστικά παραδείγματα από περιβάλλοντα μοναδικής παραμέτρου.

Παράδειγμα 3.2: Δημοπρασία ενός αντικειμένου

Σε μία δημοπρασία ενός αντικειμένου, X είναι το σύνολο των $0 - 1$ διανυσμάτων τα οποία έχουν το πολύ ένα 1 έτσι ώστε να ισχύει $\sum_{i=1}^n x_i \leq 1$.

Παράδειγμα 3.3: Δημοπρασία k αντικειμένων

Θεωρούμε μια δημοπρασία με k πανομοιότυπα αντικείμενα και τον περιορισμό ότι κάθε στρατηγικός παίχτης μπορεί να πάρει το πολύ ένα από αυτά. Το σύνολο X των $0 - 1$ διανυσμάτων περιέχει τα διανύσματα τα οποία έχουν το πολύ k στο πλήθος 1 έτσι ώστε να ισχύει $\sum_{i=1}^n x_i \leq k$.

3.5.2 Κατανομή και Κανόνες Πληρωμής

Όπως είχαμε δει στην ενότητα 3.3.3, το μοντέλο σφραγισμένων προσφορών περιέχει δύο πολύ σημαντικά βήματα. Το πρώτο αφορά την κατανομή, δηλαδή τι θα λάβει ο κάθε στρατηγικός παίχτης, ενώ το δεύτερο αφορά τον κανόνα πληρωμής, δηλαδή τι ποσό θα πληρώσει κάθε στρατηγικός παίχτης. Τα παραπάνω ορίζονται από τον πωλητή/σχεδιαστή του συστήματος και γίνονται σύμφωνα με την παρακάτω μεθοδολογία.

Μεθοδολογία 1: Κατανομή και Κανόνες Πληρωμής

1. Συλλογή των προσφορών \mathbf{b} από όλους τους στρατηγικούς παίχτες. Το διάνυσμα \mathbf{b} καλείται διάνυσμα προσφορών.
2. Επιλογή μίας εφικτής κατανομής $\mathbf{x}(\mathbf{b}) \in X \subseteq \mathbb{R}^n$, η οποία αποτελεί συνάρτηση των προσφορών.
3. Επιλογή τιμών πληρωμής $\mathbf{p}(\mathbf{b}) \in \mathbb{R}^n$, οι οποίες αποτελούν συνάρτηση των προσφορών.

Διαδικασίες τέτοιου τύπου ονομάζονται **μηχανισμοί άμεσης αποκάλυψης** (direct-revelation mechanisms) καθώς στο πρώτο βήμα της μεθοδολογίας οι στρατηγικοί παίχτες αποκαλύπτουν άμεσα τις προσφορές τους. Θεωρώντας μηχανισμό με κανόνες κατανομής και πληρωμής \mathbf{x} και \mathbf{p} αντίστοιχα, ο στρατηγικός παίχτης i λαμβάνει ωφέλεια ίση με $u_i(\mathbf{b}) = v_i x_i - p_i(\mathbf{b})$. Θα ασχοληθούμε με κανόνες πληρωμής που ικανοποιούν τη σχέση $p_i(\mathbf{b}) \in [0, b_i x_i(\mathbf{b})]$.

3.5.3 Διατύπωση και Απόδειξη Λήμματος Myerson

Πριν προχωρήσουμε στη διατύπωση και την απόδειξη του Λήμματος του Myerson αναφέρουμε δύο σημαντικούς ορισμούς.

Ορισμός 19: Υλοποιήσιμος Κανόνας Κατανομής

Ένας κανόνας κατανομής \mathbf{x} καλείται υλοποιήσιμος αν υπάρχει κανόνας πληρωμής \mathbf{p} τέτοιος ώστε ο μηχανισμός άμεσης αποκάλυψης (\mathbf{x}, \mathbf{p}) είναι DSIC.

Ορισμός 20: Μονότονος Κανόνας Κατανομής

Ένας κανόνας κατανομής \mathbf{x} για περιβάλλον μοναδικής παραμέτρου καλείται μονότονος αν για κάθε στρατηγικό παίχτη i και προσφορές \mathbf{b}_{-i} από τους υπόλοιπους στρατηγικούς παίχτες, η κατανομή $x_i(z, \mathbf{b}_{-i})$ στον i είναι μη φθίνουσα ως προς την προσφορά του z .

Παρακάτω θα ορίσουμε το Λήμμα του Myerson για περιβάλλοντα μοναδικής παραμέτρου σε 3 μέρη και στη συνέχεια θα το αποδείξουμε.

Θεώρημα 1: Λήμμα του Myerson

Έστω ένα περιβάλλον μοναδικής παραμέτρου. Ισχύει πως:

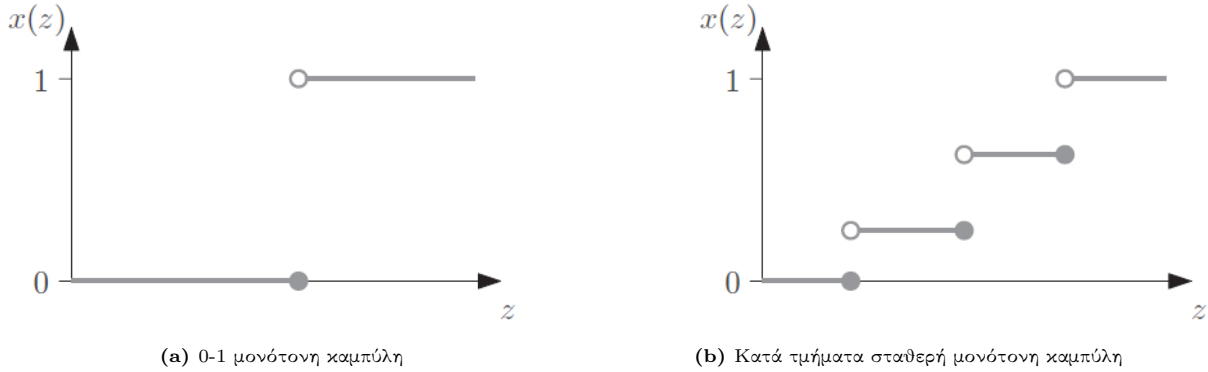
1. Ένας κανόνας κατανομής \mathbf{x} είναι υλοποιήσιμος αν και μόνο αν είναι μονότονος.
2. Αν ο \mathbf{x} είναι μονότονος, τότε υπάρχει μοναδικός κανόνας πληρωμής για τον οποίο ο μηχανισμός άμεσης αποκάλυψης (\mathbf{x}, \mathbf{p}) είναι DSIC και $p_i(\mathbf{b}) = 0$, όταν $b_i = 0$.
3. Ο πιο πάνω κανόνας πληρωμής δίνεται σε κλειστή μορφή παρακάτω.

Πάνω στο Λήμμα του Myerson έχουμε θεμελιώσει μεγάλο μέρος της διπλωματικής εργασίας. Η απόδειξη του δίνεται παρακάτω κλείνοντας με τον τρόπο αυτό την ενότητα.

ΑΠΟΔΕΙΞΗ: Έστω περιβάλλον μοναδικής παραμέτρου και κανόνας κατανομής \mathbf{x} ο οποίος δε γνωρίζουμε αν είναι μονότονος ή όχι.

Γνωρίζουμε πως για μία δημοπρασία DSIC ισχύει πως για κάθε στρατηγικό παίχτη i , για κάθε πιθανή προσωπική αξία v_i και για κάθε σύνολο προσφορών \mathbf{b}_{-i} η ωφέλεια του μεγιστοποιείται όταν προσφέρει την προσωπική του αξία.

Έστω τώρα τυχαίο i . Συμβολίζουμε για συντομία $x(z), p(z)$ την κατανομή και την πληρωμή - $x_i(z, \mathbf{b}_{-i})$ και $p_i(z, \mathbf{b}_{-i})$ αντίστοιχα - για τον στρατηγικό παίχτη i όταν προσφέρει z . Παρακάτω παρουσιάζονται παραδείγματα πιθανών συναρτήσεων κατανομής $x(z)$.



Εικόνα 3.1: Παράδειγμα καμπυλών κατανομής $x(\cdot)$. Πηγή: [20]

Έστω (\mathbf{x}, \mathbf{p}) ότι είναι DSIC και έστω $0 \leq y < z$.

Αρχικά θεωρούμε πως z είναι η προσωπική αξία του στρατηγικού παίχτη i και y μία οποιαδήποτε ψεύτικη προσφορά που μπορεί να καταθέσει. Επειδή ο μηχανισμός είναι DSIC ισχύει πως:

$$\underbrace{z \cdot x(z) - p(z)}_{\text{ωφέλεια για προσφορά } z} \geq \underbrace{z \cdot x(y) - p(y)}_{\text{ωφέλεια για προσφορά } y}$$

Αντιθέτως, θεωρούμε πως y είναι η προσωπική αξία του στρατηγικού παίχτη i και z μία οποιαδήποτε ψεύτικη προσφορά που μπορεί να καταθέσει. Άρα στην περίπτωση αυτή ισχύει πως:

$$\underbrace{y \cdot x(y) - p(y)}_{\text{ωφέλεια για προσφορά } y} \geq \underbrace{y \cdot x(z) - p(z)}_{\text{ωφέλεια για προσφορά } z}$$

Αναδιατάσσοντας τις παραπάνω ανισότητες λαμβάνουμε:

$$z \cdot [x(y) - x(z)] \leq p(y) - p(z) \leq y \cdot [x(y) - x(z)]$$

Από την παραπάνω ανισότητα βλέπουμε πως ο υλοποιησιμος κανόνας κατανομής είναι μονότονος.

Έστω τώρα πως ο κανόνας κατανομής είναι ασυνεχής όπως στην εικόνα 3.1b όπου παρατηρείται πεπερασμένος αριθμός αλμάτων. Διατηρώντας σταθερό z , θεωρώντας πως το y προσεγγίζει το z από πάνω και υποθέτοντας πως το ύψος του άλματος είναι ίσο με h το αριστερό και το δεξιό μέρος της ανισότητας τείνουν στο $z \cdot h$. Άρα έχουμε πως το άλμα στο p στο σημείο $z = z \cdot [\text{άλμα στο } x \text{ στο σημείο } z]$. Συνδυάζοντας τα παραπάνω με τη συνθήκη πως $p(0) = 0$ προκύπτει ο παρακάτω κανόνας πληρωμής:

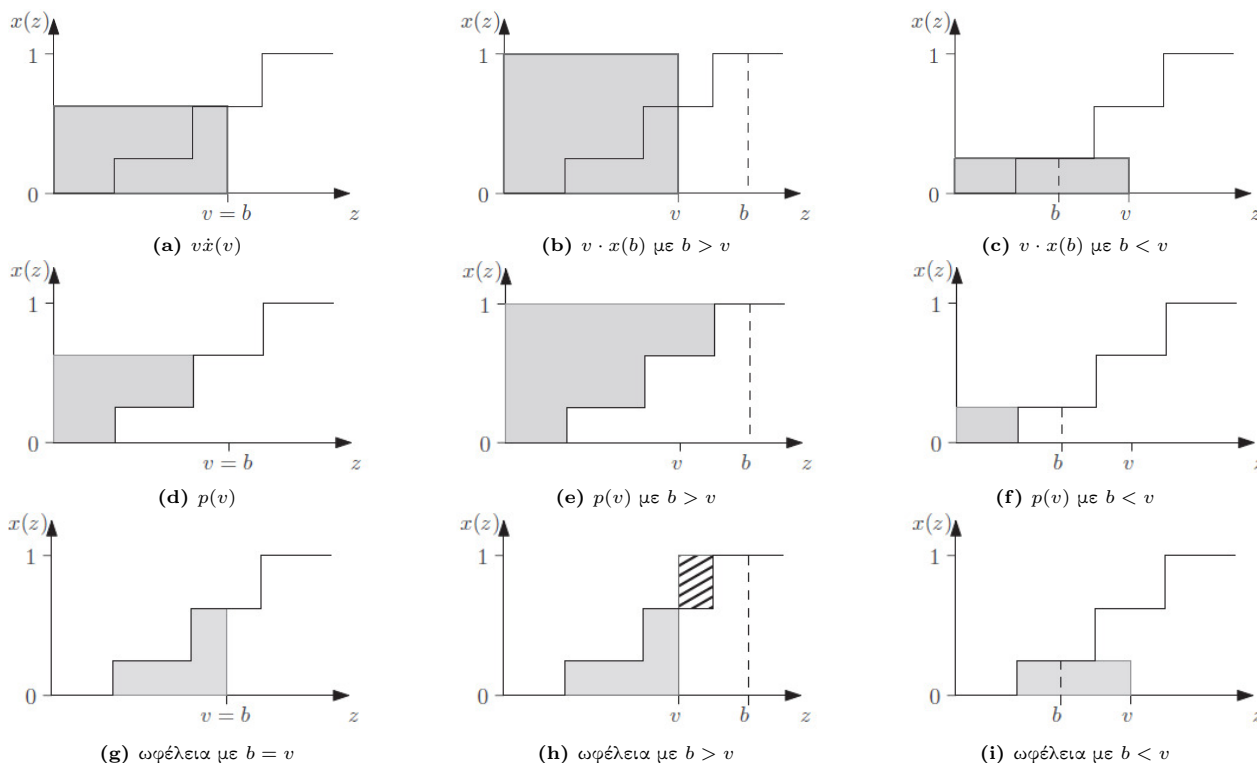
$$p_i(b_i, \mathbf{b}_{-i}) = \sum_{j=1}^l z_j \cdot [\text{άλμα στο } x_i(\cdot, \mathbf{b}_{-i}) \text{ στο σημείο } z_j]$$

όπου z_1, z_2, \dots, z_l είναι τα σημεία στα οποία παρουσιάζεται άλμα στη συνάρτηση κατανομής $x_i(\cdot, \mathbf{b}_{-i})$ στο διάστημα $[0, b_i]$.

Παρόμοια προσέγγιση γίνεται και όταν η συνάρτηση x είναι μονότονη συνεχής συνάρτηση η οποία δεν είναι κατά τμήματα σταθερή, αλλά συνεχής. Έστω ότι η x είναι παραγωγίσιμη. Τότε διαιρώντας τη διπλή ανισότητα με $y - z$ και παίρνοντας το όριο καθώς το y τείνει στο z από πάνω λαμβάνουμε τον περιορισμό: $p'(z) = z \cdot x'(z)$. Γνωρίζοντας πως $p(0) = 0$ ολοκληρώνουμε και λαμβάνουμε τον παρακάτω κανόνα πληρωμής για μονότονη συνεχή συνάρτηση, η οποία ισχύει για κάθε στρατηγικό παίχτη i , προσφορά b_i και προσφορές από τους υπολοίπους στρατηγικούς παίχτες \mathbf{b}_{-i} .

$$p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} z \cdot \frac{\partial x_i(z, \mathbf{b}_{-i})}{\partial z} dz$$

Βλέπουμε λοιπόν πως για κάθε κανόνα κατανομής \mathbf{x} , υπάρχει το πολύ ένας κανόνας πληρωμής \mathbf{p} τέτοιος ώστε ο μηχανισμός (\mathbf{x}, \mathbf{p}) να είναι DSIC. Προκειμένου να ολοκληρώσουμε την απόδειξη πρέπει να ελέγξουμε ότι αυτός ο κανόνας πληρωμής δουλεύει όταν το \mathbf{x} είναι μονότονο. Θα το αποδείξουμε με τη βοήθεια των παρακάτω εικόνων στην περίπτωση κατά τμήματα σταθερής μονότονης καμπύλης \mathbf{x} με την οποία θα ασχοληθούμε στην παρούσα διπλωματική εργασία. Παρακάτω παρατίθενται οι εικόνες και η εξήγηση αυτών.



Εικόνα 3.2: Επιχειρηματικές εικόνες για περιγραφή απόδειξης του Λήμματος Myerson. Η πρώτη στήλη μας δείχνει την περίπτωση που ο στρατηγικός παίχτης είναι φιλαλήθης. Η δεύτερη μας δείχνει την περίπτωση που ο στρατηγικός παίχτης προσφέρει παραπάνω από την προσωπική του αξία. Η τρίτη μας δείχνει την περίπτωση που ο στρατηγικός παίχτης προσφέρει λιγότερο από την προσωπική του αξία. Πηγή: [20]

Τα πιο πάνω σχήματα απεικονίζουν την ωφέλεια ενός στρατηγικού παίχτη όταν αυτός είναι φιλαλήθης (πρώτη στήλη), προσφέρει παραπάνω από την προσωπική του αξία (δεύτερη στήλη) και προσφέρει λιγότερο από την προσωπική του αξία (τρίτη στήλη). Μελετώντας τις εικόνες ανά στήλη παρατηρούμε πως αν αφαιρέσουμε το εμβαδόν της πρώτης εικόνας από αυτό της δεύτερης, λαμβάνουμε το εμβαδόν της τρίτης, η οποία αντιπροσωπεύει την ωφέλεια του στρατηγικού παίχτη. Είναι προφανές πως όταν ο στρατηγικός παίχτης είναι φιλαλήθης, η ωφέλεια του μεγιστοποιείται και είναι ίση με το εμβαδόν κάτω από την καμπύλη της κατανομής. Όπως βλέπουμε στη δεύτερη και την τρίτη στήλη η ωφέλεια που επιτυγχάνεται όταν προσφέρει περισσότερο ή λιγότερο από την προσωπική του αξία είναι πάντα μικρότερη από την ωφέλεια που επιτυγχάνεται όταν είναι φιλαλήθης. Βλέπουμε λοιπόν πως αυτός ο τρόπος πληρωμής μεγιστοποιεί την ωφέλεια του χρήστη όταν παρέχεται όταν το x είναι μονότονο. Με τον τρόπο αυτό ολοκληρώνουμε την απόδειξη. ■

3.6 Δημοπρασίες Knapsack

Στην ενότητα αυτή θα παρουσιάσουμε τις δημοπρασίες Knapsack που αποτελούν άλλο ένα παράδειγμα μηχανισμού μοναδικής παραμέτρου. Οι δημοπρασίες αυτές αποτελούν τη βάση αυτής της διπλωματικής εργασίας καθώς μοντελοποιούν φορμαλιστικά το πρόβλημα το οποίο θέλουμε να επιλύσουμε.

3.6.1 Ορισμός του Προβλήματος

Παρουσιάζεται ο ορισμός των δημοπρασιών τύπου Knapsack.

Ορισμός 21: Δημοπρασίες Knapsack

Στις δημοπρασίες τύπου Knapsack διαθέτουν:

- Ένα πωλητή που διαθέτει W πανομοιότυπα αντικείμενα προς πώληση.
- n στο πλήθος στρατηγικούς παίχτες, εκ των οποίων κάθε στρατηγικός παίχτης i διαθέτει μία προσωπική αξία v_i και ένα βάρος w_i που δηλώνει πόσα αντικείμενα επιθυμεί.
- Ένα εφικτό σύνολο X το οποίο ορίζεται από διανύσματα 0-1 (x_1, x_2, \dots, x_n) τέτοια ώστε $\sum_{i=1}^n w_i x_i \leq W$

Όποτε υπάρχουν αγαθά περιορισμένου μεγέθους προς διάθεση παρουσιάζεται ένα πρόβλημα τύπου knapsack. Στη δική μας περίπτωση, η εταιρεία που διαχειρίζεται τις υπηρεσίες cloud διαθέτει ένα περιορισμένο αριθμό από virtual machines προς μίσθωση και πολλαπλοί αγοραστές πραγματοποιούν προσφορές προκειμένου να τα αποκτήσουν. Στο σημείο αυτό, μπορούμε να τονίσουμε πως υπάρχει περίπτωση στο πρόβλημα knapsack το εφικτό σύνολο X να ορίζεται από διανύσματα (x_1, x_2, \dots, x_n) , όπου το τυχαίο $x_i \in [0, 1]$ τέτοια ώστε $\sum_{i=1}^n w_i x_i \leq W$. Το πρόβλημα αυτό ονομάζεται fractional knapsack problem ενώ το παραπάνω ονομάζεται binary knapsack problem.

3.6.2 Μεγιστοποίηση Ωφέλειας σε DSIC Δημοπρασίες Knapsack

Αφού επιθυμούμε να μεγιστοποιήσουμε την ωφέλεια, γνωρίζουμε πως ο κανόνας κατανομής θα είναι:

$$x(\mathbf{b}) = \operatorname{argmax}_X \sum_{i=1}^n b_i x_i$$

Από τον ορισμό της, βλέπουμε πως όταν οι στρατηγικοί παίχτες είναι φιλαλήθεις, ο παραπάνω κανόνας κατανομής μεγιστοποιεί την κοινωνική ωφέλεια και είναι μονότονος.

Σύμφωνα με το Λήμμα του Myerson, υπάρχει κανόνες πληρωμής \mathbf{p} τέτοιος ώστε ο μηχανισμός (\mathbf{x}, \mathbf{p}) να είναι DSIC. Ο κανόνας πληρωμής δίνεται από την κλειστή μορφή που είδαμε στην προηγούμενη ενότητα.

Στο σημείο αυτό θα μελετήσουμε κατά πόσο οι δημοπρασίες τύπου knapsack είναι ιδανικές. Όπως είδαμε και στην ενότητα 3.4.4 για να είναι μία δημοπρασία ιδανική θα πρέπει:

- Η δημοπρασία είναι DSIC.
- Η δημοπρασία μεγιστοποιεί την κοινωνική ωφέλεια.
- Η δημοπρασία μπορεί να υλοποιηθεί σε πολυωνυμικό χρόνο ως προς το μέγεθος της εισόδου, εννοώντας το πλήθος των bits τα οποία είναι απαραίτητα για να αναπαραστήσουν τα v_1, v_2, \dots, v_n .

Σύμφωνα με τον πιο πάνω ορισμό οι δημοπρασίες Knapsack δεν είναι ιδανικές. Ο λόγος είναι πως το πρόβλημα knapsack είναι NP-hard. Αυτό σημαίνει πως δεν υπάρχει αυτή τη στιγμή εφικτή λύση του προβλήματος σε πολυωνυμικό χρόνο. Για το λόγο αυτό, για την επίλυση του συγκεκριμένου προβλήματος καταφεύγουμε σε προσεγγιστικούς αλγόριθμους που χαλαρώνουν όσο το λιγότερο γίνεται το δεύτερο περιορισμός των ιδανικών δημοπρασιών προκειμένου να διατηρήσουν ακέραιους τους άλλους δύο.

Οι αλγόριθμοι που θα χρησιμοποιηθούν για την επίλυση του προβλήματος θα παρουσιαστούν αναλυτικά στο επόμενο κεφάλαιο.

3.7 Δημοπρασίες Μεγιστοποίησης Εσόδων

Στις προηγούμενες ενότητες αναφερθήκαμε σε μηχανισμούς που μεγιστοποιούν την κοινωνική ωφέλεια. Τα έσοδα στους μηχανισμούς αυτούς συνιστούν αναγκαίο κακό προκειμένου να δοθεί κίνητρο στους στρατηγικούς παίχτες να είναι φιλαλήθεις. Στην ενότητα αυτή θα παρουσιάσουμε μηχανισμούς που μεγιστοποιούν τα έσοδα για τον πωλητή.

3.7.1 Μπεϋζιανή Ανάλυση

Θα μελετήσουμε τη μεγιστοποίηση των εσόδων στο πλαίσιο της Μπεϋζιανής Ανάλυσης. Πριν προχωρήσουμε στη φορμαλιστική διατύπωση θα αναφέρουμε ένα παράδειγμα.

Παράδειγμα 3.4: Ένα αντικείμενο - Ένας στρατηγικός παίχτης

Έστω ότι έχουμε ένα αντικείμενο προς πώληση και ένα στρατηγικό παίχτη με προσωπική αξία για το αντικείμενο v . Θεωρούμε δημοπρασία DSIC άμεσης αποκάλυψης με τιμή r για το αντικείμενο. Είναι προφανές πως διακρίνονται δύο περιπτώσεις.

- Έχουμε μεγιστοποίηση κοινωνικής ωφέλειας όταν $r = 0$.
- Έχουμε μεγιστοποίηση εσόδων όταν $r = v$.

Ωστόσο σε περίπτωση που θέλουμε να μεγιστοποιήσουμε τα έσοδα, δε θα μπορούσαμε να ξέρουμε πως να θέσουμε την τιμή, καθώς η προσωπική αξία v είναι ιδιωτική για τον στρατηγικό παίχτη. Είναι προφανές πως το ζήτημα βρίσκεται στο ότι οι προσωπικές αξίες είναι άγνωστες στο σχεδιαστή του συστήματος. Για το λόγο αυτό, είναι αναγκαία η δημιουργία μηχανισμών DSIC που μεγιστοποιούν τα έσοδα ανεξάρτητα από την είσοδο.

Στη συνέχεια, ορίζουμε το μοντέλο στο οποίο θα πραγματοποιηθεί η Μπεϋζιανή Ανάλυση.

Ορισμός 22: Μοντέλο Μπεϋζιανής Ανάλυσης

Για το μοντέλο Μπεϋζιανής ανάλυσης θεωρούμε:

- Ένα περιβάλλον μοναδικής παραμέτρου θεωρώντας πως υπάρχει θετική σταθερά M τέτοια ώστε να ισχύει $x_i \leq M$, για κάθε i στο διάνυσμα εφικτής λύσης $(x_1, x_2, \dots, x_n) \in X$.
- Ανεξάρτητες συναρτήσεις κατανομής προσωπικής αξίας F_1, F_2, \dots, F_N με θετικές και συνεχείς συναρτήσεις πυκνότητας πιθανότητας f_1, f_2, \dots, f_n . Θεωρούμε πως το πεδίο ορισμού της τυχαίας F_i είναι το διάστημα $[0, v_{max}]$, όπου $v_{max} < \infty$.

Θεωρούμε πως ο σχεδιαστής του μηχανισμού γνωρίζει τις κατανομές. Στην πράξη αυτές εκτιμούνται από παλαιότερα δεδομένα. Επειδή ασχολούμαστε με DSIC μηχανισμούς, οι στρατηγικοί πράκτορες δεν έχουν ανάγκη να γνωρίζουν τις κατανομές των υπολοίπων στρατηγικών πρακτόρων. Σύμφωνα με τη συγκεκριμένη θεωρία επαναδιατυπώνουμε και μελετάμε το παραπάνω παράδειγμα.

Παράδειγμα 3.5: Ένα αντικείμενο - Ένας στρατηγικός παίχτης - Νέα Προσέγγιση

Θεωρούμε τις ίδιες συνθήκες όπως και προηγουμένως ορίζοντας ως F τη συνάρτηση κατανομής προσωπικής αξίας του στρατηγικού παίχτη.

Τα αναμενόμενα μέσα έσοδα δίνονται από την παρακάτω σχέση:

$$\underbrace{r}_{\text{έσοδα πώλησης}} \cdot \underbrace{1 - F(r)}_{\text{πιθανότητα πώλησης}}$$

Μεγιστοποιώντας αυτή την ποσότητα βρίσκουμε τη βέλτιστη τιμή.

Εφαρμογή: Έστω πως F είναι ομοιόμορφη κατανομή στο $[0, 1]$. Τα αναμενόμενα έσοδα είναι ίσα με $r(1 - r)$. Παραγωγίζοντας βλέπουμε πως η βέλτιστη τιμή είναι $r = \frac{1}{2}$ και τα βέλτιστα έσοδα $\frac{1}{4}$.

Τα παραπάνω επεκτείνονται και για πολλαπλούς στρατηγικούς παίχτες.

3.7.2 Τυπικός Ορισμός

Στόχος της ενότητας είναι να περιγράψουμε ένα βέλτιστο DSIC μηχανισμό για κάθε περιβάλλον μοναδικής παραμέτρου με κατανομές προσωπικής αξίας για τους επιμέρους στρατηγικούς παίχτες ίσες με F_1, F_2, \dots, F_n . Προκειμένου να συνοψίσουμε το πρόβλημα στόχος μας είναι να μεγιστοποιήσουμε το μέσο αναμενόμενο κέρδος για να έναν DSIC μηχανισμό (\mathbf{x}, \mathbf{p}) το οποίο φορμαλιστικά ορίζεται ως:

$$\mathbb{E}_{\mathbf{v} \sim \mathbf{F}}[\sum_{i=1}^n p_i(\mathbf{v})]$$

όπου $\mathbf{F} = F_1 \times F_2 \times \dots \times F_n$. Ωστόσο, η παραπάνω έκφραση δεν είναι εύκολο να μεγιστοποιηθεί και για το λόγο αυτό θα προχωρήσουμε παρακάτω σε μία πιο εύκολα μεγιστοποιήσιμη έκφραση.

3.7.3 Virtual Valuations

Προκειμένου να προχωρήσουμε σε μία πιο εύκολα διαχειρίσιμη μαθηματική παράσταση ορίζουμε την έννοια του virtual valuation(εικονική αξία).

Ορισμός 23: Virtual Valuation

Για ένα στρατηγικό παίχτη i με κατανομή προσωπικής αξίας F_i και προσωπική αξία v_i , το virtual valuation είναι ίσο με:

$$\varphi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$$

Παρατηρούμε πως το virtual valuation ενός στρατηγικού παίχτη εξαρτάται αποκλειστικά από την κατανομή της προσωπικής του αξίας και την προσωπική του αξία και είναι ανεξάρτητο από αυτά των υπολοίπων στρατηγικών παιχτών. Για παράδειγμα, αν $F_i(z) = z$, για $z \in [0, 1]$, δηλαδή η F_i ακολουθεί ομοιόμορφη κατανομή στο $[0, 1]$ τότε $\varphi_i(z) = z - \frac{1-z}{1} = 2z - 1$ στο $[0, 1]$.

Ο ρόλος του virtual valuation στη μεγιστοποίηση των εσόδων είναι πολύ σημαντικός. Μπορεί να θεωρηθεί πως εκφράζει τη διαφορά μεταξύ του επιθυμητού μέγιστου κέρδους μείον την απώλεια επειδή δεν ξέρεις το v_i από την αρχή. Αυτό φαίνεται και από την παρακάτω σχέση:

$$\varphi_i(v_i) = \underbrace{v_i}_{\text{επιθυμητή τιμή πώλησης}} - \underbrace{\frac{1 - F_i(v_i)}{f_i(v_i)}}_{\text{απώλεια λόγω μη γνώσης } v_i}$$

Μία ακόμα ερμηνεία στο virtual valuation είναι πως εκφράζει την κλίση μίας καμπύλης εσόδων στο σημείο v_i , όπου η καμπύλη αυτή περιγράφει τα αναμενόμενα έσοδα από ένα στρατηγικό παίχτη με προσωπική αξία που ακολουθεί κατανομή F_i ως συνάρτηση της πιθανότητας πώλησης.

3.7.4 Αναμενόμενα Έσοδα και Αναμενόμενη Εικονική Ωφέλεια

Θα δείξουμε πως τα αναμενόμενα συνολικά έσοδα είναι ίσα με την αναμενόμενη συνολική κοινωνική ωφέλεια. Για να καταλήξουμε σε αυτό το συμπέρασμα θα πρέπει πρώτα να αποδείξουμε μία πρόταση που δηλώνει πως η αναμενόμενη τιμή πληρωμής και η αναμενόμενη εικονική ωφέλεια ενός στρατηγικού παίχτη είναι ίσες.

Πρόταση 3: Ισότητα αναμενόμενης τιμής πληρωμής και αναμενόμενης εικονικής ωφέλειας στρατηγικού παίχτη

Σε κάθε περιβάλλον μοναδικής παραμέτρου με κατανομές προσωπικής αξίας F_1, F_2, \dots, F_n κάθε μηχανισμός DSIC (\mathbf{x}, \mathbf{p}) , για κάθε στρατηγικό παίχτη i και για κάθε προσωπική αξία \mathbf{v}_{-i} από τους υπολοίπους στρατηγικούς παίχτες ισχύει πως:

$$\mathbb{E}_{v_i \sim F_i} [p_i(\mathbf{v})] = \mathbb{E}_{v_i \sim F_i} [\varphi_i(v_i) \cdot x_i(\mathbf{v})]$$

ΑΠΟΔΕΙΞΗ: Σύμφωνα με το Λήμμα του Myerson που παρουσιάστηκε στην ενότητα 3.5.3, το ποσό της πληρωμής για έναν στρατηγικό παίχτη i σε ένα μηχανισμό DSIC με κανόνα κατανομής x συνεχή και παραγωγίσιμο ως προς το διάνυσμα προσωπικών αξιών \mathbf{v} δίνεται από την παρακάτω σχέση (αναφερόμαστε σε φιλαλήθεις μηχανισμούς):

$$p_i(v_i, \mathbf{v}_{-i}) = \int_0^{v_i} z \cdot \frac{\partial x_i(z, \mathbf{v}_{-i})}{\partial z} dz$$

Έστω τυχαίος στρατηγικός παίχτης i . Θα ισχύει πως:

$$\mathbb{E}_{v_i \sim F_i} [p_i(\mathbf{v})] = \int_0^{v_{max}} p_i(\mathbf{v}) f_i(v_i) dv_i = \int_0^{v_{max}} \left[\int_0^{v_i} z \cdot \frac{\partial x_i(z, \mathbf{v}_{-i})}{\partial z} dz \right] f_i(v_i) dv_i$$

Αλλάζοντας τη σειρά ολοκλήρωσης έχουμε πως:

$$\int_0^{v_{max}} \left[\int_0^{v_i} z \cdot \frac{\partial x_i(z, \mathbf{v}_{-i})}{\partial z} dz \right] f_i(v_i) dv_i = \int_0^{v_{max}} \left[\int_z^{v_{max}} f_i(v_i) dv_i \right] z \cdot \frac{\partial x_i(z, \mathbf{v}_{-i})}{\partial z} dz = \int_0^{v_{max}} (1 - F_i(z)) \cdot z \cdot \frac{\partial x_i(z, \mathbf{v}_{-i})}{\partial z} dz$$

Στη συνέχεια πραγματοποιούμε παραγοντική ολοκλήρωση και λαμβάνουμε:

$$\begin{aligned} \int_0^{v_{max}} \underbrace{(1 - F_i(z)) \cdot z}_{g(z)} \cdot \underbrace{\frac{\partial x_i(z, \mathbf{v}_{-i})}{\partial z}}_{h'(z)} dz &= \underbrace{[(1 - F_i(z)) \cdot z \cdot x_i(z, \mathbf{v}_{-i})]_0^{v_{max}}}_{=0-0} - \int_0^{v_{max}} x_i(z, \mathbf{v}_{-i}) \cdot (1 - F_i(z)) - \\ & z f_i(z) dz = \int_0^{v_{max}} \underbrace{\left(z - \frac{1 - F_i(z)}{f_i(z)} \right)}_{\varphi_i(z)} x_i(z, \mathbf{v}_{-i}) f_i(z) dz \end{aligned}$$

Γίνεται λοιπόν προφανές πως για z που ακολουθεί κατανομή F_i ισχύει πως:

$$\mathbb{E}_{v_i \sim F_i} [p_i(\mathbf{v})] = \mathbb{E}_{v_i \sim F_i} [\varphi_i(v_i) \cdot x_i(\mathbf{v})]$$

Με τη βοήθεια της παραπάνω πρότασης θα αποδείξουμε τα αναμενόμενα συνολικά έσοδα είναι ίσα με την αναμενόμενη συνολική κοινωνική ωφέλεια. ■

Θεώρημα 2: Ισότητα συνολικών αναμενόμενων εσόδων με συνολική αναμενόμενη ωφέλεια

Σε κάθε περιβάλλον μοναδικής παραμέτρου με κατανομές προσωπικής αξίας F_1, F_2, \dots, F_n και για κάθε μηχανισμός DSIC ισχύει πως:

$$\mathbb{E}_{\mathbf{v} \sim \mathbf{F}} \left[\sum_{i=1}^n p_i(\mathbf{v}) \right] = \mathbb{E}_{\mathbf{v} \sim \mathbf{F}} \left[\sum_{i=1}^n \varphi_i(v_i) \cdot x_i(\mathbf{v}) \right]$$

ΑΠΟΔΕΙΞΗ: Από την πιο πάνω πρόταση, θεωρώντας πως $\mathbf{v}_{-i} \sim \mathbf{F}_{-i}$ έχουμε πως:

$$\mathbb{E}_{v_i \sim F_i}[p_i(\mathbf{v})] = \mathbb{E}_{v_i \sim F_i}[\varphi_i(v_i) \cdot x_i(\mathbf{v})]$$

Λόγω της γραμμικότητας της μέσης τιμής έχουμε:

$$\mathbb{E}_{\mathbf{v} \sim \mathbf{F}}[\sum_{i=1}^n p_i(\mathbf{v})] = \sum_{i=1}^n \mathbb{E}_{\mathbf{v} \sim \mathbf{F}}[p_i(\mathbf{v})] = \sum_{i=1}^n \mathbb{E}_{\mathbf{v} \sim \mathbf{F}}[\varphi_i(v_i) \cdot x_i(\mathbf{v})] = \mathbb{E}_{\mathbf{v} \sim \mathbf{F}}[\sum_{i=1}^n \varphi_i(v_i) \cdot x_i(\mathbf{v})]$$

Το δεύτερο μέλος της ισότητας συνιστά την πιο εύκολα μεγιστοποιήσιμη μαθηματική έκφραση. Πρακτικά παρατηρούμε πως είναι σαν να έχουμε αντικαταστήσει τα v_i με τα $\varphi_i(v_i)$. Βλέπουμε πως η μεγιστοποίηση των εσόδων για DSIC μηχανισμούς ανάγεται στην μεγιστοποίηση της εικονικής ωφέλειας των στρατηγικών παιχτών. ■

3.7.5 Μεγιστοποίηση Αναμενόμενης Εικονικής Ωφέλειας

Στην ενότητα αυτή θα μελετήσουμε πως θα μεγιστοποιήσουμε την έκφραση $\mathbb{E}_{\mathbf{v} \sim \mathbf{F}}[\sum_{i=1}^n \varphi_i(v_i) \cdot x_i(\mathbf{v})]$. Μία προσέγγιση είναι να μεγιστοποιήσουμε για κάθε ξεχωριστό \mathbf{v} επιλέγοντας κατάλληλο κανόνα κατανομής $\mathbf{x}(\mathbf{v})$ που μεγιστοποιεί την εικονική ωφέλεια $\sum_{i=1}^n \varphi_i(v_i)x_i(\mathbf{v})$. Αυτός ο κανόνας κατανομής ονομάζεται **κανόνας κατανομής μεγιστοποίησης εικονικής ωφέλειας**. Ένα παράδειγμα τέτοιου κανόνα δίνεται παρακάτω μελετώντας ξανά το γνωστό παράδειγμα της ενότητας.

Παράδειγμα 3.6: Ένα αντικείμενο - Ένας στρατηγικός παίχτης - Κανόνας κατανομής

Θεωρούμε τις ίδιες συνθήκες όπως και προηγουμένως.

- Το virtual valuation δίνεται από τη σχέση $\varphi_i(v_i) = 2v_i - 1$, για κάθε \mathbf{v} .
- Είναι εύκολο να φανταστεί κανείς πως το αντικείμενο θα δοθεί στον στρατηγικό παίχτη με το υψηλότερο virtual valuation.
- Αυτό δεν ισχύει πάντοτε καθώς στο διάστημα $[0, 1]$ το virtual valuation μπορεί να πάρει και αρνητικές τιμές και έτσι δε μεγιστοποιείται η εικονική ωφέλεια.
- Συνεπώς, για να μεγιστοποιηθεί η εικονική ωφέλεια ο κανόνας κατανομής μεγιστοποίησης εικονικής ωφέλειας θα πρέπει να δίνει το αντικείμενο στον στρατηγικό παίχτη το υψηλότερο **θετικό** virtual valuation.

Αν ο κανόνας κατανομής μεγιστοποίησης εικονικής ωφέλειας είναι μονότονος, μπορεί ολόκληρος ο μηχανισμός να μετατραπεί σε ένα DSIC μηχανισμό πάνω στον οποίο θα εφαρμόσουμε το Λήμμα του Myerson προκειμένου να βρεθεί ο κατάλληλος κανόνας πληρωμής που μεγιστοποιεί τα αναμενόμενα έσοδα. Για να έχουμε μονότονο κανόνα κατανομής θα πρέπει οι πιθανοτικές κατανομές των προσωπικών αξιών να πληρούν μία ιδιότητα. Παρακάτω περιγράφεται η ιδιότητα αυτή.

Ορισμός 24: Κανονικές κατανομές

Μια κατανομή F καλείται κανονική όταν το virtual valuation της $v - \frac{1-F(v)}{f(v)}$ είναι μη φθίνουσα συνάρτηση.

Πρόταση 4: Μονότονος Κανόνας Κατανομής Μεγιστοποίησης Κοινωνικής Ωφέλειας

Αν η προσωπική αξία κάθε στρατηγικού παίχτη ακολουθεί κανονική κατανομή, τότε ο κανόνας κατανομής μεγιστοποίησης εικονικής ωφέλειας είναι μονότονος.

Παρακάτω παρουσιάζεται ο μηχανισμός που ακολουθείται για τη μεγιστοποίηση της αναμενόμενης εικονικής ωφέλειας και κατά συνέπεια για τη μεγιστοποίηση των αναμενόμενων κερδών.

Μεθοδολογία 2: Μηχανισμός Μεγιστοποίησης Εικονικής Ωφέλειας

Θεωρούμε πως η κατανομή προσωπικής αξίας F_i για κάθε στρατηγικό παίχτη είναι κανονική.

1. Μετατρέπουμε την προσφορά $b_i = v_i$ του κάθε στρατηγικού παίχτη i στο αντίστοιχο virtual valuation $\varphi_i(v_i)$.
2. Επιλέγουμε τη εφικτή κατανομή (x_1, x_2, \dots, x_n) που μεγιστοποιεί την εικονική ωφέλεια $\sum_{i=1}^n \varphi_i(v_i)x_i(\mathbf{v})$.
3. Χρησιμοποιούμε τον κανόνα πληρωμής που μας προσφέρει το Λήμμα του Myerson.

Παρατηρούμε πως ο μηχανισμός μεγιστοποίησης αναμενόμενων εσόδων είναι σχεδόν ίδιος με τη μεγιστοποίηση με αυτόν της μεγιστοποίησης κοινωνικής ωφέλειας που μελετήσαμε σε προηγούμενη ενότητα.

Ο μηχανισμός αυτός μπορεί να απλοποιηθεί ακόμα περισσότερο. Αν επιστρέψουμε στις δημοπρασίες μοναδικού αντικειμένου και θεωρήσουμε πως όλοι οι στρατηγικοί παίχτες ακολουθούν ανεξάρτητες και ισόνομες κανονικές κατανομές F μπορούμε να χρησιμοποιήσουμε κανόνα κατανομής μεγιστοποίησης εικονικής ωφέλειας που δίνει το αντικείμενο στο στρατηγικό παίχτη με το υψηλότερο θετικό virtual valuation. Ο κανόνας κατανομής είναι ίδιος με αυτόν που χρησιμοποιείται σε δημοπρασία second-price με reserve price $= \varphi^{-1}(0)$, δηλαδή την τιμή για την οποία το virtual valuation γίνεται 0. Επομένως, με τον πολύ απλό αυτό τρόπο μπορούμε να μεγιστοποιήσουμε τα αναμενόμενα έσοδα. Την τεχνική αυτή χρησιμοποίησε με επιτυχία η εταιρεία Yahoo το 2008 προκειμένου να μεγιστοποιήσει τα έσοδά της στις δημοπρασίες ηλεκτρονικών διαφημίσεων θεωρώντας πως όλοι οι στρατηγικοί παίχτες έχουν προσωπικές αξίες που ακολουθούν κατανομή lognormal.

Ολοκληρώνουμε την ενότητα αυτή αναφέροντας ένα θεώρημα το οποίο προσφέρει ένα πολύ σημαντικό ποιοτικό αποτέλεσμα για τους σχεδιαστές μηχανισμών με στόχο τη μεγιστοποίηση εσόδων.

Θεώρημα 3: Θεώρημα Bulow-Klemperer [6]

Έστω F κανονική κατανομή και n θετικός αέριαιος. Έστω επίσης \mathbf{p} και \mathbf{p}^* οι κανόνες πληρωμής μίας δημοπρασίας second-price με $n + 1$ στρατηγικούς παίχτες και μίας βέλτιστης ως προς τη μεγιστοποίηση του κέρδους δημοπρασίας (ως προς την F) με στρατηγικούς παίχτες. Αποδεικνύεται πως:

$$\mathbb{E}_{\mathbf{v} \sim F^{n+1}} [\sum_{i=1}^{n+1} p_i(\mathbf{v})] \geq \mathbb{E}_{\mathbf{v} \sim F^n} [\sum_{i=1}^n p_i^*(\mathbf{v})]$$

Το θεώρημα Bulow-Klemperer υποδηλώνει πως είναι πιο σημαντικό για τη μεγιστοποίηση εσόδων το να αυξάνονται οι συμμετέχοντες σε μια δημοπρασία παρά το να χρησιμοποιείς το βέλτιστο μηχανισμό. Ακόμα και για έναν παραπάνω στρατηγικό παίχτη στη δημοπρασία τα έσοδα είτε αυξάνονται είτε στη χειρότερη περίπτωση παραμένουν σταθερά. Το αποτέλεσμα αυτό επιβεβαιώνεται από το πιο πάνω θεώρημα και έχει πολύ μεγάλη πρακτική εφαρμογή. Επειδή το παραπάνω θεώρημα μας ενδιαφέρει μόνο ποιοτικά η απόδειξη του παραλείπεται.

