ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΜΠΕΡΙΦΟΡΑΣ ΠΑΡΑΓΩΓΩΝ ΣΤΗΝ ΑΓΟΡΑ ΕΠΟΜΕΝΗΣ ΜΕΡΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Δ. Κεντρωτής

Επιβλέπων: Ιωάννης Ψαρράς

Καθηγητής ΕΜΠ

Αθήνα, Απρίλιος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΜΠΕΡΙΦΟΡΑΣ ΠΑΡΑΓΩΓΩΝ ΣΤΗΝ ΑΓΟΡΑ ΕΠΟΜΕΝΗΣ ΜΕΡΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Δ. Κεντρωτής

Επιβλέπων: Ιωάννης Ψαρράς

Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14η Απριλίου 2014 .

.....	.....	.....
Ιωάννης Ψαρράς	Δημήτριος Ασκούνης	Βασίλειος Ασημακόπουλος
Καθηγητής ΕΜΠ	Επίκουρος Καθηγητής	Καθηγητής ΕΜΠ

Αθήνα, Απρίλιος 2014

.....

Κωνσταντίνος Δ. Κεντρωτής

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

Ευχαριστίες

Καταρχάς, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας, κο Ιωάννη Ψαρρά, για την εμπιστοσύνη που έδειξε προς το πρόσωπό μου κατά την ανάθεση αυτής, καθώς και για την πολύτιμη καθοδήγησή του. Επίσης, θερμές ευχαριστίες χρωστάω στον κο Σωτήρη Παπαδέλη, του οποίου η συνεισφορά και η βοήθεια ήταν εξαιρετικά σημαντικές για την επιτυχή ολοκλήρωση της εργασίας. Τέλος, θέλω να εκφράσω την ευγνωμοσύνη μου προς τους γονείς μου, Δημήτρη και Ευαγγελία, για τη συνεχή στήριξη και συμπαράστασή τους όλα αυτά τα χρόνια των σπουδών μου.

Κωνσταντίνος Δ. Κεντρωτής

Αθήνα, 14 Απριλίου 2014

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει την υπολογιστική μοντελοποίηση της συμπεριφοράς των παραγωγών ηλεκτρικής ενέργειας σε ανταγωνιστικό περιβάλλον, μέσω μεθόδων ενισχυτικής μάθησης και συγκεκριμένα μέσω του τροποποιημένου αλγορίθμου των Roth & Erev. Αρχικά γίνεται αναφορά στη σημασία αλλά και στις προκλήσεις που έχει η μοντελοποίηση της αγοράς ηλεκτρικής ενέργειας και επισημαίνονται τα βασικά χαρακτηριστικά της ελληνικής αγοράς: ο Ημερήσιος Ενεργειακός Προγραμματισμός (ΗΕΠ) και η Οριακή Τιμή Συστήματος (ΟΤΣ). Εν συνεχεία, περιγράφεται η έννοια της μάθησης στις υπολογιστικές μηχανές, ορίζονται οι απαιτούμενες έννοιες και αναφέρονται επιγραμματικά οι τρεις βασικοί τύποι της. Ακολουθεί πλήρης παρουσίαση των μεθόδων ενισχυτικής μάθησης, τόσο με το απαράιτητο θεωρητικό υπόβαθρο, όσο και με το αντίστοιχο μαθηματικό τους μοντέλο. Έπειτα, γίνεται παρουσίαση του αρχικού αλγορίθμου των Roth και Erev (RE αλγόριθμος), αλλά και του αντίστοιχου τροποποιημένου αλγορίθμου (MRE), που αποτελεί βελτίωση του αρχικού. Παράλληλα, τονίζονται οι λόγοι για τους οποίους οι δυο αλγόριθμοι προτιμώνται για την επιλύση του προβλήματος της μοντελοποίησης της συμπεριφοράς των παραγωγών ηλεκτρικής ενέργειας και γίνεται η απαραίτητη προσαρμογή τους στο εν λόγω πρόβλημα. Τέλος, δίδεται ο κώδικας των RE και MRE αλγορίθμων στη γλώσσα προγραμματισμού Python.

Λέξεις Κλειδιά

Μοντελοποίηση τιμής αγοράς ηλεκτρικής ενέργειας, μέθοδοι μάθησης, μέθοδοι ενισχυτικής μάθησης, αλγόριθμος Roth Erev, τροποποιημένος αλγόριθμος Roth Erev, προσομοίωση πρακτόρων βασισμένη στις κοινωνικές επιστήμες.