3.8 Τυχαίες Δημοπρασίες

Κλείνουμε το κεφάλαιο κάνοντας μια εισαγωγή στις τυχαίες (randomized) δημοπρασίες. Διατηρώντας το συμβολισμό από τις προηγούμενες ενότητες παρουσιάζουμε βασικούς ορισμούς.

3.8.1 Ορισμός

Αρχικά, δίνουμε τον ορισμό της τυχαίας δημοπρασίας.

Ορισμός 25: Τυχαίες δημοπρασίες

Μία δημοπρασία αποκαλείται τυχαία αν η διαδικασία από την οποία υπολογίζεται η κατανομή των αντικειμένων στους στρατηγικούς παίχτες και η τιμή πώλησης αυτών συνιστά τυχαία διαδικασία. Αν μία δημοπρασία είναι τυχαία τότε τα έσοδα της, οι τιμές πώλησης και η κατανομή των αντικειμένων είναι τυχαίες μεταβλητές.

Στη συνέχεια θα ασχοληθούμε με τη φιλαλήθεια στις τυχαίες δημοπρασίες.

3.8.2 Φιλαλήθεια στις Τυχαίες Δημοπρασίες

Προκειμένου να μιλήσουμε για φιλαλήθεια στις τυχαίες δημοπρασίες θα πρέπει να ορίσουμε την έννοια της ανεξαρτησίας προσφοράς.

Ορισμός 26: Δημοπρασίες ανεξάρτητες από την προσφορά

Έστω στρατηγικός παίχτης i , \mathbf{b}_{-i} το διάνυσμα των προσφορών των στρατηγικών παιχτών δίχως την προσφορά του στρατηγικού παίχτη i και συνάρτηση $f : \mathbf{b}_{-i} \rightarrow \mathbb{R}_{\geq 0}$.

Μία δημοπρασία καλείται ανεξάρτητη από την προσφορά αν έχει την παρακάτω μορφή:

- Για κάθε στρατηγικό παίχτη i υπολογίζει κατώφλι $t_i = f(\mathbf{b}_{-i})$.
- Αν $b_i \geq t_i$, τότε ο στρατηγικός παίχτης i κερδίζει το αντικείμενο και πληρώνει τιμή ίση με t_i .
- Σε αντίθετη περίπτωση ο στρατηγικός παίχτης χάνει.

Στη συνέχεια, ορίζουμε ξανά τη φιλαλήθεια των ντετερμινιστικών δημοπρασιών με τη βοήθεια των δημοπρασιών ανεξάρτητων από την προσφορά.

Πρόταση 5: Φιλαλήθεια και ντετερμινιστικές δημοπρασίες

Κάθε ντετερμινιστική δημοπρασία ανεξάρτητη από την προσφορά είναι φιλαλήθης.

ΑΠΟΔΕΙΞΗ: Έστω στρατηγικός παίχτης i με κατώφλι t_i .

- Αν ποντάρει πάνω από v_i και $v_i \geq t_i$ τότε κερδίζει και η ωφέλεια του γίνεται $v_i - t_i$.
- Αν ποντάρει πάνω από v_i και $v_i < t_i$ υπάρχει περίπτωση η ωφέλεια του να είναι αρνητική.
- Αντίθετα, αν ποντάρει κάτω από v_i και $v_i \geq t_i$ υπάρχει περίπτωση να χάσει και η ωφέλεια του να είναι 0 αντί για θετική.
- Αντίθετα, αν ποντάρει κάτω από v_i και $v_i < t_i$ η ωφέλεια του παραμένει 0.

Σε κάθε περίπτωση η ωφέλεια του γίνεται μέγιστη με σιγουριά όταν ο στρατηγικός παίχτης είναι φιλαλήθης. ■

Πρόταση 6: Ντετερμινιστικές δημοπρασίες και ντετερμινιστικές ανεξάρτητες από την προσφορά δημοπρασίες

Κάθε φιλαλήθης ντετερμινιστική δημοπρασία είναι ισοδύναμη με μία ντετερμινιστική δημοπρασία ανεξάρτητη από την προσφορά.

ΑΠΟΔΕΙΞΗ: Δοθείσας μιας τυχαία ντετερμινιστικής δημοπρασίας \mathcal{A} μπορούμε να προσδιορίσουμε συνάρτηση f τέτοια ώστε η ντετερμινιστική δημοπρασία ανεξάρτητη από την προσφορά \mathcal{A}_f να είναι ισοδύναμη με την \mathcal{A} .

Έστω $\mathbf{b}_i^x = (b_1, \dots, b_{i-1}, x, b_{i+1}, \dots, b_n)$ το διάνυσμα προσφορών που προκύπτει άμα αντικαταστήσουμε το b_i με το x . Αν υπάρχει κάποια τιμή x^* τέτοια ώστε στη $\mathcal{A}(\mathbf{b}_i^{x^*})$ έτσι ώστε ο στρατηγικός παίχτης i να κερδίζει και να πληρώνει p τότε ορίζουμε $f(\mathbf{b}_{-i}) = p$.

Δεδομένης αυτής της τιμής p θα δείξουμε για την $\mathcal{A}(\mathbf{b}_i^x)$ πως:

- Αν ο στρατηγικός παίχτης i κερδίσει, πληρώνει p .
- Ο στρατηγικός παίχτης i αν κάνει προσφορά $> p$ κερδίζει.

Για το πρώτο, έστω πως υπάρχει κάποια προσφορά y τέτοια ώστε η $\mathcal{A}(\mathbf{b}_i^y)$ οδηγεί στον i να κερδίσει και να πληρώσει $q \neq p$. Δίχως βλάβη γενικότητας θεωρούμε πως $q > p$ και συνεπώς ένας στρατηγικός παίχτης με μεγαλύτερη προσφορά y θα έπρεπε να κερδίζει περισσότερα από ότι αν προσέφερε x^* . Αυτό έρχεται ενάντια στο φιλαλήθες σύστημα της \mathcal{A} και για το λόγο αυτό είναι ΑΤΟΠΟ.

Για το δεύτερο, έστω πως υπάρχει προσφορά $y \in (p, \infty)$ τέτοια ώστε ο στρατηγικός παίχτης i να μη κερδίζει προσφέροντας y . Παρατηρούμε πως στην περίπτωση αυτή η ωφέλεια θα έπρεπε να ήταν μεγαλύτερη από ότι αν πόνταρε x^* . Αυτό έρχεται ενάντια στο φιλαλήθες σύστημα της \mathcal{A} και για το λόγο αυτό είναι ΑΤΟΠΟ. ■

Συνδυάζοντας τις δύο πιο πάνω προτάσεις μπορούμε να καταλήξουμε στο ακόλουθο συμπέρασμα.

Θεώρημα 4: Φιλαλήθεια Ντετερμινιστικής Δημοπρασίας

Μία ντετερμινιστική δημοπρασία είναι φιλαλήθης αν και μόνο αν είναι ισοδύναμη με μία ντετερμινιστική δημοπρασία ανεξάρτητη προσφοράς.

Ολοκληρώνουμε την ενότητα συνδυάζοντας όλα τα παραπάνω για να ορίσουμε την έννοια της φιλαλήθειας στις τυχαίες δημοπρασίες.

Ορισμός 27: Φιλαλήθεια σε τυχαίες δημοπρασίες

Μία τυχαία δημοπρασία ανεξάρτητη προσφοράς είναι μία πιθανοτική κατανομή πάνω στο σύνολο των ντετερμινιστικών δημοπρασιών ανεξάρτητων προσφορών.

Συνεπώς, μια τυχαία δημοπρασία είναι φιλαλήθης αν και μόνο αν είναι ισοδύναμη με μία τυχαία δημοπρασία ανεξάρτητη προσφοράς.

Κεφάλαιο 4

Τεχνικές Εκτίμησης Κατανομών Προσωπικής Αξίας

Στο κεφάλαιο αυτό πραγματοποιείται μία εισαγωγή στις τεχνικές εκτίμησης των κατανομών προσωπικής αξίας των στρατηγικών παιχτών στις δημοπρασίες για εικονικές μηχανές. Παρουσιάζεται το θεωρητικό υπόβαθρο για την κλασική εκτίμηση παραμέτρων σε κατανομές, για τη lognormal κατανομή, για την απόσταση Kullback-Leibler, ενώ γίνεται μία εισαγωγή στα νευρωνικά δίκτυα, στους αλγόριθμους βελτιστοποίησης και περιγράφεται ένα νευρωνικό δίκτυο που θα χρησιμοποιηθεί για τη μεγιστοποίηση κερδών στις δημοπρασίες.

Στην ενότητα 4.1 γίνεται μια εισαγωγή στην κλασική εκτίμηση παραμέτρων εστιάζοντας στις εκτιμήτριες συναρτήσεις, στην εκτίμηση μέγιστης πιθανοφάνειας και στην εκτίμηση μέσης τιμής και διακύμανσης μίας τυχαίας μεταβλητής .

Στην ενότητα 4.2 παρουσιάζεται η κατανομή lognormal και ο εκτιμητής μέγιστης πιθανοφάνειας της.

Στην ενότητα 4.3 παρουσιάζεται ο αλγόριθμος Expectation-Maximization και μία τροποποίησή του για δημοπρασίες.

Στην ενότητα 4.4 παρουσιάζεται η εντροπία και η απόσταση Kullback-leibler.

Στην ενότητα 4.5 παρουσιάζονται τα νευρωνικά δίκτυα, εστιάζοντας στο απλό Perceptron, στα νευρωνικά δίκτυα πολλαπλών στρωμάτων και στις εκπαίδευση αυτών.

Στην ενότητα 4.6 παρουσιάζονται οι παραλλαγές των αλγορίθμων βελτιστοποίησης Gradient Descent εστιάζοντας στον Batch Gradient Descent, στον Stochastic Gradient Descent και στον Mini-Batch Gradient Descent εστιάζοντας παράλληλα στη μέθοδο Adaptive Moment Estimation (Adam)

Στην ενότητα 4.7 πραγματοποιείται μία εισαγωγή σε ένα νευρωνικό δίκτυο που στόχο έχει τη μεγιστοποίηση των κερδών σε δημοπρασίες ενός αντικειμένου που στο επόμενο κεφάλαιο θα προσαρμοστεί σε δημοπρασίες Knapsack.

4.1 Κλασική Εκτίμηση Παραμέτρων

Στην ενότητα αυτή θα παρουσιάσουμε την κλασική εκτίμηση παραμέτρων όπου η άγνωστη παράμετρος θ μίας κατανομής δεν είναι τυχαία, αλλά θεωρείται πως είναι άγνωστη σταθερά. Θα αναφερθούμε σε εκτιμήτριες συναρτήσεις, στις ιδιότητες αυτών και στην εκτίμηση μέγιστης πιθανοφάνειας ενώ παράλληλα θα παρουσιάσουμε και μερικά παραδείγματα κατανόησης.

4.1.1 Εκτιμήτριες Συναρτήσεις

Έστω ένα διάνυσμα παρατηρήσεων $X = (X_1, \dots, X_n)$. Μία εκτιμήτρια συνάρτηση $\hat{\theta}$ μίας άγνωστης παραμέτρου θ είναι μία τυχαία μεταβλητή της μορφή $\hat{\theta} = g(X)$, για κάποια συνάρτηση g . Αφού η κατανομή της X εξαρτάται από τη θ τότε και η $\hat{\theta}$ εξαρτάται από αυτή.

Όταν ενδιαφερόμαστε για τον αριθμό των παρατηρήσεων n , συμβολίζουμε εναλλακτικά την εκτιμήτρια συνάρτηση ως $\hat{\theta}_n$. Για κάθε τιμή του n έχουμε μία διαφορετική συνάρτηση, οπότε η $\hat{\theta}_n$ θεωρείται ακολουθία εκτιμητριών. Συμβολίζουμε τη μέση τιμή και τη διασπορά της $\hat{\theta}_n$ ως $\mathbb{E}_\theta[\hat{\theta}_n]$ και $\text{var}_\theta[\hat{\theta}_n]$ αντίστοιχα και ορίζονται με το γνωστό τρόπο. Τα δύο αυτά μεγέθη είναι προφανώς συναρτήσεις της άγνωστης παραμέτρου θ . Στη συνέχεια, παρουσιάζουμε ένα χρήσιμο σύνολο ορολογιών για τις εκτιμήτριες συναρτήσεις.

Ορισμός 28: Ορολογία εκτιμητριών συναρτήσεων

Έστω $\hat{\theta}_n$ μία εκτιμήτρια άγνωστης παραμέτρου θ , δηλαδή μία συνάρτηση από ένα διάνυσμα παρατηρήσεων $X = (X_1, \dots, X_n)$ των οποίων η κατανομή εξαρτάται από την άγνωστη παράμετρο θ . Για την εκτιμήτρια συνάρτηση ορίζουμε τα ακόλουθα:

- Το σφάλμα εκτίμησης $\tilde{\theta}_n$, ορίζεται ως $\tilde{\theta}_n = \hat{\theta}_n - \theta$.
- Η μεροληψία της εκτιμήτριας $b_\theta(\hat{\theta}_n)$ είναι η μέση τιμή του σφάλματος εκτίμησης, δηλαδή $b_\theta(\hat{\theta}_n) = \mathbb{E}_\theta[\tilde{\theta}_n] - \theta$.
- Η μέση τιμή, η διασπορά και η μεροληψία $b_\theta(\hat{\theta}_n)$ εξαρτώνται από την άγνωστη παράμετρο θ , ενώ το σφάλμα εκτίμησης εξαρτάται από τις παρατηρήσεις (X_1, \dots, X_n) .
- Η $\hat{\theta}_n$ καλείται αμερόληπτη αν ισχύει πως $\mathbb{E}_\theta[\hat{\theta}_n] = \theta$, για κάθε δυνατή τιμή της θ .
- Η $\hat{\theta}_n$ καλείται ασυμπτωτικά αμερόληπτη αν ισχύει πως $\lim_{n \rightarrow \infty} \mathbb{E}_\theta[\hat{\theta}_n] = \theta$, για κάθε δυνατή τιμή της θ .
- Η $\hat{\theta}_n$ καλείται συνεπής αν η ακολουθία $\hat{\theta}_n$ συγκλίνει στην πραγματική τιμή της άγνωστης παραμέτρου θ , κατά πιθανότητα, για κάθε δυνατή τιμή της θ .

Στις περισσότερες των περιπτώσεων δε μπορούμε να περιμένουμε η εκτιμήτρια να είναι ακριβώς ίση με την άγνωστη θ καθώς εξαρτάται από το διάνυσμα τυχαίων παρατηρήσεων $X = (X_1, \dots, X_n)$. Συνεπώς, το σφάλμα της εκτίμησης είναι πιθανό να μην είναι μηδενικό. Αν το μέσο σφάλμα της εκτίμησης είναι μηδέν για κάθε δυνατή τιμή της θ τότε η εκτιμήτρια είναι αμερόληπτη. Η ασυμπτωτική αμεροληψία πρακτικά εκφράζει πως όσο αυξάνεται το πλήθος των παρατηρήσεων τόσο η εκτιμήτρια γίνεται αμερόληπτη.

Πέρα από αμεροληψία πολλές φορές ενδιαφερόμαστε για το μέσο μέγεθος του σφάλματος εκτίμησης. Αυτό αποτυπώνεται στο μέγιστο τετραγωνικό σφάλμα το οποία δίνεται από τη σχέση: $\hat{\theta}_n^2 = b_\theta(\hat{\theta}_n^2) + \text{var}_\theta[\hat{\theta}_n]$. Πρόκειται για μία πολύ σημαντική σχέση καθώς οι όροι στη δεξιά πλευρά είναι αντικρουόμενοι. Συνήθως μία μείωση στη διασπορά οδηγεί σε αύξηση στη μεροληψία και για το λόγο αυτό μία καλή εκτιμήτρια συνάρτηση έχει μικρούς και τους δύο όρους.

Στη συνέχεια θα παρουσιάσουμε την εκτίμηση μέγιστης πιθανοφάνειας παρουσιάζοντας παραδείγματα εκτίμησης μέσης τιμής και διασποράς για τυχαίες μεταβλητές.

4.1.2 Εκτίμηση Μέγιστης Πιθανοφάνειας

Έστω πως το διάνυσμα παρατηρήσεων $X = (X_1, \dots, X_n)$ περιγράφεται από μία από κοινού συνάρτηση μάζας πιθανότητας $p_X(x; \theta)$, της οποίας η μορφή εξαρτάται από μία παράμετρο θ . Θεωρώντας ως x μία τυχαία τιμή του X μία εκτίμηση μέγιστης πιθανοφάνειας είναι μία τιμή της παραμέτρου που μεγιστοποιεί την αριθμητική συνάρτηση $p_X(x_1, \dots, x_n; \theta)$ ως προς θ . Δηλαδή:

$$\hat{\theta}_n = \arg \max_{\theta} p_X(x_1, \dots, x_n; \theta)$$

Για συνεχείς τυχαίες μεταβλητές έχουμε αντίστοιχα:

$$\hat{\theta}_n = \arg \max_{\theta} f_X(x_1, \dots, x_n; \theta)$$

Αναφερόμαστε στη $p_X(x; \theta)$ (ή τη $f_X(x; \theta)$ για X συνεχή) ως συνάρτηση πιθανοφάνειας. Σε περίπτωση που θεωρούμε πως οι παρατηρήσεις X_i είναι ανεξάρτητες έχουμε πως $p_X(x_1, \dots, x_n; \theta) = \prod_{i=1}^n p_{X_i}(x_i; \theta)$ (για διακριτές X_i). Στην περίπτωση αυτή είναι πιο εύκολο να μεγιστοποιήσουμε το λογάριθμο της ορίζοντας με τον τρόπο αυτό τη συνάρτηση λογαριθμικής πιθανοφάνειας η οποία για διακριτή τυχαία μεταβλητή δίνεται από την παρακάτω σχέση:

$$\log p_X(x_1, \dots, x_n; \theta) = \log \prod_{i=1}^n p_{X_i}(x_i; \theta) = \sum_{i=1}^n \log p_{X_i}(x_i; \theta)$$

Όταν η X είναι συνεχής ομοίως έχουμε :

$$\log f_X(x_1, \dots, x_n; \theta) = \log \prod_{i=1}^n f_{X_i}(x_i; \theta) = \sum_{i=1}^n \log f_{X_i}(x_i; \theta)$$

Για τη δεδομένη τιμή x η πιθανοφάνεια είναι η πιθανότητα να προκύψει η τιμή x που παρατηρήσαμε όταν η παράμετρος είναι ίση με θ . Η μέγιστη πιθανοφάνεια εκφράζει την τιμή της άγνωστης παραμέτρου θ υπό την οποία οι παρατηρήσεις έχουν τη μεγαλύτερη πιθανότητα να προκύψουν. Στη συνέχεια, παρουσιάζουμε ένα παράδειγμα εύρεσης μέσης τιμής και διασποράς με τη μέθοδο της μέγιστης πιθανοφάνειας σε κανονική κατανομή.

Παράδειγμα 4.1: Εύρεση μέσης τιμής και διασποράς με τη μέθοδο της μέγιστης πιθανοφάνειας σε κανονική κατανομή

Έστω n ανεξάρτητες παρατηρήσεις κανονικής κατανομής X_1, \dots, X_n . Θεωρούμε ως τη ζητούμενη παράμετρο το διάνυσμα $\theta = (\mu, v)$. Η συνάρτηση πιθανοφάνειας είναι η

$$f_X(x; \mu, v) = \prod_{i=1}^n f_{X_i}(x_i; \mu, v) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi v}} e^{-(x_i - \mu)^2 / 2v}$$

Μετά από κάποιους υπολογισμούς μπορεί να γραφτεί ως:

$$f_X(x; \mu, v) = \frac{1}{(2\pi v)^{n/2}} \cdot \exp\left\{-\frac{ns_n^2}{2v}\right\} \cdot \exp\left\{-\frac{n(m_n - \mu)^2}{2v}\right\}$$

όπου m_n η αριθμητική τιμή της τυχαίας μεταβλητής $M_n = \frac{1}{n} \sum_{i=1}^n X_i$ και s_n^2 η αριθμητική τιμή της τυχαίας μεταβλητής $\bar{S}_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - M_n)^2$.

Η λογαριθμική συνάρτηση πιθανοφάνειας είναι η

$$\log f_X(x; \mu, v) = -\frac{n}{2} \cdot \log(2\pi) - \frac{n}{2} \cdot \log(v) - \frac{ns_n^2}{2v} - \frac{n(m_n - \mu)^2}{2v}$$

Παραγωγίζοντας την παραπάνω συνάρτηση ως προς μ και v και θέσουμε τις παραγώγους ίσες με το μηδέν προκύπτουν οι εκτιμήσεις μέγιστης πιθανοφάνειας. Λαμβάνουμε:

$$\hat{\theta}_n = (m_n, s_n^2) \text{ και } \hat{\Theta}_n = (M_n, \bar{S}_n^2)$$

Η M_n είναι η δειγματική μέση τιμή και η \bar{S}_n^2 είναι η δειγματική διασπορά.

4.1.3 Εκτίμηση της Μέσης Τιμής και Διακύμανσης μίας Τυχαίας Μεταβλητής

Κλείνουμε την ενότητα κάνοντας αναφορά στο πρόβλημα της εκτίμησης της μέσης τιμής και διασποράς μίας κατανομής πιθανότητας χρησιμοποιώντας πάλι διάνυσμα παρατηρήσεων και χωρίς να έχουμε γνώση της κατανομής που ακολουθείται.

Έστω πάλι το διάνυσμα παρατηρήσεων $X = (X_1, \dots, X_n)$ με παρατηρήσεις ανεξάρτητες που ακολουθούν την ίδια κατανομή με κοινή άγνωστη μέση τιμή. Η πιο φυσική εκτιμήτρια της θ είναι η δειγματική μέση τιμή $M_n = \frac{X_1 + \dots + X_n}{n}$. Πρόκειται για αμερόληπτη εκτιμήτρια καθώς $\mathbb{E}_\theta[M_n] = \mathbb{E}_\theta[X] = \theta$. Το μέσο τετραγωνικό σφάλμα της M_n ισούται με τη διασπορά της v/n , όπου v κοινή διασπορά των X_i . Δεν είναι αναγκαστικά η εκτιμήτρια με τη μικρότερη διασπορά καθώς αν θεωρήσουμε την $\hat{\Theta}_n = 0$, η διασπορά της $\hat{\Theta}_n$ είναι μηδέν αλλά η μεροληψία της είναι $-\theta$. Το τετραγωνικό σφάλμα εξαρτάται από το θ και ισούται με θ^2 . Παρατηρούμε πως διαφορετικές εκτιμήτριες παρουσιάζουν διαφορετικές ιδιότητες. Παρακάτω παρουσιάζεται ένα παράδειγμα χρήσης διαφορετικών εκτιμητών για την εκτίμηση της μέσης τιμής και διασποράς μίας τυχαίας μεταβλητής κλείνοντας με τον τρόπο αυτό την ενότητα.

Παράδειγμα 4.2: Εκτίμηση της Μέσης Τιμής και Διασποράς μίας Τυχαίας Μεταβλητής - Πολλαπλοί εκτιμητές

Έστω το διάνυσμα παρατηρήσεων $X = (X_1, \dots, X_n)$ με παρατηρήσεις ανεξάρτητες που ακολουθούν την ίδια κατανομή με κοινή άγνωστη μέση τιμή θ και άγνωστη διασπορά v .

Αρχικά, ορίζουμε την παρακάτω εκτιμήτρια για τη μέση τιμή.

$$\hat{\Theta}_n = \frac{X_1 + \dots + X_n}{n+1}$$

Η εκτιμήτρια αυτή δεν είναι αμερόληπτη όπως η δειγματική μέση τιμή καθώς $\mathbb{E}_\theta[\hat{\Theta}_n] = \frac{n\theta}{n+1}$ και $b_\theta(\hat{\Theta}_n) = -\frac{\theta}{n+1}$. Όμως, $\lim_{n \rightarrow \infty} b_\theta(\hat{\Theta}_n) = 0$ και συνεπώς η $\hat{\Theta}_n$ είναι ασυμπτωτικά αμερόληπτη. Επιπλέον, έχει διασπορά μικρότερη αυτής της δειγματικής μέσης τιμής ίση $\text{var}_\theta(\hat{\Theta}_n) = \frac{vn}{(n+1)^2}$ και είναι ανεξάρτητη του θ . Το μέσο τετραγωνικό σφάλμα είναι ίσο με $\tilde{\Theta}_n^2 = b_\theta(\hat{\Theta}_n^2) + \text{var}_\theta[\hat{\Theta}_n] = \frac{\theta^2}{(n+1)^2} + \frac{vn}{(n+1)^2}$.

Βλέπουμε λοιπόν πως διαφορετικές εκτιμήτριες παρουσιάζουν διαφορετικές ιδιότητες για τη μέση τιμή. Το ίδιο θα δείξουμε παρακάτω και για τη διασπορά.

Έστω η φυσική εκτιμήτρια για τη διασπορά $\bar{S}_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - M_n)^2$. Γνωρίζοντας πως $\mathbb{E}_{(u,v)}[M_n] = \theta$ και $\mathbb{E}_{(u,v)}[X_i^2] = \theta^2 + v$ και $\mathbb{E}_{(u,v)}[M_n^2] = \theta^2 + \frac{v}{n}$ προκύπτει πως $\mathbb{E}_{(u,v)}[S_n^2] = \frac{n-1}{n}v$. Συνεπώς, βλέπουμε πως η S_n^2 δεν είναι αμερόληπτη παρόλο που όπως φαίνεται είναι ασυμπτωτικά αμερόληπτη. Μετά από κλιμάκωση μία αμερόληπτη εκτιμήτρια διασποράς είναι η $\hat{S}_n^2 = \frac{n}{n-1} \bar{S}_n^2$.

Βλέπουμε πως παρόλο η φυσική εκτιμήτρια της μέσης τιμής είναι αμερόληπτη, η φυσική εκτιμήτρια για τη διασπορά δεν είναι. Επιβεβαιώνεται και για τη διασπορά πως διαφορετικές εκτιμήτριες παρουσιάζουν διαφορετικές ιδιότητες.

4.2 Lognormal Κατανομή

Στην ενότητα αυτή θα μελετήσουμε ορισμένες ιδιότητες της lognormal κατανομής. Η συγκεκριμένη κατανομή έχει χρησιμοποιηθεί για τη μεγιστοποίηση των κερδών σε δημοπρασίες καθώς έχει βρεθεί πως η προσωπική αξία στρατηγικών παιχτών πολλές φορές ακολουθεί τη lognormal κατανομή.

4.2.1 Ορισμός

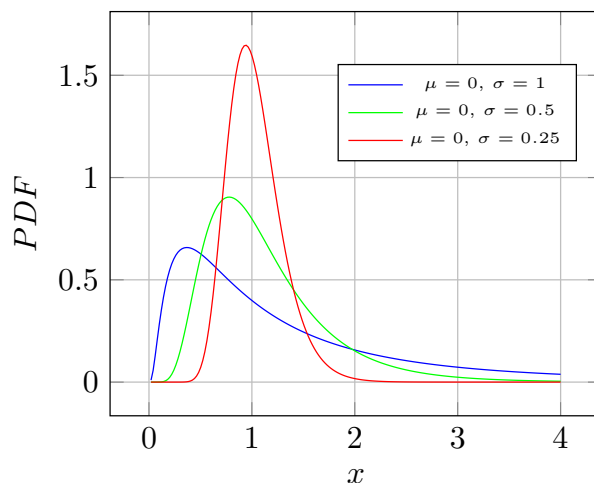
Πρόκειται για μίας συνεχή κατανομή μίας τυχαίας μεταβλητής της οποίας ο λογάριθμος ακολουθεί κανονική κατανομή. Συνεπώς, αν θεωρήσουμε πως η τυχαία μεταβλητή X ακολουθεί lognormal κατανομή, τότε η τυχαία μεταβλητή $Y = \ln(X)$ ακολουθεί κανονική κατανομή.

Προκύπτει πως για τυχαία μεταβλητή $X \sim \text{Lognormal}(\mu, \sigma^2)$ ισχύουν τα στοιχεία που εμφανίζονται στον παρακάτω πίνακα.

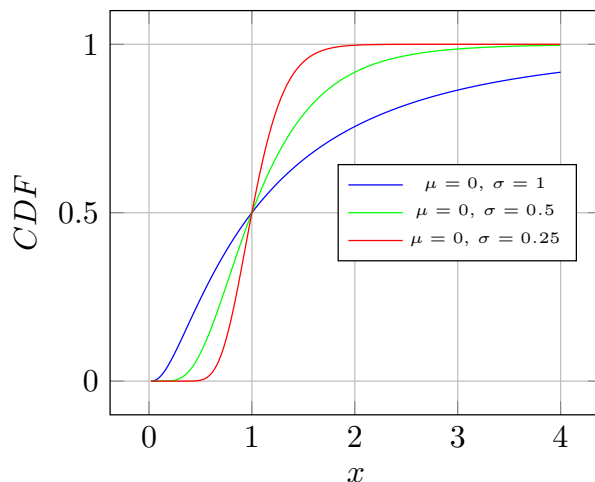
| Ιδιότητα | Τυπικός Συμβολισμός |
|----------------------------------|---|
| Συμβολισμός | $\text{Lognormal}(\mu, \sigma^2)$ |
| Παράμετροι | $\mu \in (-\infty, +\infty)$ και $\sigma > 0$ |
| Πεδίο ορισμού | $x > 0$ |
| Συνάρτηση Πυκνότητας Πιθανότητας | $\frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right)$ |
| Συνάρτηση Κατανομής Πιθανότητας | $\frac{1}{2} + \frac{1}{2} \text{erf}\left[\frac{\ln x - \mu}{\sqrt{2}\sigma}\right]$ |
| Αναμενόμενη Τιμή | $\exp\left(\mu + \frac{\sigma^2}{2}\right)$ |
| Διακύμανση | $[\exp(\sigma^2) - 1][\exp(2\mu + \sigma^2)]$ |

Πίνακας 4.1: Ιδιότητες Lognormal Κατανομής

Στη συνέχεια, παρουσιάζουμε 2 γραφικές παραστάσεις. Στην πρώτη απεικονίζεται η συνάρτηση πυκνότητας πιθανότητας της lognormal (PDF) και στη δεύτερη η συνάρτηση κατανομής πιθανότητας της lognormal (CDF). Στις δύο αυτές περιπτώσεις θεωρούμε σταθερό $\mu = 0$ και λαμβάνουμε τις τιμές $\{1, 0.5, 0.25\}$ για το σ .



Εικόνα 4.1: Συνάρτηση πυκνότητας πιθανότητας Lognormal



Εικόνα 4.2: Συνάρτηση κατανομής πιθανότητας Lognormal

Προχωράμε περιγράφοντας τη διαδικασία για την εύρεση εκτιμητριών συναρτήσεων για τον προσδιορισμό των παραμέτρων της κατανομής Lognormal πραγματοποιώντας εκτίμηση μέγιστης πιθανοφάνειας.

4.2.2 Εκτιμητής Μέγιστης Πιθανοφάνειας

Θεωρούμε το πρόβλημα της εκτίμησης της μέσης τιμής μ και της διασποράς σ μιας lognormal κατανομής χρησιμοποιώντας n ανεξάρτητες παρατηρήσεις X_1, \dots, X_n .

Έστω πως για την τυχαία μεταβλητή X έχουμε $X \sim \text{Lognormal}(\mu, \sigma^2)$ και φ η συνάρτηση πυκνότητας της κανονικής κατανομής $N(\mu, \sigma^2)$. Η συνάρτηση πιθανοφάνειας της Lognormal είναι η:

$$f_X(x; \mu, \sigma) = \prod_{i=1}^n \frac{1}{x_i} \varphi_{\mu, \sigma}(\ln x_i)$$

ενώ η λογαριθμική συνάρτηση πιθανοφάνειας είναι η:

$$\log f_X(x; \mu, \sigma) = -\sum_{k=1}^n \ln x_k + \sum_{k=1}^n \ln(\varphi_{\mu, \sigma}(\ln x_k))$$

Παραγωγίζοντας την παραπάνω συνάρτηση ως προς τις άγνωστες παραμέτρους και εξισώνοντας τις παραγώγους με το μηδέν, λαμβάνουμε τις εκτιμήσεις των ζητούμενων παραμέτρων που μεγιστοποιούν την πιθανοφάνεια. Αυτές είναι:

$$\hat{\mu} = \frac{\sum_{k=1}^n \ln x_k}{n}, \quad \hat{\sigma}^2 = \frac{\sum_{k=1}^n (\ln x_k - \hat{\mu})^2}{n}$$

Με την παραπάνω μελέτη κλείνουμε την ενότητα. Τα συγκεκριμένα αποτελέσματα θα χρησιμοποιηθούν στις προσεγγίσεις μας για την επίλυση του ζητούμενου προβλήματος.

4.3 Αλγόριθμος Expectation-Maximization (EM)

Στην ενότητα αυτή θα παρουσιάσουμε τον αλγόριθμο Expectation-Maximization [13] ή αλλιώς EM. Όπως και στην προηγούμενη ενότητα, στόχος μας είναι να εκτιμήσουμε όσο το δυνατόν καλύτερα γίνεται άγνωστες παραμέτρους πιθανοτικών κατανομών προκειμένου να βελτιστοποιήσουμε τα έσοδα του πωλητή της δημοπρασίας. Ο συγκεκριμένος αλγόριθμος είναι ιδιαίτερα χρήσιμος όταν δεδομένα λείπουν (missing data) από τις παρατηρήσεις μας. Πραγματοποιεί μία επαναληπτική διαδικασία στην οποία εκτιμά ανά επανάληψη τις άγνωστες παραμέτρους χρησιμοποιώντας τα υπάρχοντα δεδομένα.

4.3.1 Ορισμός

Έστω ένα σύνολο παρατηρήσεων $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ από μία κατανομή. Θεωρούμε πως κάποια χαρακτηριστικά λείπουν. Αυτά που λείπουν τα θεωρούμε "bad" (b) ενώ αυτά που υπάρχουν "good" (g). Κάθε τυχαία παρατήρηση \mathbf{x}_k μπορεί να γραφεί ως $\mathbf{x}_k = \{\mathbf{x}_{kg}, \mathbf{x}_{kb}\}$. Διαχωρίζουμε τα χαρακτηριστικά σε δύο σύνολα D_g και D_b για τα οποία ισχύει $D_g \cup D_b = D$. Στη συνέχεια, ορίζουμε την παρακάτω συνάρτηση:

$$Q(\theta; \theta^i) = \mathbb{E}_{D_b} [\ln p(D_g, D_b; \theta) | D_g; \theta^i]$$

Το semicolon δηλώνει για το αριστερό μέλος ότι το $Q(\theta; \theta^i)$ είναι συνάρτηση του θ θεωρώντας το θ^i σταθερό, ενώ για το δεξί μέλος δηλώνει ότι η αναμενόμενη τιμή υπολογίζεται πάνω στις τιμές που λείπουν θεωρώντας πως το θ^i εκφράζει τις πραγματικές παραμέτρους που περιγράφουν την πλήρη κατανομή.

Πιο συγκεκριμένα, το θ^i εκφράζει την τρέχουσα για την επανάληψη καλύτερη εκτίμηση για την πλήρη κατανομή, ενώ το θ πρόκειται για ένα υποψήφιο διάνυσμα παραμέτρων για μία καλύτερη εκτίμηση. Δεδομένου το διάνυσματος θ το δεξί μέλος της σχέσης υπολογίζει την πιθανοφάνεια των δεδομένων συμπεριλαμβανοντας τα άγνωστα χαρακτηριστικά D_b δεδομένου της τρέχουσας καλύτερης κατανομής που περιγράφεται από το θ^i . Ο αλγόριθμος επιλέγει το καλύτερο υποψήφιο διάνυσμα θ , δηλαδή αυτό που μεγιστοποιεί τη σχέση, για την επόμενη επανάληψη ονομάζοντας το θ^{i+1} . Παρακάτω δίνεται σε μορφή ψευδοκώδικα ο αλγόριθμος.

Algorithm 1 Expectation-Maximization

```

1: procedure EM-ALGORITHM
2:   initialize:  $\theta^0, T, i = 0$ 
3:   do  $i \leftarrow i + 1$ 
4:     E step: compute  $Q(\theta; \theta^i)$ 
5:     M step: compute  $\theta^{i+1} \leftarrow \arg \max_{\theta} Q(\theta; \theta^i)$ 
6:   while  $Q(\theta^{i+1}; \theta^i) - Q(\theta^i; \theta^{i-1}) > T$ 
7:   return  $\theta^{i+1}$ 

```

Ο αλγόριθμος Expectation-Maximization είναι ιδιαίτερα χρήσιμος όταν η βελτιστοποίηση της συνάρτησης $Q(\cdot, \cdot)$ είναι πιο εύκολη από τη βελτιστοποίηση της συνάρτησης μέγιστης πιθανοφάνειας. Επιπλέον, μας προσφέρει μίας πολύ σημαντική ιδιότητα. Σε κάθε βήμα, ο αλγόριθμος εγγυάται πως η λογαριθμική πιθανοφάνεια των "good" δεδομένων αυξάνεται μονότονα. Στη συνέχεια θα παρουσιάσουμε ένα ενδεικτικό παράδειγμα του αλγορίθμου Expectation-Maximization.

Παράδειγμα 4.3: Παράδειγμα αλγορίθμου EM

Έστω ότι το σύνολο των παρατηρήσεών μας αποτελείται από 4 σημεία σε δύο διαστάσεις, με ένα μόνο χαρακτηριστικό, το x_{41} , που απουσιάζει στην παρατήρηση \mathbf{x}_4 . Έχουμε $D = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\} = \left\{ \binom{0}{2}, \binom{1}{0}, \binom{2}{2}, \binom{*}{4} \right\}$. Το σύνολο D_b περιέχει το χαρακτηριστικό x_{41} , ενώ το D_g όλα τα υπόλοιπα. Θεωρούμε πως τα χαρακτηριστικά ακολουθούν δισδιάστατη κανονική κατανομή με διαγώνια συνδιακύμανση και άγνωστες μέσες τιμές. Συνεπώς, θεωρούμε το διάνυσμα παραμέτρων θ για το οποίο ισχύει

$$\theta = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \sigma_1^2 \\ \sigma_2^2 \end{bmatrix} \quad (4.1)$$

Θεωρούμε αναίρετα ως θ^0 το διάνυσμα:

$$\theta^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (4.2)$$

Υπολογίζουμε τη συνάρτηση $Q(\theta; \theta^0)$. Έχουμε

$$Q(\theta; \theta^0) = \mathbb{E}_{x_{41}}[\ln p(\mathbf{x}_g, \mathbf{x}_b; \theta | \theta^0; D_g)] = \int_{-\infty}^{+\infty} [\sum_{k=1}^3 \ln p(\mathbf{x}_k | \theta) + \ln p(\mathbf{x}_4 | \theta)] p(x_{41} | \theta^0; x_{42} = 4) dx_{41} =$$

$$\sum_{k=1}^3 [\ln p(\mathbf{x}_k | \theta)] + \int_{-\infty}^{+\infty} \ln p\left(\binom{x_{41}}{4} | \theta\right) \underbrace{\frac{p\left(\binom{x_{41}}{4} | \theta^0\right)}{\int_{-\infty}^{+\infty} p\left(\binom{x'_{41}}{4} | \theta^0\right) dx'_{41}}}_{=K} dx_{41}$$

Η σταθερά K βγαίνει εκτός ολοκληρώματος και επομένως έχουμε πως $Q(\theta; \theta^0)$ είναι

$$Q(\theta; \theta^0) = \sum_{k=1}^3 [\ln p(\mathbf{x}_k | \theta)] - \frac{1+\mu_1^2}{2\sigma_1^2} - \frac{(4-\mu_2)^2}{2\sigma_2^2} - \ln(2\pi\sigma_1\sigma_2)$$

Με τον τρόπο αυτό ολοκληρώνεται το βήμα E. Στη συνέχεια, για το βήμα M μεγιστοποιούμε με το γνωστό τρόπο την πιο πάνω συνάρτηση και λαμβάνουμε τις παραμέτρους του επόμενου βήματος. Αυτές είναι:

$$\theta^1 = \begin{bmatrix} 0.75 \\ 2.0 \\ 0.938 \\ 2.0 \end{bmatrix} \quad (4.3)$$

Πραγματοποιώντας μερικές ακόμα επαναλήψεις του αλγορίθμου παρατηρούμε πως ο αλγόριθμος συγκλίνει στις παραμέτρους $\boldsymbol{\mu} = \binom{1}{2}$ και $\boldsymbol{\Sigma} = \begin{pmatrix} 0.667 & 0 \\ 0 & 2.0 \end{pmatrix}$.

Παρατηρούμε πως ο αλγόριθμος EM μας οδηγεί στη μέγιστη πιθανοφάνεια για τα καλά δεδομένα (good). Για διαφορετικές τιμές κακών δεδομένων είναι πιθανό να είχαμε σύγκλιση σε διαφορετική λύση με ακόμα υψηλότερη πιθανοφάνεια.

Στη συνέχεια προχωράμε στην παρουσίαση της τροποποίησης που πραγματοποιήσαμε στον αλγόριθμο EM για την περίπτωση δημοπρασιών, η οποία θα μας βοηθήσει στην καλύτερη εκτίμηση των παραμέτρων των κατανομών προσωπικής αξίας των στρατηγικών παιχτών.

4.3.2 Αλγόριθμος Expectation-Maximization για Δημοπρασίες

Στην υποενότητα αυτή θα πραγματοποιήσουμε μια τροποποίηση του αλγορίθμου Expectation-Maximization για δημοπρασίες [10]. Προσπαθούμε να εκτιμήσουμε την κατανομή της προσωπικής αξίας κάθε στρατηγικού παίχτη. Ωστόσο, πολλοί στρατηγικοί παίχτες όταν εισέρχονται σε μία δημοπρασία με σκοπό να προσφέρουν αξία μικρότερη από την τρέχουσα reserve price, δεν το πραγματοποιούν καθώς γνωρίζουν πως σίγουρα δε θα κερδίσουν το επιθυμητό αγαθό. Αυτοί οι χρήστες περιέχουν προσφορές χρήσιμες για την εκτίμηση της κατανομής προσωπικής αξίας κάθε στρατηγικού παίχτη. Για το λόγο αυτό, όταν εκτιμούμε τις άγνωστες παραμέτρους μίας κατανομής πρέπει να λάβουμε υπόψη και τα δεδομένα που λείπουν.

Έστω πως όλοι οι στρατηγικοί παίχτες ακολουθούν κατανομή με συνάρτηση πυκνότητας πιθανότητας $f(x)$ που παραμετροποιείται από το διάνυσμα παραμέτρων $\theta: f(x|\theta)$ και ο ρυθμός αφίξεων τους ακολουθεί κατανομή με συνάρτηση πυκνότητας πιθανότητας $g(x)$ που παραμετροποιείται από το διάνυσμα παραμέτρων $\lambda: g(x|\lambda)$. Δεδομένων των παρατηρήσεων (X_1, \dots, X_n) που εκφράζουν προσφορές ίσες με τις προσωπικές αξίες κάθε στρατηγικού παίχτη που προσφέρει θέλουμε να βρούμε τις εκτιμήσεις μέγιστης πιθανοφάνειας των παραμέτρων. Για συντομία, χωρίζουμε τα δεδομένα σε (x_v, x_h) όπου τα x_v είναι αυτά που παρατηρούμε και x_h τα δεδομένα που απουσιάζουν.

Έστω m στρατηγικοί παίχτες συνολικά, n στρατηγικοί παίχτες που έκαναν προσφορά και $m-n$ το πλήθος των προσφορών που δεν έχουμε παρατηρήσει. Τυπικά, σύμφωνα με τον αλγόριθμο EM για το διάνυσμα παραμέτρων θ θα πρέπει στο βήμα E να υπολογίσουμε το ολοκλήρωμα

$$Q(\theta) = \int \log(p(x_h, x_v|\theta))p(x_h|x_v, \theta^{(old)}, \lambda^{(old)})dx_h$$

και στο βήμα M να υπολογίσουμε το νέο διάνυσμα παραμέτρων.

$$\theta^{(new)} = \arg \max_{\theta} (Q(\theta))$$

Παρόμοια διαδικασία χρησιμοποιείται και για το διάνυσμα παραμέτρων λ .