Abstract

This thesis examines the computational modeling of producers' behavior in a competitive electricity market, throughout methods of reinforcement learning and particularly, using the Modified Roth Erev algorithm. At first, the importance and the challenges of modeling the electricity market are mentioned. Also, the basic traits of Greek electricity market, the Day Ahead Schedule (DAS) and the System Marginal Price (SMP) are presented. Later on, the definition of Learning in computational machines is given and its basic three types are briefly described. The full presentation of the reinforcement learning methods, along with their theoretical and mathematical background follows. Moreover, both Roth Erev (RE) and Modified Roth Erev (MRE) algorithms are described and analysed and the reasons for which these two algorithms are used for modeling the strategy of day-ahead electricity market are given. Finally, the implementation of the two algorithms into Python source code is presented.

Key Words

Modeling price of electricity market, learning methods, reinforcement learning methods, Roth Erev algorithm, Modified Roth Erev algorithm, agent based simulation based in the social sciences.

Πίνακας Περιεχομένων

Κεφάλαιο 1: Εισαγωγή

Στο κεφάλαιο αυτό εξηγούνται οι λόγοι για τους οποίους η μαθηματική μοντελοποίηση της αγοράς ηλεκτρικής ενέργειας είναι απαραίτητη και παρουσιάζονται οι προκλήσεις που καλείται κανείς να αντιμετωπίσει στην προσπάθειά του να την υλοποιήσει.

1.1 Χρησιμότητα και προκλήσεις μοντελοποίησης αγοράς ηλεκτρικής ενέργειας

Η ηλεκτρική ενέργεια αποτελεί ένα εξαιρετικά πολύτιμο αγαθό δεδομένου ότι είναι άρρηκτα συνδεδεμένο με την ποιότητα ζωής του ανθρώπου. Με τον πληθυσμό του πλανήτη να ανέρχεται κοντά στα 7 δισεκατομμύρια ανθρώπων και με τις εκτιμήσεις να αναμένουν να φθάσει τα 9 δισεκατομμύρια μέχρι το 2050, η ζήτηση του αγαθού αυτού αναμένεται να αυξηθεί στο μέλλον ακόμη περισσότερο από τις 20,132 περίπου TWh (10^{12} Wh) που απαιτούνταν το 2009 σύμφωνα με τη μελέτη των IEA/OECD (International Energy Agency/Organisation for Economic Co-operation and Development) [1]. Το γεγονός αυτό σε συνδυασμό με τη δραματική μείωση των αποθεμάτων πετρελαίου και άλλων πρωτογενών αγαθών που καταναλώνονται κατά το ένα τρίτο τους ετησίως μόνο για παραγωγή ηλεκτρικής ενέργειας, οδήγησαν στην ανάγκη για την όλο και αυξανόμενη διείσδυση των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) στην ηλεκτροπαραγωγή. Για κάποιον που επιθυμεί να μοντελοποιήσει μαθηματικά το σύστημα αγοράς ηλεκτρικής ενέργειας, τα δεδομένα αυτά και οι πολιτικές (οικονομικές και κοινωνικές) που τα διέπουν, αποτελούν νέες μεταβλητές καθιστώντας το πρόβλημα της μοντελοποίησης ακόμη πιο σύνθετο, καθώς η παράμετρος της στοχαστικής συμπεριφοράς του συστήματος αυξάνεται.

Ωστόσο το ερώτημα είναι τί καθιστά την ηλεκτρική ενέργεια ένα τόσο ιδιαίτερο εμπορικό αγαθό. Η απάντηση στο ερώτημα αυτό δεν είναι άλλη απ' την αδυναμία μας να την αποθηκεύουμε, σε αντίθεση με τα περισσότερα εμπορεύσιμα αγαθά, κάτι που σημαίνει ότι πρέπει να περνάει απευθείας απ' την παραγωγή στην κατανάλωση με όλες τις δυσκολίες ή τις ιδιαιτερότητες που μπορεί να προκύπτουν απ' το γεγονός αυτό.

Μέχρι τη δεκαετία του 1990, στις περισσότερες χώρες στον κόσμο, η αγορά ηλεκτρικής ενέργειας ήταν μονοπωλιακή. Ωστόσο, τα τελευταία χρόνια η δομή της εν λόγω αγοράς έχει υποστεί σημαντικές δομικές αλλαγές τείνοντας να καταστεί πλήρως ανταγωνιστική [2]. Ο σκοπός της μετάβασης αυτής ήταν η βελτίωση της αξιοπιστίας και της επάρκειας του δικτύου, καθώς και η μείωση των τιμών για τους καταναλωτές. Έτσι, η σύγχρονη αγορά ηλεκτρικής ενέργειας εκλαμβάνεται ως ανταγωνιστική και το πρόβλημα ανάγεται στη μοντελοποίηση μιας ανταγωνιστικής αγοράς ηλεκτρικής ενέργειας.