Το παραπάνω ολοκλήρωμα είναι δύσκολο να υπολογιστεί λόγω της πολύπλοκης διαδικασίας προσφορών που ακολουθείται στις δημοπρασίες. Σύμφωνα με το μοντέλο μας, στο βήμα E του αλγορίθμου πρέπει να δημιουργήσουμε δείγματα από την κατανομή $p(x_h|x_v, \theta^{(old)}, \lambda^{(old)})$. Παρακάτω παρουσιάζουμε μία μέθοδο που προσομοιώνει αυτή τη διαδικασία.

Μεθοδολογία 3: Δημιουργία άγνωστων προσφορών στο βήμα E

1. Ταξινομούμε τις προσφορές x_v σε αύξουσα σειρά και ονομάζουμε την μικρότερη p .
2. Λαμβάνουμε m από την κατανομή $g(m|\lambda^{(old)})$.
3. Αν $m < n$, απορρίπτουμε το m και γυρνάμε πίσω στο βήμα 2.
4. Στη συνέχεια προσομοιώνουμε τη διαδικασία προσφορών χρησιμοποιώντας τις προσφορές x_v και τις $m - n$ προσφορές που λείπουν.
5. Όσο $|x_h| < m - n$ λαμβάνουμε μία τυχαία προσφορά από την κατανομή $f(x, \theta^{(old)})$ και τη συγκρίνουμε με την p . Αν είναι μικρότερη από την p τότε κρατάμε την προσφορά στο σύνολο x_h . Αλλιώς απορρίπτουμε. Επαναλαμβάνουμε μέχρι $|x_h| = m - n$.

Έχοντας λάβει τα δείγματα προχωράμε στο βήμα M στο οποίο πραγματοποιούμε εκτίμηση μέγιστης πιθανοφάνειας. Προφανώς, η εκτίμηση αυτή διαφέρει ανάλογα με το είδος της κάθε κατανομής. Σε κάποιες κατανομές (κανονική, lognormal κ.τ.λ.) υπάρχουν κλειστοί τύποι που μας δίνουν τις τιμές των παραμέτρων που οδηγούν σε μέγιστη πιθανοφάνεια, ενώ σε κάποιες άλλες η εύρεση τους πραγματοποιείται με αριθμητικές μεθόδους. Ο αλγόριθμος τερματίζει όταν οι τιμές των παραμέτρων θ και λ συγκλίνουν.

Στην επόμενη ενότητα θα παρουσιάσουμε μία μετρική προκειμένου να υπολογίσουμε ποσοτικά το πόσο "καλή" είναι η εκτίμησή των παραμέτρων που πραγματοποιούμε τη βοήθεια του αλγορίθμου Expectation-Maximization.

4.4 Απόσταση Kullback–Leibler

Στην ενότητα αυτή θα παρουσιάσουμε την απόσταση Kullback–Leibler[11] (ή σχετική εντροπία) που αποτελεί ένα μέτρο της απόστασης μεταξύ δύο κατανομών. Αρχικά, παρουσιάζουμε την έννοια της εντροπίας [22].

4.4.1 Εντροπία

Παρακάτω δίνεται ο ορισμός της εντροπίας.

Ορισμός 29: Ορισμός εντροπίας για διακριτή τυχαία μεταβλητή

Έστω X μία διακριτή τυχαία μεταβλητή με πεδίο ορισμού/αλφάβητο \mathcal{X} με συνάρτηση μάζας πιθανότητας $p_X(x) = p(x) = Pr\{X = x\}$. Η εντροπία $H(X)$ της διακριτής τυχαίας μεταβλητής X δίνεται από τη σχέση:

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

Η εντροπία είναι μέτρο της μέσης αβεβαιότητας μίας τυχαίας μεταβλητής καθώς επίσης και ο μέσος αριθμός από bits που χρειάζονται για την περιγραφή μίας τυχαίας μεταβλητής. Επιπλέον, η εντροπία μπορεί να ερμηνευθεί ως η αναμενόμενη τιμή της τυχαίας μεταβλητής $\log \frac{1}{p(x)}$, όπου η X λαμβάνεται σύμφωνα με τη συνάρτηση μάζας πιθανότητας $p(x)$. Συνεπώς,

$$H(X) = \mathbb{E}_p[\log \frac{1}{p(X)}]$$

Σημείωση: Για την εντροπία θεωρούμε πως $\log(\cdot) = \log_2(\cdot)$ και χρησιμοποιούμε τη σύμβαση $0 \log 0 = 0$, η οποία δικαιολογείται εύκολα βάσει της συνέχειας, αφού $x \log x \rightarrow 0$ καθώς $x \rightarrow 0$. Επιπρόσθετα, επειδή $0 \leq p(x) \leq 1$, για κάθε $x \in \mathcal{X}$ προκύπτει πως $-\log p(x) = \log \frac{1}{p(x)} \geq 0$. Επομένως, ισχύει πάντα πως $H(X) \geq 0$.

Ομοίως, έστω X μία συνεχής τυχαία μεταβλητή με πεδίο ορισμού/αλφάβητο \mathcal{X} με συνάρτηση πυκνότητας πιθανότητας $f_X(x) = f(x)$. Η εντροπία $H(X)$ της συνεχούς τυχαίας μεταβλητής X δίνεται από τη σχέση:

$$H(X) = - \int_{\mathcal{X}} f(x) \log f(x)$$

Παρακάτω παρουσιάζεται ένα απλό παράδειγμα υπολογισμού εντροπίας για μία διακριτή τυχαία μεταβλητή.

Παράδειγμα 4.4: Υπολογισμός εντροπίας διακριτής τυχαίας μεταβλητής

Έστω η τυχαία μεταβλητή X για την οποία ισχύει πως:

$$X = \begin{cases} a, & \text{με πιθανότητα } \frac{1}{2} \\ b, & \text{με πιθανότητα } \frac{1}{4} \\ c, & \text{με πιθανότητα } \frac{1}{8} \\ d, & \text{με πιθανότητα } \frac{1}{8} \end{cases}$$

Η εντροπία της τυχαίας μεταβλητής X θα είναι:

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) = -\frac{1}{2} \log \frac{1}{2} - \frac{1}{4} \log \frac{1}{4} - \frac{1}{8} \log \frac{1}{8} - \frac{1}{8} \log \frac{1}{8} = \frac{7}{4} \text{ bits.}$$

4.4.2 Ορισμός Απόστασης Kullback–Leibler

Όπως προαναφέραμε, η εντροπία μίας τυχαίας μεταβλητής είναι ένα μέτρο της αβεβαιότητας καθώς και ένα μέτρο της μέσης ποσότητας πληροφορίας που απαιτείται για να περιγραφεί η τυχαία μεταβλητή. Ωστόσο, μόνη της δε μας προσφέρει τη δυνατότητα να συγκρίνουμε μεταξύ τους διαφορετικές κατανομές. Για το λόγο αυτό, χρησιμοποιούμε την απόσταση Kullback–Leibler.

Αντίθετα η απόσταση Kullback–Leibler ή αλλιώς σχετική εντροπία είναι ένα μέτρο της απόστασης μεταξύ δύο κατανομών. Πρόκειται για τον αναμενόμενο λογάριθμο του λόγου πιθανοφανειών δύο κατανομών. Η σχετική εντροπία $D(p||q)$ είναι ένα μέτρο του πόσο άστοχο είναι να θεωρήσουμε ότι η κατανομή είναι η q ενώ η πραγματική κατανομή είναι η p . Παρακάτω δίνουμε τον τυπικό ορισμό της απόστασης Kullback–Leibler ή σχετικής εντροπίας.

Ορισμός 30: Ορισμός απόστασης Kullback–Leibler

Έστω δύο διακριτές τυχαίες μεταβλητές X, Y με κοινό πεδίο ορισμού/αλφάβητο \mathcal{X} και με συναρτήσεις μάζας πιθανότητας $p(x)$ και $q(x)$ αντίστοιχα. Η απόσταση Kullback–Leibler ή σχετική εντροπία μεταξύ των δύο συναρτήσεων μάζας πιθανότητας ορίζεται ως εξής:

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = \mathbb{E}_p \log \frac{p(X)}{q(X)}$$

Στον παραπάνω ορισμό χρησιμοποιούμε τη σύμβαση $0 \log \frac{0}{0} = 0$, καθώς και τις συμβάσεις $0 \log \frac{0}{q} = 0$ και $p \log \frac{p}{0} = \infty$. Επομένως αν υπάρχει $x \in \mathcal{X}$ τέτοιο ώστε $p(x) > 0$ και $q(x) = 0$, τότε έχουμε $D(p||q) = \infty$.

Στο σημείο αυτό να τονίσουμε πως δεν πρόκειται για μία πραγματική απόσταση μεταξύ κατανομών, διότι δεν είναι συμμετρική και δεν ικανοποιεί την τριγωνική ανισότητα. Παρόλα αυτά, αποτελεί ένα πολύ καλό μέσο για τη σύγκριση ομοιότητας μεταξύ κατανομών. Προφανώς, όσο μικρότερη είναι η απόσταση Kullback–Leibler, τόσο πιο όμοιες είναι οι δύο κατανομές.

Ομοίως, έστω δύο συνεχείς τυχαίες μεταβλητές X, Y με κοινό πεδίο ορισμού/αλφάβητο \mathcal{X} και με συναρτήσεις πυκνότητας πιθανότητας $p(x)$ και $q(x)$ αντίστοιχα. Η απόσταση Kullback–Leibler ή σχετική εντροπία μεταξύ των δύο συναρτήσεων πυκνότητας πιθανότητας ορίζεται ως εξής:

$$D(p||q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

Παρακάτω παρουσιάζεται ένα απλό παράδειγμα υπολογισμού απόστασης Kullback–Leibler για δύο διακριτές τυχαίες μεταβλητές.

Παράδειγμα 4.5: Υπολογισμός απόστασης Kullback–Leibler

Έστω $\mathcal{X} = \{0, 1\}$ και δύο κατανομές p και q με πεδίο ορισμού το \mathcal{X} . Θεωρούμε πως $p(0) = 1 - r$, $p(1) = r$, $q(0) = 1 - s$ και $q(1) = s$. Τότε έχουμε:

$$D(p||q) = (1 - r) \log \frac{1-r}{1-s} + r \log \frac{r}{s} \text{ και } D(q||p) = (1 - s) \log \frac{1-s}{1-r} + s \log \frac{s}{r}$$

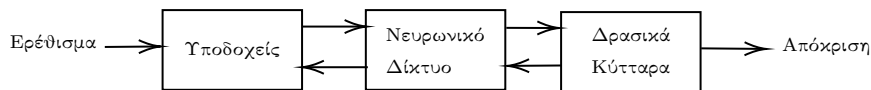
Αν τώρα $r = s$, προκύπτει πως $D(p||q) = D(q||p) = 0$ διότι προφανώς οι δύο κατανομές είναι ίδιες. Αν $r = \frac{1}{2}$ και $s = \frac{1}{4}$ τότε έχουμε:

$$D(p||q) = \frac{1}{2} \log \frac{\frac{1}{2}}{\frac{3}{4}} + \frac{1}{2} \log \frac{\frac{1}{2}}{\frac{1}{4}} = 0.2075 \text{ bits και } D(q||p) = \frac{3}{4} \log \frac{\frac{3}{4}}{\frac{1}{2}} + \frac{1}{4} \log \frac{\frac{1}{4}}{\frac{1}{2}} = 0.1887 \text{ bits}$$

Στο παράδειγμα αυτό βλέπουμε πως όταν οι κατανομές είναι ίδιες $D(p||q) = D(q||p) = 0$, ενώ γένει ισχύει πως $D(p||q) \neq D(q||p)$.

4.5 Νευρωνικά Δίκτυα

Στην ενότητα αυτή θα κάνουμε μία εισαγωγή στα νευρωνικά δίκτυα, τα οποία συνιστούν μία προσέγγιση στα προβλήματα μηχανικής μάθησης. Το πεδίο των νευρωνικών δικτύων βασίστηκε στο ότι ο ανθρώπινος εγκέφαλος εκτελεί υπολογισμούς εντελώς διαφορετικά από ότι ένας ψηφιακός υπολογιστής. Ο εγκέφαλος του ανθρώπου συνιστά ένα πολύπλοκο μη γραμμικό παράλληλο υπολογιστή. Το ανθρώπινο νευρικό σύστημα μπορεί να αντιμετωπιστεί ως ένα σύστημα τριών σταδίων όπως απεικονίζεται στην παρακάτω εικόνα.

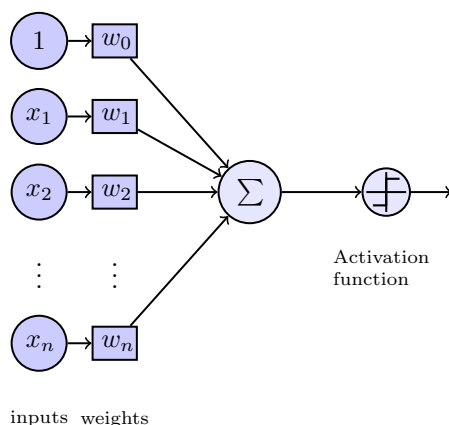


Εικόνα 4.3: Σχηματική Αναπαράσταση Νευρικού Συστήματος

Το κέντρο του συστήματος είναι εγκέφαλος ο οποίος αναπαρίσταται από το νευρωνικό δίκτυο. Αυτός προσλαμβάνει συνεχώς τις πληροφορίες, τις επεξεργάζεται και λαμβάνει τις κατάλληλες αποφάσεις. Τα βέλη που απεικονίζονται στο παραπάνω σχήμα έχουν δυο κατευθύνσεις. Αυτά με κατεύθυνση από τα δεξιά προς τα αριστερά υποδεικνύουν μετάδοση πληροφορίας προς τα εμπρός (πρόσθια τροφοδότηση) και αυτά από τα αριστερά προς τα δεξιά υποδεικνύουν την παρουσία ανάδρασης στο σύστημα. Οι υποδοχείς μετατρέπουν τα ερεθίσματα που προέρχονται από το ανθρώπινο σώμα ή το εξωτερικό περιβάλλον σε ηλεκτρικά σήματα που μεταφέρουν πληροφορία στο νευρικό δίκτυο, Τα δραστικά κύτταρα μετατρέπουν τα ηλεκτρικά σήματα που παράγονται στο νευρικό δίκτυο σε αισθητικές αποκρίσεις. Τα νευρωνικά δίκτυα κατασκευάστηκαν με στόχο να μιμηθούν τη λειτουργία του εγκεφάλου μας. Συνεχίζουμε την εισαγωγή στα νευρωνικά δίκτυα με την περιγραφή του Perceptron (μοντέλο ενός νευρώνα) που αποτέλεσε την πρώτη προσπάθεια για τη δημιουργία νευρωνικού δικτύου.

4.5.1 Το Perceptron

Το perceptron [19] [12] αποτελεί την απλούστερη δυνατή μορφή ενός νευρωνικού δικτύου και χρησιμοποιείται για την ταξινόμηση προτύπων τα οποία λέγεται πως είναι γραμμικά διαχωρίσιμα. Αποτελείται από ένα μεμονωμένο νευρώνα με προσαρμόσιμα συναπτικά βάρη και πόλωση (bias). Το μοντέλο αυτό προτάθηκε το 1958 από τον Rosenblatt ως το πρώτο μοντέλο μάθησης με τη συνδρομή ενός εκπαιδευτή (επιβλεπόμενη μάθηση). Σχηματικά απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 4.4: Perceptron

Το παραπάνω μοντέλο διαθέτει τρία βασικά χαρακτηριστικά. Αρχικά, αποτελείται από ένα σύνολο συνάψεων κάθε μία από εκ των οποίων διαθέτει το δικό της βάρος w_i . Ένα σήμα x_j στην είσοδο της σύναψης j που συνδέεται με το νευρώνα πολλαπλασιάζεται πρώτα επί το συναπτικό βάρος w_j . Επιπλέον, διαθέτει ένα αθροιστή για την

άθροιση των σημάτων εισόδου πραγματοποιώντας λειτουργίες γραμμικού συνδυαστή. Τέλος, αποτελείται από μία συνάρτηση ενεργοποίησης (activation function) για τον περιορισμό του πλάτους του σήματος εξόδου ενός νευρώνα. Υπάρχουν διάφορα είδη συναρτήσεων ενεργοποίησης τα οποία θα αναλυθούν παρακάτω. Στο σημείο αυτό θα πρέπει να επισημάσουμε την ύπαρξη της εξωτερικά εφαρμοζόμενης πόλωσης b η οποία στο παραπάνω σχήμα περιγράφεται ως είσοδος μεγέθους 1 σταθμισμένη με βάρος w_0 .

Έστω $\mathbf{x} = [+1, x_1, \dots, x_n]^T$ το διάνυσμα εισόδων και $\mathbf{w} = [b, w_1, \dots, w_n]^T$ το διάνυσμα συναπτικών βαρών. Η έξοδος v του γραμμικού συνδυαστή θα δίνεται από τη σχέση:

$$v = \sum_{i=0}^n w_i x_i = \mathbf{w}^T \mathbf{x}$$

Η εξίσωση $\mathbf{w}^T \mathbf{x} = 0$ αν απεικονιστεί σε ένα n -διάστατο χώρο με συντεταγμένες x_1, \dots, x_n ορίζει ένα υπερεπίπεδο ως διαχωριστική επιφάνεια απόφασης μεταξύ δύο διαφορετικών κλάσεων εισόδων.

Προκειμένου το perceptron να λειτουργήσει σωστά θα πρέπει οι δύο αυτές κλάσεις να είναι γραμμικά διαχωρίσιμες, δηλαδή τα προς ταξινόμηση πρότυπα να είναι επαρκώς διαχωρισμένα το ένα από το άλλο για να διασφαλιστεί ότι η επιφάνεια απόφασης αποτελείται από ένα υπερεπίπεδο. Δοθέντων διανυσμάτων και από τις δύο κατηγορίες για την εκπαίδευση του ταξινομητή, η διαδικασία εκπαίδευσης απαιτεί την προσαρμογή του διανύσματος βαρών \mathbf{w} με τέτοιο τρόπο ώστε οι δύο κλάσεις να είναι γραμμικά διαχωρίσιμες. Δηλαδή, να βρεθεί διάνυσμα βαρών \mathbf{w} τέτοιο ώστε:

$$\begin{aligned} \mathbf{w}^T \mathbf{x} &\geq 0, \text{ για κάθε διάνυσμα εισόδων } \mathbf{x} \text{ που ανήκει στην κλάση 1} \\ \mathbf{w}^T \mathbf{x} &< 0, \text{ για κάθε διάνυσμα εισόδων } \mathbf{x} \text{ που ανήκει στην κλάση 2} \end{aligned}$$

Στη συγκεκριμένη περίπτωση συνάρτηση ενεργοποίησης είναι η συνάρτηση προσήμου, για την οποία ισχύει πως:

$$f(v) = \begin{cases} 1, & \text{αν } v \geq 0 \\ -1, & \text{αν } v < 0 \end{cases}$$

Αποδεικνύεται πως το perceptron μετά το πέρας της εκπαίδευσης του μπορεί να ταξινομήσει σωστά πρότυπα από δύο γραμμικά διαχωρίσιμες κλάσεις. Για να συμβεί αυτό θα πρέπει να δοθεί κατάλληλος αριθμός εποχών. Σε κάθε μία εποχή το perceptron δέχεται όλα τα πρότυπα εισόδου. Αν σε μία εποχή δεν υπάρξει ούτε μία λανθασμένη ταξινόμηση τότε η διαδικασία εκπαίδευσης έχει ολοκληρωθεί. Πριν περιγράψουμε τη διαδικασία εκπαίδευσης του perceptron, στον παρακάτω πίνακα παρουσιάζουμε μερικές χρήσιμες συναρτήσεις ενεργοποίησης.

| Όνομα | Τυπικός Συμβολισμός | Ομαλότητα |
|--------------------|--|------------|
| Binary Step | $f(x) = \begin{cases} 1 & \text{αν } x \geq 0 \\ 0 & \text{αν } x < 0 \end{cases}$ | C^{-1} |
| Rectifier | $f(x) = \max\{x, 0\}$ | C^{-1} |
| Logistic (Sigmoid) | $f(x) = \frac{1}{1+e^{-x}}$ | C^∞ |
| TanH | $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ | C^∞ |
| ArcTan | $f(x) = \tan^{-1}(x)$ | C^∞ |

Πίνακας 4.2: Βασικές Συναρτήσεις Ενεργοποίησης

Η διαδικασία εκπαίδευσης του perceptron παρουσιάζεται παρακάτω.

Μεθοδολογία 4: Αλγόριθμος εκπαίδευσης perceptron

Μεταβλητές και Παράμετροι:

- $\mathbf{x}(n) =$ διάνυσμα εισόδων $(m + 1)$ επί $1 = [1, x_1, \dots, x_m]^T$, $\mathbf{w}(n) =$ διάνυσμα συναπτικών βαρών $(m + 1)$ επί $1 = [1, w_1, \dots, w_m]^T$
- $b =$ πόλωση, $\eta =$ παράμετρος ρυθμού μάθησης
- $y(n) =$ πραγματική απόκριση, $d(n) =$ επιθυμητή απόκριση
- Όρισε μέγιστο αριθμό εποχών.

Τα βήματα που πραγματοποιούνται είναι τα ακόλουθα:

1. Αρχικοποίηση με τυχαίες τιμές του διανύσματος $\mathbf{w}(n)$. Στη συνέχεια, εκτέλεσε τα ακόλουθα βήματα για χρονικές στιγμές $n = 1, 2, \dots$
2. Επανάλαβε έως ότου δεν γίνει καμία αλλαγή βαρών ή έχει συμπληρωθεί ο μέγιστος αριθμός εποχών. Τη χρονική στιγμή n , ενεργοποίησε το perceptron εφαρμόζοντας το διάνυσμα εισόδων $\mathbf{x}(n)$ και την επιθυμητή απόκριση $d(n)$.
3. Υπολόγισε την πραγματική απόκριση του perceptron $y(n) = \text{sgn}[\mathbf{w}^T \mathbf{x}]$.
4. Ενημέρωσε το διάνυσμα βαρών ως εξής: $\mathbf{w}(n + 1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n)$, όπου

$$d(n) = \begin{cases} 1, & \text{αν } \mathbf{x}(n) \text{ ανήκει στην κλάση 1} \\ -1, & \text{αν } \mathbf{x}(n) \text{ ανήκει στην κλάση 2} \end{cases}$$

5. Αν έχει συμπληρωθεί ο μέγιστος αριθμός εποχών επέστρεψε error, αλλιώς επέστρεψε τα βάρη.

Βλέπουμε λοιπόν πως το απλό perceptron αρκεί για την επίλυση του προβλήματος ταξινόμησης προτύπων που ανήκουν σε δύο γραμμικά διαχωρίσιμες κλάσεις. Παρόλα αυτά, παρουσιάζει αδυναμία επίλυσης πιο σύνθετων προβλημάτων. Για το λόγο αυτό, κατασκευάζουμε πολλαπλών στρωμάτων νευρωνικά δίκτυα με πολλά perceptron διατεταγμένα ανά στρώμα.

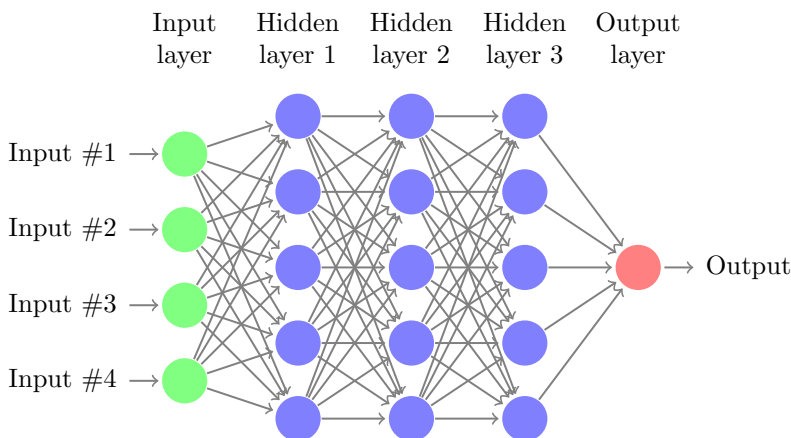
4.5.2 Νευρωνικά Δίκτυα Πολλαπλών Στρωμάτων

Στην ενότητα αυτή θα μιλήσουμε για perceptron πολλαπλών στρωμάτων-επιπέδων. Στην παρακάτω εικόνα φαίνεται ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο πολλαπλών στρωμάτων 4 χαρακτηριστικών εισόδου, μίας εξόδου και 3 κρυφών επιπέδων. Τα δίκτυα στο οποία θα αναφερθούμε είναι όλα πλήρως συνδεδεμένα, εννοώντας πως ένας νευρώνας σε οποιοδήποτε επίπεδο του δικτύου συνδέεται με όλους τους νευρώνες του προηγούμενου επιπέδου.

Σε ένα perceptron πολλαπλών στρωμάτων-επιπέδων παρατηρούμε δύο είδη σημάτων. Το πρώτο είδος είναι τα λειτουργικά σήματα (function signal) δηλαδή σήματα εισόδου το οποία διαδίδονται προς τα εμπρός και εμφανίζονται τελικά στην έξοδο του δικτύου. Το σήμα εισόδου εκτελεί μία χρήσιμη λειτουργία στην έξοδο του δικτύου και σε κάθε νευρώνα του ενδιάμεσου δικτύου το σήμα αυτό υπολογίζεται ως συνάρτηση των εισόδων και των σχετιζομένων βαρών που εφαρμόζονται σε αυτό τον νευρώνα. Η δεύτερη κατηγορία αφορά τα σήματα σφάλματος (error signal). Ένα τέτοιο σήμα προέρχεται από ένα νευρώνα εξόδου του δικτύου και διαδίδεται προς τα πίσω διαμέσου του δικτύου. Αποκαλείται σήμα σφάλματος επειδή ο υπολογισμός του απαιτεί τη χρήση μίας σχετικής συνάρτησης σφάλματος.

Οι κρυφοί νευρώνες απαρτίζουν τα κρυφά επίπεδα του δικτύου. Δεν ανήκουν ούτε στο επίπεδο εισόδου ούτε στο επίπεδο εξόδου. Για το λόγο ονομάζονται και κρυφοί. Δρουν ως "ανιχνευτές χαρακτηριστικών" και

διαδραματίζουν κρίσιμο ρόλο στη λειτουργία του νευρωνικού δικτύου. Κατά την πρόοδο της διαδικασίας μάθησης οι κρυφοί νευρώνες αρχίζουν σταδιακά να ανακαλύπτουν τα εξέχοντα χαρακτηριστικά που χαρακτηρίζουν τα δεδομένα εκπαίδευσης.



Εικόνα 4.5: Παράδειγμα Multilayer Perceptron με 3 Κρυφά Επίπεδα και Ένα Κόμβο Εξόδου

Στη συνέχεια θα μιλήσουμε για την εκπαίδευση ενός perceptron πολλαπλών επιπέδων.

4.5.3 Εκπαίδευση Νευρωνικού Δικτύου Πολλαπλών Στρωμάτων

Έστω $\{\mathbf{x}(n), \mathbf{d}(n)\}_{n=1}^N$ το δείγμα εκπαίδευσης. Έστω πως $y_j(n)$ η έξοδος του νευρώνα j στο επίπεδο εξόδου. Τότε, το σήμα σφάλματος που παράγεται στην έξοδο του νευρώνα j θα είναι:

$$e_j(n) = d_j(n) - y_j(n)$$

όπου d_j το j -οστό στοιχείο του διανύσματος επιθυμητών αποκρίσεων $\mathbf{d}(n)$. Η στιγμιαία ενέργεια σφάλματος του νευρώνα ορίζεται ως:

$$E_j(n) = \frac{1}{2} e_j^2(n)$$

ενώ η συνολική στιγμιαία ενέργεια σφάλματος του δικτύου ως

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

όπου το C περιλαμβάνει όλους τους νευρώνες στο επίπεδο εξόδου. Τέλος, καθώς το δείγμα εκπαίδευσης αποτελείται από N παραδείγματα, μέση ενέργεια σφάλματος ορίζεται ως:

$$E_{av}(N) = \frac{1}{N} \sum_{n=1}^N E(n) = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n)$$

Ένας από τους πιο σημαντικούς αλγόριθμους για την εκπαίδευση ενός νευρωνικού δικτύου είναι ο αλγόριθμος **Back Propagation**. Πριν περάσουμε στην περιγραφή του Back Propagation κάνουμε μία σύντομη περιγραφή του αλγόριθμου **Gradient Descent**.

Ο Gradient Descent είναι ένας επαναληπτικός αλγόριθμος βελτιστοποίησης για την εύρεση του ελαχίστου μίας συνάρτησης. Η ιδέα του βασίζεται στην πραγματοποίηση βημάτων προς το αρνητικό της κλίσης σε ένα συγκεκριμένο σημείο της συνάρτησης προς βελτιστοποίηση.

Έστω λοιπόν μία αντικειμενική συνάρτηση $\mathbf{F}(\mathbf{x})$ πολλαπλών μεταβλητών και έστω ένα σημείο \mathbf{a} στο οποίο η συνάρτηση $\mathbf{F}(\mathbf{x})$ ορίζεται και είναι παραγωγίσιμη σε μία γειτονιά του σημείου αυτού. Η αντικειμενική συνάρτηση $\mathbf{F}(\mathbf{x})$ μειώνεται γρηγορότερα αν μετακινηθεί από το σημείο \mathbf{a} σε σημείο αντίθετα της κατεύθυνσης της κλίσης της $\mathbf{F}(\mathbf{x})$ στο σημείο αυτό. Προκύπτει πως για $\gamma \in \mathbb{R}_+$ αρκετά μικρό το σημείο

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla \mathbf{F}(\mathbf{a}_n)$$

μας οδηγεί στη σχέση $\mathbf{F}(\mathbf{a}_n) \geq \mathbf{F}(\mathbf{a}_{n+1})$. Αφαιρώντας τον όρο $\gamma \nabla \mathbf{F}(\mathbf{a}_n)$ από το \mathbf{a} κινούμαστε προς το ελάχιστο της συνάρτησης. Επομένως, θεωρώντας αρχικό σημείο \mathbf{x}_0 τοπικό ελαχίστου λαμβάνουμε τα σημεία $\mathbf{x}_0, \mathbf{x}_1, \dots$ για τα οποία ισχύει:

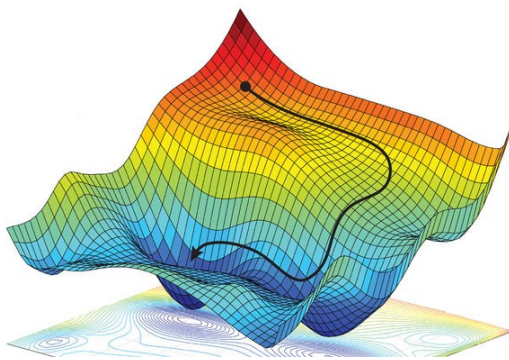
$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla \mathbf{F}(\mathbf{x}_n)$$

και λαμβάνουμε την ακολουθία:

$$\mathbf{F}(\mathbf{x}_0) \geq \mathbf{F}(\mathbf{x}_1) \geq \dots$$

η οποία πιστεύουμε πως θα μας οδηγήσει σε τοπικό ελάχιστο. Η σταθερά γ μπορεί να αλλάζει σε κάθε επανάληψη. Αποδεικνύεται πως αν γνωρίζουμε κάποιες συνθήκες για τη συνάρτηση $\mathbf{F}(\mathbf{x})$ (π.χ. $\mathbf{F}(\mathbf{x})$ κυρτή και $\nabla \mathbf{F}(\mathbf{x})$ Lipschitz) υπάρχει γ τέτοιο ώστε η $\mathbf{F}(\mathbf{x})$ συγγλίνει σίγουρα σε τοπικό ελάχιστο. Αν μάλιστα η $\mathbf{F}(\mathbf{x})$ είναι κυρτή τότε το τοπικό ελάχιστο είναι και ολικό.

Παρόλα αυτά, ο αλγόριθμος gradient descent παρουσιάζει κάποιους περιορισμούς. Είναι αργός όταν πλησιάζει το ελάχιστο, ενώ παράλληλα για ορισμένα προβλήματα είναι πιθανό να παρουσιάσει κινήσεις τύπου "ζικ-ζακ" γύρω από το ελάχιστο. Επιπλέον, δεν ορίζεται για μη διαφορίσιμες συναρτήσεις. Για να γίνει άρση κάποιων περιορισμών έχουν δημιουργηθεί διάφορες παραλλαγές του αλγορίθμου που θα αναλυθούν σε πιο κάτω ενότητα. Παρακάτω παρουσιάζεται σχηματικά η πορεία του αλγορίθμου gradient descent σε μία τρισδιάστατη κυρτή συνάρτηση.



Εικόνα 4.6: Πορεία Αλγορίθμου Gradient Descent σε Τρισδιάστατη Κυρτή Συνάρτηση.

Προχωράμε τώρα στην ανάλυση του αλγορίθμου Back Propagation. Ο αλγόριθμος Back Propagation αποτελείται από δύο περάσματα. Το πρώτο γίνεται με κατεύθυνση προς τα εμπρός και το δεύτερο με κατεύθυνση προς τα πίσω.

Στο πέρασμα με κατεύθυνση προς τα εμπρός, τα συναπτικά βάρη παραμένουν αμετάβλητα σε όλη την έκταση του δικτύου και τα λειτουργικά σήματα του δικτύου υπολογίζονται σε βάση νευρώνα προς νευρώνα έως ότου φτάσουμε στους κόμβους εξόδου. Έστω τυχαίος νευρώνας j . Το τοπικό πεδίο v_j που παράγεται στην είσοδο της συνάρτησης ενεργοποίησης που σχετίζεται με το νευρώνα j είναι

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n)$$

όπου m ο συνολικός αριθμός εισόδων που εφαρμόζονται στον νευρώνα j . Το συναπτικό βάθος w_{j0} ισούται με την πόλωση b_j που εφαρμόζεται στον νευρώνα j . Επομένως, το λειτουργικό σήμα $y_j(n)$ που εμφανίζεται στην έξοδο του νευρώνα j κατά την επανάληψη n είναι $y_j(n) = \varphi_j(v_j(n))$ όπου $\varphi_j(\cdot)$ η συνάρτηση ενεργοποίησης.

Στο πέρασμα με κατεύθυνση προς τα πίσω, ξεκινάμε από το επίπεδο εξόδου στέλνοντας τα σήματα σφάλματος προς τα αριστερά, σε όλα τα επίπεδα του δικτύου, επίπεδο προς επίπεδο, υπολογίζοντας την τοπική κλίση για κάθε νευρώνα μετατρέποντας με τον τρόπο αυτό κατάλληλα τα συναπτικά βάρη του δικτύου. Πιο συγκεκριμένα, εφαρμόζεται μία διόρθωση $\Delta w_{ji}(n)$ στο συναπτικό βάρος $w_{ji}(n)$, η οποία είναι ανάλογη με τη μερική παράγωγο $\frac{\partial E(n)}{\partial w_{ji}(n)}$. Σύμφωνα με τον κανόνα αλυσίδας ισχύει πως:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial y_j(n)}{\partial w_{ji}(n)}$$

Προκύπτει πως:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n), \quad \frac{\partial e_j(n)}{\partial y_j(n)} = -1, \quad \frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'_j(v_j(n)) \quad \text{και} \quad \frac{\partial y_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

Χρησιμοποιώντας όλα τα παραπάνω λαμβάνουμε πως:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi'_j(v_j(n)) y_i(n)$$

Επομένως, η διόρθωση $\Delta w_{ij}(n)$ που εφαρμόζεται στο $w_{ji}(n)$ ορίζεται από τον κανόνα:

$$\Delta w_{ij}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} = \eta \delta_j(n) y_i(n)$$

όπου η η παράμετρος μάθησης του αλγορίθμου Back Propagation και $\delta_j(n)$ η τοπική κλίση που ορίζεται ως $\delta_j(n) = \frac{\partial E(n)}{\partial v_j(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} = e_j(n) \varphi'_j(v_j(n))$. Η παράμετρος η είναι πρακτικά η παράμετρος γ του αλγορίθμου Gradient Descent και η χρήση του αρνητικού προσήμου υποδηλώνει τη χρήση του αλγορίθμου αυτού.

Όταν ο υπό μελέτη νευρώνας j είναι ένας κόμβος εξόδου τροφοδοτείται με την δική του επιθυμητή απόκριση και το είναι εύκολο να υπολογίσουμε το σήμα σφάλματος. Όταν ο νευρώνας j είναι ένας κρυφός νευρώνας τότε δεν υπάρχει καθορισμένη επιθυμητή απόκριση για αυτόν. Συνεπώς, το σήμα σφάλματος για ένα κρυφό νευρώνα πρέπει να καθοριστεί αναδρομικά, δουλεύοντας προς τα πίσω. Επαναπροσδιορίζοντας την τοπική κλίση και χρησιμοποιώντας τους τύπους ενέργειας που ορίσαμε παραπάνω προκύπτει πως:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial y_j(n)} \varphi'_j(v_j(n))$$

όπου $\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)}$ όπου k κόμβος εξόδου. Προκύπτει πως $e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n))$ και επομένως $\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi'_k(v_k(n))$. Συνεπώς για το νευρώνα k το τοπικό πεδίο δίνεται από τη σχέση $v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n)$ όπου m ο συνολικός αριθμός εισόδων που εφαρμόζονται στον νευρώνα k . Παραγωγίζοντας ως προς $y_j(n)$ λαμβάνουμε $\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$ και χρησιμοποιώντας τις πιο πάνω εξισώσεις καταλήγουμε πως $\frac{\partial E(n)}{\partial y_j(n)} = -\sum_k \delta_k(n) w_{kj}(n)$. Τέλος λαμβάνουμε τον τύπο οπισθοδιάδοσης για την τοπική κλίση $\delta_j(n)$ ο οποίος περιγράφεται από την

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

όπου ο νευρώνας j είναι κρυφός.

Ο εξωτερικός παράγοντας $\varphi'_j(v_j(n))$ που χρησιμοποιείται στον υπολογισμό της τοπικής κλίσης $\delta_j(n)$ εξαρτάται αποκλειστικά τη συνάρτηση ενεργοποίηση που σχετίζεται με τον κρυφό νευρώνα j . Ο άλλος παράγοντας για τον υπολογισμό αυτό εξαρτάται από δύο σύνολα όρων. Το πρώτο σύνολο όρων $\delta_k(n)$ απαιτεί γνώση των σημάτων σφάλματος $e_k(n)$ για όλους τους νευρώνες που βρίσκονται στο αμέσως δεξιότερο επίπεδο του κρυφού νευρώνα j και συνδέονται με τον νευρώνα j . Το δεύτερο σύνολο $w_{kj}(n)$ αποτελείται από τα συναπτικά βάρη που σχετίζονται με αυτές τις συνδέσεις.

Συνοπτικά λοιπόν έχουμε πως η διόρθωση $\Delta w_{ij}(n)$ που εφαρμόζεται στο συναπτικό βάρος που συνδέει το νευρώνα j με το νευρώνα i ορίζεται από τον παρακάτω κανόνα :

$$\Delta w_{ij}(n) = \left(\text{παράμετρος ρυθμού μάθησης} \eta \right) \times \left(\text{τοπική κλίση} \delta_j(n) \right) \times \left(\text{σήμα εισόδου νευρώνα} j y_i(n) \right)$$

και πως η τοπική κλίση $\delta_j(n)$ εξαρτάται από το εάν ο νευρώνας j είναι ένας νευρώνας εξόδου ή ένας κρυφός κόμβος.

1. Αν ο νευρώνας j είναι ένας κόμβος εξόδου, το $\delta_j(n)$ ισούται με το γινόμενο της παραγώγου $\varphi'(v_j(n))$ και του σήματος σφάλματος $e_j(n)$. Που σχετίζονται με τον νευρώνα j .
2. Αν ο νευρώνας j είναι ένας κρυφός κόμβος, το $\delta_j(n)$ ισούται με γινόμενο της σχετιζόμενης παραγώγου $\varphi'(v_j(n))$ και του σταθμισμένου αθροίσματος των δ που υπολογίζονται για τους νευρώνες του επόμενου κρυφού επιπέδου ή επιπέδου εξόδου και οι οποίοι συνδέονται με τον νευρώνα j .

Στο σημείο αυτό ολοκληρώνουμε την περιγραφή του αλγορίθμου Back Propagation, κλείνουμε την ενότητα και συνεχίζοντας περιγράφοντας κάποιες χρήσιμες παραλλαγές του αλγορίθμου Gradient Descent.

4.6 Περιγραφή Αλγορίθμων Βελτιστοποίησης Gradient Descent

Στην προηγούμενη ενότητα αναφερθήκαμε στον αλγόριθμο του Back Propagation κάνοντας αναφορά και στο γενικό αλγόριθμο του Gradient Descent. Στην ενότητα αυτή, θα περιγράψουμε διαφορετικούς αλγορίθμους Gradient Descent [21] αναλύοντας τη συμπεριφορά τους. Ξεκινάμε κάνοντας μία σύντομη υπενθύμιση της βασικής θεωρίας χρησιμοποιώντας λίγο διαφορετικό συμβολισμό.

Έστω μία αντικειμενική συνάρτηση $\mathbf{J}(\theta)$ πολλαπλών μεταβλητών και έστω ένα σημείο θ στο οποίο η συνάρτηση $\mathbf{J}(\theta)$ ορίζεται και είναι παραγωγίσιμη σε μία γειτονιά του σημείου αυτού. Η αντικειμενική συνάρτηση $\mathbf{J}(\theta)$ μειώνεται γρηγορότερα αν μετακινηθεί από το σημείο θ σε σημείο αντίθετα της κατεύθυνσης της κλίσης της $\mathbf{J}(\theta)$ στο σημείο αυτό. Ο ρυθμός μάθησης η καθορίζει το πλήθος των βημάτων που πραγματοποιούνται μέχρι να φτάσουμε σε (τοπικό ελάχιστο).

Παρακάτω παρουσιάζουμε τις τρεις παραλλαγές του αλγορίθμου Gradient Descent.

4.6.1 Batch Gradient Descent

Ο Batch Gradient Descent υπολογίζει την κλίση της αντικειμενικής συνάρτησης για ολόκληρο το διαθέσιμο σύνολο δεδομένων εκπαίδευσης. Τυπικά αυτό περιγράφεται ως εξής:

$$\theta = \theta - \eta \cdot \nabla_{\theta} \mathbf{J}(\theta)$$

Στην περίπτωση του Batch Gradient Descent για να ενημερώσουμε τις παραμέτρους μία φορά θα πρέπει να υπολογίσουμε τις κλίσεις για όλο το σύνολο δεδομένων εκπαίδευσης. Η διαδικασία αυτή είναι πολύ αργή για μεγάλα σύνολα δεδομένων εκπαίδευσης, ενώ υπάρχει και η δυνατότητα τα πολύ μεγάλα σύνολα δεδομένων εκπαίδευσης να μη χωρούν ολόκληρα στη μνήμη του υπολογιστή. Θεωρώντας σταθερό αριθμό εποχών *epochs* παρακάτω δίνεται ψευδοκώδικας που προσομοιώνει τη λειτουργία του αλγορίθμου Batch Gradient Descent.

Algorithm 2 Batch Gradient Descent

```

1: procedure BATCH GRADIENT DESCENT
2:   for i in range(epochs) do
3:     params_grad = evaluate_gradient(obj_function , data , params )
4:     params = params - learning_rate * params_grad

```

Αρχικά, υπολογίζουμε το διάνυσμα παραμέτρων της κλίσης της αντικειμενικής συνάρτησης πάνω σε όλα τα δεδομένα εκπαίδευσης και έπειτα υπολογίζουμε τις νέες παραμέτρους.

4.6.2 Stochastic Gradient Descent

Σε αντίθεση με τον Batch Gradient Descent, ο αλγόριθμος Stochastic Gradient Descent υπολογίζει την κλίση της αντικειμενικής συνάρτησης για κάθε ένα δείγμα εκπαίδευσης $x^{(i)}$ με ετικέτα $y^{(i)}$ από το διαθέσιμο σύνολο δεδομένων εκπαίδευσης ξεχωριστά κάθε φορά. Τυπικά αυτό περιγράφεται ως εξής:

$$\theta = \theta - \eta \cdot \nabla_{\theta} \mathbf{J}(\theta; x^{(i)}; y^{(i)})$$

Σε αντίθεση με τον Batch Gradient Descent, ο αλγόριθμος Stochastic Gradient Descent είναι αρκετά πιο γρήγορος και υπολογιστικά εφικτός. Επιπρόσθετα, το γεγονός ότι υπολογίζει την κλίση της αντικειμενικής συνάρτησης για κάθε ένα δείγμα εκπαίδευσης ξεχωριστά οδηγεί σε συχνές ενημερώσεις υψηλής διακύμανσης. Αυτό σημαίνει πως μπορεί να μας οδηγήσει σε νέα ελάχιστα. Παρόλα αυτά, η συγκεκριμένη ιδιαιτερότητα δυσχεραίνει τη γρήγορη σύγκλιση. Έχει αποδειχθεί πως αν μειώνουμε το ρυθμό μάθησης σταδιακά ο Stochastic Gradient Descent συγκλίνει σχεδόν πάντα σε ελάχιστο, δείχνοντας παρόμοια συμπεριφορά με τον Batch Gradient Descent. Θεωρώντας σταθερό αριθμό εποχών *epochs* παρακάτω δίνεται ψευδοκώδικας που προσομοιώνει τη λειτουργία του αλγορίθμου Stochastic Gradient Descent.

Algorithm 3 Stochastic Gradient Descent

```

1: procedure STOCHASTIC GRADIENT DESCENT
2:   for i in range(epochs) do
3:     data = shuffle(data)
4:     for example in data do
5:       params_grad = evaluate_gradient(obj_function , example , params )
6:       params = params - learning_rate * params_grad

```

Στο συγκεκριμένο ψευδοκώδικα, απλά πραγματοποιούμε ένα ανακάτεμα (shuffle) των δεδομένων πριν τον υπολογισμό της κλίσης η οποία αυτή τη φορά γίνεται για ένα δείγμα (example) του συνόλου δεδομένων κάθε φορά. Ο λόγος που πραγματοποιούμε ανακάτεμα παρουσιάζεται πιο κάτω.

4.6.3 Mini-Batch Gradient Descent

Ο αλγόριθμος Mini-Batch Gradient Descent είναι ένας υβριδικός συνδυασμός των πιο πάνω αλγορίθμων καθώς υπολογίζει την κλίση και ενημερώνει τις παραμέτρους για n δείγματα και ετικέτες από το διαθέσιμο σύνολο δεδομένων εκπαίδευσης. Τυπικά αυτό περιγράφεται ως εξής:

$$\theta = \theta - \eta \cdot \nabla_{\theta} \mathbf{J}(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$

Με τον τρόπο αυτό επιτυγχάνεται μικρότερη διακύμανση, σταθερότερη σύγκλιση και ευκολία στην υπολογιστική διαχείριση. Πολλές state-of-the-art deep learning βιβλιοθήκες χρησιμοποιούν τον αλγόριθμο αυτό. Τυπικά μεγέθη για batch size είναι μεταξύ του 50 και του 256. Θεωρώντας σταθερό αριθμό εποχών *epochs* παρακάτω δίνεται ψευδοκώδικας που προσομοιώνει τη λειτουργία του αλγορίθμου Mini-Batch Gradient Descent.

Algorithm 4 Mini-Batch Gradient Descent

```

1: procedure MINI-BATCH GRADIENT DESCENT
2:   for i in range(epochs) do
3:     data = shuffle(data)
4:     for batch in get_batches(data, batch_size) do
5:       params_grad = evaluate_gradient(obj_function , batch , params )
6:       params = params - learning_rate * params_grad

```

Παρόλα αυτά υπάρχουν ορισμένα ζητήματα τα οποία πρέπει να αναφερθούν. Για παράδειγμα, το να επιλέξεις ρυθμό μάθησης είναι δύσκολο καθώς μικρός ρυθμός μάθησης οδηγεί σε αργή σύγκλιση ενώ υψηλός ρυθμός αυξάνει την διακύμανση και την ταλάντωση γύρω από το ελάχιστο. Επίσης, είναι δύσκολο να προγραμματίσεις εκ των προτέρων πως θα πρέπει να αλλάξει κατά τη διάρκεια εκπαίδευσης ο ρυθμός μάθησης καθώς κάθε φορά θα πρέπει ο ρυθμός μάθησης να προσαρμόζεται στο εκάστοτε σύνολο δεδομένων εκπαίδευσης. Επίσης, επειδή ο ίδιος ρυθμός μάθησης εφαρμόζεται σε όλες τις παραμέτρους μπορεί πολλές φορές τα χαρακτηριστικά του συνόλου δεδομένων εκπαίδευσης να μην εμφανίζονται στην ίδια συχνότητα και για το λόγο αυτό να μη θέλουμε να τα ενημερώσουμε όλα στον ίδιο βαθμό. Τέλος, δεν πρέπει να ξεχνάμε πως σε μία συνάρτηση μη κυρτή είναι πολύ πιθανό το νευρωνικό μας δίκτυο να παγιδευτεί σε υπο-βέλτιστα τοπικά ελάχιστα, ιδιαίτερα μάλιστα σε σημεία σέλας (saddle points).

Για τους παραπάνω λόγους, υπάρχουν διάφορες τροποποιήσεις που αντιμετωπίζουν κάποια από τα προβλήματα αυτά. Στο πλαίσιο της παρούσας διπλωματικής εργασίας χρησιμοποιούμε τη βελτιστοποίηση Adaptive Moment Estimation για την εκπαίδευση του νευρωνικού μας δικτύου.

4.6.4 Adaptive Moment Estimation (Adam)

Η μέθοδος Adaptive Moment Estimation (Adam) υπολογίζει προσαρμοστικούς ρυθμούς μάθησης για κάθε μία από τις παραμέτρους του νευρωνικού μας δικτύου. Η μέθοδος διατηρεί έναν εκθετικά μειούμενο μέσο όρο προηγούμενων κλίσεων. Ορίζουμε ως $g_{t,i}$ την κλίση της αντικειμενικής συνάρτησης ως προς την παράμετρο θ_i τη χρονική στιγμή t .

$$g_{t,i} = \nabla_{\theta_t} \mathbf{J}(\theta_{t,i})$$

Ορίζουμε ως g_t το διάνυσμα που αποτελείται από όλα τα $g_{t,i}$. Επιπλέον, ορίζουμε ως m_t και v_t τις εκτιμήσεις της πρώτης ροπής (μέση τιμή) και της δεύτερης ροπής (διακύμανσης) των κλίσεων. Πρόκειται για διανύσματα που διατηρούν πληροφορία για κάθε χαρακτηριστικό και είναι αρχικοποιημένα στο 0. Τυπικά περιγράφονται ως:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \text{ και} \\ v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

όπου β_1 και β_2 θετικές σταθερές. Οι δημιουργοί της μεθόδου παρατήρησαν ότι οι παραπάνω ποσότητες δεν είναι αμερόληπτοι εκτιμητές και για το λόγο τις τροποποίησαν όπως φαίνεται παρακάτω.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \text{ και} \\ \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Ο κανόνας εκπαίδευσης περιγράφεται τυπικά ως:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

Οι δημιουργοί της μεθόδου προτείνουν τις τιμές $\beta_1 = 0.9$, $\beta_2 = 0.999$ και $\epsilon = 10^{-8}$ καθώς εμπειρικές μέθοδοι έχουν δείξει πως οι τιμές αυτές δίνουν καλά αποτελέσματα στην πράξη.

Με την παραπάνω πειραφή κλείνουμε την ενότητα και συνεχίζουμε περιγράφοντας το μοντέλο του νευρωνικού δικτύου που θα χρησιμοποιήσουμε για την επίλυση του προβλήματός μας.

4.7 Νευρωνικό Δίκτυο για Δημοπρασίες

Στην ενότητα αυτή θα περιγράψουμε το νευρωνικό δίκτυο που χρησιμοποιήσαμε σε αυτή τη διπλωματική εργασία για την εκτίμηση των virtual valuations των στρατηγικών παιχτών με σκοπό πάντα τη μεγιστοποίηση των κερδών για των πωλητή εικονικών μηχανών.

4.7.1 Η Δομή του Δικτύου

Θα περιγραφεί η δομή του δικτύου [8] για δημοπρασία ενός αντικειμένου, ενώ στο επόμενο κεφάλαιο όπου παρουσιάζονται αναλυτικά όλες οι προσεγγίσεις μας για την επίλυση του προβλήματος θα περιγραφεί η χρήση του για την επίλυση του προβλήματος της δημοπρασίας Knapsack.

Έστω το γνωστό πρόβλημα δημοπρασιών όπου ένας πωλητής διαθέτει ένα αντικείμενο προς πώληση σε n στρατηγικούς παίχτες. Κάθε στρατηγικός παίχτης έχει προσωπική αξία $v_i \in \mathbb{R}_{\geq 0}$. Θεωρούμε τυχαίο μηχανισμό (\mathbf{g}, \mathbf{p}) ο οποίος αντιστοιχεί το διάνυσμα προσφορών \mathbf{b} σε ένα διάνυσμα πιθανοτήτων $\mathbf{g}(\mathbf{b}) \in \mathbb{R}_{\geq 0}^n$, όπου το τυχαίο $g_i(\mathbf{b}) \in \mathbb{R}_{\geq 0}$ εκφράζει την πιθανότητα ο στρατηγικός παίχτης i να λάβει το αντικείμενο. Προφανώς, ισχύει πως $\sum_{i=1}^n g_i(\mathbf{b}) = 1$. Ο κανόνας πληρωμής για τον στρατηγικό παίχτη i ορίζεται ως $p_i(\mathbf{b}) = g_i(\mathbf{b})t_i(\mathbf{b})$ όπου $t(\mathbf{b})$ συνάρτηση πληρωμής δεδομένου νικητή η οποία ορίζεται ως: $t_i : \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}_{\geq 0}$. Τα αναμενόμενα έσοδα αυτού του μηχανισμού είναι $rev(\mathbf{g}, \mathbf{p}) = \mathbb{E}_{\mathbf{v} \sim \mathbf{F}}[\sum_{i=1}^n g_i(\mathbf{v})t_i(\mathbf{v})]$, όπου $\mathbf{F} = F_1 \times F_2 \times \dots \times F_n$ με κατανομές προσωπικής αξίας για τους επιμέρους στρατηγικούς παίχτες ίσες με F_1, F_2, \dots, F_n που πραγματοποιούν φιλαλήθεις προσφορές.

Θεωρώντας πως οι στρατηγικοί παίχτες διαθέτουν κανονικές κατανομές προσωπικής αξίας, γνωρίζουμε πως λαμβάνουμε μέγιστα έσοδα με τη βοήθεια του υπολογισμού των virtual valuations όπως αναφέρθηκε και στην ενότητα 3.7.5. Στόχος μας είναι να κατασκευάσουμε ένα νευρωνικό δίκτυο που να προσεγγίζει όσο καλύτερα γίνεται τα virtual valuations των αγνώστων σε εμάς κατανομών προσωπικών αξιών των στρατηγικών παιχτών.

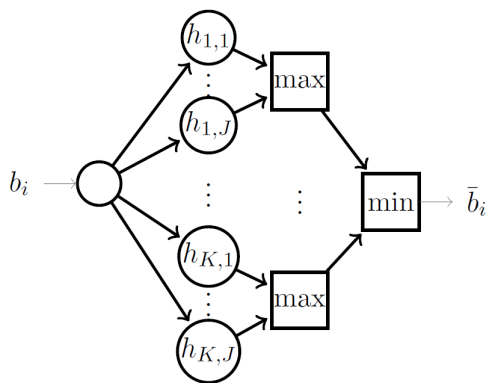
Για την επίλυση του πιο πάνω προβλήματος μοντελοποιούμε κάθε άγνωστη συνάρτηση virtual valuation ως ένα νευρωνικό δίκτυο δύο επιπέδων με \min και \max πράξεις πάνω σε γραμμικές συναρτήσεις. Θεωρούμε K ομάδες που αποτελούνται από J γραμμικές συναρτήσεις με κλίσεις $w_{kj}^i \in \mathbb{R}_{>0}$ με $k = 1, \dots, K$ και $j = 1, \dots, J$ και σημεία τομής με τον άξονα των y ίσα με β_{kj}^i και ορίζουμε την εκτίμηση για το virtual valuation του στρατηγικού παίχτη i ως εξής:

$$\bar{\phi}_i(b_i) = \min_{k \in [K]} \max_{j \in [J]} w_{kj}^i b_i + \beta_{kj}^i \quad (4.4)$$

Από ότι βλέπουμε η $\bar{\phi}_i(b_i)$ είναι μία μη φθίνουσα συνάρτηση. Αποδεικνύεται πως μπορούμε να θέσουμε $w_{kj}^i = e^{a_{kj}^i}$ όπου a_{kj}^i ανήκει σε φραγμένο διάστημα της μορφής $[-B, B]$. Για αρκετά μεγάλα K και J το συγκεκριμένο νευρωνικό δίκτυο έχει δείξει [24] πως προσεγγίζει οποιαδήποτε συνεχή και φραγμένη μονότονη συνάρτηση σε αυθαίρετα μικρό βαθμό ακρίβειας. Με τη βοήθεια της πιο πάνω έκφρασης μπορούμε εύκολα κιόλας να βρούμε και το αντίστροφο του virtual valuation το οποίο θα είναι:

$$\bar{\phi}_i^{-1}(b_i) = \max_{k \in [K]} \min_{j \in [J]} e^{-a_{kj}^i} (y - \beta_{kj}^i) \quad (4.5)$$

Το παραπάνω νευρωνικό δίκτυο απεικονίζεται σχηματικά παρακάτω:

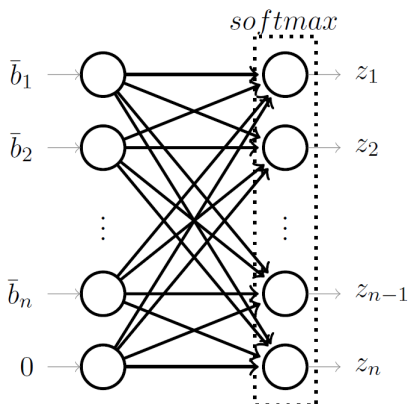


Εικόνα 4.7: Νευρωνικό δίκτυο που εκτιμά και προσεγγίζει τη συνάρτηση virtual valuation $\bar{\phi}_i(b_i)$ για τον στρατηγικό παίχτη i . Πηγή: [8]

Στη συνέχεια, για τη δημοπρασία μας χρειαζόμαστε επιπλέον να δημιουργήσουμε ένα εικονικό στρατηγικό παίχτη που θα διαθέτει virtual valuation ίσο με 0, έτσι ώστε όταν όλα τα virtual valuation είναι μικρότερα του μηδενός να μην κερδίζει κανείς στη δημοπρασία. Ο πιθανοτικός κανόνας κατανομής \mathbf{g} υλοποιείται με τη βοήθεια της συνάρτησης softmax πάνω στα virtual valuations $\bar{b}_1, \dots, \bar{b}_n$ και στη μηδενική τιμή $\bar{b}_{n+1} = 0$. Η συνάρτηση softmax πάνω στα virtual valuations για τον στρατηγικό παίχτη i δίνεται πάνω τη σχέση:

$$g_i(\bar{\mathbf{b}}) = \frac{e^{\kappa \bar{b}_i}}{\sum_{j=1}^{n+1} e^{\kappa \bar{b}_j}}$$

όπου $i \in 1, \dots, n$ και $\kappa > 0$ μία σταθερά επιλεγμένη εκ των προτέρων. Όσο υψηλότερη είναι η τιμή του κ τόσο καλύτερη γίνεται η προσέγγιση αλλά τόσο λιγότερο ομαλή γίνεται η συνάρτηση κατανομής και επομένως είναι δυσκολότερο να τη βελτιστοποιήσεις. Ο κανόνας κατανομής παρουσιάζεται σχηματικά παρακάτω.

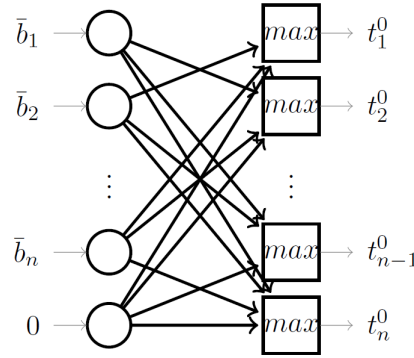


Εικόνα 4.8: Υλοποίηση πιθανοτικού κανόνα κατανομής. Πηγή: [8]

Τέλος για λόγους πληρότητας, αν και δε θα χρησιμοποιηθεί, αναφέρουμε τον κανόνα πληρωμής για τη second price δημοπρασία ενός αντικειμένου. Η τιμή p_i δίνεται από τη σχέση $p_i = \bar{\phi}_i^{-1}(t_i(\bar{\mathbf{b}}))$, όπου

$$t_i(\bar{\mathbf{b}}) = \max\{\max_{j \neq i} \bar{b}_j, 0\} \tag{4.6}$$

όπου $i \in 1, \dots, n$. Παρακάτω παρουσιάζεται σχηματικά η διαδικασία κατασκευής του κανόνα πληρωμής t .

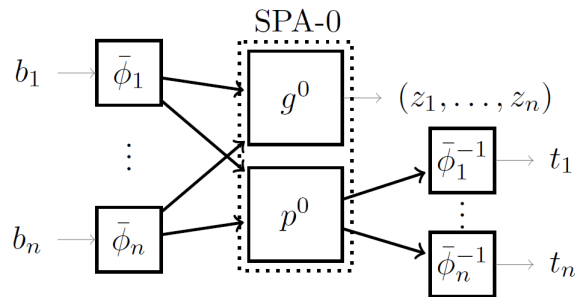


Εικόνα 4.9: Υλοποίηση κανόνα πληρωμής t . Πηγή: [8]

Συνεπώς, αν ορίσουμε ως $g^{\alpha, \beta}$ και $t^{\alpha, \beta}$ τον κανόνα κατανομής και τον κανόνα πληρωμής αντίστοιχα δεδομένων των παραμέτρων α, β και δοθέντος ενός συνόλου δεδομένων προσφορών $S = \{v^{(1)}, \dots, v^{(L)}\}$ που ακολουθούν την $\mathbf{F} = F_1 \times F_2 \times \dots \times F_n$ με κατανομές προσωπικής αξίας για τους επιμέρους στρατηγικούς παίχτες ίσες με F_1, F_2, \dots, F_n βελτιστοποιούμε τις παραμέτρους α, β θέλοντας να βρούμε το ελάχιστο της παρακάτω συνάρτησης (με αρνητικό πρόσημο).

$$\widehat{rev}(\mathbf{g}, \mathbf{p}) = \frac{1}{L} \sum_{l=1}^L \sum_{i=1}^m g_i^{\alpha, \beta}(v^{(l)}) t_i^{\alpha, \beta}(v^{(l)})$$

Κλείνουμε την ενότητα και το κεφάλαιο παρουσιάζοντας σχηματικά το συνολικό δίκτυο προς εκπαίδευση. Η εκπαίδευση πραγματοποιείται με τη βοήθεια mini-batch stochastic gradient descent χρησιμοποιώντας τη μέθοδο Adaptive Moment Estimation (Adam).



Εικόνα 4.10: Συνολικό νευρωνικό δίκτυο. Πηγή: [8]

όπου SPA-0 η μοντελοποίηση της διαδικασίας second price auction (S.P.A.) με τον εικονικό χρήστη που θέτει προσφορά ίση με μηδέν (SPA-0).

Κεφάλαιο 5

Προσεγγίσεις για την Επίλυση του Προβλήματος

Στο κεφάλαιο αυτό παρουσιάζονται οι προσεγγίσεις μας για την επίλυση του προβλήματος για την κατανομή των εικονικών μηχανών στους χρήστες με απώτερο στόχο τη μεγιστοποίηση των εσόδων για τον πωλητή/πάροχο. Παρουσιάζονται τέσσερις διαφορετικές προσεγγίσεις. Η πρώτη προσέγγιση έχει ως στόχο τη μεγιστοποίηση της κοινωνικής ωφέλειας για τους χρήστες και για το λόγο αυτό δεν είναι εφαρμόσιμη στην πραγματικότητα αλλά γίνεται μόνο για λόγους πληρότητας της εργασίας. Στις επόμενες τρεις διαφορετικές προσεγγίσεις έχουμε ως στόχο τη μεγιστοποίηση των εσόδων για τον πωλητή/πάροχο.

Στην ενότητα [5.1](#) παρουσιάζεται η μέθοδος που χρησιμοποιείται για τη μεγιστοποίηση της κοινωνικής ωφέλειας για τους χρήστες σε δημοπρασίες τύπου Knapsack. Η μέθοδος αυτή για το πρόβλημα του limited supply βασίζεται σε ένα προσεγγιστικό αλγόριθμο ο οποίος μας δίνει θεωρητικά κάτω φράγματα για την απόδοση του, ενώ η μέθοδος για το πρόβλημα του unlimited supply βασίζεται στη μέθοδο του reserve pricing.

Στην ενότητα [5.2](#) παρουσιάζεται η μέθοδος που χρησιμοποιείται για τη μεγιστοποίηση των εσόδων για τον πωλητή σε δημοπρασίες τύπου Knapsack. Η μέθοδος αυτή για το πρόβλημα του limited supply βασίζεται σε ένα προσεγγιστικό αλγόριθμο ο οποίος μας δίνει θεωρητικά κάτω φράγματα για την απόδοση του, ενώ η μέθοδος για το πρόβλημα του unlimited supply βασίζεται σε κυρτό συνδυασμό δύο τυχαίων δημοπρασιών που επίσης μας δίνουν θεωρητικά κάτω φράγματα για την απόδοσή τους.

Στην ενότητα [5.2](#) παρουσιάζεται άλλη μία μέθοδος που χρησιμοποιείται για τη μεγιστοποίηση των εσόδων για τον πωλητή σε δημοπρασίες τύπου Knapsack. Η μέθοδος αυτή για το πρόβλημα του limited supply βασίζεται σε ένα προσεγγιστικό αλγόριθμο ο οποίος χρησιμοποιεί εκτιμήσεις πάνω στα virtual valuations των χρηστών. Οι εκτιμήσεις αυτές πραγματοποιούνται πάνω σε προηγούμενα δεδομένα και υλοποιούνται με τη βοήθεια των τεχνικών εκτίμησης κατανομών προσωπικής αξίας που είδαμε στο κεφάλαιο [4](#). Η μέθοδος για το πρόβλημα του unlimited supply βασίζεται στη μέθοδο του reserve pricing που καθορίζεται και πάλι με τη βοήθεια των τεχνικών εκτίμησης κατανομών προσωπικής αξίας.

Στην ενότητα [5.4](#) παρουσιάζεται η τελευταία "υβριδική" μέθοδος που χρησιμοποιείται για τη μεγιστοποίηση των εσόδων για τον πωλητή σε δημοπρασίες τύπου Knapsack. Η μέθοδος αυτή για το πρόβλημα του limited supply βασίζεται στο νευρωνικό δίκτυο που παρουσιάσαμε στην ενότητα [4.7](#) το οποίο προσεγγίζει τα virtual valuations των χρηστών. Για το πρόβλημα του unlimited supply βασίζεται σε κυρτό συνδυασμό των δύο τυχαίων δημοπρασιών που προαναφέρθηκαν.

5.1 Μεγιστοποίηση Κοινωνικής Ωφέλειας

Στην πρώτη μας προσέγγιση για την επίλυση του προβλήματος της βέλτιστης κατανομής των εικονικών μηχανών σε χρήστες θα προσπαθήσουμε να μεγιστοποιήσουμε την κοινωνική ωφέλεια. Δεν ξεκινάμε άμεσα με προσεγγίσεις που έχουν ως στόχο τη μεγιστοποίηση των κερδών προκειμένου να παρουσιάσουμε μία πλήρη εικόνα του προβλήματος. Τονίζεται πως το μοντέλο αυτό είναι μη-ρεαλιστικό καθώς δεν υπάρχει κάποιος προφανής λόγος την πρακτική εφαρμογή του. Παρουσιάζουμε στη συνέχεια τους αλγόριθμους για την επίλυση του προβλήματος.

5.1.1 Αλγόριθμοι Μεγιστοποίησης Κοινωνικής Ωφέλειας

Το πρόβλημα της βέλτιστης κατανομής των εικονικών μηχανών σε χρήστες μπορεί να μοντελοποιηθεί ως μία δημοπρασία Knapsack, όπως παρουσιάστηκε στην ενότητα 3.6. Σε κάθε γύρο, ο οποίος αντιστοιχεί σε ένα ορισμένο χρονικό διάστημα (π.χ. μία ώρα), οι χρήστες ζητούν ένα συγκεκριμένο πλήθος εικονικών μηχανών από τον πωλητή. Αν το πλήθος των ζητούμενων εικονικών μηχανών είναι μεγαλύτερο από τις συνολικές διαθέσιμες εικονικές μηχανές του πωλητή, τότε έχουμε να λύσουμε μία δημοπρασία Knapsack περιορισμένης προμήθειας (limited supply). Σε αντίθετη περίπτωση, έχουμε να λύσουμε μία δημοπρασία άπειρης προμήθειας (infinite supply). Στη συγκεκριμένη περίπτωση, θέλουμε να λύσουμε το πρόβλημα με στόχο τη μεγιστοποίηση της κοινωνικής ωφέλειας. Παρακάτω δίνονται οι προτάσεις μας για τις δύο αυτές περιπτώσεις.

Εξετάζουμε την περίπτωση limited supply. Έστω \mathbf{b} το διάνυσμα προσφορών σύμφωνα με το οποίο ο μηχανισμός μας επιλέγει ένα εφικτό σύνολο νικητών S , έτσι ώστε το συνολικό για το συνολικό μέγεθος να ισχύει πως $\sum_{i \in S} w_i \leq W$, όπου w_i η αίτηση σε πόρους του κάθε χρήστη i και W το σύνολο των διαθέσιμων εικονικών μηχανών. Παρακάτω παρουσιάζουμε τα βήματα ενός άπληστου αλγόριθμου για την επίλυση του προβλήματος.

Μεθοδολογία 5: Άπληστος αλγόριθμος για limited supply πρόβλημα

1. Ταξινόμησε τους χρήστες σε φθίνουσα σειρά έτσι ώστε να ισχύει:

$$\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \dots \geq \frac{b_n}{w_n}$$

2. Ξεκίνα να επιλέγεις χρήστες με βάση την παραπάνω σειρά έως ότου δε μπορείς να εξυπηρετήσεις άλλους.
3. Επέστρεψε ως λύση ή την παραπάνω λύση ή τον χρήστη με την υψηλότερη προσφορά ανάλογα με το ποιά λύση επιτυγχάνει υψηλότερη κοινωνική ωφέλεια.

Ο παραπάνω κανόνας κατανομής είναι μονότονος και επομένως αν χρησιμοποιήσουμε το Λήμμα του Myerson, δημιουργούμε ένα μηχανισμό DSIC ο οποίος τρέχει σε πολυωνυμικό χρόνο. Επιπλέον, ο παραπάνω αλγόριθμος αποτελεί $\frac{1}{2}$ -approximation, δηλαδή σε κάθε περίπτωση επιστρέφει εφικτή λύση που επιτυγχάνει τουλάχιστον κατά 50% τη βέλτιστη δυνατή κοινωνική ωφέλεια. Μάλιστα, αν για κάθε i ισχύει πως $w_i \leq aW$, με $a \in (0, \frac{1}{2}]$, τότε ο παραπάνω αλγόριθμος αποτελεί $1 - a$ approximation ακόμα και αν το τρίτο βήμα του αλγόριθμου παραλειφθεί. Συνεπώς, υπάρχουν περιπτώσεις (οι οποίες είναι και οι πιο συνήθεις) στις οποίες αυτός ο απλός προσεγγιστικός άπληστος αλγόριθμος πετυχαίνει ακόμη καλύτερα αποτελέσματα. Στο σημείο αυτό, καλό είναι να τονίσουμε πως δεν είναι όλοι οι "λογικοί" κανόνες κατανομής μονότονοι. Για παράδειγμα, για κάθε $\epsilon > 0$ υπάρχει προσεγγιστικός αλγόριθμος $1 - \epsilon$ που επιλύει το πρόβλημα Knapsack σε πολυωνυμικό χρόνο. Παρόλα αυτά, ο κανόνας κατανομής δεν είναι μονότονος. Επομένως, για προβλήματα NP-hard πρέπει πρώτα να ελέγχουμε αν οι καλύτεροι προσεγγιστικοί αλγόριθμοι οδηγούν σε DSIC μηχανισμούς. Αν όχι, θα πρέπει να καταφύγουμε σε άλλους προσεγγιστικούς αλγόριθμους.

Στην περίπτωση του unlimited supply θεωρούμε ένα απλό μοντέλο με *reserve_price* = 0 προκειμένου να μεγιστοποιήσουμε την κοινωνική ωφέλεια. Πρόκειται για ένα μοντέλο μη-ρεαλιστικό το οποίο μελετάται για λόγους πληρότητας. Στο επόμενο κεφάλαιο θα προχωρήσουμε στην πειραματική αξιολόγηση του μοντέλου.

5.2 Μεγιστοποίηση Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών

Στην ενότητα αυτή θα παρουσιάσουμε μία προσέγγιση για την επίλυση του προβλήματος της βέλτιστης κατανομής των εικονικών μηχανών σε χρήστες, με στόχο τη μεγιστοποίηση των κερδών για τον πωλητή/πάροχο. Η προσέγγισή μας βασίζεται στη χρήση προσεγγιστικού αλγορίθμου για την επίλυση του προβλήματος του limited supply και τυχαίων δημοπρασιών για την επίλυση του προβλήματος του unlimited supply. Παρουσιάζουμε στη συνέχεια τους αλγορίθμους για την επίλυση του προβλήματος.

5.2.1 Προσεγγιστικός Αλγόριθμος για Limited Supply

Μοντελοποιούμε το πρόβλημα όπως στην προηγούμενη ενότητα, δηλαδή η βέλτιστη κατανομή των εικονικών μηχανών σε χρήστες μπορεί να μοντελοποιηθεί ως μία δημοπρασία Knapsack. Παρακάτω παρουσιάζουμε έναν προσεγγιστικό αλγόριθμο - χρησιμοποιώντας το γνωστό συμβολισμό - για την επίλυση του προβλήματος με σκοπό τη μεγιστοποίηση των εσόδων για τον πωλητή και στη συνέχεια δίνουμε ένα κάτω φράγμα για να περιγράψουμε την απόδοσή του.

Μεθοδολογία 6: Προσεγγιστικός αλγόριθμος για limited supply πρόβλημα

1. Αγνοούμε τις προσφορές για τις οποίες ισχύει $w_i > \frac{W}{2}$.
2. Ταξινομήσε τους χρήστες σε φθίνουσα σειρά έτσι ώστε να ισχύει:

$$\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \dots \geq \frac{b_n}{w_n}$$

3. Ξεκίνα να επιλέγεις χρήστες με βάση την παραπάνω σειρά έως ότου δε μπορείς να εξυπηρετήσεις άλλους.
4. Έστω d^* η μεγαλύτερη ανά μονάδα τιμή η οποία δεν επιλέχθηκε. Η τιμή που θα πληρώσει ο χρήστης i θα είναι ίση με $p_i(\mathbf{b}) = d^* x_i(\mathbf{b}) w_i$.

Αποδεικνύεται [2] η παρακάτω πρόταση σχετικά με τον συγκεκριμένο προσεγγιστικό αλγόριθμο.

Πρόταση 7: Απόδοση Προσεγγιστικού Αλγορίθμου

Θεωρώντας N στο σύνολο προτάσεις, ο παραπάνω αλγόριθμος επιλέγει εφικτό σύνολο H τέτοιο ώστε να ισχύει $OPT(H) \geq \frac{OPT(N)}{3} - h$, όπου OPT τα έσοδα που προκύπτουν από τη βέλτιστη συνάρτηση πληρωμής και h άνω φράγμα στην υψηλότερη προσωπική αξία χρήστη.

Αποδεικνύουμε την πρόταση για 0-1 πρόβλημα Knapsack. Για κλασματικό πρόβλημα λαμβάνουμε ακόμα καλύτερα αποτελέσματα.

ΑΠΟΔΕΙΞΗ: Έστω $N' \subset N$ στο οποίο περιέχονται τα $w_i \leq \frac{W}{2}$. Το πολύ μόνο ένα αντικείμενο από το σύνολο $N \setminus N'$ μπορεί να "χωρέσει" στο knapsack. Συνεπώς, ασχολούμαστε με το σύνολο N' χωρίς να χάσουμε πάνω από ένα έξτρα όρο h .

Αν όλα τα αντικείμενα του N' "χωράνε" στο knapsack, τότε η πρόταση αποδεικνύεται όπως θα φανεί και παρακάτω. Αν όχι, γνωρίζουμε πως τα αντικείμενα του συνόλου H καταλαμβάνουν χώρο τουλάχιστον ίσο με $\frac{W}{2}$. Αυτό προκύπτει από το ότι υπάρχει αντικείμενο στο N' το οποίο δε χωράει στο υπόλοιπο knapsack και από το ότι κάθε αντικείμενο στο N' έχει μέγεθος το πολύ $\frac{W}{2}$. Επομένως, ο κανόνας πληρωμής πετυχαίνει κέρδος τουλάχιστον $d^* \frac{W}{2}$ καθώς η ανά μονάδα προσωπική αξία ανά χρήστη είναι τουλάχιστον d^* . Επομένως $OPT(H) \geq d^* \frac{W}{2}$.

Έστω $L = N' \setminus H$ το σύνολο των αντικειμένων που δε "χώρεσαν" στο knapsack, τα οποία έχουν το πολύ ανά μονάδα προσωπική αξία ανά χρήστη ίση με d^* . Προφανώς, ισχύει $OPT(L) \leq d^*W \leq 2OPT(H) \implies 3OPT(H) \geq OPT(H) + OPT(L) \geq OPT(N')$. Επομένως, αποδεικνύεται πως: $OPT(H) \geq \frac{OPT(N)}{3} - h$. ■

Βλέπουμε, πως ο συγκεκριμένος προσεγγιστικός αλγόριθμος μας δίνει μία πολύ καλή λύση που προσεγγίζει τα μέγιστα δυνατά έσοδα για το συγκεκριμένο πρόβλημα. Προχωράμε ορίζοντας τη λύση για το πρόβλημα Unlimited Supply.

5.2.2 Τυχαίες Δημοπρασίες για Unlimited Supply

Θεωρούμε το πρόβλημα knapsack unlimited supply για το οποίο ισχύει πως $W = \infty$. Προκειμένου να λύσουμε το πρόβλημα λαμβάνοντας καλές θεωρητικές εγγυήσεις θα επιλέξουμε να υλοποιήσουμε ένα κυρτό συνδυασμό δύο ξεχωριστών λύσεων, ο οποίος θα διατηρεί τη βασική ιδιότητα της φιλαλήθειας. Πριν περάσουμε στη περιγραφή των αλγορίθμων θα κάνουμε μία θεωρητική εισαγωγή στα εκθετικά διαστήματα.

Ορισμός 31: Κανόνας Πληρωμής Εκθετικών Διαστημάτων

Ένας μονότονος κανόνας πληρωμής με εκθετικά διαστήματα είναι ένας κανόνας πληρωμής στον οποίο οι νικητές μπορούν να χωριστούν σε διαστήματα ίσων τιμών σύμφωνα με τις προσφορές έτσι ώστε το i -οστό διάστημα (σε φθίνουσα σειρά σύμφωνα με τις προσφορές) να περιέχει τουλάχιστον 2^{i-1} νικητές.

Πρόταση 8: Ισοδυναμία Κανόνων Πληρωμής Εκθετικών Διαστημάτων

Δοθέντος ενός οποιουδήποτε κανόνα πληρωμών $p(\cdot)$, ο οποίος προσφέρει συνολικά έσοδα ίσα με P πάνω σε ένα σύνολο προσφορών $(v_1, \dots, v_n; w_1, \dots, w_n; W = \infty)$, υπάρχει μονότονος κανόνας πληρωμής με εκθετικά διαστήματα $p'(\cdot)$ με έσοδα τουλάχιστον $\frac{P}{2}$.

ΑΠΟΔΕΙΞΗ: Ταξινομούμε τους νικητές του $p(\cdot)$ σε φθίνουσα σειρά και χωρίζουμε τις προσφορές σε διαστήματα τέτοια ώστε το i -οστό διάστημα να περιέχει τουλάχιστον 2^{i-1} νικητές αλλά αυστηρά λιγότερους από 2^{i-1} νικητές με μέγεθος αυστηρά μεγαλύτερο από το μικρότερο νικητή στο i . Έστω w_i το μικρότερο αντικείμενο στο διάστημα i . Ορίζουμε ως $p'(\cdot)$ τον κανόνα πληρωμής τέτοιο ώστε όλα τα αντικείμενα στο διάστημα i να προσφέρονται στην τιμή $p(w_i)$.

Θα δείξουμε ότι τα έσοδα του $p(\cdot)$ δεν είναι παραπάνω από τα διπλάσια του $p'(\cdot)$. Για τυχαίο διάστημα i η απώλεια είναι αυτή στα έσοδα μεταξύ του $p(\cdot)$ και του $p'(\cdot)$ στο διάστημα $[w_i, w_{-1})$. Δεν υπάρχει κάποια απώλεια από αντικείμενο με μέγεθος w_i και η υπόλοιπη απώλεια από τα υπόλοιπα αντικείμενα στο διάστημα i φράσσεται από την ποσότητα $p(w_{i-1}) - p(w_i)$. Αφού το διάστημα i περιέχει λιγότερα από 2^{i-1} αντικείμενα με μέγεθος ακριβώς παραπάνω από w_i η συνολική απώλεια είναι μικρότερη ή ίση με την ποσότητα $(2^{i-1} - 1) \times (p(w_{i-1}) - p(w_i))$. "Χρεώνουμε" αυτή την απώλεια στους νικητές σε όλα τα προηγούμενα διαστήματα. Υπάρχουν τουλάχιστον $\sum_{j=1}^{i-1} 2^{j-1} = 2^{i-1} - 1$ νικητές και επομένως κάθε νικητής χρεώνεται παραπάνω το πολύ κατά $p(w_{i-1}) - p(w_i)$ έκαστος. Οι χρεώσεις σε κάθε νικητή στο διάστημα i λόγω επόμενων διαστημάτων αθροίζουν το πολύ σε $p(w_i)$ και συνεπώς η συνολική απώλεια συνοψίζεται από τα έσοδα του κανόνα $p'(\cdot)$. Επομένως, τα έσοδα του $p'(\cdot)$ είναι τουλάχιστον $\frac{P}{2}$. ■

Στη συνέχεια παρουσιάζουμε τις δύο λύσεις του κυρτού συνδυασμού που θα χρησιμοποιήσουμε.

Μεθοδολογία 7: Random Sampling Knapsack Auction

1. Χωρίζουμε τους χρήστες σε δύο σύνολα A και B ομοιόμορφα στην τύχη.
2. Υπολόγισε τον μονότονο κανόνα πληρωμής για εκθετικά διαστήματα (περιορίζοντας σε δυνάμεις του 2) για κάθε σύνολο. Έστω πως ο εκθετικό κανόνας πληρωμής για το σύνολο A είναι p_A και για το B είναι p_B .
3. Χρησιμοποίησε τον κανόνα πληρωμής p_A για το B και τον p_B για το A .

Αποδεικνύεται πως η παραπάνω λύση διατηρεί την ιδιότητα της φιλαλήθειας.

Μεθοδολογία 8: General Attribute Auction

Στη θέση των attributes θεωρούμε τις προσφορές.

1. Ταξινομήσε τους χρήστες σε φθίνουσα σειρά έτσι ώστε να ισχύει:

$$\frac{b_1}{w_1} \geq \frac{b_2}{w_2} \geq \dots \geq \frac{b_n}{w_n}$$

2. Θέτουμε τις παραμέτρους $a = 2$ και $p = \frac{1}{2}$. Έστω h άνω φράγμα στην υψηλότερη προσωπική αξία χρήστη. Ορίζουμε τη μεταβλητή experts ως $\log_a h + 1$. Εκφράζει εικονικούς ειδικούς που πραγματοποιούν την κατάλληλη προσφορά για τα VMs.
3. Για κάθε expert j , ορίζεις εικονικό σκορ $s_j = ka_j$ με πιθανότητα $(1-p)^k p$, όπου ως k μπορούμε να ορίσουμε το πλήθος των ανεξάρτητων δοκιμών Bernoulli μέχρι να έρθει το πρώτο ενδεχόμενο με πιθανότητα p .
4. Για κάθε χρήστη i , επιλέγουμε τον ειδικό j με το μεγαλύτερο σκορ μέχρι στιγμής (στις ισοπαλίες επιλέγουμε σταθερά αυθαίρετα) και προσφέρουμε στον i την τιμή a^j του ειδικού.
5. Ανανεώνουμε τα σκορ των ειδικών που θα μπορούσαν να είναι κάνει πώληση, δηλαδή ισχύει $a^j \leq b_i$, σύμφωνα με τον κανόνα $s_j \leftarrow s_j + a^j$ και επιστρέφουμε στο βήμα 4 έως ότου τελειώσει ο γύρος.

Αφού έχουμε παρουσιάσει τις δύο λύσεις κλείνουμε την ενότητα παρουσιάζοντας την τυχαία δημοπρασία που συνιστά κυρτό συνδυασμό αυτών των δύο δημοπρασιών καθώς και ένα κάτω φράγμα της απόδοσής της.

Μεθοδολογία 9: Unlimited Supply Knapsack Auction

1. Τρέξε το πρώτο βήμα του αλγορίθμου RSK.
2. Σύμφωνα με προκαθορισμένη πιθανότητα p τρέξε General Attribute Auction στα σύνολα A και B ξεχωριστά και με πιθανότητα $1-p$ υλοποίησε τα βήματα του αλγορίθμου RSK.

Αποδεικνύεται πως ο παραπάνω μηχανισμός διατηρεί την ιδιότητα της φιλαλήθειας και τα έσοδα που προκύπτουν από αυτά είναι $\alpha OPT - \gamma h(\log \log \log n_A + \log \log \log n_B + 19)$, όπου α, γ σταθερές και h άνω φράγμα στην υψηλότερη προσωπική αξία χρήστη.

5.3 Μεγιστοποίηση Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας

Στην ενότητα αυτή θα παρουσιάσουμε μία προσέγγιση για την επίλυση του προβλήματος της βέλτιστης κατανομής των εικονικών μηχανών σε χρήστες, με στόχο τη μεγιστοποίηση των κερδών για τον πωλητή/πάροχο χρησιμοποιώντας τεχνικές εκτίμησης των κατανομών προσωπικής αξίας. Η προσέγγισή μας βασίζεται σε τεχνικές για την εκτίμηση των αγνώστων παραμέτρων των κατανομών προσωπικής αξίας των χρηστών οι οποίες θα χρησιμοποιηθούν τόσο για την επίλυση του προβλήματος του limited supply όσο και του unlimited supply. Παρουσιάζουμε στη συνέχεια τους αλγόριθμους για την επίλυση του προβλήματος.

5.3.1 Αλγόριθμοι Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπική Αξίας

Εξετάζουμε αρχικά και πάλι την περίπτωση limited supply. Προκειμένου να μεγιστοποιήσουμε τα έσοδά μας θα χρησιμοποιήσουμε τον προσεγγιστικό αλγόριθμο της ενότητας 5.1 ωστόσο στη θέση των προσφορών θα χρησιμοποιήσουμε τα virtual valuations σύμφωνα με τη μεθοδολογία της ενότητας 3.7.5.

Προκειμένου να εκτιμήσουμε τα virtual valuations θα πρέπει να χρησιμοποιήσουμε παλαιότερα δεδομένα από αντίστοιχες δημοπρασίες. Σε αυτά θα χρησιμοποιήσουμε τις τεχνικές που αναλύσαμε στο Κεφάλαιο 4 εστιάζοντας στην κλασική εκτίμηση παραμέτρων ενότητα 4.1 και στον αλγόριθμο expectation-maximization ενότητα 4.3, θεωρώντας πως όλοι οι χρήστες ακολουθούν την ίδια κατανομή. Τα αποτελέσματα θα αναλυθούν και θα συγκριθούν με τα πραγματικά δεδομένα. Η μέθοδος που ακολουθείται είναι η ακόλουθη.

Μεθοδολογία 10: Αλγόριθμος Μεγιστοποίησης Εσόδων με Εκτίμηση Παραμέτρων για Limited Supply Πρόβλημα

1. Εκτίμησε το διάνυσμα παραμέτρων $\hat{\theta}$ με βάση τα υπάρχοντα δεδομένα.
2. Μετέτρεψε την προσφορά $b_i = v_i$ του κάθε χρήστη i στο αντίστοιχο virtual valuation $\hat{\varphi}_i(v_i)$.
3. Ταξινόμησε τους χρήστες σε φθίνουσα σειρά κρατώντας μόνο τις θετικές ανά μονάδα εικονικές αξίες έτσι ώστε να ισχύει:

$$\frac{\hat{\varphi}_1(v_1)}{w_1} \geq \frac{\hat{\varphi}_2(v_2)}{w_2} \geq \dots \geq \frac{\hat{\varphi}_n(v_n)}{w_n}$$

4. Ξεκίνα να επιλέγεις χρήστες με βάση την παραπάνω σειρά έως ότου δε μπορείς να εξυπηρετήσεις άλλους.
5. Επέστρεψε ως λύση ή την παραπάνω λύση ή τον χρήστη με την υψηλότερη προσφορά ανάλογα με το ποιά λύση επιτυγχάνει υψηλότερα έσοδα.