Αυτό που μένει τώρα να διευκρινιστεί είναι ο λόγος για τον οποίο η μαθηματική μοντελοποίηση της αγοράς της ηλεκτρικής ενέργειας αποκτά χρησιμότητα. Ο πρώτος λόγος έχει να κάνει με την αδυναμία έμπρακτου πειραματισμού σχετικά με τις διάφορες παραμέτρους της αγοράς, καθότι το αγαθό που εμπορεύεται είναι τόσο σημαντικό και ιδιαίτερο που δεν επιδέχεται πειραματισμών. Έτσι, είναι σημαντικό μέσω κάποιου μοντέλου να μπορούμε να προσομοιώσουμε την αγορά ούτως ώστε να

είμαστε σε θέση να πειραματιζόμαστε σε σχέση με τις όποιες παραμέτρους μας απασχολούν. Ο δεύτερος λόγος είναι καθαρά πρακτικός, αφού μέσα απ' την προσομοίωση μπορούμε να προβλέπουμε, σε βραχυπρόθεσμη σχετικά βάση, στοιχεία ζήτησης, τιμών ή όποιες άλλες παραμέτρους μας ενδιαφέρουν. Ο τρίτος λόγος που μπορεί να χρησιμεύσει η μαθηματική μοντελοποίηση της αγοράς ενέργειας, είναι η διαμόρφωση σεναρίων, μιας διαδικασίας αρκετά σύνθετης που έχει ως βασικό της ζητούμενο όχι τόσο το να προβλέψουμε, όσο το να είμαστε σε θέση να γνωρίζουμε τις τάσεις που μπορεί να προκύψουν στο μέλλον ή πράγματα που θα θέλαμε να αποφευχθούν.

Εν κατακλείδι, η σημαντικότερη πρόκληση που αντιμετωπίζει κάποιος στην προσπάθειά του να μοντελοποιήσει μαθηματικά την αγορά ηλεκτρικής ενέργειας είναι το γεγονός οτι καλείται στην ουσία να μοντελοποιήσει την ίδια την ανθρώπινη συμπεριφορά, τον τρόπο που λαμβάνει ο άνθρωπος αποφάσεις και τις στρατηγικές που ακολουθεί ανάλογα με τις ισχύουσες συνθήκες.

Κεφάλαιο 2: Ελληνική Αγορά Ηλεκτρικής Ενέργειας

Στο κεφάλαιο αυτό γίνεται αναφορά στους βασικούς μηχανισμούς λειτουργίας της ελληνικής χονδρεμπορικής αγοράς ηλεκτρικής ενέργειας.

2.1 Χονδρεμπορική Αγορά Ηλεκτρικής Ενέργειας

Χονδρεμπορική αγορά ηλεκτρικής ενέργειας καλείται ο μηχανισμός συναλλαγών ή αλλιώς *διμερών δημοπρασιών* (*Double Auctions*) μεταξύ των πωλητών (παραγωγών και εισαγωγέων) και των αγοραστών (προμηθευτών), οι οποίοι ως επί το πλείστον είναι εκείνοι που προμηθεύουν την ενέργεια στους τελικούς καταναλωτές.

Στο ελληνικό σύστημα ηλεκτρικής ενέργειας το ρόλο του ανεξάρτητου Διαχειριστή του Συστήματος τον έχει ο Διαχειριστής Ελληνικού Συστήματος Μεταφοράς Ηλεκτρικής Ενέργειας (ΔΕΣΜΗΕ) που φροντίζει για την ομαλή διεξαγωγή της διαδικασίας Κατανομής Φορτίου, η οποία επιβάλει το ποιος σταθμός θα παράγει και πόση ποσότητα ηλεκτρικής ενέργειας. Η Κατανομή Φορτίου πρέπει να γίνεται με τρόπο που να διασφαλίζεται η ευστάθεια και η ασφαλής λειτουργία του δικτύου, καθώς και να τηρούνται όλες οι συμφωνηθείσες εμπορικές συμφωνίες.

2.2 Ημερήσιος Ενεργειακός Προγραμματισμός (ΗΕΠ)

Ο Ημερήσιος Ενεργειακός Προγραμματισμός (ΗΕΠ) αποτελεί το μοντέλο βάσει του οποίου οργανώνεται η χονδρεμπορική αγορά ενέργειας και πραγματοποιούνται οι διμερείς δημοπρασίες που αφορούν στην ενέργεια που θα διακινηθεί και θα καταναλωθεί την επόμενη ημέρα στην Ελλάδα. Ως *διμερή δημοπρασία* ορίζουμε τη συναλλαγή στην οποία συμμετέχουν πολυάριθμοι πωλητές αλλά και πολυάριθμοι αγοραστές. Ο λόγος για τον οποίο οι διμερείς δημοπρασίες αποτελούν αντικείμενο μελέτης είναι το γεγονός ότι η αλληλεπίδραση όλων αυτών των παραγόντων οδηγεί σε απρόβλεπτες μεταβολές τιμών της ηλεκτρικής ενέργειας. Οι δημοπρασίες διέπονται από συγκεκριμένους κανόνες και υλοποιούνται με διαφορετικό αλγόριθμο ανάλογα με τη χώρα και το Σύστημα, χωρίς ωστόσο να υπάρχει κάποιος αλγόριθμος που να θεωρείται ‘ιδανικός’ ως προς το αποτέλεσμα.