Ισχύουν παρόμοια αποτελέσματα με βάση αυτά της ενότητας 5.1. Ο κανόνας κατανομής είναι μονότονος και επομένως αν χρησιμοποιήσουμε το Λήμμα του Myerson, δημιουργούμε ένα μηχανισμό DSIC ο οποίος τρέχει σε πολυωνυμικό χρόνο. Επιπλέον, ο παραπάνω αλγόριθμος για αβεγάδιαστη εκτίμηση των virtual valuations αποτελεί $\frac{1}{2}$ -approximation, δηλαδή σε κάθε περίπτωση επιστρέφει εφικτή λύση που επιτυγχάνει τουλάχιστον κατά 50% τα βέλτιστα έσοδα. Μάλιστα, αν για κάθε i ισχύει πως $w_i \leq aW$, με $a \in (0, \frac{1}{2}]$, τότε ο παραπάνω αλγόριθμος αποτελεί $1 - a$ approximation ακόμα και αν το πέμπτο βήμα του αλγορίθμου παραλειφθεί.

Στην περίπτωση του unlimited supply θεωρούμε ένα μοντέλο με $reserve_price = \hat{\varphi}^{-1}(0)$ προκειμένου να μεγιστοποιήσουμε τα έσοδα χρησιμοποιώντας την τεχνική που εφάρμοσε η εταιρεία Yahoo το 2008 με επιτυχία προκειμένου να μεγιστοποιήσει τα έσοδα της.

5.4 Μεγιστοποίηση Εσόδων με Χρήση Νευρωνικών Δικτύων

Στην ενότητα αυτή θα παρουσιάσουμε την τελευταία μας προσέγγιση για την επίλυση του προβλήματος της βέλτιστης κατανομής των εικονικών μηχανών σε χρήστες, με στόχο τη μεγιστοποίηση των κερδών για τον πωλητή/πάροχο χρησιμοποιώντας αυτή τη φορά μία υβριδική τεχνική βασισμένη σε νευρωνικά δίκτυα και τυχαίες δημοπρασίες. Η προσέγγισή μας χρησιμοποιεί το νευρωνικό δίκτυο που παρουσιάστηκε στην ενότητα 4.7 για την επίλυση του προβλήματος του limited supply καθώς και τις τυχαίες δημοπρασίες της ενότητας 5.2 για την επίλυση του προβλήματος του unlimited supply. Παρουσιάζουμε στη συνέχεια τους αλγορίθμους για την επίλυση του προβλήματος.

5.4.1 Αλγόριθμοι Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων

Εξετάζουμε αρχικά και πάλι την περίπτωση limited supply. Προκειμένου να μεγιστοποιήσουμε τα έσοδά μας θα χρησιμοποιήσουμε τον προσεγγιστικό αλγόριθμο της ενότητας 5.1 σε συνδυασμό με τον νευρωνικό δίκτυο που παρουσιάστηκε στην ενότητα 4.7 το οποίο χρησιμοποιείται για την καλύτερη δυνατή προσέγγιση των virtual valuations των χρηστών.

Προκειμένου να εκτιμήσουμε τα virtual valuations θα πρέπει να χρησιμοποιήσουμε παλαιότερα δεδομένα από αντίστοιχες δημοπρασίες. Με βάση αυτά εκπαιδεύουμε το νευρωνικό μας δίκτυο θεωρώντας αυτή τη φορά πως όλοι οι χρήστες δεν ακολουθούν την ίδια κατανομή αλλά πως κάθε ένας ακολουθεί μία διαφορετική κατανομή που πρέπει να εκτιμηθεί μέσω των virtual valuations της. Το νευρωνικό δίκτυο της ενότητας 4.7 είναι κατασκευασμένο για ένα αντικείμενο. Λύνουμε το πρόβλημα στη βάση του προσεγγιστικού αλγορίθμου της ενότητας 5.1 σύμφωνα με τον οποίο οι νικητές επιλέγονται με βάση τη μεγαλύτερη ανά μονάδα προσφορά τους. Οι κατανομές που υπολογίζουμε είναι πάντα για την ανά μονάδα προσφορά. Η συνάρτηση κόστους που θέλουμε να ελαχιστοποιήσουμε είναι το αρνητικό των εσόδων από τη δημοπρασία ενός αντικειμένου. Προφανώς, υπάρχουν μικρές τροποποιήσεις αν μιλάμε για δημοπρασία Knapsack τύπου 0-1 ή fractional. Με τον τρόπο αυτό, όπως θα δούμε και στο επόμενο κεφάλαιο γίνονται πάρα πολύ καλές προσεγγίσεις των virtual valuations που χρησιμοποιούνται σε μελλοντικά δεδομένα για μεγιστοποίηση εσόδων σε δημοπρασίες τύπου Knapsack χρηστών διαφορετικών κατανομών. Η μέθοδος που ακολουθείται είναι η ακόλουθη.

Μεθοδολογία 11: Αλγόριθμος Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικού δικτύου

1. Εκπαίδευση Δικτύου
 - (a) Δώσε τυχαίες μικρές αρχικές τιμές για να αρχικοποιήσεις το δίκτυο.
 - (b) Μετέτρεψε την προσφορά $b_i = v_i$ του κάθε χρήστη i στο αντίστοιχο virtual valuation $\widehat{\varphi}_i(v_i)$ με τη βοήθεια του νευρωνικού δικτύου.
 - (c) Ταξινόμησε τους χρήστες σε φθίνουσα σειρά κρατώντας μόνο τις θετικές ανά μονάδα εικονικές αξίες έτσι ώστε να ισχύει:

$$\frac{\widehat{\varphi}_1(v_1)}{w_1} \geq \frac{\widehat{\varphi}_2(v_2)}{w_2} \geq \dots \geq \frac{\widehat{\varphi}_n(v_n)}{w_n}$$
 - (d) Ξεκίνα να επιλέγεις χρήστες με βάση την παραπάνω σειρά έως ότου δε μπορείς να εξυπηρετήσεις άλλους.
 - (e) Επέστρεψε τη λύση, δώσε τιμές στους χρήστες, σύγκρινε την με τη σωστή λύση που μεγιστοποιεί τα αναμενόμενα έσοδα και προσαρμόσε αντίστοιχα τα βάρη.
2. Στη συνέχεια επέστρεψε ως λύση το κατάλληλο διάνυσμα βαρών για να προσεγγίσεις τα virtual valuations των χρηστών.
3. Τρέξε τον προσεγγιστικό αλγόριθμο της ενότητας 5.2.

Ο κανόνας κατανομής είναι μονότονος και επομένως αν χρησιμοποιήσουμε το Λήμμα του Myerson, δημιουργούμε ένα μηχανισμό DSIC ο οποίος τρέχει σε πολυωνυμικό χρόνο.

Στην περίπτωση του unlimited supply δικαιολογούμε για ποιο λόγο ονομάσαμε το μοντέλο μας υβριδικού. Στην περίπτωση αυτή, χρησιμοποιούμε τη μεθοδολογία τυχαίων δημοπρασιών της ενότητας 5.2 καθώς όπως θα δούμε και στη συνέχεια μας δίνει βέλτιστα αποτελέσματα σε συνδυασμό με την προσέγγιση του νευρωνικού δικτύου.

Κεφάλαιο 6

Πειραματική Αξιολόγηση

Στο κεφάλαιο αυτό παρουσιάζονται τα πειραματικά αποτελέσματα των προσεγγίσεων που παρουσιάστηκαν προηγουμένως. Αρχικά, γίνεται μία περιγραφή του περιβάλλοντος προσομοίωσης εστιάζοντας στις παραμέτρους του, στις κατανομές προσφορών και αιτημάτων χρηστών, στο σύνολο πραγματικών δεδομένων και στις μετρικές αξιολόγησης. Έπειτα παρουσιάζονται τα αποτελέσματα από τις προσομοιώσεις μας για τις τέσσερις προσεγγίσεις μας τόσο για το binary (0-1) όσο και για το fractional πρόβλημα.

Στην ενότητα 6.1 παρουσιάζεται το περιβάλλον προσομοίωσης. Περιγράφονται οι παράμετροι που θα χρησιμοποιηθούν σε κάθε περίπτωση, οι κατανομές προσφορές και αιτημάτων χρηστών που επιλέχθηκαν, το σύνολο πραγματικών δεδομένων Google Trace και οι επιμέρους μετρικές αξιολόγησης

Στην ενότητα 6.2 παρουσιάζονται τα πειραματικά αποτελέσματα για την προσέγγιση μεγιστοποίησης κοινωνικής ωφέλειας.

Στην ενότητα 6.3 παρουσιάζονται τα πειραματικά αποτελέσματα για την προσέγγιση μεγιστοποίησης εσόδων με χρήση προσεγγιστικού αλγορίθμου και τυχαίων δημοπρασιών.

Στην ενότητα 6.4 παρουσιάζονται τα πειραματικά αποτελέσματα για την προσέγγιση μεγιστοποίησης εσόδων εκτιμώντας τις κατανομές προσωπική αξίας των χρηστών.

Στην ενότητα 6.5 παρουσιάζονται τα πειραματικά αποτελέσματα για την προσέγγιση μεγιστοποίησης εσόδων με χρήση της υβριδικής τεχνικής που περιλαμβάνει νευρωνικά δίκτυα και τυχαίες δημοπρασίες.

6.1 Περιβάλλον Προσομοίωσης

Στην ενότητα αυτή θα περιγράψουμε τις παραμέτρους του περιβάλλοντος στο οποίο βασίστηκαν οι προσομοιώσεις μας. Θα αναφερθούμε στις βασικές παραμέτρους του συστήματος, στην κατανομή που ακολουθούν οι προσφορές των χρηστών καθώς και στην κατανομή που ακολουθούν τα αιτήματα τους για εικονικές μηχανές, ενώ φυσικά θα αναφερθούμε στο σύνολο δεδομένων πάνω στο οποίο πραγματοποιείται η προσομοίωση καθώς και στις μετρικές στις οποίες θα λάβουμε για την αξιολόγηση των αποτελεσμάτων. Συνεχίζουμε με την περιγραφή των παραμέτρων.

6.1.1 Παραμετροποίηση

Πραγματοποιήσαμε διαφορετικές μεταξύ τους προσομοιώσεις προκειμένου να μελετήσουμε τη συμπεριφορά του μηχανισμού τόσο σε περιπτώσεις limited supply όσο και σε περιπτώσεις unlimited supply καθώς και σε συνδυασμό αυτών. Για το λόγο αυτό ορίσαμε ως πλήθος Virtual Machines του πωλητή ίσο με 40, πλήθος γύρων στο οποίο θα μελετήσουμε τη δημοπρασία ίσο με 72 το οποίο σε περίπτωση που πραγματοποιούμε τη δημοπρασία ανά μία ώρα (όπως και στο σύστημα Spot Instances της Amazon) ισοδυναμεί με 3 μέρες και αριθμό χρηστών ίσο με 10, 20 και 50. Στον παρακάτω πίνακα παρουσιάζονται οι βασικές παράμετροι.

| Παράμετροι | Τιμές |
|-------------------|------------|
| Εικονικές Μηχανές | 40 |
| Πλήθος Γύρων | 72 |
| Πλήθος Χρηστών | 10, 20, 50 |

Πίνακας 6.1: Βασικές Παράμετροι Προσομοίωσης

6.1.2 Κατανομές Προσφορών και Αιτημάτων Χρηστών

Σε ότι αφορά τις προσομοιώσεις, μπορούμε να τις χωρίσουμε σε δύο κατηγορίες. Στην πρώτη κατηγορία, χρησιμοποιούμε συνθετικό σύνολο δεδομένων σε ότι αφορά τα αιτήματα των χρηστών και πιο απλοϊκό μοντέλο προσφορών. Στη δεύτερη κατηγορία, χρησιμοποιούμε ένα πραγματικό σύνολο δεδομένων σε ότι αφορά τα αιτήματα των χρηστών το οποίο προέρχεται από το Google Trace [18] και ένα πιο σύνθετο μοντέλο προσφορών.

Στην πρώτη κατηγορία, για κάθε χρήστη i τα αιτήματα για εικονικές μηχανές ακολουθούν το απόλυτο ημιτονικής συνάρτησης με συχνότητα $2.5i$, ενώ οι ανά μονάδα προσφορές ακολουθούν $Lognormal(2, 0.25^2)$ κατανομή για κάθε χρήστη i .

Στη δεύτερη κατηγορία, για κάθε χρήστη i τα αιτήματα για εικονικές μηχανές βασίζονται στο προαναφερθέν Google Trace και θα περιγραφούν αναλυτικά στην επόμενη ενότητα, ενώ οι ανά μονάδα προσφορές ακολουθούν $Lognormal(2 + 0.05i - Y, 0.25^2)$ κατανομή για κάθε χρήστη i , με τη μέση τιμή $2 + 0.05i$ να μειώνεται σύμφωνα με την τυχαία μεταβλητή $Y = 0.01(\text{init_total_demands} - \text{remaining_total_demands}) \cdot |X|$, με $X \sim N(0, 1)$. Η παραπάνω κατανομή ανά μονάδα προσφοράς επιλέχθηκε με τέτοιο τρόπο έτσι ώστε κάθε χρήστης να ακολουθεί διαφορετική κατανομή και επιπλέον να αντικατοπτρίζει τη ρεαλιστική συμπεριφορά των περισσοτέρων χρηστών, η οποία δηλώνει ότι οι χρήστες εν γένει προσφέρουν λιγότερα όταν το έργο τους προς εκτέλεση δείχνει να φτάνει στο τέλος του. Η μικρότερη προσφορά δηλώνεται με τον όρο που αφαιρείται από την μέση τιμή, ο οποίος είναι ανάλογος με τη διαφορά των συνολικών εικονικών μηχανών που επιθυμεί ο χρήστης μείον αυτές που επιθυμεί ακόμα, επί το απόλυτο μιας τυχαίας τυπικής κανονικής μεταβλητής που αντικατοπτρίζει λευκό θόρυβο.

Στον παρακάτω πίνακα παρουσιάζονται τα προαναφερθέντα για κάθε χρήστη i .

| Προσεγγίση | Κατανομή Προσφορών | Κατανομή Αιτημάτων |
|------------|---|---|
| Πρώτη | $Lognormal(2, 0.25^2)$ | Απόλυτο Ημιτονικής Συνάρτησης με συχνότητα $2.5i$ |
| Δεύτερη | $Lognormal(2 + 0.05i - Y, 0.25^2)$, με $Y = 0.01(init - remaining) X $, με $X \sim N(0, 1)$ | Αιτήσεις από Google Trace |

Πίνακας 6.2: Κατανομές Προσφορών και Αιτημάτων Χρηστών

6.1.3 Σύνολο Δεδομένων Google Trace

Στο σημείο αυτό θα παρουσιάσουμε το σύνολο δεδομένων το οποίο χρησιμοποιήσαμε στις προσομοιώσεις μας. Πρόκειται για το σύνολο δεδομένων Google Trace/Google Cluster Data [18] το οποίο αντικατοπτρίζει 29 μέρες πληροφορία σχετικά με ένα cluster της Google το οποίο αποτελείται από 12.500 εικονικές μηχανές. Το συγκεκριμένο trace file αποτελείται από πολλαπλά datasets. Κάθε dataset αποτελείται από ένα σχεσιακό πίνακα, οποίος περιέχει ένα primary key (timestamp). Πριν προχωρήσουμε στην περιγραφή των πινάκων θα πρέπει να τονίσουμε πως τα περισσότερα δεδομένα είναι κρυπτογραφημένα ή παρουσιάζονται κανονικοποιημένα για λόγους προσωπικών δικαιωμάτων. Στη δική μας περίπτωση έχουμε πως τα μεγέθη *CPU* (πλήθος πυρήνων ή πλήθος πυρήνων ανά sec), *memory* (bytes) και *disk space* (bytes) είναι κανονικοποιημένα με μέγιστη τιμή ίση με 1 και ελάχιστη ίση με 0. Συνεχίζοντας παρουσιάζουμε τους πίνακες οι οποίοι είναι οι παρακάτω:

- Machine Events
- Machine Attributes
- Job Events
- Task Events
- Task Constraints
- Task Usage

Οι δύο πρώτοι πίνακες αφορούν τις εικονικές μηχανές του cluster. Ο πίνακας **Machine Events** περιέχει έξι διαφορετικές κατηγορίες εγγραφών. Αυτές είναι: *timestamp*, *machine ID*, *event type*, *platform ID*, *capacity: CPU*, *capacity: memory*. Ο πίνακας **Machine Attributes** περιέχει πέντε διαφορετικές κατηγορίες εγγραφών. Αυτές είναι: *timestamp*, *machine ID*, *attribute name*, *attribute value*, *attribute deleted*.

Στη συνέχεια, ασχολούμαστε με τους πίνακες που αφορούν τα tasks και τα jobs. Υπενθυμίζουμε πως ένα job μπορεί να περιέχει πολλά tasks τα οποία τρέχουν σειριακά ή παράλληλα. Ο πίνακας **Job Events** περιέχει οχτώ διαφορετικές κατηγορίες εγγραφών. Αυτές είναι: *timestamp*, *missing info*, *job ID*, *event type*, *user name*, *scheduling class*, *job name*, *logical job name*. Ο πίνακας **Task Events** περιέχει δεκατρείς διαφορετικές κατηγορίες εγγραφών. Αυτές είναι: *timestamp*, *missing info*, *job ID*, *task index within the job*, *machine ID*, *event type*, *username*, *scheduling class*, *priority*, *resource request for CPU cores*, *resource request for RAM*, *resource request for disk space*, *different-machine constraint*. Ο πίνακας **Task Constraints** περιέχει έξι διαφορετικές κατηγορίες εγγραφών. Αυτές είναι: *timestamp*, *job ID*, *task index within the job*, *attribute name*, *attribute value*, *comparison operator*. Τέλος, ο πίνακας **Task Usage** περιέχει είκοσι διαφορετικές κατηγορίες εγγραφών. Αυτές είναι: *start time of the measurement period*, *end time of the measurement period*, *job ID*, *task index*, *machine ID*, *mean CPU usage rate*, *canonical memory usage*, *assigned memory usage*, *unmapped page cache memory usage*, *total page cache memory usage*, *mean disk I/O time*, *mean local disk space used*, *maximum CPU usage*, *maximum disk IO time*, *cycles per instruction*, *memory accesses per instruction (MAI)*, *sample portion*, *aggregation type* (1 if maximums from subcontainers were summed), *sampled CPU usage: mean CPU usage during a random 1s sample in the measurement period*.

Στο πλαίσιο της συγκεκριμένης διπλωματικής εργασίας και της συγκεκριμένης προσομοίωσης χρησιμοποιήσαμε την τεχνική που συναντάμε σε διεθνείς εργασίες [25], [23]. Κάθε job αποτελείται από διάφορα tasks. Κάθε task πραγματοποιεί κάποια request σε πόρους (cpu, ram, disk) τα οποία όμως είναι κανονικοποιημένα. Θεωρούμε σύμφωνα με την υπηρεσία Amazon Spot Instances ένα vm ως βάση για τα requests. Η εικονική μηχανή που χρησιμοποιούμε εμείς είναι η Amazon EC2 Instance Type a1.medium, η οποία χρησιμοποιείται ευρέως για οικονομικούς λόγους και τα χαρακτηριστικά της φαίνονται στον παρακάτω πίνακα.

| Χαρακτηριστικά | Τιμή |
|--------------------------|------|
| Πλήθος Εικονικών Πυρήνων | 1 |
| Μνήμη | 2 Gb |

Πίνακας 6.3: Χαρακτηριστικά Amazon EC2 Instance Type a1.medium

Επιπρόσθετα θεωρούμε πως το μέγιστο που μπορεί κάποιο task να αιτηθεί σε πόρους είναι 32 εικονικούς πυρήνες και 128 Gb Ram. Συνεπώς, ο αριθμός των vms που απαιτεί κάθε job το οποίο αποτελείται από πολλαπλά tasks βρίσκεται από τη σχέση: $max(\lceil \frac{normalized_cpu * 32}{1} \rceil, \lceil \frac{normalized_ram * 128}{2} \rceil)$ όπου *normalized_cpu* και *normalized_ram* η κανονικοποιημένη τιμή που απαιτεί ένα job. Για να αποκτήσουμε τις προαναφερθείσες τιμές πραγματοποιήσαμε inner join πάνω στο πεδίο *job ID* μεταξύ των πινάκων **Task Events** και **Job Events**. Στη συνέχεια, για κάθε *job ID* κρατήσαμε τις τιμές σχετικά με τα ζητούμενα resources από κάθε task μέσα στο Job, τις αθροίσαμε (ελέγχοντας να μην παίρνουν τιμές μεγαλύτερες του 1) και βρήκαμε τις ζητούμενες παραμέτρους.

Με τον τρόπο αυτό, δημιουργούμε μέσα από το Google Trace τα demands σε εικονικές μηχανές που θα πραγματοποιήσουν οι χρήστες μας στη δεύτερη κατηγορία προσομοιώσεων. Προχωράμε κλείνοντας την ενότητα περιγράφοντας τις μετρικές αξιολόγησης που θα χρησιμοποιήσουμε.

6.1.4 Μετρικές Αξιολόγησης

Για την αξιολόγηση των προσομοιώσεων μας κρατάμε ορισμένες μετρικές. Οι μετρικές αυτές είναι η κοινωνική ωφέλεια συνολικά, ανά γύρο και ανά χρήστη, τα έσοδα συνολικά και ανά γύρο, η CPU Utilization(%) ανά γύρο και για τους προσεγγιστικούς αλγόριθμους για τους οποίους έχουμε θεωρητικά κάτω φράγματα για τους λόγους προσέγγισης, την τιμή του λόγου προσέγγισης (approximation ratio) ανά γύρο. Επιπρόσθετα, για την αξιολόγηση της ομοιότητας μεταξύ των κατανομών για τον αλγόριθμο Expectation-Maximization χρησιμοποιούμε την απόσταση Kullback-Leibler. Συνοπτικά λοιπόν οι μετρικές είναι:

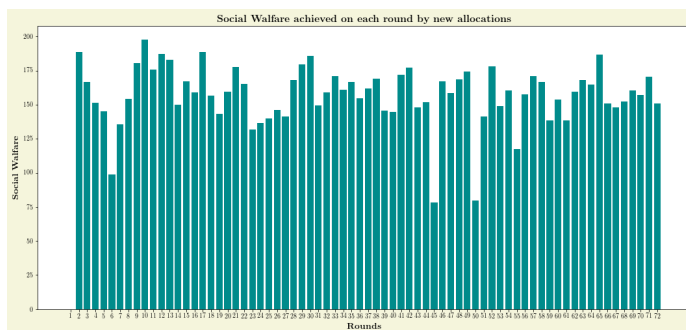
1. Social Welfare (συνολικά, ανά γύρο, ανά χρήστη)
2. Έσοδα (συνολικά, ανά γύρο)
3. CPU Utilization(%) ανά γύρο
4. Approximation Ratio (ανά γύρο)
5. Απόσταση Kullback-Leibler

6.2 Αποτελέσματα Μεγιστοποίησης Κοινωνικής Ωφέλειας

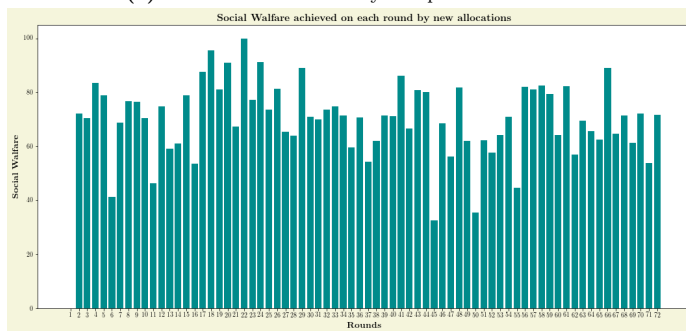
Στην ενότητα αυτή θα παραθέσουμε τα αποτελέσματα των προσομοιώσεων μας για την προσέγγιση της μεγιστοποίησης της κοινωνικής ωφέλειας. Αρχικά, θα παρουσιάσουμε τα αποτελέσματα για την πρώτη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου όπως αναφέρθηκε και προηγουμένως, ενώ στη συνέχεια θα παρουσιάσουμε τα αποτελέσματα για τη δεύτερη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές βασίζονται στο Google Trace. Ξεκινάμε παραθέτοντας τα αποτελέσματα της πρώτης προσέγγισης.

6.2.1 Πρώτη Προσέγγιση

Στην πρώτη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου. Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Στη συγκεκριμένη περίπτωση, παραθέτουμε διαγράμματα για το λόγο προσέγγισης όπως δίνεται από τη σχέση στην ενότητα 5.1.1. Ξεκινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο, τη χρησιμοποίηση CPU ανά γύρο και το λόγο προσέγγισης ανά γύρο.

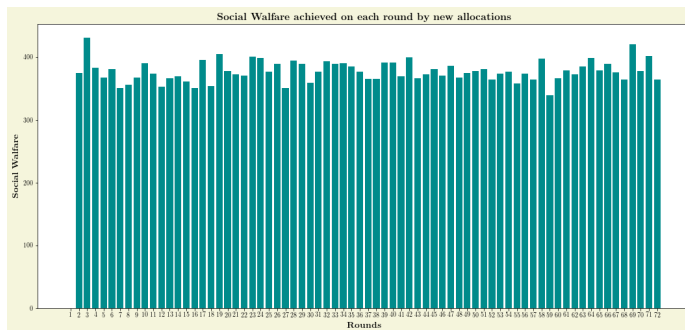


(a) Social Welfare - Binary Knapsack - 10 Users



(b) Social Welfare - Binary Knapsack - 20 Users

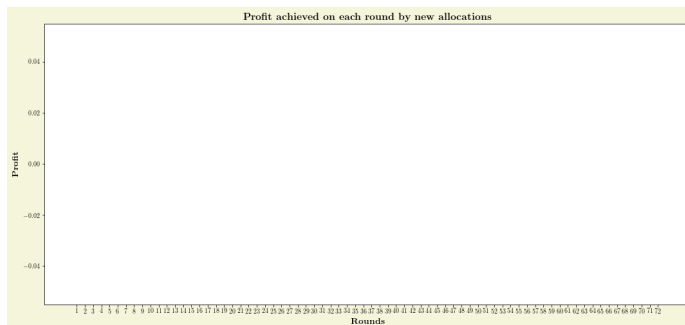
Εικόνα 6.1: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



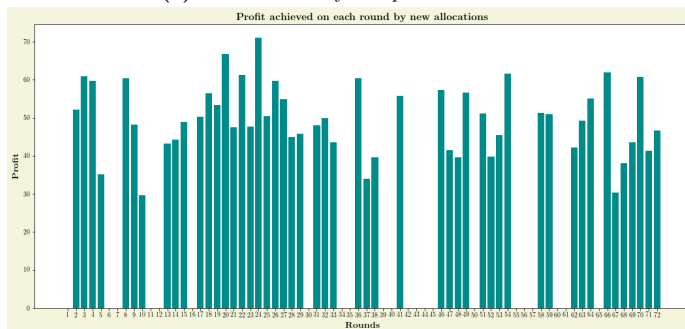
(c) Social Welfare - Binary Knapsack - 50 Users

Εικόνα 6.1: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack (συνέχεια).

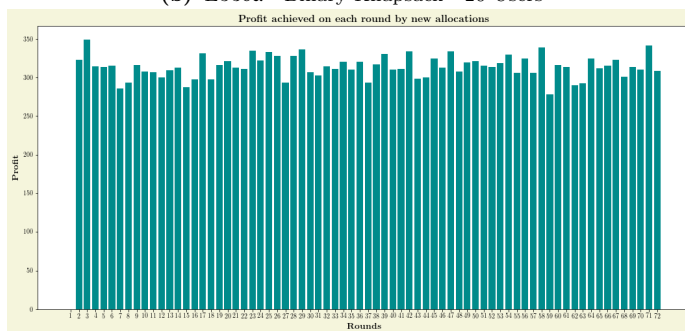
Στη συνέχεια, παρουσιάζουμε τα διαγράμματα σχετικά με τα έσοδα ανά γύρο.



(a) Έσοδα - Binary Knapsack - 10 Users



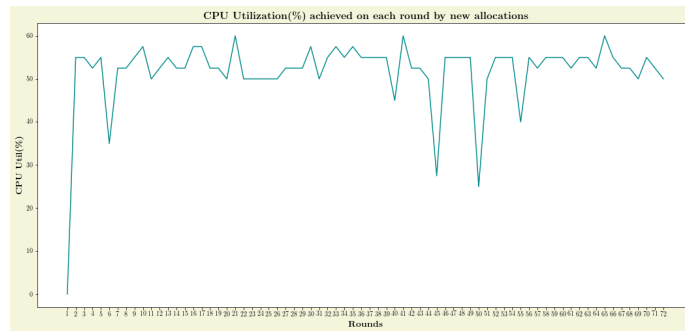
(b) Έσοδα - Binary Knapsack - 20 Users



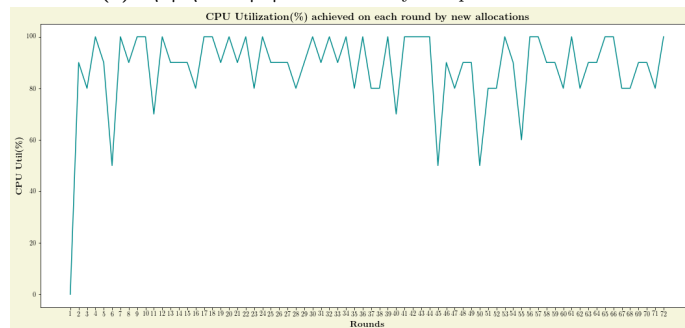
(c) Έσοδα - Binary Knapsack - 50 Users

Εικόνα 6.2: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

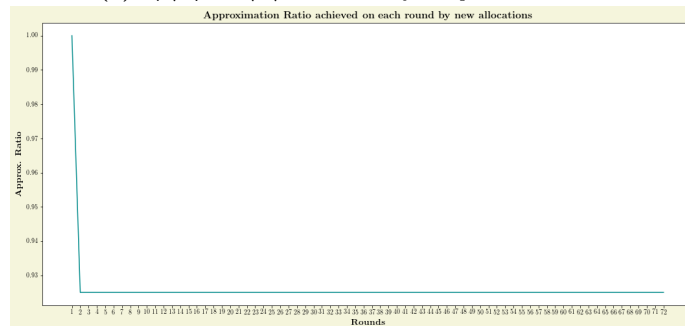
Ακολουθούν τα διαγράμματα σχετικά με τη χρησιμοποίηση CPU ανά γύρο.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users



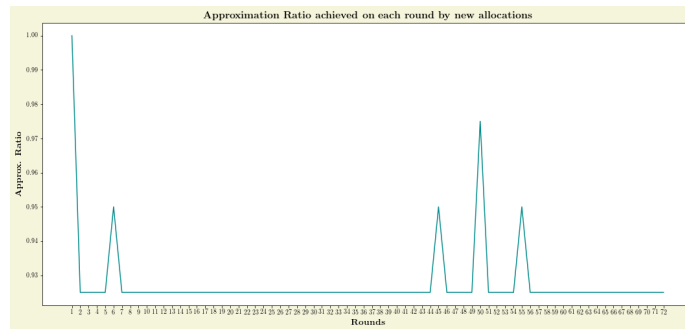
(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users



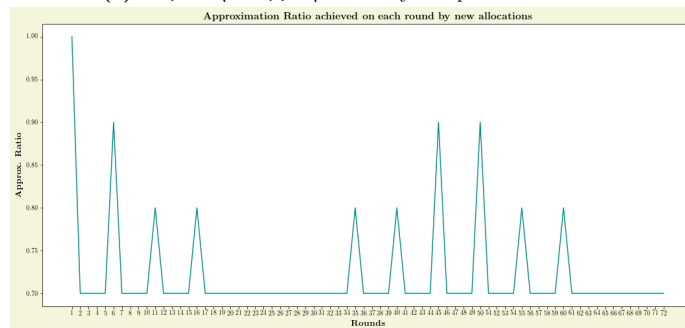
(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

Εικόνα 6.3: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

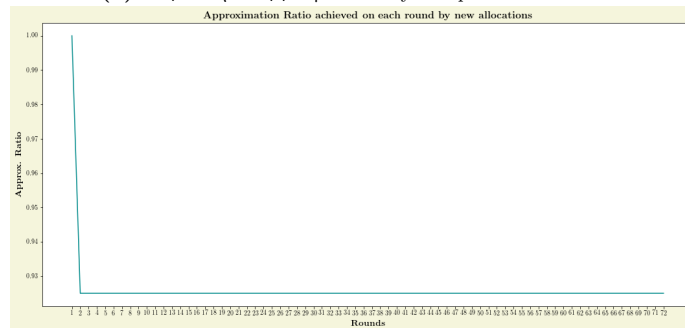
Τέλος, παρουσιάζονται τα διαγράμματα σχετικά με το λόγο προσέγγισης ανά γύρο.



(a) Λόγος Προσέγγισης - Binary Knapsack - 10 Users



(b) Λόγος Προσέγγισης - Binary Knapsack - 20 Users



(c) Λόγος Προσέγγισης - Binary Knapsack - 50 Users

Εικόνα 6.4: Διαγράμματα λόγου προσέγγισης ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

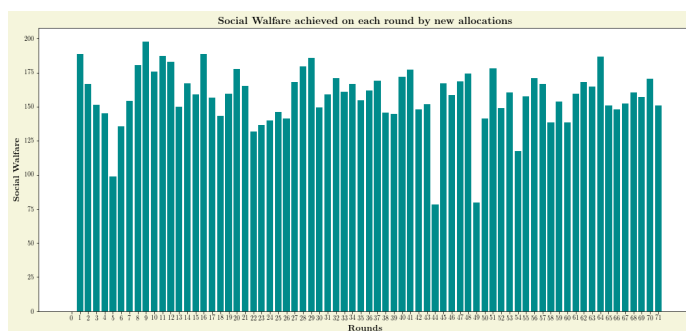
Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 0 | 11164.34 | 15.506 |
| 20 | 2488.47 | 4986.14 | 13.850 |
| 50 | 22290.01 | 26801.81 | 7.445 |

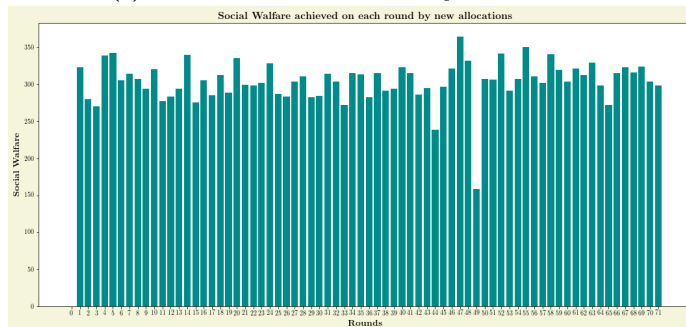
Πίνακας 6.4: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας

Σχολιασμός: Από τις παραπάνω προσομοιώσεις λαμβάνουμε τα αναμενόμενα με βάση τη θεωρία αποτελέσματα. Παρατηρούμε έχουμε μεγάλη κοινωνική ωφέλεια, πάντα μεγαλύτερη από τα έσοδα του πωλητή. Ιδιαίτερα στην περίπτωση του unlimited supply παρατηρούμε πως τα έσοδα είναι 0, δηλαδή πως ο πωλητής διαθέτει τις εικονικές μηχανές δωρεάν, γεγονός που πράγματι μεγιστοποιεί την κοινωνική ωφέλεια. Βλέπουμε πως όσο αυξάνεται ο αριθμός των χρηστών, τόσο μειώνεται η μέση κοινωνική ωφέλεια καθώς αρχίζουμε και έχουμε περιπτώσεις limited supply. Στους 20 χρήστες έχουμε και limited και unlimited supply, ενώ στους 50 μόνο limited. Στην περίπτωση του Limited supply εισέρχονται ως "αναγκαίο κακό" οι πληρωμές προκειμένου να έχουμε μία δίκαιη κατανομή των εικονικών μηχανών στους χρήστες και προφανώς η μέση κοινωνική ωφέλεια μειώνεται. Σε ότι αφορά τη χρησιμοποίηση CPU, αυτή όπως αναμενόταν αυξάνεται καθώς αυξάνονται και οι χρήστες καθώς σε περιπτώσεις unlimited supply σημαίνει πως πολλές εικονικές μηχανές δεν κατανέμονται σε χρήστες, ενώ σε περιπτώσεις limited supply σχεδόν όλες οι εικονικές μηχανές κατανέμονται σε χρήστες. Τέλος, το θεωρητικό approximation ratio μας λαμβάνει πάρα πολύ καλές τιμές μεταξύ 0.79 και 0.975.

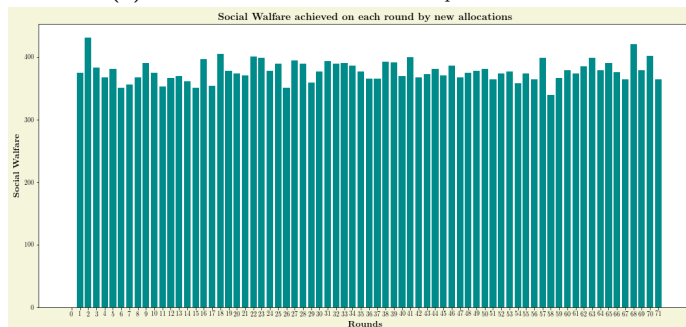
Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

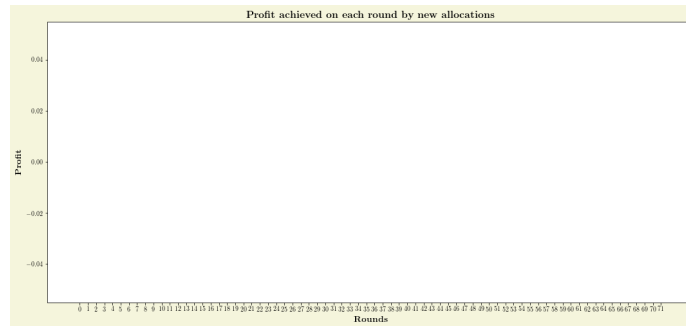


(b) Social Welfare - Fractional Knapsack - 20 Users

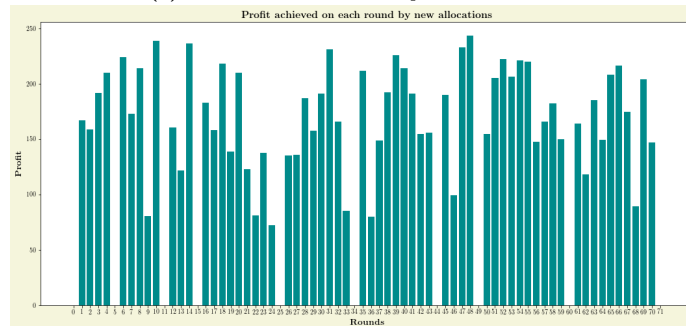


(c) Social Welfare - Fractional Knapsack - 50 Users

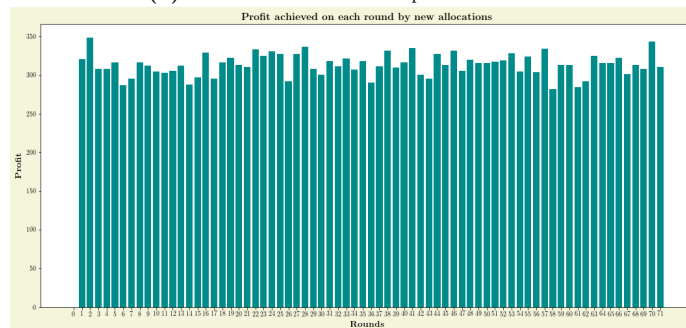
Εικόνα 6.5: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

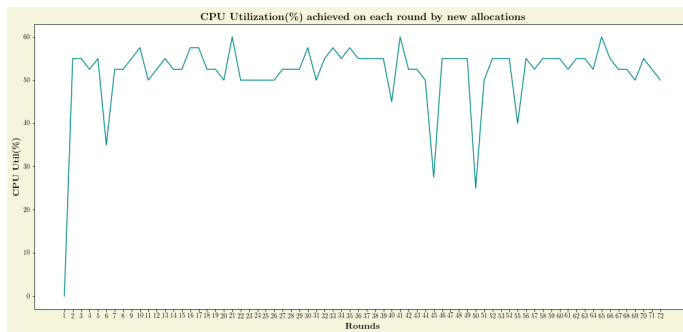


(b) Έσοδα - Fractional Knapsack - 20 Users

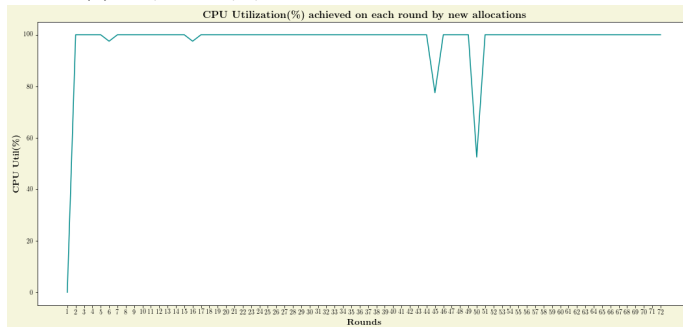


(c) Έσοδα - Fractional Knapsack - 50 Users

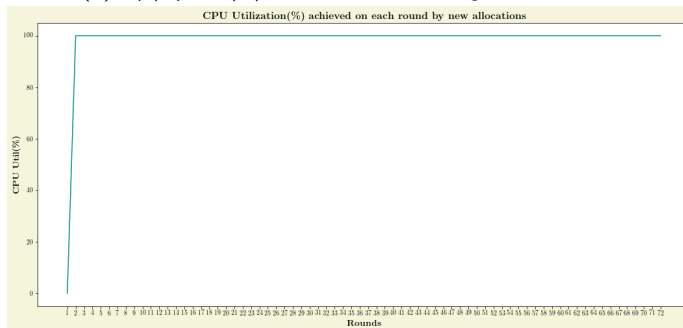
Εικόνα 6.6: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Χρησιμοποίηση CPU - Fractional Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Fractional Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Fractional Knapsack - 50 Users

Εικόνα 6.7: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

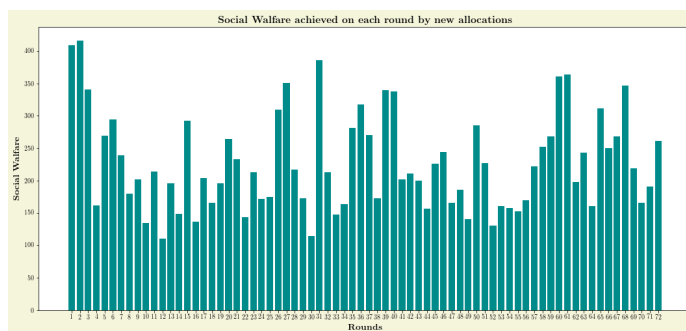
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 0 | 11164.34 | 15.506 |
| 20 | 10660.31 | 21601.17 | 15.0 |
| 50 | 22263.16 | 26815.97 | 7.449 |

Πίνακας 6.5: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας

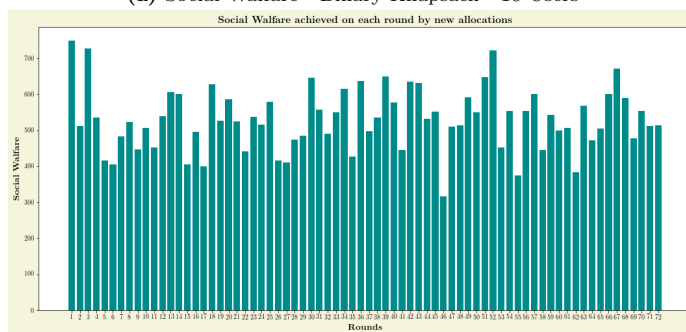
Σχολιασμός: Παρατηρούμε παρόμοια αποτελέσματα με αυτά του binary προβλήματος. Όπως αναμενόταν είναι ελαφρώς καλύτερα καθώς σύμφωνα με τη fractional προσέγγιση μπορούμε να εξυπηρετούμε κάποιους παραπάνω χρήστες που δε θα μπορούσαμε στην binary προσέγγιση.

6.2.2 Δεύτερη Προσέγγιση

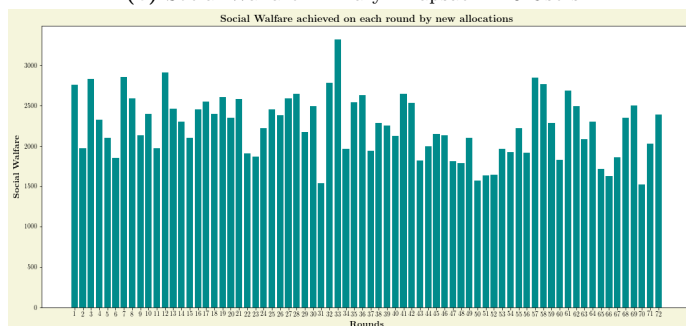
Στη δεύτερη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές παράγονται όπως αναφέρθηκε στην ενότητα 6.1.3 με βάση το Google Trace. Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Ξεκινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο, τη χρησιμοποίηση CPU ανά γύρο και το λόγο προσέγγισης ανά γύρο.



(a) Social Welfare - Binary Knapsack - 10 Users

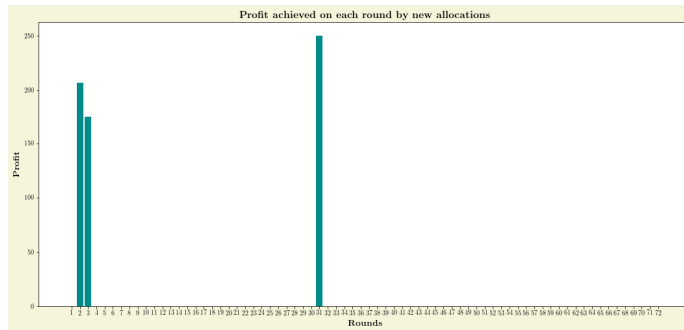


(b) Social Welfare - Binary Knapsack - 20 Users

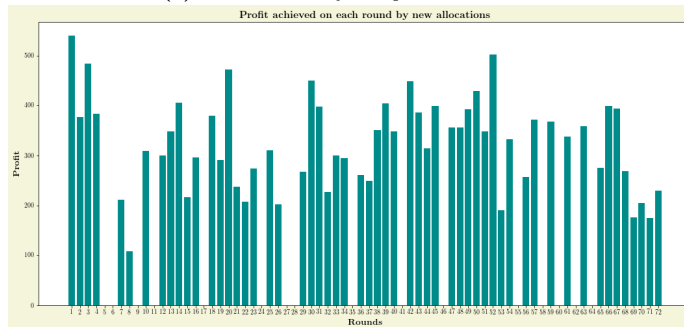


(c) Social Welfare - Binary Knapsack - 50 Users

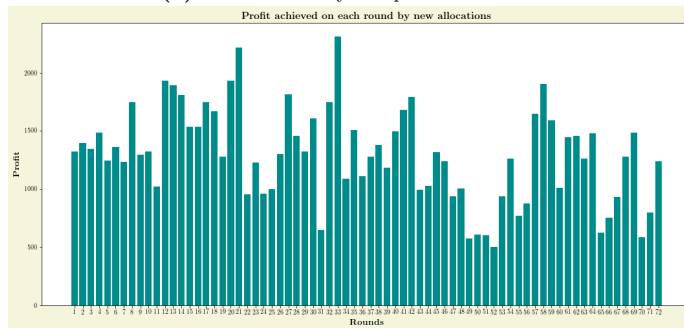
Εικόνα 6.8: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary knapsack.



(a) Έσοδα - Binary Knapsack - 10 Users

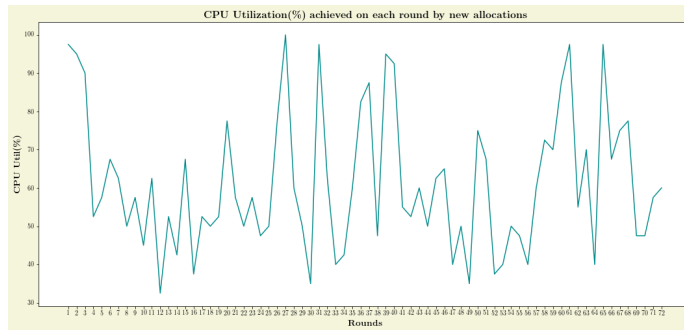


(b) Έσοδα - Binary Knapsack - 20 Users

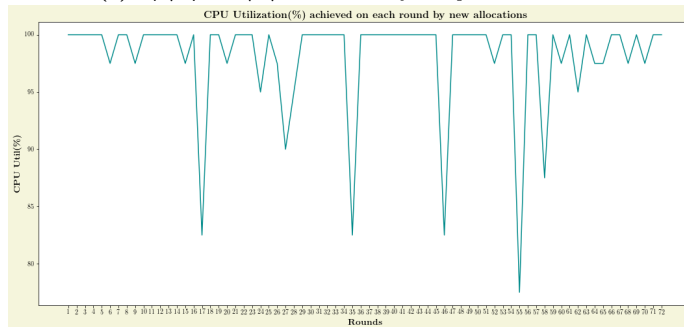


(c) Έσοδα - Binary Knapsack - 50 Users

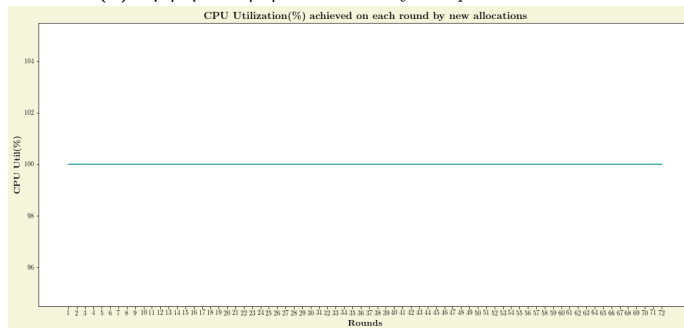
Εικόνα 6.9: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary knapsack.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users

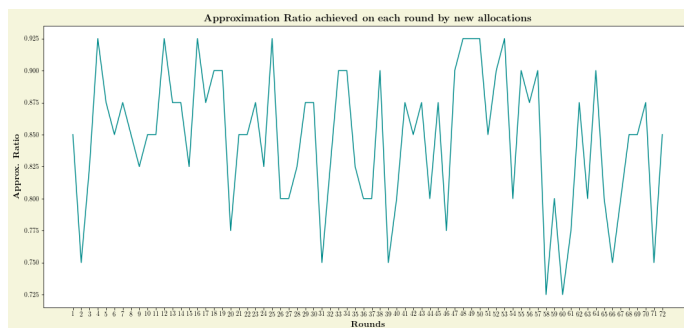


(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users

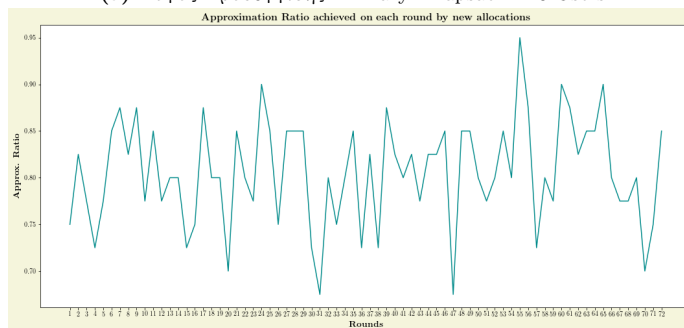


(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

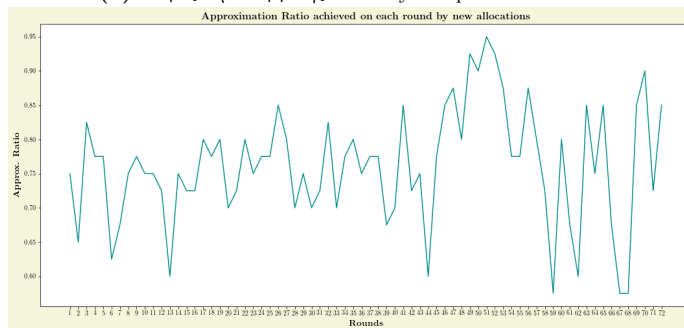
Εικόνα 6.10: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary knapsack.



(a) Λόγος Προσέγγισης - Binary Knapsack - 10 Users



(b) Λόγος Προσέγγισης - Binary Knapsack - 20 Users



(c) Λόγος Προσέγγισης - Binary Knapsack - 50 Users

Εικόνα 6.11: Διαγράμματα λόγου προσέγγισης ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

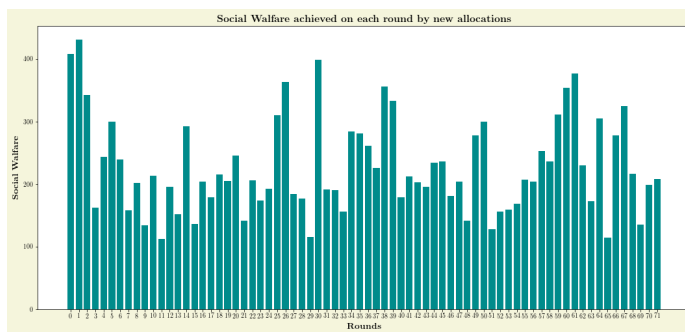
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 631.72 | 16404.03 | 22.783 |
| 20 | 18165.43 | 38130.29 | 26.47 |
| 50 | 93339.73 | 161845.75 | 44.95 |

Πίνακας 6.6: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας

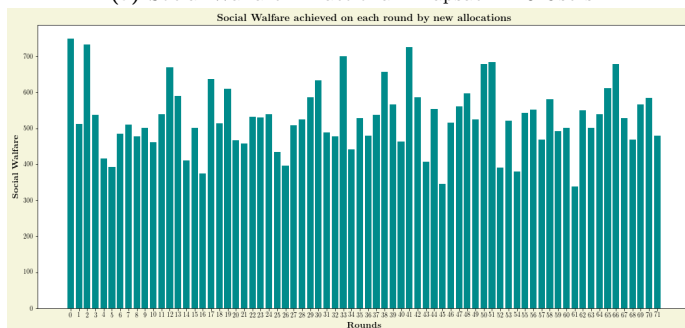
Σχολιασμός: Από τις παραπάνω προσομοιώσεις λαμβάνουμε τα αναμενόμενα με βάση τη θεωρία αποτελέσματα. Παρατηρούμε έχουμε μεγάλη κοινωνική ωφέλεια, πάντα μεγαλύτερη από τα έσοδα του πωλητή. Ιδιαίτερα στην περίπτωση του unlimited supply παρατηρούμε πως τα έσοδα είναι 0, δηλαδή πως ο πωλητής διαθέτει τις

εικονικές μηχανές δωρεάν, γεγονός που πράγματι μεγιστοποιεί την κοινωνική ωφέλεια. Βλέπουμε πως όσο αυξάνεται ο αριθμός των χρηστών, δεν μειώνεται η μέση κοινωνική ωφέλεια όπως στην πρώτη προσέγγιση. Αυτό συμβαίνει καθώς κάθε χρήστης έχει διαφορετική κατανομή ανά μονάδα προσφοράς και καθώς αρχίζουμε να έχουμε περιπτώσεις limited supply αρχίζουν και υπερισχύουν οι χρήστες με τις μεγαλύτερες ανά μονάδα προσφορές. Σε ότι αφορά τη χρησιμοποίηση CPU, παρουσιάζει πολλές διακυμάνσεις λόγω των ασύμμετρων αιτήσεων για εικονικές μηχανές ωστόσο όπως αναμενόταν αυξάνεται καθώς αυξάνονται και οι χρήστες καθώς σε περιπτώσεις unlimited supply σημαίνει πως πολλές εικονικές μηχανές δεν κατανέμονται σε χρήστες, ενώ σε περιπτώσεις limited supply σχεδόν όλες οι εικονικές μηχανές κατανέμονται σε χρήστες. Τέλος, το θεωρητικό approximation ratio μας λαμβάνει πάρα πολύ καλές τιμές μεταξύ 0.58 και 0.95.

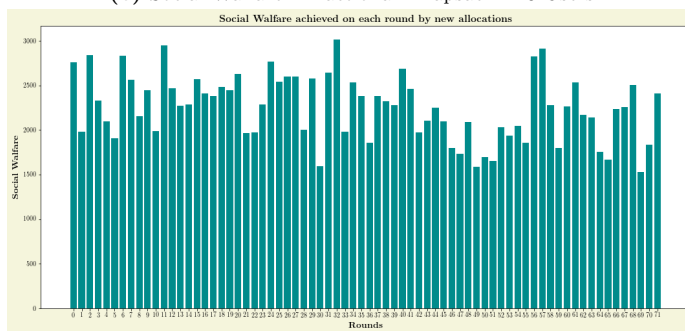
Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

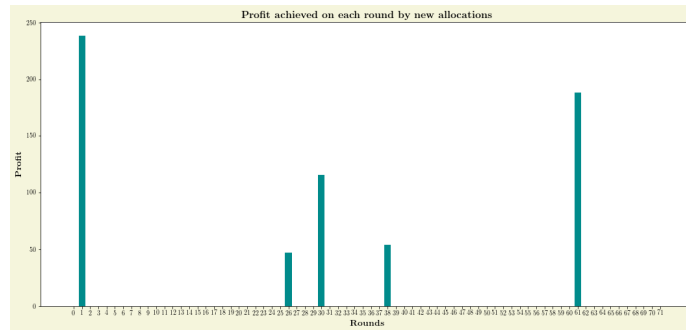


(b) Social Welfare - Fractional Knapsack - 20 Users

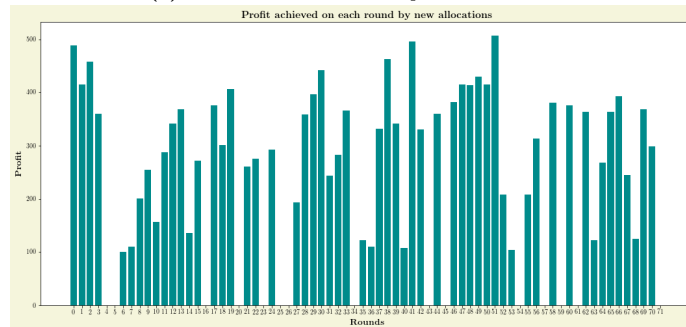


(c) Social Welfare - Fractional Knapsack - 50 Users

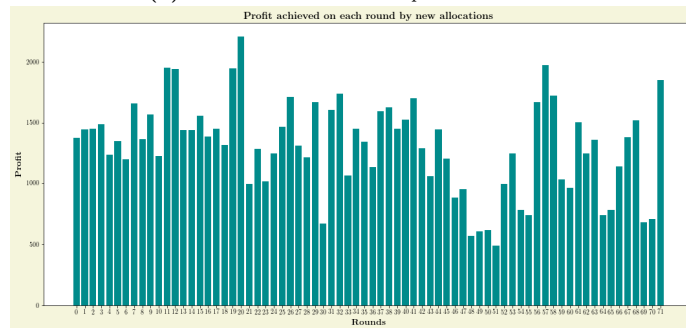
Εικόνα 6.12: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

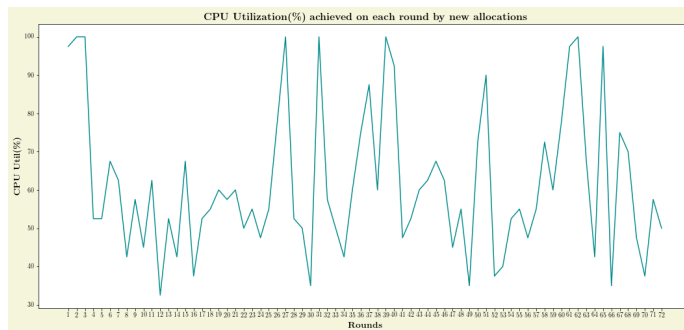


(b) Έσοδα - Fractional Knapsack - 20 Users

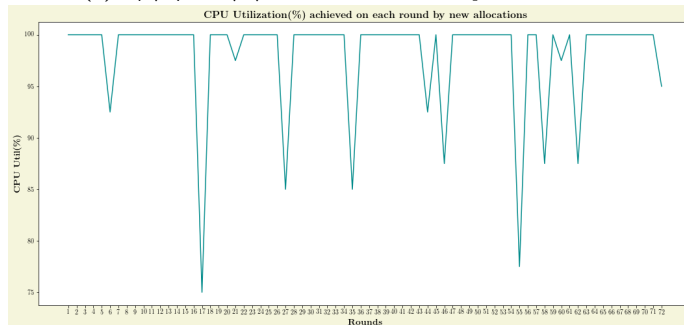


(c) Έσοδα - Fractional Knapsack - 50 Users

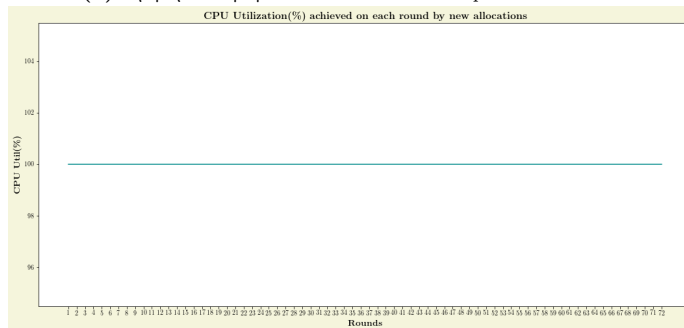
Εικόνα 6.13: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Χρησιμοποίηση CPU - Fractional Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Fractional Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Fractional Knapsack - 50 Users

Εικόνα 6.14: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 643.63 | 16421.45 | 22.81 |
| 20 | 17500.9 | 38029.67 | 26.41 |
| 50 | 94045.42 | 162175.46 | 45.05 |

Πίνακας 6.7: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Κοινωνικής Ωφέλειας

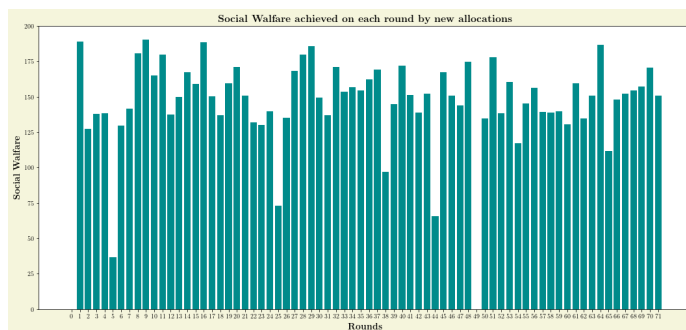
Σχολιασμός: Παρατηρούμε παρόμοια αποτελέσματα με αυτά του binary προβλήματος. Όπως αναμενόταν είναι ελαφρώς καλύτερα στις περιπτώσεις με 10 και 50 χρήστες καθώς σύμφωνα με τη fractional προσέγγιση μπορούμε να εξυπηρετούμε κάποιους παραπάνω χρήστες που δε θα μπορούσαμε στην binary προσέγγιση.

6.3 Αποτελέσματα Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών

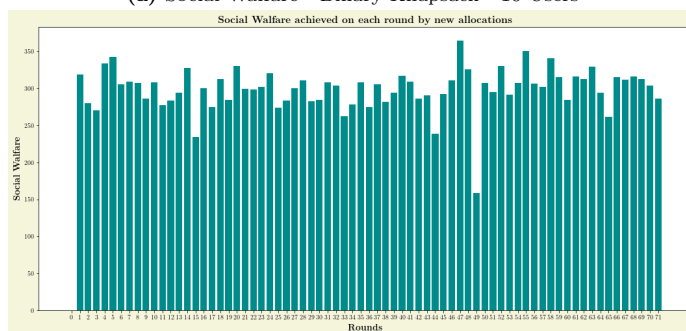
Στην ενότητα αυτή θα παραθέσουμε τα αποτελέσματα των προσομοιώσεων μας για την προσέγγιση της μεγιστοποίησης εσόδων με χρήση προσεγγιστικού αλγορίθμου και τυχαίων δημοπρασιών όπως αναλύθηκε στην ενότητα 5.2. Αρχικά, θα παρουσιάσουμε τα αποτελέσματα για την πρώτη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου όπως αναφέρθηκε και προηγουμένως, ενώ στη συνέχεια θα παρουσιάσουμε τα αποτελέσματα για τη δεύτερη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές βασίζονται στο Google Trace. Ξεκινάμε παραθέτοντας τα αποτελέσματα της πρώτης προσέγγισης.

6.3.1 Πρώτη Προσέγγιση

Στην πρώτη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου. Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Ξεκινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο και τη χρησιμοποίηση CPU ανά γύρο.

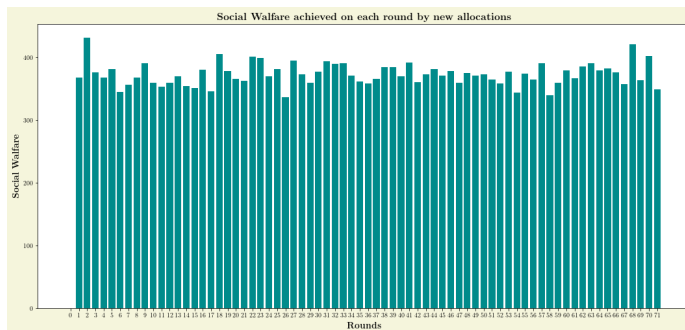


(a) Social Welfare - Binary Knapsack - 10 Users



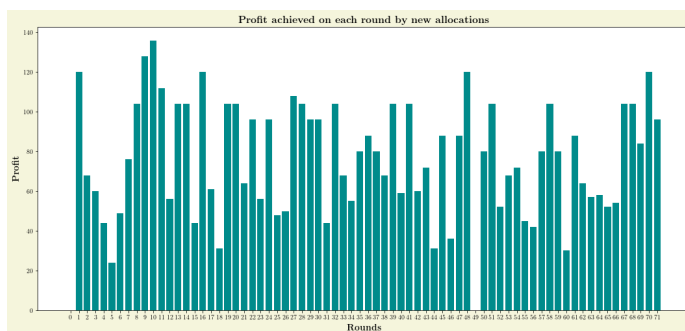
(b) Social Welfare - Binary Knapsack - 20 Users

Εικόνα 6.15: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

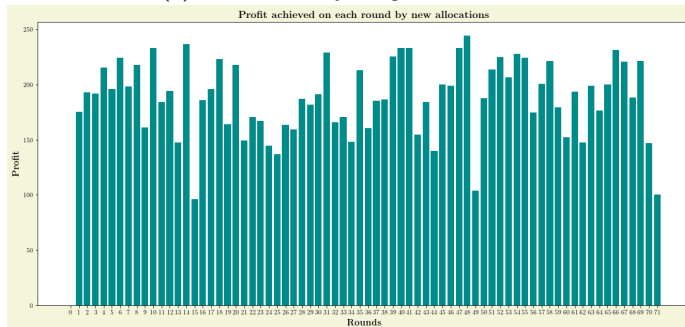


(c) Social Welfare - Binary Knapsack - 50 Users

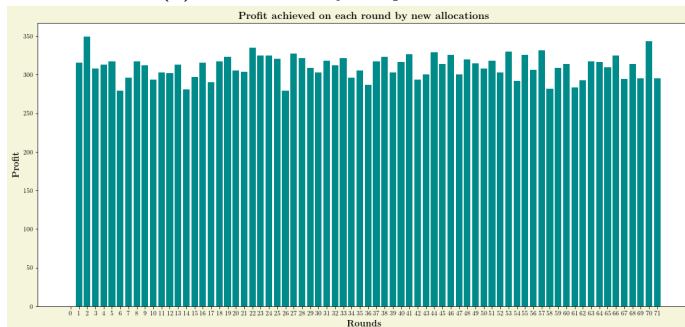
Εικόνα 6.15: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack (συνέχεια).



(a) Έσοδα - Binary Knapsack - 10 Users

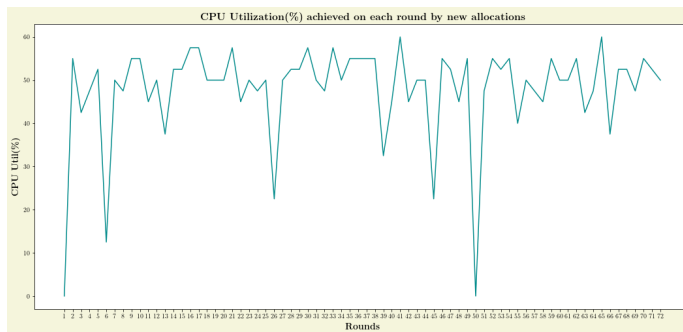


(b) Έσοδα - Binary Knapsack - 20 Users

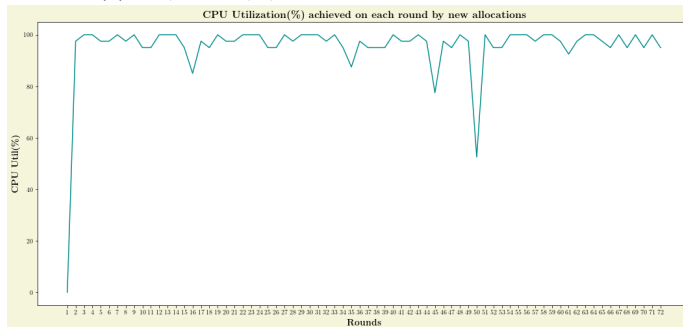


(c) Έσοδα - Binary Knapsack - 50 Users

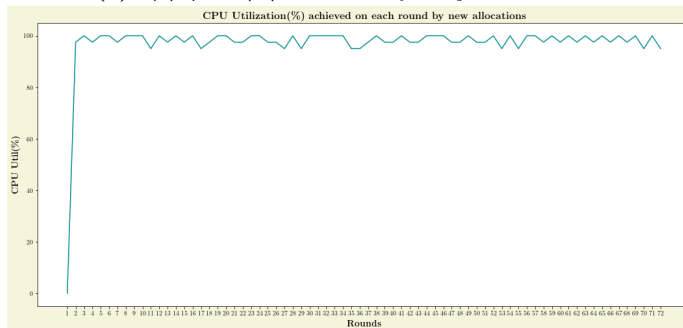
Εικόνα 6.16: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

Εικόνα 6.17: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

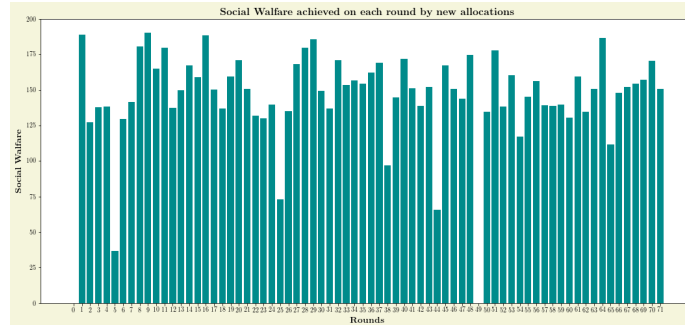
Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 5450 | 10399.21 | 14.44 |
| 20 | 13349.43 | 21230.9 | 14.74 |
| 50 | 22023.97 | 26461.94 | 7.35 |

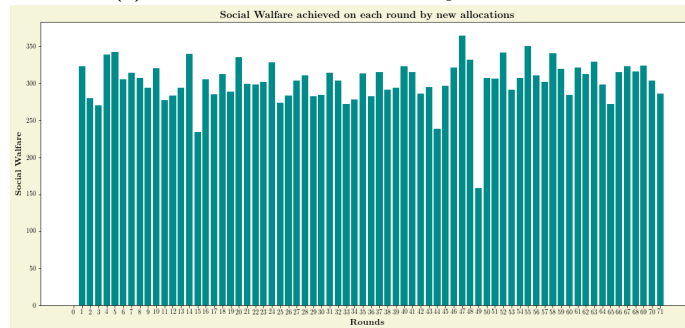
Πίνακας 6.8: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών

Σχολιασμός: Από τις παραπάνω προσομοιώσεις λαμβάνουμε τα αναμενόμενα με βάση τη θεωρία αποτελέσματα. Παρατηρούμε ότι έχουμε μεγάλη αύξηση των κερδών σε σχέση με την προσέγγιση μεγιστοποίησης κοινωνικής ωφέλειας ιδιαίτερα στην περίπτωση του unlimited supply, γεγονός που αποδεικνύει το πόσο καλά λειτουργούν οι τυχαίες δημοπρασίες. Η κοινωνική ωφέλεια παραμένει σε σχετικά υψηλά επίπεδα. Σε ότι αφορά τη χρησιμοποίηση CPU, αυτή όπως αναμενόταν αυξάνεται καθώς αυξάνονται και οι χρήστες καθώς σε περιπτώσεις unlimited supply σημαίνει πως πολλές εικονικές μηχανές δεν κατανέμονται σε χρήστες, ενώ σε περιπτώσεις limited supply σχεδόν όλες οι εικονικές μηχανές κατανέμονται σε χρήστες.

Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

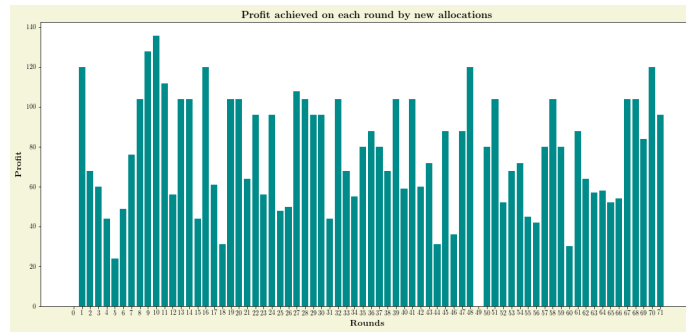


(b) Social Welfare - Fractional Knapsack - 20 Users

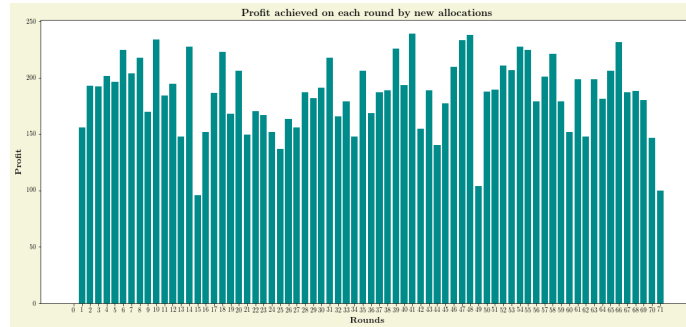


(c) Social Welfare - Fractional Knapsack - 50 Users

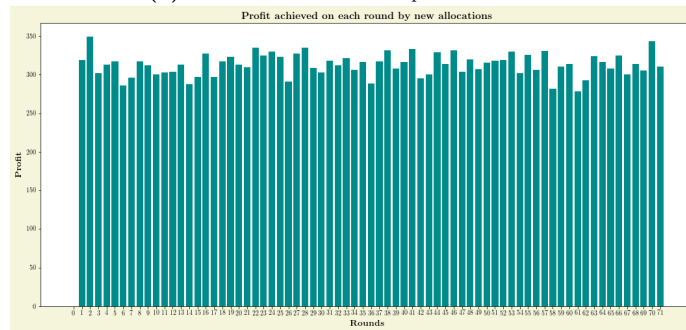
Εικόνα 6.18: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

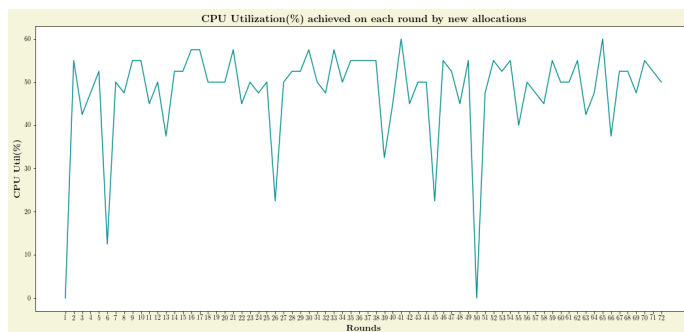


(b) Έσοδα - Fractional Knapsack - 20 Users

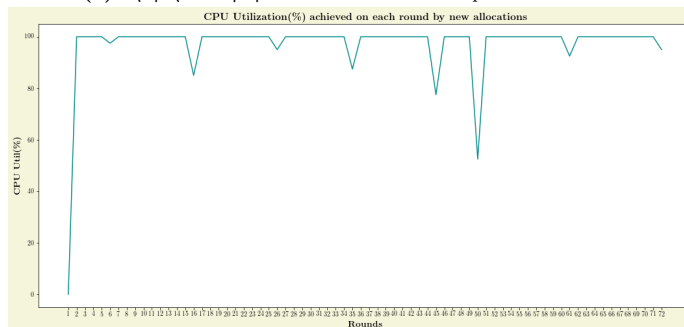


(c) Έσοδα - Fractional Knapsack - 50 Users

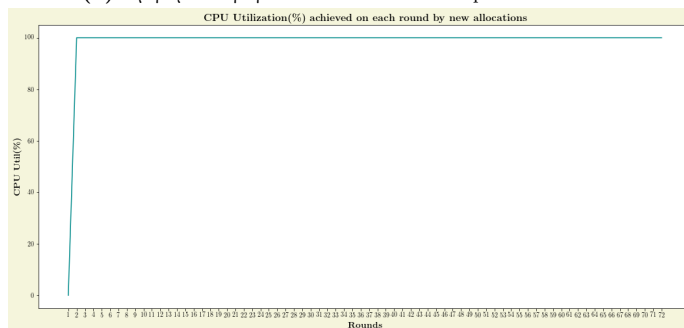
Εικόνα 6.19: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Χρησιμοποίηση CPU - Fractional Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Fractional Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Fractional Knapsack - 50 Users

Εικόνα 6.20: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

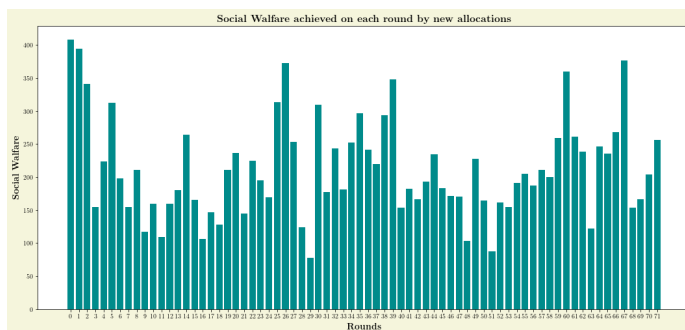
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 5450 | 10399.21 | 14.44 |
| 20 | 13136.2 | 21478.07 | 14.92 |
| 50 | 22220.4 | 26815.97 | 7.45 |

Πίνακας 6.9: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών

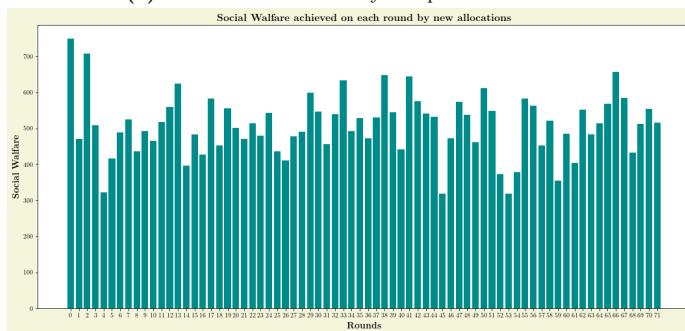
Σχολιασμός: Παρατηρούμε παρόμοια αποτελέσματα με αυτά του binary προβλήματος με μικρές αποκλίσεις.

6.3.2 Δεύτερη Προσέγγιση

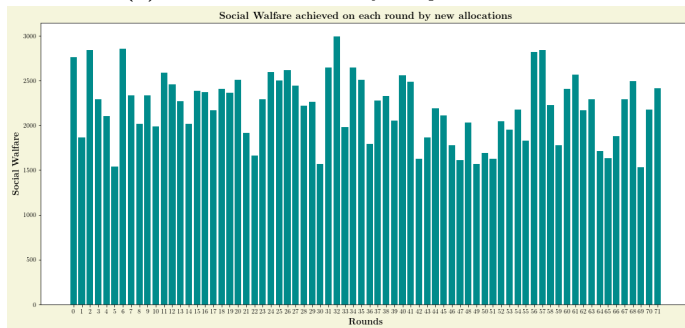
Στη δεύτερη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές παράγονται όπως αναφέρθηκε στην ενότητα 6.1.3 με βάση το Google Trace. Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Ξεκινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο και τη χρησιμοποίηση CPU ανά γύρο.



(a) Social Welfare - Binary Knapsack - 10 Users

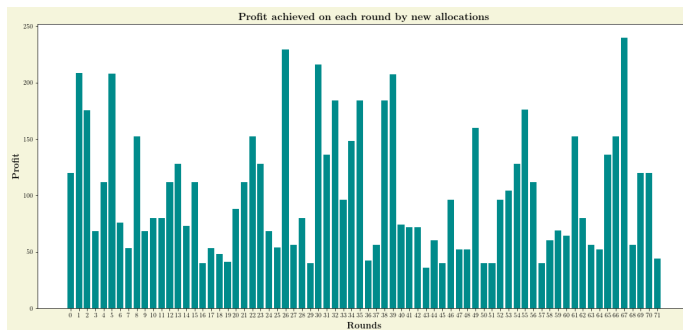


(b) Social Welfare - Binary Knapsack - 20 Users

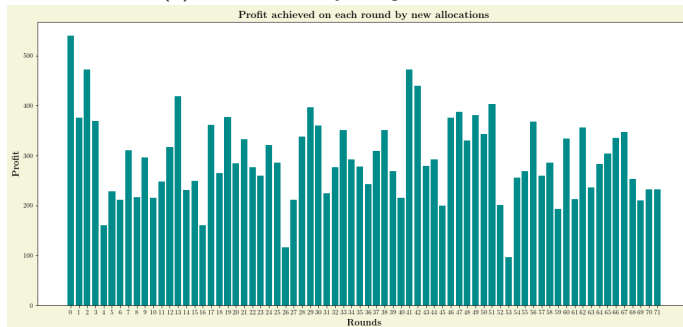


(c) Social Welfare - Binary Knapsack - 50 Users

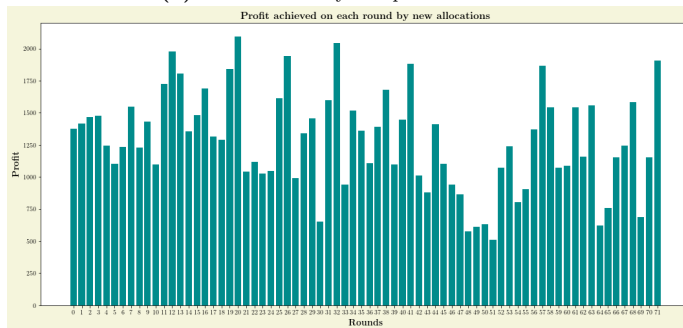
Εικόνα 6.21: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Έσοδα - Binary Knapsack - 10 Users

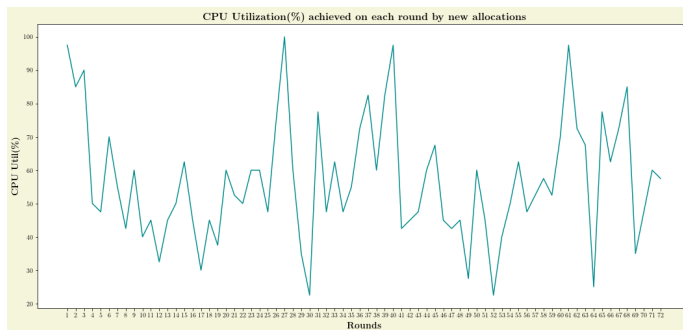


(b) Έσοδα - Binary Knapsack - 20 Users

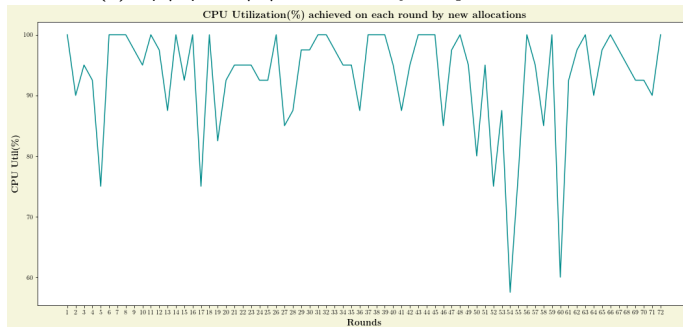


(c) Έσοδα - Binary Knapsack - 50 Users

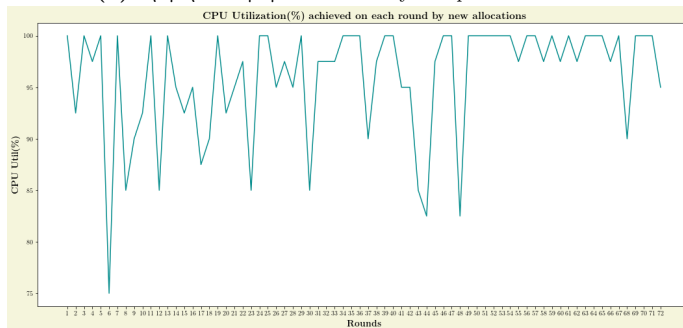
Εικόνα 6.22: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

Εικόνα 6.23: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

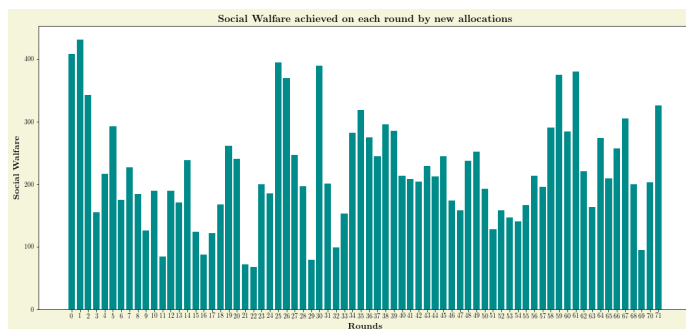
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 7319.82 | 15326.63 | 21.29 |
| 20 | 21167.37 | 36551.79 | 25.38 |
| 50 | 92358.8 | 158150.38 | 43.93 |

Πίνακας 6.10: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών

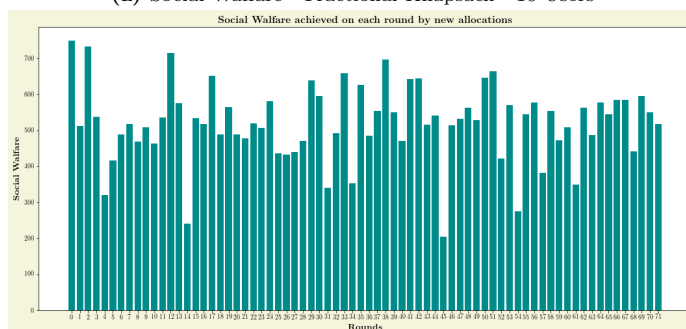
Σχολιασμός: Από τις παραπάνω προσομοιώσεις λαμβάνουμε τα αναμενόμενα με βάση τη θεωρία αποτελέσματα. Παρατηρούμε ότι έχουμε μεγάλη αύξηση των κερδών σε σχέση με την προσέγγιση μεγιστοποίησης κοινωνικής ωφέλειας ιδιαίτερα στην περίπτωση του unlimited supply, γεγονός που αποδεικνύει το πόσο καλά

λειτουργούν οι τυχαίες δημοπρασίες. Η κοινωνική ωφέλεια παραμένει σε υψηλά επίπεδα. Η χρησιμοποίηση CPU όπως αναμενόταν αυξάνεται καθώς αυξάνονται και οι χρήστες καθώς σε περιπτώσεις unlimited supply σημαίνει πως πολλές εικονικές μηχανές δεν κατανέμονται σε χρήστες, ενώ σε περιπτώσεις limited supply σχεδόν όλες οι εικονικές μηχανές κατανέμονται σε χρήστες.

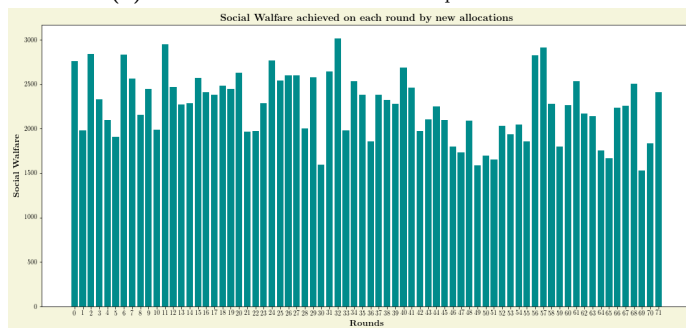
Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

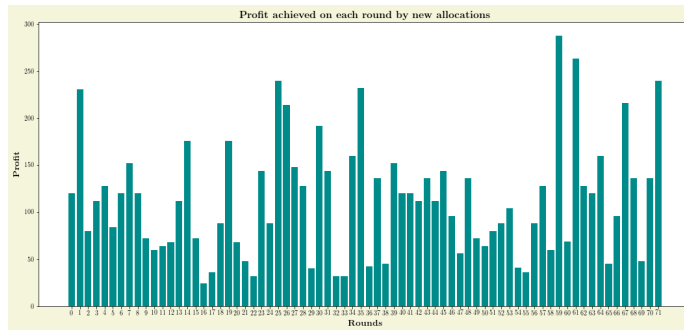


(b) Social Welfare - Fractional Knapsack - 20 Users

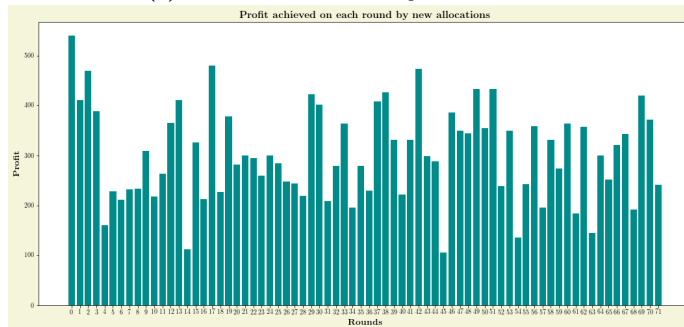


(c) Social Welfare - Fractional Knapsack - 50 Users

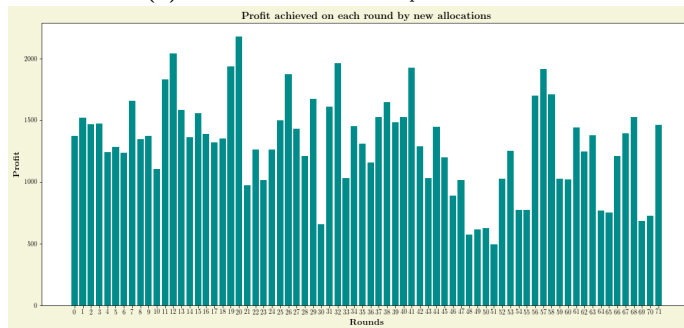
Εικόνα 6.24: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

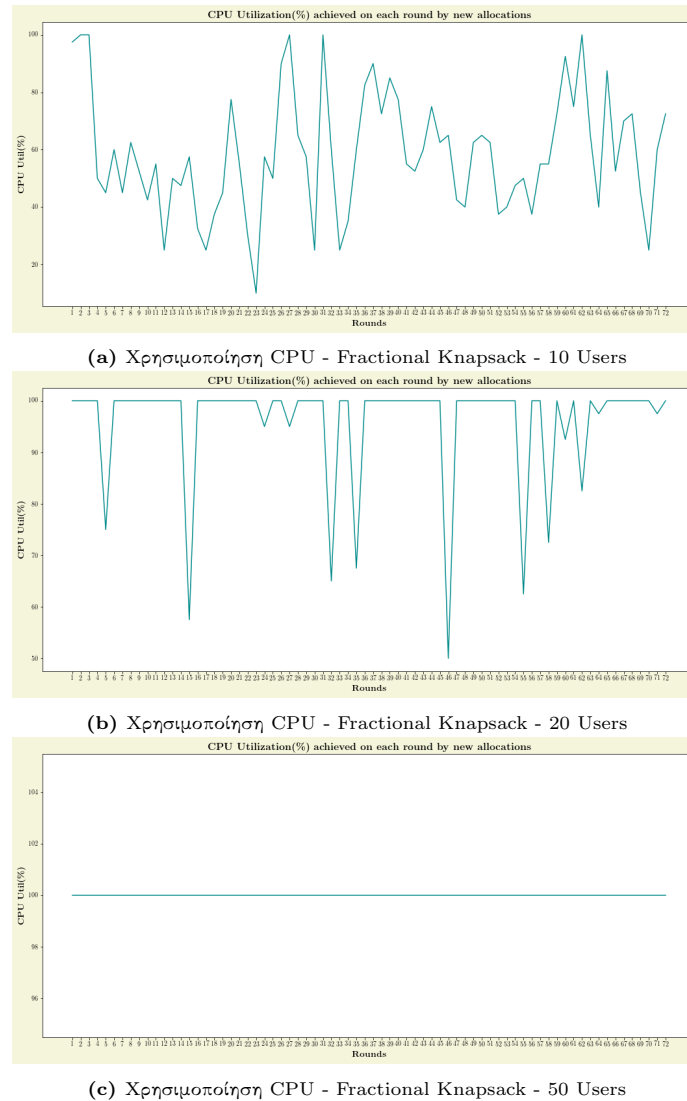


(b) Έσοδα - Fractional Knapsack - 20 Users



(c) Έσοδα - Fractional Knapsack - 50 Users

Εικόνα 6.25: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



Εικόνα 6.26: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 8150 | 15844.06 | 22.01 |
| 20 | 21780.2 | 37438.44 | 26 |
| 50 | 94105.2 | 162175.46 | 45.05 |

Πίνακας 6.11: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Προσεγγιστικού Αλγορίθμου και Τυχαίων Δημοπρασιών

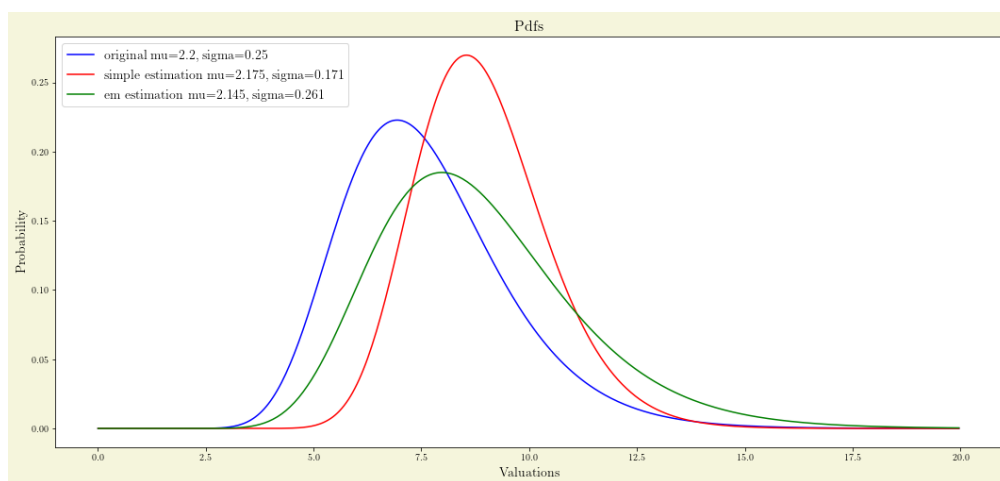
Σχολιασμός: Παρατηρούμε παρόμοια αποτελέσματα με αυτά του binary προβλήματος με μικρές αποκλίσεις.

6.4 Αποτελέσματα Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας

Στην ενότητα αυτή θα παραθέσουμε τα αποτελέσματα των προσομοιώσεων μας για την προσέγγιση της μεγιστοποίησης εσόδων εκτιμώντας κατανομές προσωπικής αξίας όπως αναλύθηκε στην ενότητα 5.3. Αρχικά, θα παρουσιάσουμε τα αποτελέσματα για την πρώτη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου όπως αναφέρθηκε και προηγουμένως, ενώ στη συνέχεια θα παρουσιάσουμε τα αποτελέσματα για τη δεύτερη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές βασίζονται στο Google Trace. Ξεκινάμε παραθέτοντας τα αποτελέσματα της πρώτης προσέγγισης.

6.4.1 Πρώτη Προσέγγιση

Στην πρώτη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου. Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Ξεκινάμε παραθέτοντας τα αποτελέσματα σχετικά με την εκτίμηση παραμέτρων που πραγματοποιήσαμε χρησιμοποιώντας τις τεχνικές των ενοτήτων 4.1 και 4.3 θεωρώντας πως κάθε χρήστης ακολουθεί την ίδια κατανομή προσωπικής αξίας. Θεωρούμε πως χάνεται ένα 15% των χαμηλότερων προσφορών (πολύ καλά, ακόμα καλύτερα, αποτελέσματα λαμβάνουμε και για άλλες τιμές). Παρακάτω δίνονται σε διαγράμματα οι επιμέρους εκτιμήσεις και οι αποστάσεις Kullback-Leibler από την πραγματική κατανομή.



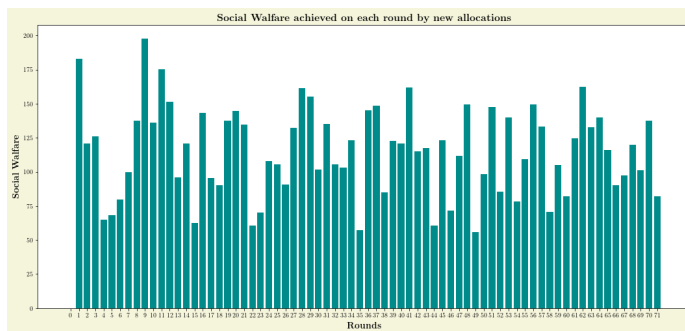
Εικόνα 6.27: Εκτιμήσεις της κατανομής προσωπικής αξίας των χρηστών

| | Μέση τιμή μ | Τυπική απόκλιση σ | Απόσταση Kullback-Leibler |
|--|-----------------|--------------------------|---------------------------|
| Απλή Προσέγγιση Εκτίμησης Μέγιστης Πιθανοφάνειας | 2.174610 | 0.170541 | 0.716046 |
| Εκτίμηση με Αλγόριθμο EM | 2.144711 | 0.261355 | 0.154773 |

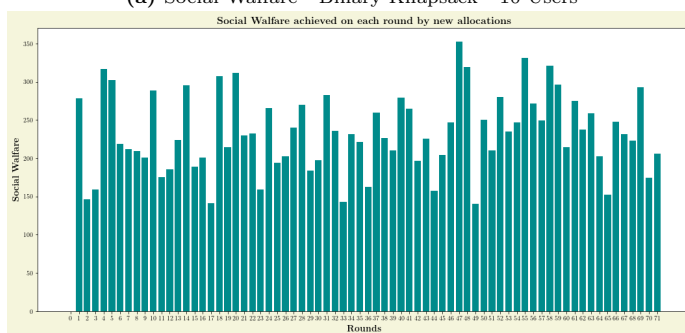
Πίνακας 6.12: Αποτελέσματα Εκτίμησης της Κατανομής Προσωπικής Αξίας των Χρηστών

Παρατηρούμε πως τόσο σχηματικά όσο και με βάση την απόσταση Kullback-Leibler η εκτίμηση που πραγματοποιήθηκε με τον αλγόριθμο Expectation-Maximization δίνει καλύτερα αποτελέσματα σε σχέση με την εκτίμηση μέγιστης πιθανοφάνειας. Για το λόγο αυτό, από εδώ και πέρα θα θεωρήσουμε πως οι χρήστες έχουν κατανομές προσωπικής αξίας που ακολουθούν $\text{Lognormal}(2.140843, 0.193239^2)$.

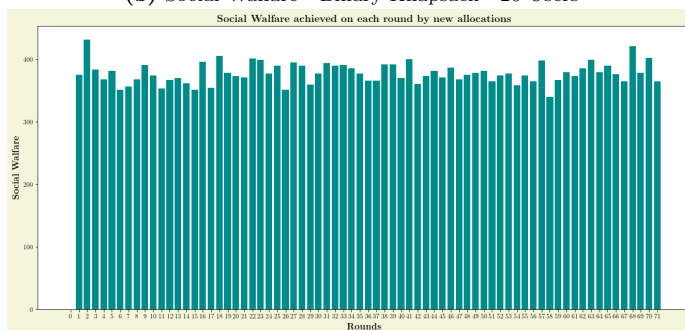
Ξεινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο και τη χρησιμοποίηση CPU ανά γύρο.



(a) Social Welfare - Binary Knapsack - 10 Users

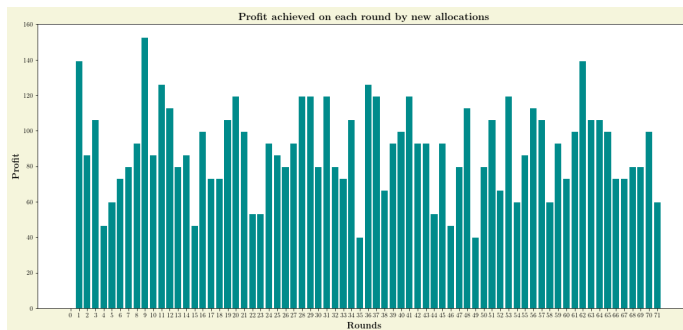


(b) Social Welfare - Binary Knapsack - 20 Users

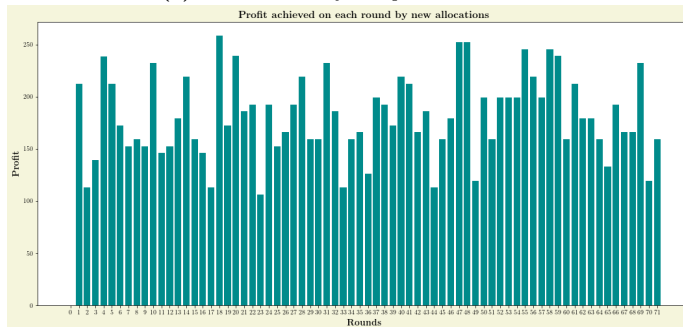


(c) Social Welfare - Binary Knapsack - 50 Users

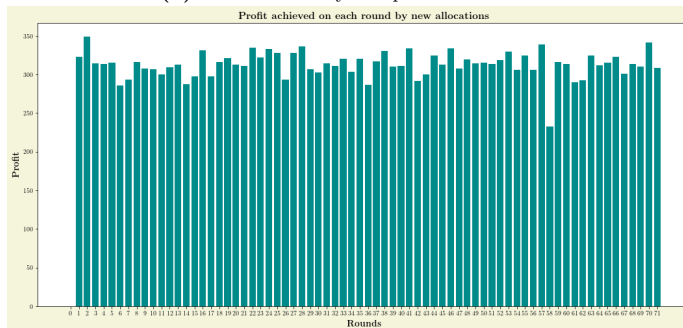
Εικόνα 6.28: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Έσοδα - Binary Knapsack - 10 Users

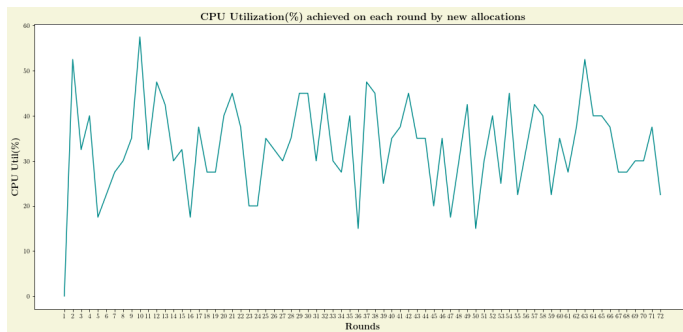


(b) Έσοδα - Binary Knapsack - 20 Users

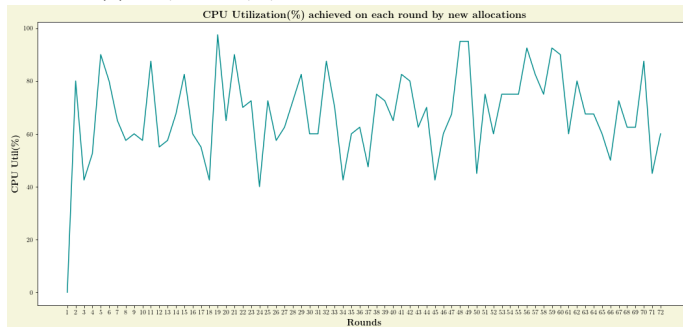


(c) Έσοδα - Binary Knapsack - 50 Users

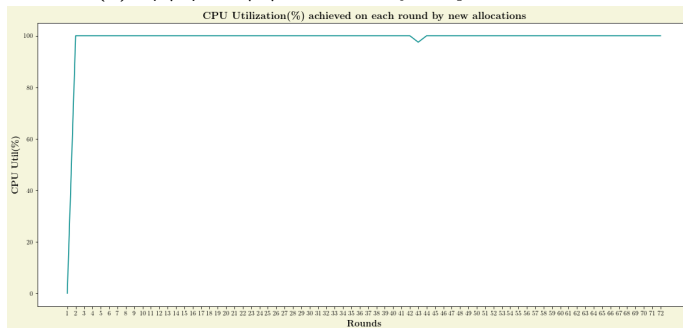
Εικόνα 6.29: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτησικές εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

Εικόνα 6.30: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

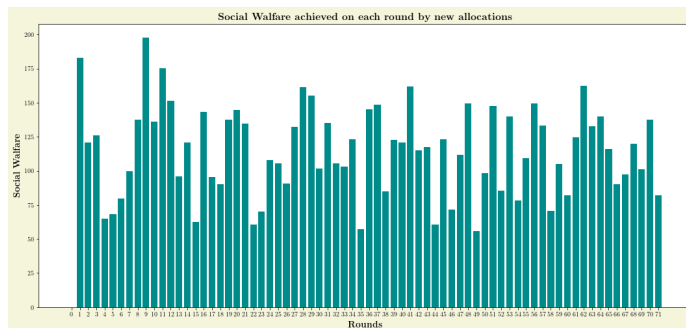
Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 6348.26 | 8171.47 | 11.35 |
| 20 | 12855.73 | 16516.74 | 11.47 |
| 50 | 22218.24 | 26795.18 | 7.44 |

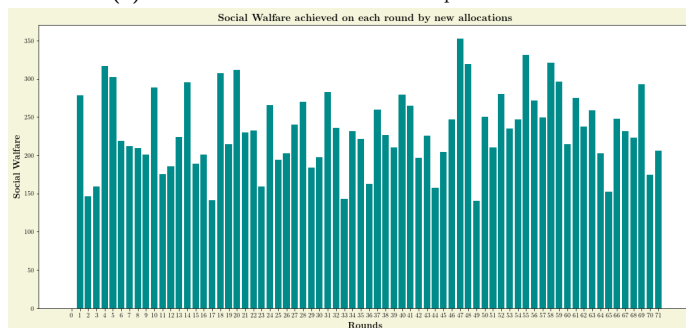
Πίνακας 6.13: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας.

Σχολιασμός: Από τις παραπάνω προσομοιώσεις λαμβάνουμε τα αναμενόμενα με βάση τη θεωρία αποτελέσματα. Σε σχέση με την προηγούμενη προσέγγιση παρατηρούμε πως έχουμε συνολικά αύξηση των εσόδων ιδιαίτερα μάλιστα στην περίπτωση του unlimited supply. Η κοινωνική ωφέλεια σε σχέση με την προηγούμενη προσέγγιση είναι χαμηλότερη όπως αναμενόταν. Σε ότι αφορά τη χρησιμοποίηση CPU, αυτή όπως αναμενόταν αυξάνεται καθώς αυξάνονται και οι χρήστες.

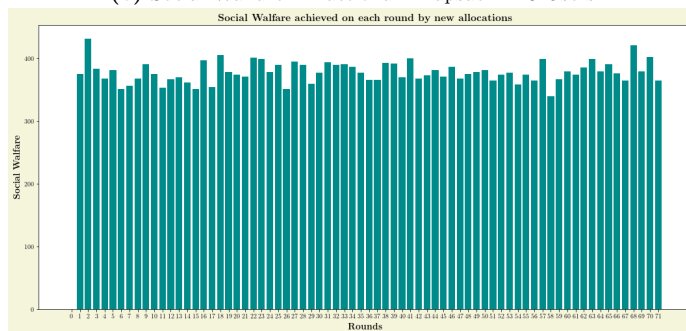
Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

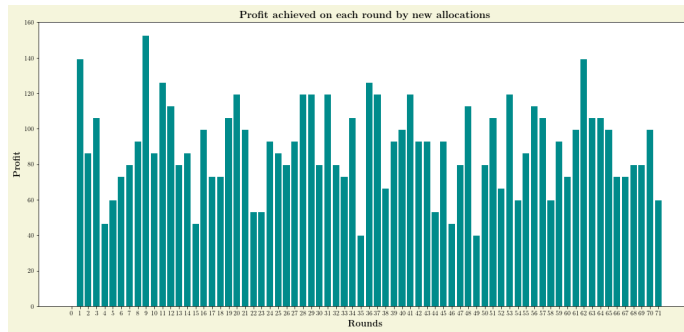


(b) Social Welfare - Fractional Knapsack - 20 Users

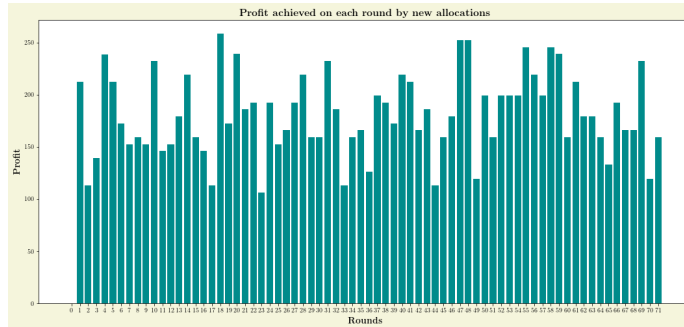


(c) Social Welfare - Fractional Knapsack - 50 Users

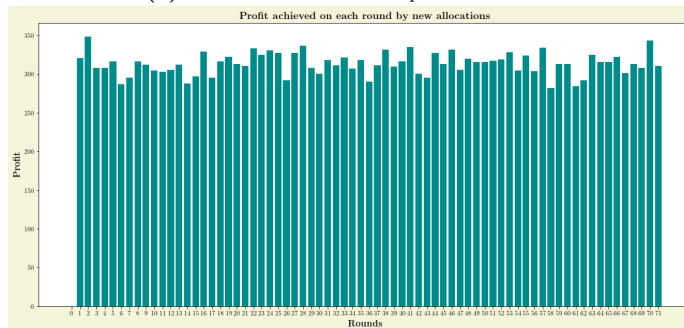
Εικόνα 6.31: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

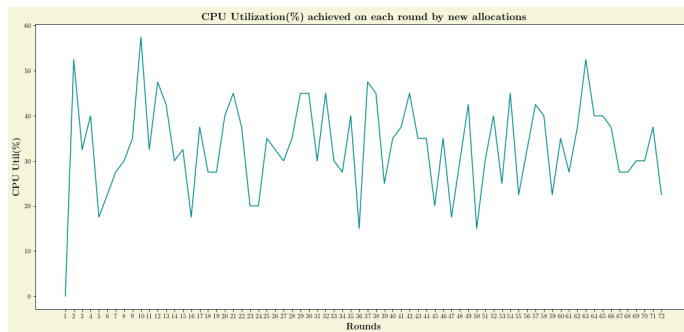


(b) Έσοδα - Fractional Knapsack - 20 Users

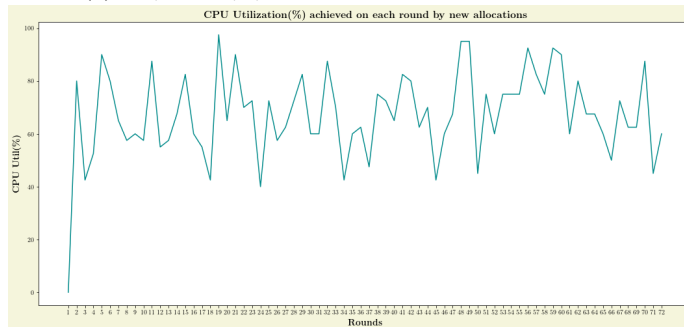


(c) Έσοδα - Fractional Knapsack - 50 Users

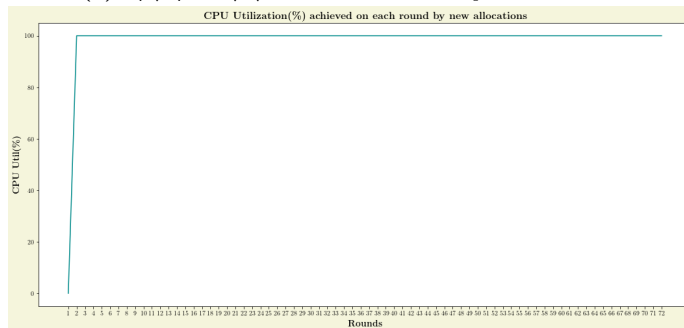
Εικόνα 6.32: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Χρησιμοποίηση CPU - Fractional Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Fractional Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Fractional Knapsack - 50 Users

Εικόνα 6.33: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

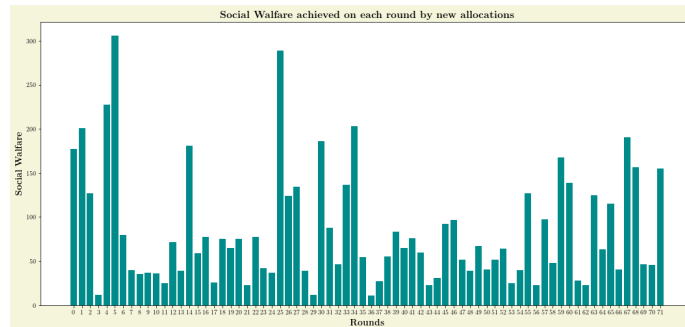
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 6348.26 | 8171.47 | 11.35 |
| 20 | 12855.73 | 16516.74 | 11.47 |
| 50 | 22263.16 | 26815.97 | 7.45 |

Πίνακας 6.14: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας.

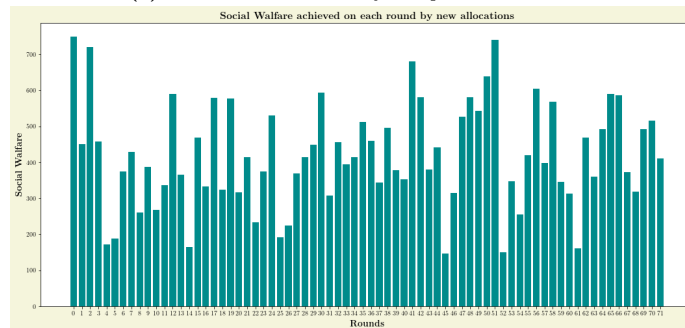
Σχολιασμός: Παρατηρούμε παρόμοια αποτελέσματα με αυτά του binary προβλήματος με μικρές αποκλίσεις.

6.4.2 Δεύτερη Προσέγγιση

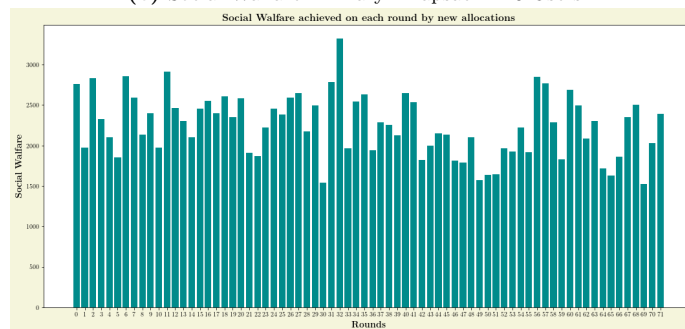
Στη δεύτερη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές παράγονται όπως αναφέρθηκε στην ενότητα 6.1.3 με βάση το Google Trace. Πραγματοποιούμε και πάλι εκτίμηση παραμέτρων χρησιμοποιώντας τον αλγόριθμο Expectation-Maximization θεωρώντας πως και πάλι οι χρήστες ακολουθούν την ίδια κατανομή. Προκύπτει πως οι χρήστες έχουν κατανομές προσωπικής αξίας που ακολουθούν $\text{Lognormal}(2.661505, 0.264519^2)$. Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Ξεκινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο και τη χρησιμοποίηση CPU ανά γύρο.



(a) Social Welfare - Binary Knapsack - 10 Users

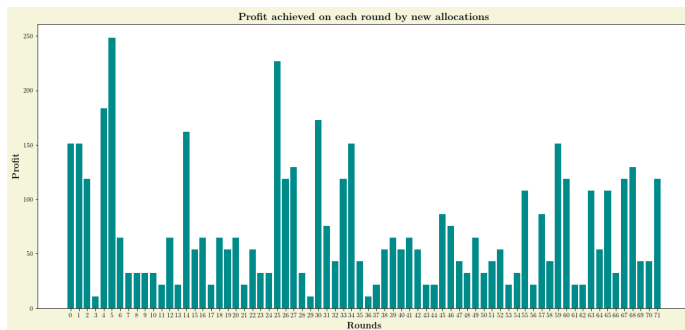


(b) Social Welfare - Binary Knapsack - 20 Users

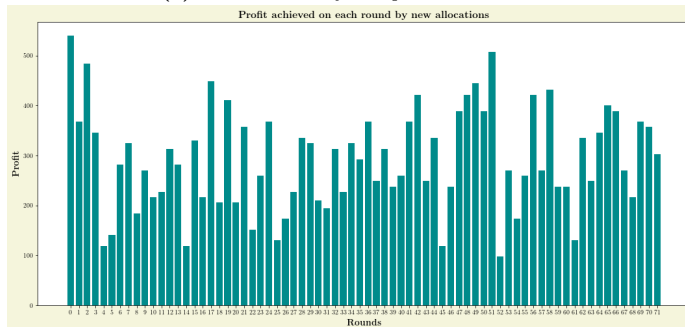


(c) Social Welfare - Binary Knapsack - 50 Users

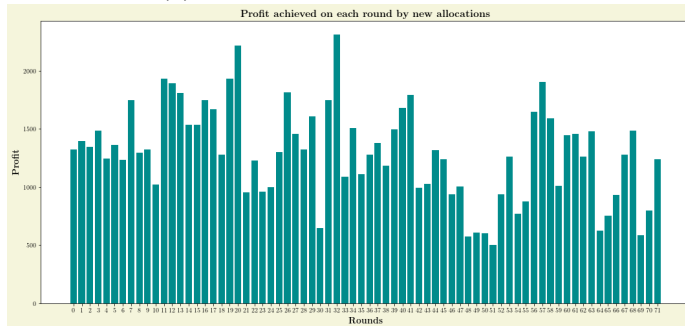
Εικόνα 6.34: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Έσοδα - Binary Knapsack - 10 Users

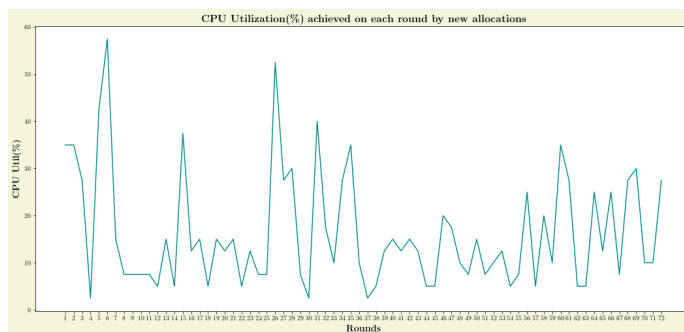


(b) Έσοδα - Binary Knapsack - 20 Users

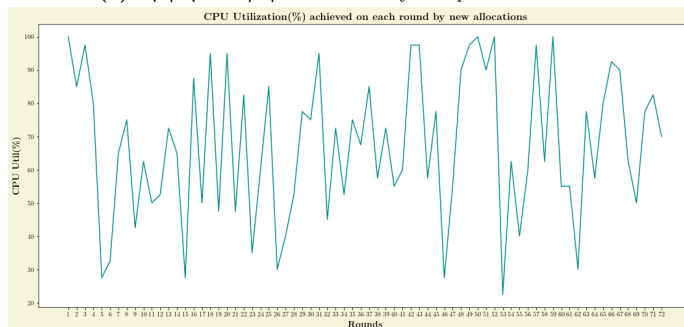


(c) Έσοδα - Binary Knapsack - 50 Users

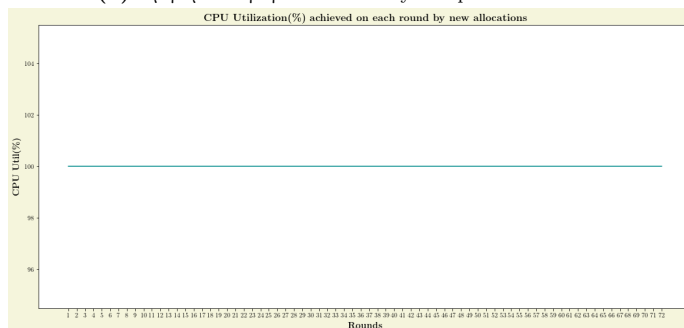
Εικόνα 6.35: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

Εικόνα 6.36: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

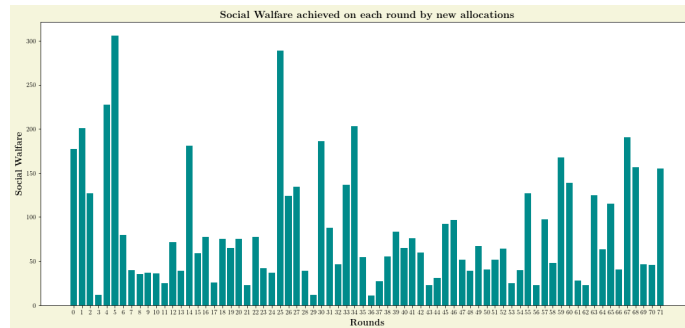
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 5100.54 | 6052.59 | 8.41 |
| 20 | 20978.3 | 30150.74 | 20.94 |
| 50 | 93339.73 | 161845.75 | 44.95 |

Πίνακας 6.15: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εχτιμώντας Κατανομές Προσωπικής Αξίας.

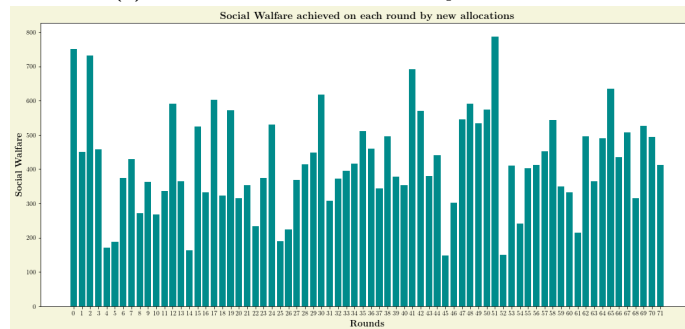
Σχολιασμός: Παρατηρώντας τις προσομοιώσεις βλέπουμε πως σε σχέση με την προηγούμενη προσέγγιση παρατηρούμε πως έχουμε μείωση των εσόδων για 10 και 50 χρήστες ενώ μικρή αύξηση για την περίπτωση των 20 χρηστών. Η κοινωνική ωφέλεια σε σχέση με την προηγούμενη προσέγγιση είναι χαμηλότερη για την περίπτωση

των 10 και 20 χρηστών ενώ λίγο ανεβασμένη στην περίπτωση των 50 χρηστών. Τα αποτελέσματα αυτά οφείλονται στο ότι στη συγκεκριμένη προσομοίωση υποθέσαμε πως όλοι οι χρήστες ακολουθούν την ίδια κατανομή που προφανώς δεν ισχύει. Για την επίλυση του προβλήματος παρουσιάζουμε στην επόμενη ενότητα την υβριδική προσέγγιση με χρήση νευρωνικών δικτύων και τυχαίων δημοπρασιών. Σε ότι αφορά τη χρησιμοποίηση CPU, αυτή όπως αναμενόταν παρουσιάζει τα γνωστά αποτελέσματα.

Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

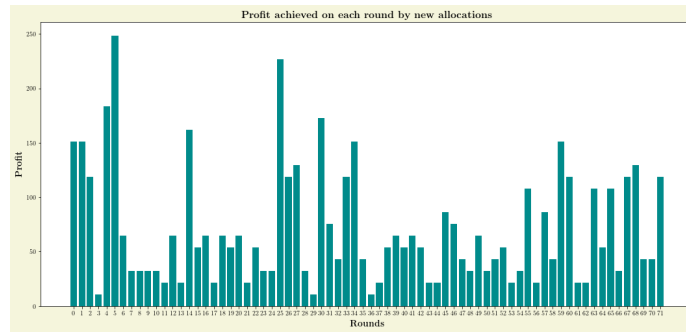


(b) Social Welfare - Fractional Knapsack - 20 Users

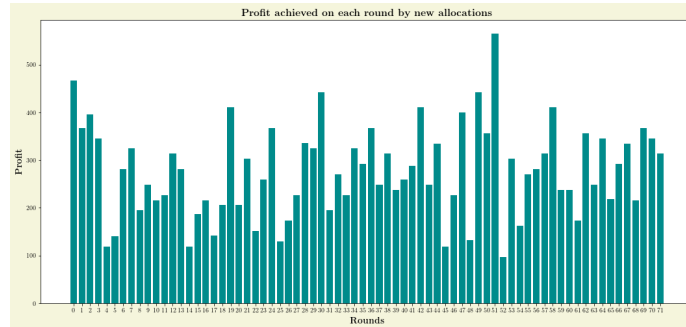


(c) Social Welfare - Fractional Knapsack - 50 Users

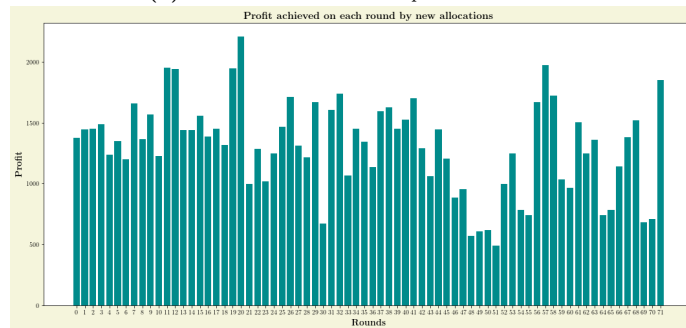
Εικόνα 6.37: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

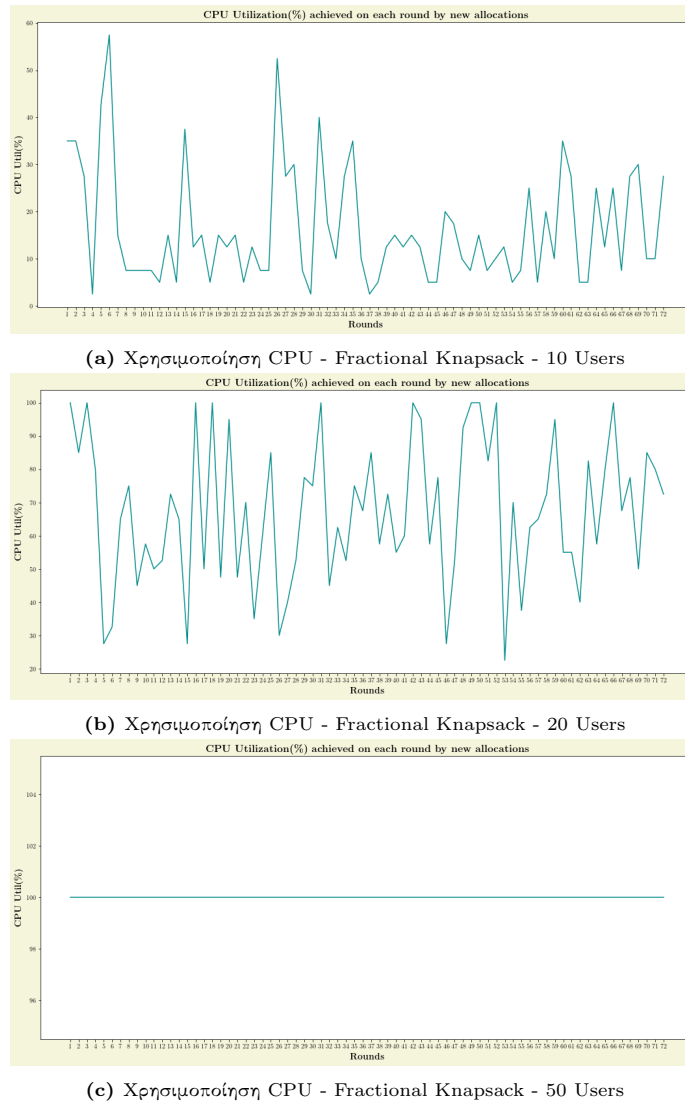


(b) Έσοδα - Fractional Knapsack - 20 Users



(c) Έσοδα - Fractional Knapsack - 50 Users

Εικόνα 6.38: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



Εικόνα 6.39: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 5100.54 | 6052.59 | 8.41 |
| 20 | 19899.94 | 30112.8 | 20.91 |
| 50 | 94045.42 | 162175.46 | 45.05 |

Πίνακας 6.16: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων Εκτιμώντας Κατανομές Προσωπικής Αξίας.

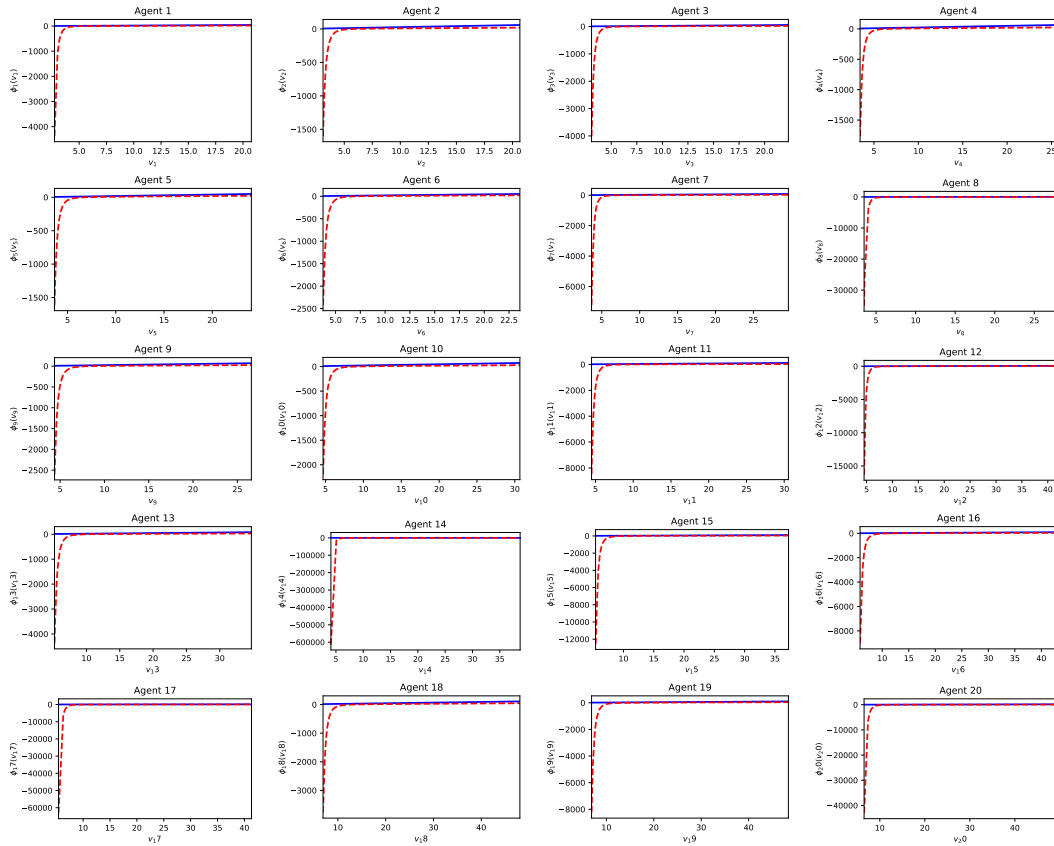
Σχολιασμός: Παρατηρούμε παρόμοια αποτελέσματα με αυτά του binary προβλήματος με μικρές αποκλίσεις.

6.5 Αποτελέσματα Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων

Στην ενότητα αυτή θα παραθέσουμε τα αποτελέσματα των προσομοιώσεων μας για την προσέγγιση της μεγιστοποίησης εσόδων με χρήση νευρωνικών δικτύων όπως αναλύθηκε στην ενότητα 5.4. Αρχικά, θα παρουσιάσουμε τα αποτελέσματα για την πρώτη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου όπως αναφέρθηκε και προηγουμένως, ενώ στη συνέχεια θα παρουσιάσουμε τα αποτελέσματα για τη δεύτερη προσέγγιση στην οποία οι αιτήσεις των χρηστών για εικονικές μηχανές βασίζονται στο Google Trace. Ξεκινάμε παραθέτοντας τα αποτελέσματα της πρώτης προσέγγισης.

6.5.1 Πρώτη Προσέγγιση

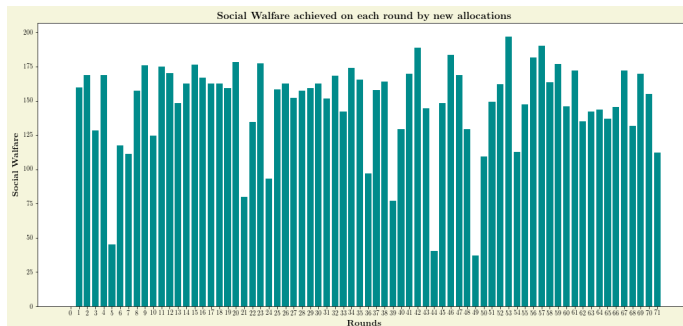
Στην πρώτη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές ακολουθούν συνάρτηση ημίτονου. Χρησιμοποιούμε την προσέγγιση που χρησιμοποιήθηκε στην ενότητα 5.4. Εκπαιδεύουμε το νευρωνικό μας δίκτυο χρησιμοποιώντας τις τιμές $J = 20, K = 25$ για 10 χρήστες, $J = 30, K = 35$ για 20 χρήστες και $J = 50, K = 55$ για 50 χρήστες. Παραθέτουμε τα διαγράμματα που μας δείχνουν τις προσεγγίσεις του νευρωνικού μας δικτύου για τα virtual valuations των πρώτων 20 χρηστών.



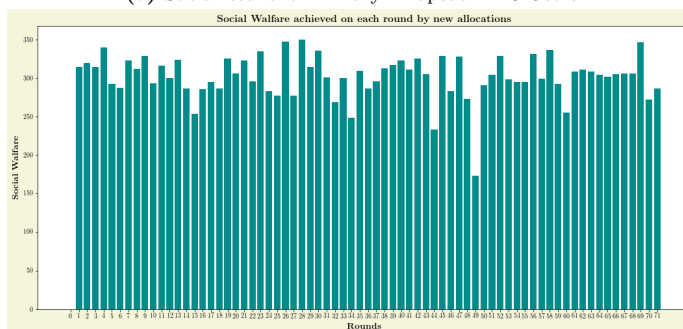
Εικόνα 6.40: Εκτιμήσεις των virtual valuations των πρώτων 20 χρηστών από το νευρωνικό δίκτυο

Παρατίθενται τα διαγράμματα μόνο για τους πρώτους 20 χρήστες για εξοικονόμηση χώρου καθώς όλα τα διαγράμματα δίνουν παραπλήσια αποτελέσματα. Με μπλε γραμμή βλέπουμε την εκτίμησή μας και με κόκκινη διακεκομμένη το πραγματικό virtual valuation. Βλέπουμε πως οι προσεγγίσεις είναι πάρα πολύ καλές καθώς στο μεγαλύτερο μέρος ταυτίζονται με την πραγματική κατανομή. Χάνεται μόνο ένα μικρό κομμάτι στην αρχή λόγω πολύ απότομης καμπής.

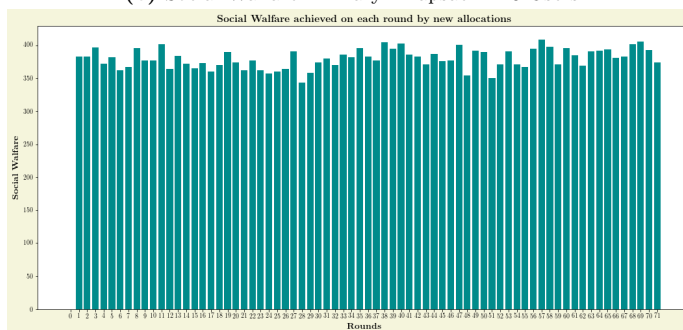
Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Ξεκινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο και τη χρησιμοποίηση CPU ανά γύρο.



(a) Social Welfare - Binary Knapsack - 10 Users

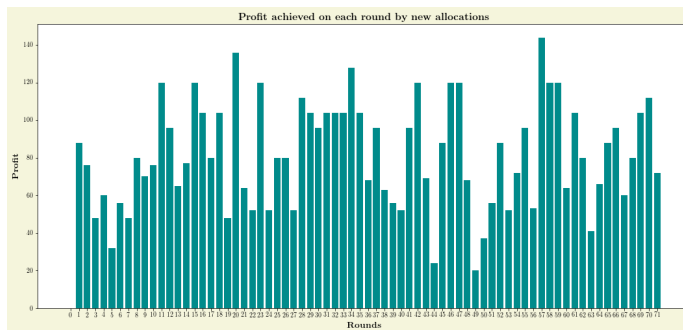


(b) Social Welfare - Binary Knapsack - 20 Users

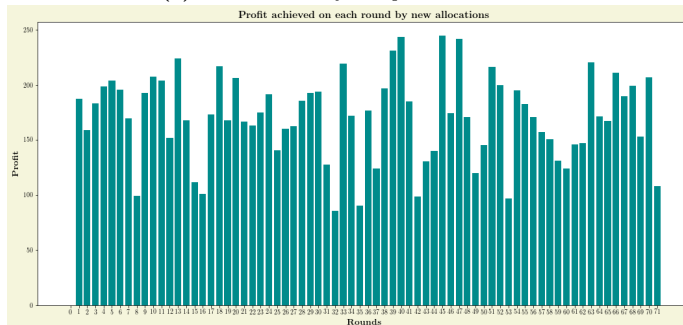


(c) Social Welfare - Binary Knapsack - 50 Users

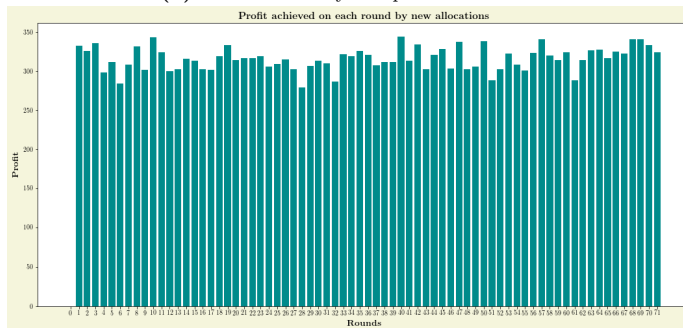
Εικόνα 6.41: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Έσοδα - Binary Knapsack - 10 Users

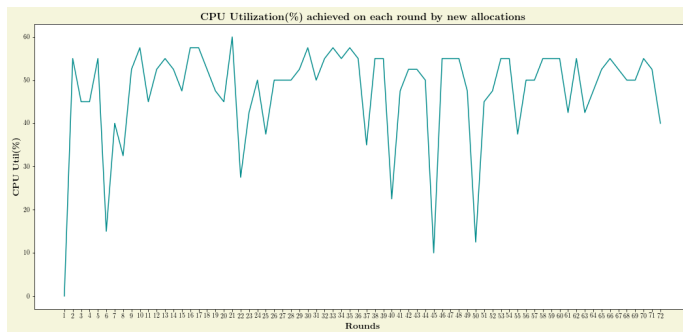


(b) Έσοδα - Binary Knapsack - 20 Users

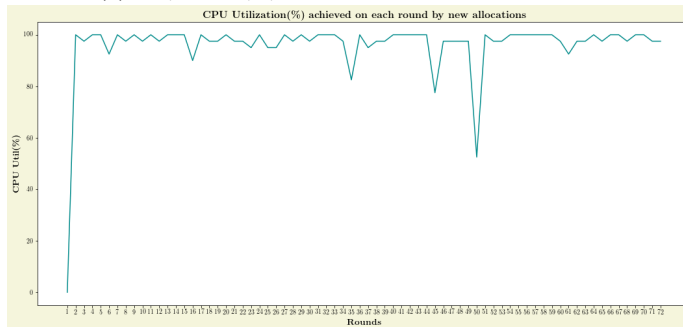


(c) Έσοδα - Binary Knapsack - 50 Users

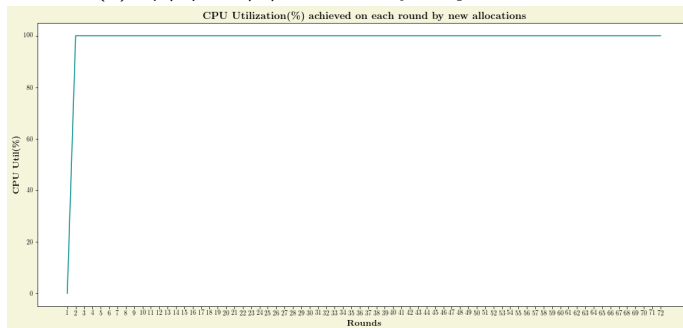
Εικόνα 6.42: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

Εικόνα 6.43: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

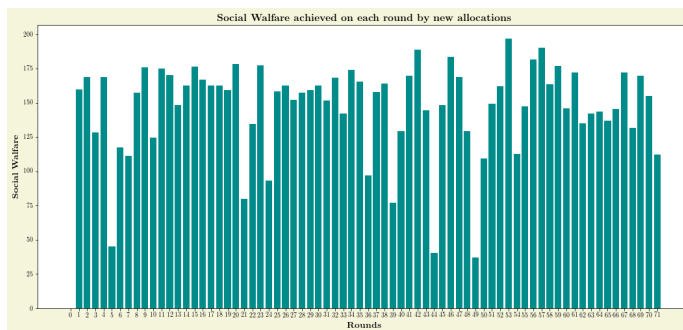
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 5805 | 10446.53 | 14.50 |
| 20 | 12125.49 | 21438.20 | 14.88 |
| 50 | 22427.03 | 26969.65 | 7.49 |

Πίνακας 6.17: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων

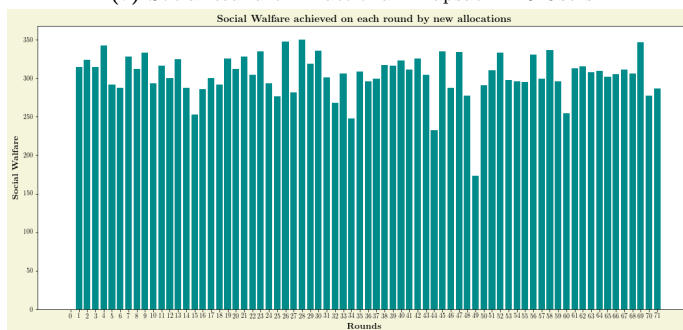
Σχολιασμός: Σε σχέση με τις δύο προηγούμενες προσεγγίσεις μεγιστοποίησης εσόδων παρατηρούμε πως έχουμε σχεδόν παραπλήσια έσοδα στο πρόβλημα του unlimited supply για 10 και 20 χρήστες ενώ στο πρόβλημα του limited supply με 50 χρήστες έχουμε τα μεγαλύτερα έσοδα. Η κοινωνική ωφέλεια σε σχέση

με την προσέγγιση των προσεγγιστικών αλγορίθμων είναι σχεδόν ίδια ενώ σε σχέση με την προσέγγιση της εκτίμησης προσωπικής αξίας χρηστών είναι μεγαλύτερη ιδιαίτερα για 10 και 20 χρήστες. Σε ότι αφορά τη χρησιμοποίηση CPU, αυτή όπως αναμενόταν αυξάνεται καθώς αυξάνονται και οι χρήστες.

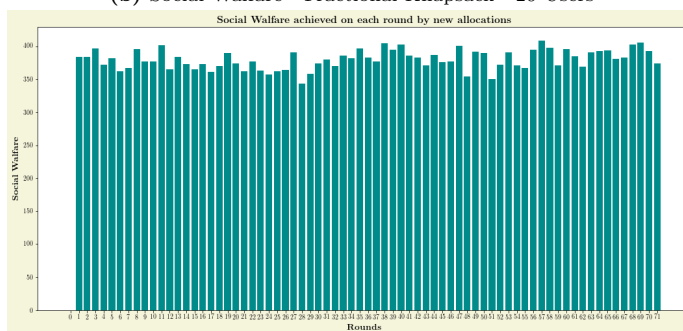
Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

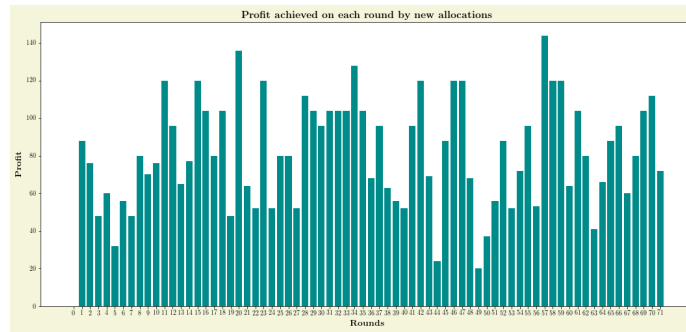


(b) Social Welfare - Fractional Knapsack - 20 Users

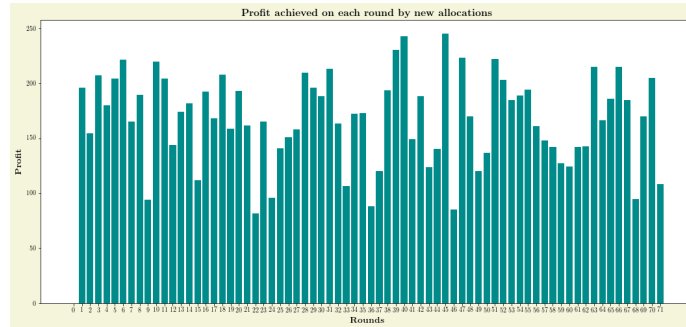


(c) Social Welfare - Fractional Knapsack - 50 Users

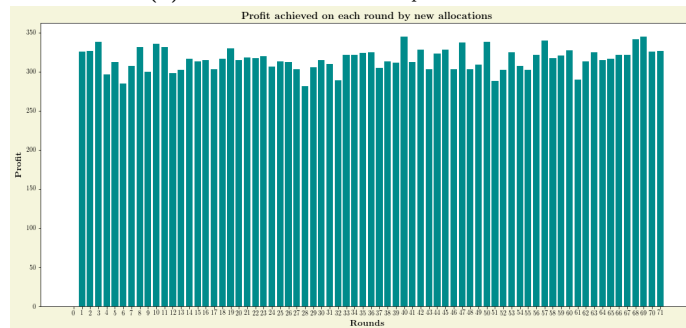
Εικόνα 6.44: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

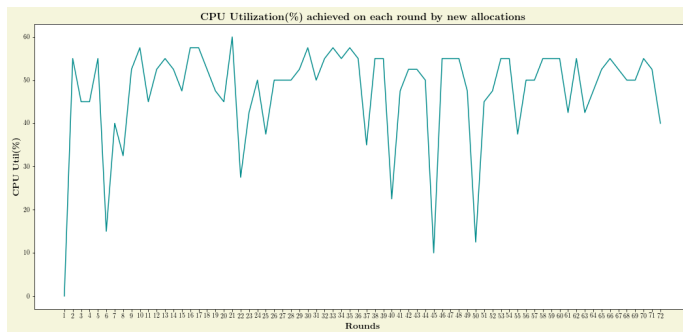


(b) Έσοδα - Fractional Knapsack - 20 Users

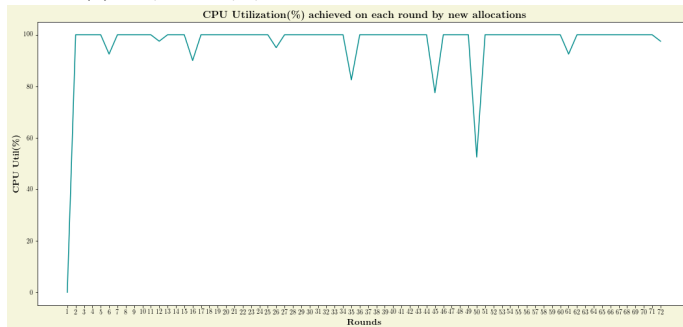


(c) Έσοδα - Fractional Knapsack - 50 Users

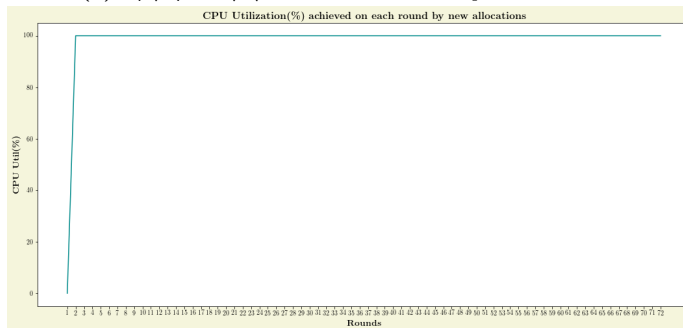
Εικόνα 6.45: Διαγράμματα εσόδων ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Χρησιμοποίηση CPU - Fractional Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Fractional Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Fractional Knapsack - 50 Users

Εικόνα 6.46: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για ημιτονικές αιτήσεις εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 5805 | 10446.53 | 14.50 |
| 20 | 11923.63 | 21595.65 | 15 |
| 50 | 22424.56 | 26988.89 | 7.5 |

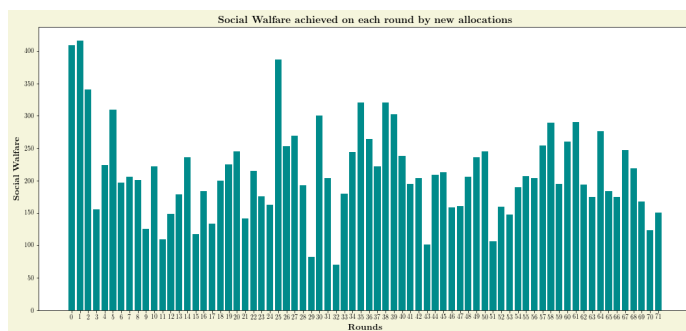
Πίνακας 6.18: Αποτελέσματα Προσομοίωσης Πρώτης Προσέγγισης για Fractinal Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων

Σχολιασμός: Παρατηρούμε παρόμοια αποτελέσματα με αυτά του binary προβλήματος με μικρές αποκλίσεις.

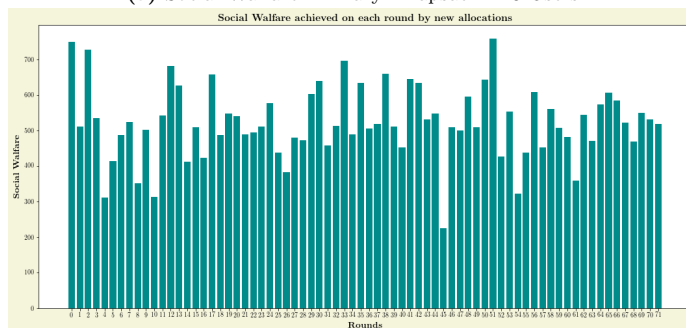
6.5.2 Δεύτερη Προσέγγιση

Στη δεύτερη προσέγγιση μας οι αιτήσεις των χρηστών για εικονικές μηχανές παράγονται με βάση το Google Trace. Χρησιμοποιούμε και πάλι το νευρωνικό μας δίκτυο αναγνωρίζοντας αυτή τη φορά πως το δίκτυο έχει εκπαιδευτεί για διαφορετική κατανομή ανά χρήστη. Οι προσεγγίσεις virtual valuations είναι παρόμοιες με αυτές της ενότητας 6.4. Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές.

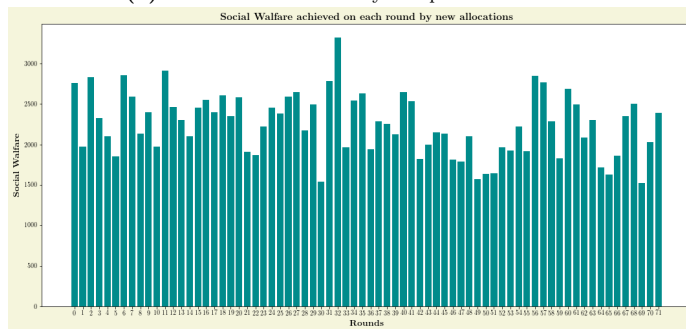
Παρακάτω παρατίθενται τα σχετικά διαγράμματα ομαδοποιημένα ανά κατηγορία για 10, 20 και 50 χρήστες, ενώ στη συνέχεια παρατίθενται σε μορφή πίνακα οι υπόλοιπες μετρικές. Ξεκινάμε παραθέτοντας τα αποτελέσματα για την περίπτωση του προβλήματος Knapsack 0-1 (binary) σχετικά με την κοινωνική ωφέλεια ανά γύρο, τα έσοδα ανά γύρο και τη χρησιμοποίηση CPU ανά γύρο.



(a) Social Welfare - Binary Knapsack - 10 Users

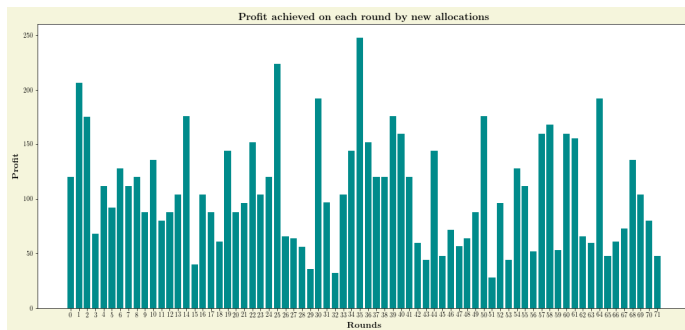


(b) Social Welfare - Binary Knapsack - 20 Users

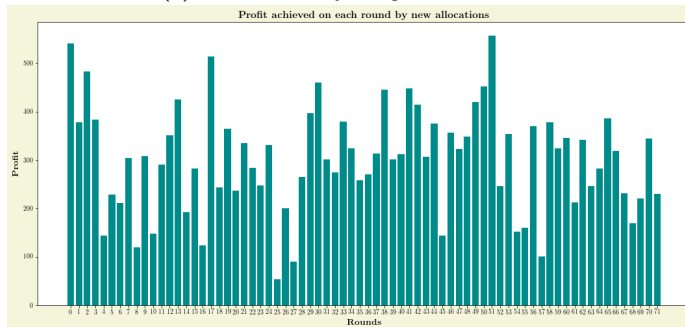


(c) Social Welfare - Binary Knapsack - 50 Users

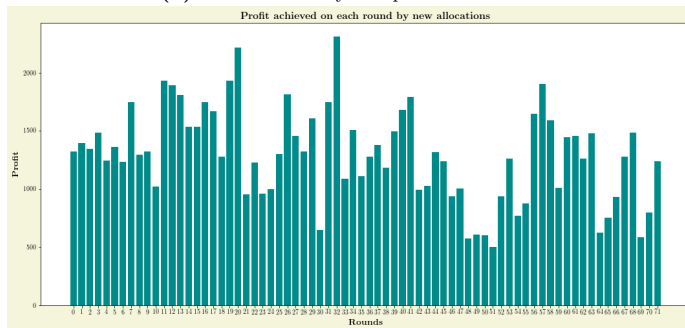
Εικόνα 6.47: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Έσοδα - Binary Knapsack - 10 Users

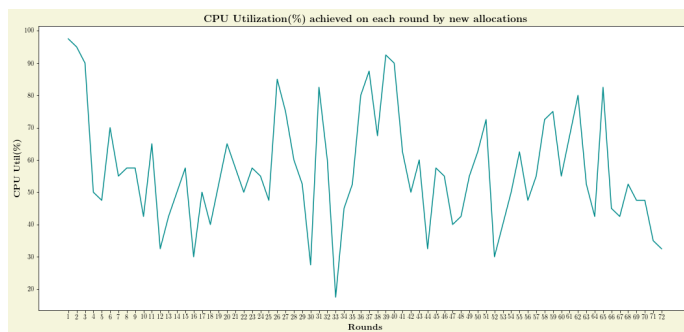


(b) Έσοδα - Binary Knapsack - 20 Users

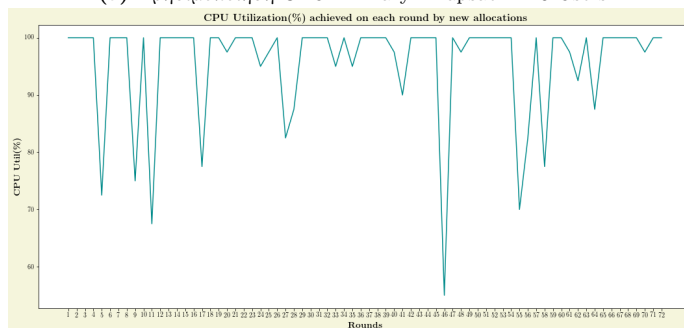


(c) Έσοδα - Binary Knapsack - 50 Users

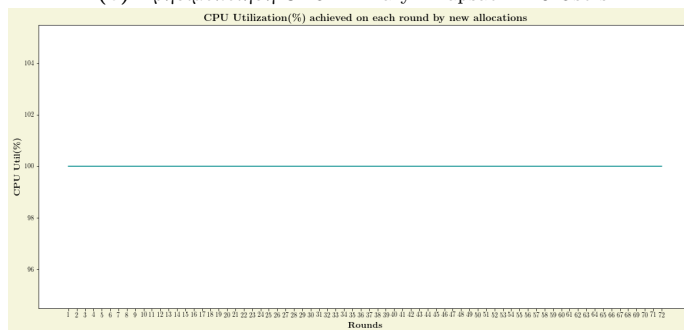
Εικόνα 6.48: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.



(a) Χρησιμοποίηση CPU - Binary Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Binary Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Binary Knapsack - 50 Users

Εικόνα 6.49: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του binary (0-1) knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

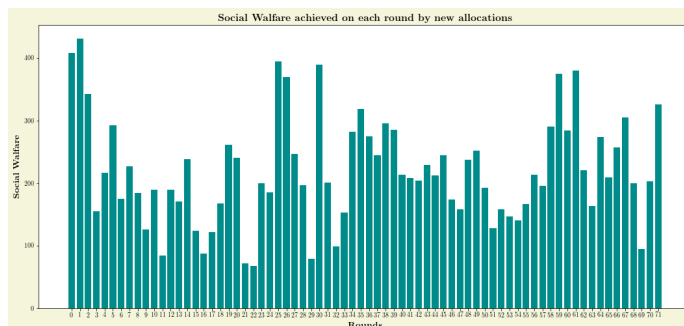
| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 7690.9 | 15273.09 | 21.21 |
| 20 | 21679.69 | 37573.52 | 26.09 |
| 50 | 94045.42 | 162175.46 | 45.05 |

Πίνακας 6.19: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Binary Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων

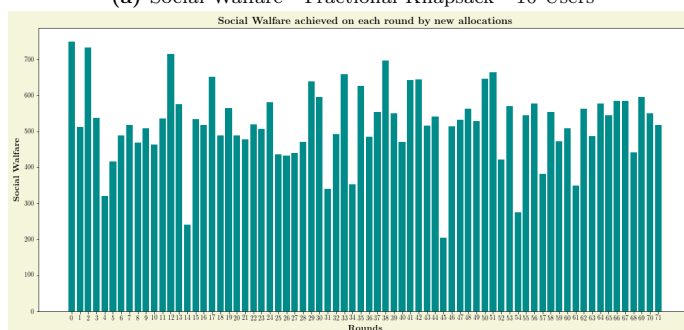
Σχολιασμός: Σε σχέση με τις δύο προηγούμενες προσεγγίσεις μεγιστοποίησης εσόδων παρατηρούμε πως έχουμε υψηλότερα έσοδα από κάθε άλλη περίπτωση. Η κοινωνική ωφέλεια επίσης είναι σχεδόν υψηλότερη

από κάθε άλλη περίπτωση. Η χρησιμοποίηση CPU όπως αναμενόταν αυξάνεται καθώς αυξάνονται και οι χρήστες.

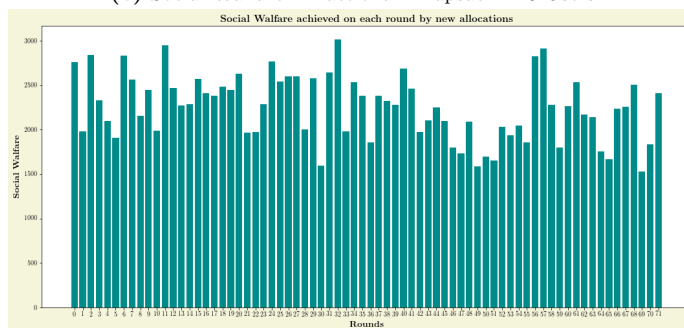
Στη συνέχεια, παρατίθενται τα ίδια αποτελέσματα για το fractional problem.



(a) Social Welfare - Fractional Knapsack - 10 Users

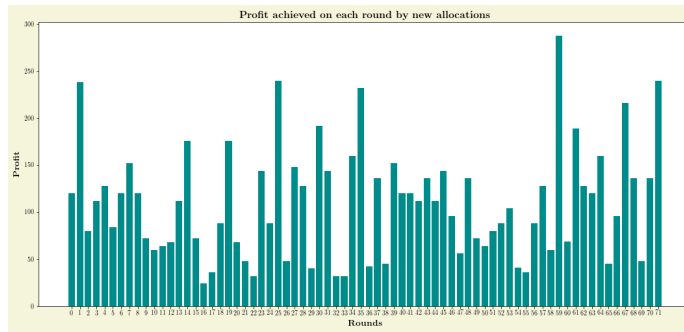


(b) Social Welfare - Fractional Knapsack - 20 Users

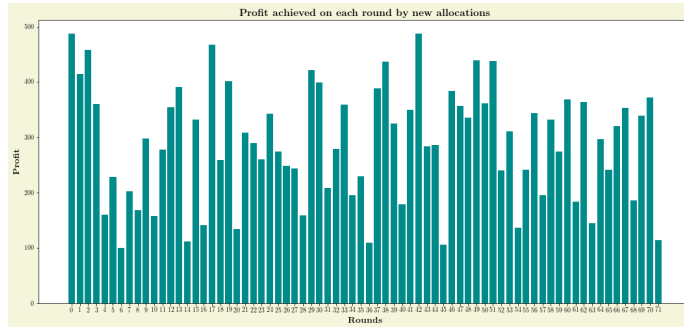


(c) Social Welfare - Fractional Knapsack - 50 Users

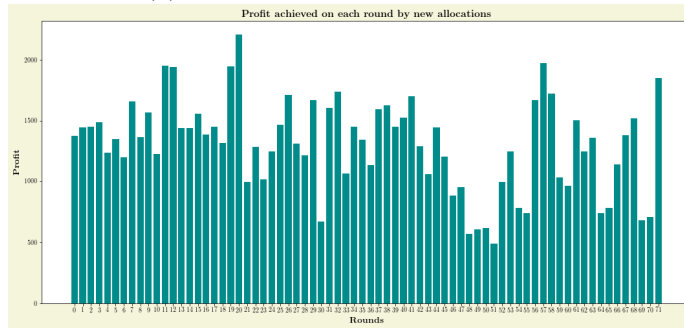
Εικόνα 6.50: Διαγράμματα κοινωνικής ωφέλειας ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Έσοδα - Fractional Knapsack - 10 Users

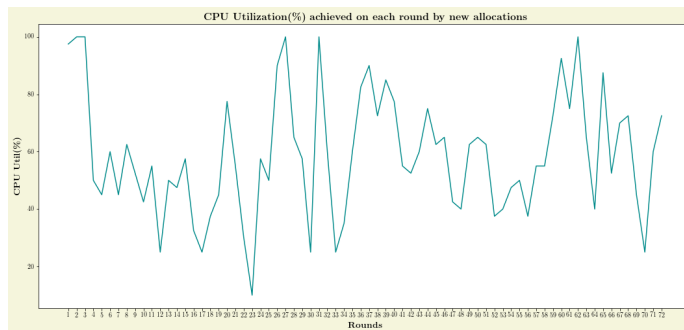


(b) Έσοδα - Fractional Knapsack - 20 Users

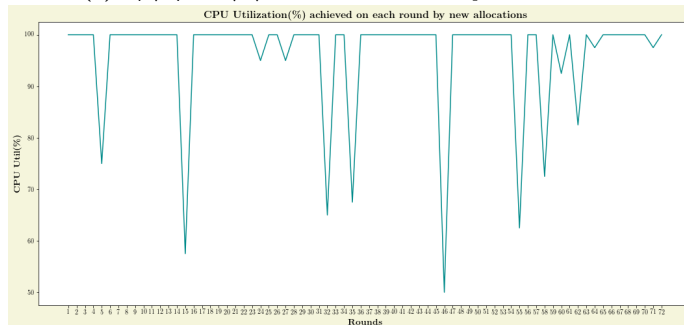


(c) Έσοδα - Fractional Knapsack - 50 Users

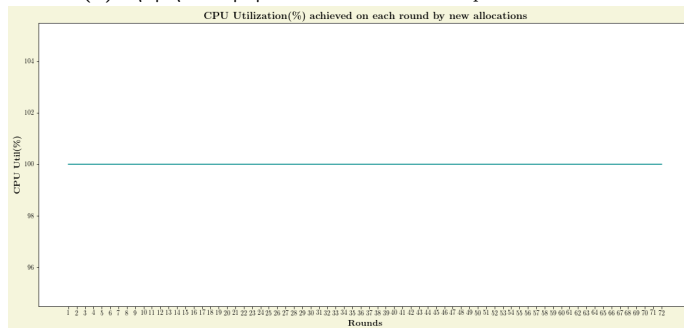
Εικόνα 6.51: Διαγράμματα εσόδων ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.



(a) Χρησιμοποίηση CPU - Fractional Knapsack - 10 Users



(b) Χρησιμοποίηση CPU - Fractional Knapsack - 20 Users



(c) Χρησιμοποίηση CPU - Fractional Knapsack - 50 Users

Εικόνα 6.52: Διαγράμματα χρησιμοποίησης CPU ανά γύρο για αιτήσεις από το Google Trace εικονικών μηχανών από χρήστες στο πρόβλημα του fractional knapsack.

Έχοντας παρουσιάσει όλες τις γραφικές παραστάσεις για τη συγκεκριμένη περίπτωση παραθέτουμε σε μορφή πίνακα τα συνολικά αποτελέσματα.

| Αριθμός Χρηστών | Συνολικά Έσοδα | Κοινωνική Ωφέλεια | Μέση Κοινωνική Ωφέλεια |
|-----------------|----------------|-------------------|------------------------|
| 10 | 7918.0 | 15844.06 | 22.01 |
| 20 | 20753.5 | 37438.44 | 26 |
| 50 | 93339.73 | 161845.75 | 44.95 |

Πίνακας 6.20: Αποτελέσματα Προσομοίωσης Δεύτερης Προσέγγισης για Fractional Knapsack στην Προσέγγιση Μεγιστοποίησης Εσόδων με Χρήση Νευρωνικών Δικτύων

Κλείνουμε το κεφάλαιο αυτό τονίζοντας πως η τελευταία προσέγγιση είναι αυτή που θα επιλέγαμε εμείς προκειμένου να αντικαταστήσει το σύστημα που χρησιμοποιεί η υπηρεσία Spot Instances. Πέρα από ότι η μέθοδος αυτή δίνει καλύτερα αποτελέσματα στο πραγματικό σύνολο δεδομένων Google Trace, σημαντικό στοιχείο το οποίο μας οδήγησε στην επιλογή της είναι το γεγονός πως μπορεί να προσαρμόζεται σε οποιαδήποτε αλλαγή της κατανομής προσωπικής αξίας του χρήστη. Επιπρόσθετα, το νευρωνικό δίκτυο που χρησιμοποιείται έχει τη δυνατότητα να εκπαιδευτεί και συνεπώς να βελτιστοποιηθεί περαιτέρω χρησιμοποιώντας ισχυρότερους υπολογιστικούς πόρους από ότι χρησιμοποιήσαμε εμείς. Προκειμένου να συλλεχθούν αυτά τα απαραίτητα δεδομένα για τους χρήστες θα μπορούσαμε για αρχή να χρησιμοποιήσουμε την προσέγγιση επίλυσης με τη χρήση προσεγγιστικού αλγορίθμου που δίνει επίσης πολύ καλά αποτελέσματα.

Κεφάλαιο 7

Επίλογος

Στο κεφάλαιο αυτό κλείνουμε την παρούσα διπλωματική εργασία. Πραγματοποιούμε μία σύνοψη παρουσιάζοντας τα τελικά συμπεράσματα και αναφέρουμε μελλοντικές επεκτάσεις της εργασίας

Στην ενότητα [7.1](#) πραγματοποιείται μία σύνοψη της εργασίας και παρουσιάζονται τα βασικά συμπεράσματα αυτής.

Στην ενότητα [7.2](#) παρουσιάζονται πιθανές μελλοντικές επεκτάσεις της παρούσας διπλωματικής εργασίας.

7.1 Σύνοψη και Συμπεράσματα

Το ζήτημα της δυναμικής καταχώρησης δυναμικών μηχανών και η κατάλληλη τιμολόγηση αυτών είναι ιδιαίτερα σημαντικό για τις εταιρείες που παρέχουν υπηρεσίες Cloud Computing. Η ραγδαία ανάπτυξη που παρουσιάζεται σε αυτές τις υπηρεσίες αποτελεί έναυσμα για την εξέλιξη νέων τεχνικών που θα εξυπηρετούν τόσο τον πάροχο μεγιστοποιώντας τα έσοδά του, όσο και τους χρήστες διανέμοντας τους υπολογιστικούς πόρους με δίκαιο τρόπο.

Η παρούσα διπλωματική εργασία παρουσιάζει διάφορες προσεγγίσεις προς το σκοπό αυτό. Τα επιμέρους μοντέλα βασίστηκαν πάνω στη θεωρία του σχεδιασμού μηχανισμών και των δημοπρασιών. Κατασκευάστηκαν μοντέλα με βασικό κοινό στοιχείο την ιδιότητα της φιλαλήθειας, γεγονός που κάνει πιο εύκολη τη μελέτη του προβλήματος για τον σχεδιαστή. Οι δημοπρασίες τύπου Knapsack χρησιμοποιήθηκαν για τη βασική μοντελοποίηση του προβλήματος, ενώ οι δημοπρασίες μεγιστοποίησης εσόδων και οι τυχαίες δημοπρασίες χρησιμοποιήθηκαν προς όφελος του παρόχου.

Το πρώτο μοντέλο που πραγματοποιήσαμε είχε ως στόχο τη μεγιστοποίηση της κοινωνικής ωφέλειας των χρηστών. Για την επίλυση αυτού του προβλήματος χρησιμοποιήθηκε προσεγγιστικός αλγόριθμος. Αν και το μοντέλο αυτό δεν είναι ρεαλιστικό και εφαρμόσιμο αποτέλεσε βάση για τις υπόλοιπες προσεγγίσεις, ενώ παράλληλα δίνει ένα πιο πλήρες - θεωρητικά μιλώντας - πρόσωπο στην εργασία.

Το δεύτερο μοντέλο που πραγματοποιήσαμε είχε ως στόχο τη μεγιστοποίηση των εσόδων για τον πάροχο. Βασιστήκαμε σε προσεγγιστικούς αλγορίθμους και τυχαίες δημοπρασίες για την επίτευξη του στόχου.

Το τρίτο μοντέλο που πραγματοποιήσαμε είχε ως στόχο τη μεγιστοποίηση των εσόδων για τον πάροχο. Για την υλοποίηση του χρησιμοποιήσαμε τη θεωρία δημοπρασιών μεγιστοποίησης εσόδων και βασιστήκαμε στη θεωρία πιθανοτήτων και πιο συγκεκριμένα στην εκτίμηση αγνώστων παραμέτρων από υπάρχοντα δεδομένα για την εύρεση της κατανομής προσφορών των χρηστών.

Το τέταρτο και τελευταίο μοντέλο που πραγματοποιήσαμε είχε ως στόχο τη μεγιστοποίηση των εσόδων για τον πάροχο. Πρόκειται για μία υβριδική λύση που συνδυάζει τυχαίες δημοπρασίες και νευρωνικά δίκτυα. Το νευρωνικό δίκτυο που χρησιμοποιείται εκπαιδεύεται προκειμένου να πραγματοποιεί εκτιμήσεις της κατανομής προσφορών κάθε χρήστη ξεχωριστά.

Τα παραπάνω μοντέλα αξιολογήθηκαν και συγκρίθηκαν μεταξύ τους τόσο σε συνθετικά όσο και σε πραγματικά δεδομένα. Τα τρία τελευταία ήταν αυτά που μεγιστοποίησαν τα έσοδα του παρόχου. Το τελευταίο μοντέλο ήταν αυτό που παρουσίασε τα μεγαλύτερα έσοδα και είναι αυτό που τελικά συστήνουμε. Ωστόσο, ο λόγος για τον οποίο προτείνεται αυτό το μοντέλο δεν έγκειται αποκλειστικά στη μεγιστοποίηση εσόδων. Το συγκεκριμένο νευρωνικό δίκτυο προσαρμόζεται για κάθε χρήστη ξεχωριστά και επομένως με ενδεχόμενη καλύτερη εκπαίδευση και παραμετροποίηση είναι πιθανό να παρουσιάσει ακόμα καλύτερα αποτελέσματα.

Συνολικά, μέσα από αυτή την διπλωματική εργασία ήρθαμε σε επαφή και κατανοήσαμε πολλές διαφορετικές έννοιες όπως η αλγοριθμική θεωρία παιγνίων, ο σχεδιασμός μηχανισμών, οι δημοπρασίες, οι μέθοδοι για εκτίμηση αγνώστων παραμέτρων, τα νευρωνικά δίκτυα, οι προσεγγιστικοί αλγόριθμοι κ.τ.λ. γεγονός που τονίζει τόσο το πολύπλευρο όσο και τον εκπαιδευτικό χαρακτήρα της.

7.2 Μελλοντική Εργασία

Κλείνουμε την παρούσα εργασία παρουσιάζοντας πιθανές κατευθύνσεις μελλοντικής εργασίας.

- Αρχικά, η παρούσα διπλωματική εργασία θα μπορούσε να επεκταθεί σε περιπτώσεις όπου ο πάροχος θα διαθέτει προς μίσθωση εικονικές μηχανές διαφορετικού τύπου. Ένα τέτοιο πρόβλημα έγκειται στην κατηγορία των Combinatorial Auctions και παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς υποστηρίζει το πλέον ρεαλιστικό σενάριο.
- Μία άλλη προσέγγιση, θα ήταν να κατασκευαστεί ένα πλήρες end-to-end σύστημα στο οποίο ο πάροχος θα χρησιμοποιεί ένα σύστημα παρόμοιο με το δικό μας, ενώ ο χρήστης θα χρησιμοποιεί ένα "έξυπνο πράκτορα" [4] στη θέση του, του οποίου αναθέτοντας του ένα συνολικό budget θα ανταγωνίζεται ενάντια στους υπόλοιπους χρήστες προκειμένου να καλύψει τις υπολογιστικές ανάγκες του χρήστη.
- Επιπρόσθετα, εφόσον υπάρξουν διαθέσιμα δεδομένα από την υπηρεσία Spot Instances θα μπορούσαμε να αξιολογήσουμε και έπειτα να βελτιστοποιήσουμε το μοντέλο μας εφαρμόζοντας το πλήρως πάνω στην υπάρχουσα υπηρεσία.
- Επιπλέον, με τη βοήθεια των κατάλληλων υπολογιστικών πόρων και πλήθους δεδομένων εκπαίδευσης θα μπορούσαμε να πραγματοποιήσουμε ένα fine tuning στο νευρωνικό μας δίκτυο για την επίτευξη ακόμα καλύτερα προσεγγίσεων στις κατανομές προσφορές των χρηστών.
- Τέλος, ιδιαίτερο θεωρητικό ενδιαφέρον θα είχε η επίλυση του προβλήματος Knapsack δίχως προσεγγιστικούς αλγόριθμους με τη βοήθεια ισχυρών υπολογιστικών συστημάτων και η σύγκριση των αποτελεσμάτων αυτών με τα δικά μας.

Κεφάλαιο 8

Βιβλιογραφία

- [1] Amazon Elastic Compute Cloud - User Guide for Linux Instances. page 1068.
- [2] Gagan Aggarwal and Jason D. Hartline. Knapsack auctions. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm - SODA '06*, pages 1083–1092, Miami, Florida, 2006. ACM Press.
- [3] D.P. Bertsekas and J.N. Tsitsiklis. *Introduction to Probability*. Athena Scientific optimization and computation series. Athena Scientific, 2008.
- [4] C. Bitsakos, I. Konstantinou, and N. Koziris. Derp: A deep reinforcement learning cloud system for elastic resource provisioning. In *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 21–29, Dec 2018.
- [5] Avrim Blum and Jason D. Hartline. Near-optimal online auctions. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pages 1156–1163, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
- [6] Jeremy Bulow and Paul Klemperer. Auctions vs. negotiations. Working Paper 4608, National Bureau of Economic Research, January 1994.
- [7] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, New York, NY, USA, 2006.
- [8] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai Srivatsa Ravindranath. Optimal Auctions through Deep Learning. *arXiv:1706.03459 [cs]*, June 2017. arXiv: 1706.03459.
- [9] Simon S. Haykin. *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition, 2009.
- [10] Albert Xin Jiang and Kevin Leyton-Brown. Estimating Bidders' Valuation Distributions in Online Auctions. page 16.
- [11] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- [12] Warren Mcculloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:127–147, 1943.
- [13] T.K. Moon. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13:47 – 60, 12 1996.
- [14] Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6(1):58–73, February 1981.
- [15] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*, volume 1 of *MIT Press Books*. The MIT Press, March 1994.

-
- [16] Michael Ostrovsky and Michael Schwarz. Reserve Prices in Internet Advertising Auctions: A Field Experiment. *SSRN Electronic Journal*, 2009.
- [17] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. *ACM Trans. Econ. Comput.*, 1(4):18:1–18:26, December 2013.
- [18] Charles Reiss, John Wilkes, and Joseph Hellerstein. Google cluster-usage traces: format + schema. page 14.
- [19] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [20] Tim Roughgarden. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 1st edition, 2016.
- [21] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747 [cs]*, September 2016. arXiv: 1609.04747.
- [22] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 7 1948.
- [23] W. Shi, L. Zhang, C. Wu, Z. Li, and F. C. M. Lau. An online auction framework for dynamic resource provisioning in cloud computing. *IEEE/ACM Transactions on Networking*, 24(4):2060–2073, Aug 2016.
- [24] Joseph Sill. Monotonic Networks. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 661–667. MIT Press, 1998.
- [25] L. Zhang, Z. Li, and C. Wu. Dynamic resource provisioning in cloud computing: A randomized auction approach. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 433–441, April 2014.