Σκοπός του μοντέλου του ΗΕΠ είναι η ελαχιστοποίηση του συνολικού κόστους που απαιτείται για την κάθε *Ημέρα Κατανομής*. Ως Ημέρα Κατανομής ορίζεται η χρονική περίοδος 24 ωρών που συμπίπτει με μια ωρολογιακή ημέρα και κατά την οποία δίνεται στους συναλλασσόμενους το δικαίωμα υποβολής προσφορών για την επόμενη. Η ελαχιστοποίηση, όμως, του κόστους δεν αποτελεί αυτοσκοπό. Πρέπει να γίνει με τέτοιο τρόπο ώστε να διασφαλίζεται η αξιοπιστία και η ασφαλής λειτουργία του Συστήματος, γι’ αυτό υπάρχουν και τεχνικές προδιαγραφές ως προς την επαρκή εφεδρεία και τη μέγιστη διείσδυση από ΑΠΕ που πρέπει πάντα να πληρούνται.

Ο προσδιορισμός της τιμής ηλεκτρικής ενέργειας μέσω του ΗΕΠ είναι αποτέλεσμα της βελτιστοποίησης της αντικειμενικής συνάρτησης κόστους και απαιτεί μια σειρά από σύνθετες παραμέτρους που είτε θέτει είτε ελέγχει ο Ρυθμιστής του Συστήματος. Κάθε παραγωγός υποβάλει μια υποχρεωτική προσφορά για το σύνολο της ισχύος του και αντίστοιχα κάθε προμηθευτής υποβάλει μια προσφορά για το σύνολο της ζήτησής

του, χωρίς να επιτρέπονται διμερή συμβόλαια φυσικής παράδοσης μεταξύ παραγωγών και προμηθευτών-αγοραστών (*mandatory pool model*). [20]

Τον Αύγουστο του 2013, το μητρώο συμμετοχόντων στον ΕΗΠ σύμφωνα με το Μηνιαίο Δελτίο Συστήματος Συναλλαγών ΗΕΠ Αυγούστου 2013 του Λειτουργού Αγοράς Ηλεκτρικής Ενέργειας (ΛΑΓΗΕ) [21] είχε ως εξής:

Σε όλους τους παρακάτω πίνακες με γκρι χρώμα ή με αστερίσκο σημειώνονται αυτοί που δραστηριοποιήθηκαν στον ΗΕΠ.

Πίνακας 2.1 Πίνακας **Παραγωγών** Μητρώου ΗΕΠ, Αυγούστου 2013

Α/Α	ΕΠΩΝΥΜΙΑ	ΣΥΝΤΟΜΟΓΡΑΦΙΑ
1*	ELPEDISON ΕΝΕΡΓΕΙΑΚΗ Α.Ε.	ELPEDISON_POWER
2	ENELCO A.E.	ENELCO
3	PROTERGIA A.E.	PROTERGIA
4	ΑΛΟΥΜΙΝΙΟΝ ΑΝΩΝΥΜΟΣ ΕΤΑΙΡΕΙΑ	ALUMINIUM S.A.
5	ΔΗΜΟΣΙΑ ΕΠΙΧΕΙΡΗΣΗ ΗΛΕΚΤΡΙΣΜΟΥ Α.Ε.	PPC
6	ΗΡΩΝ ΘΕΡΜΟΗΛΕΚΤΡΙΚΗ Α.Ε.	HERON
7	ΗΡΩΝ ΙΙ ΒΟΙΩΤΙΑΣ Α.Ε.	HERON_II_VIOTIAS
8*	ΚΟΡΙΝΘΟΣ POWER Α.Ε.	KORINTHOS POWER
9	ΜΟΤΟΡ ΟΙΛ (ΕΛΛΑΣ) ΔΙΥΛΙΣΤΗΡΙΑ ΚΟΡΙΝΘΟΥ Α.Ε.	MOTOROIL

* Η ELPEDISON_POWER και η ΚΟΡΙΝΘΟΣ POWER Α.Ε., έχοντας άδεια παραγωγής, δραστηριοποιούνται ως Προμηθευτές εκπροσωπώντας τα γενικά βοηθητικά των μονάδων τους.

Πίνακας 2.2 Πίνακας **Προμηθευτών** Μητρώου ΗΕΠ, Αυγούστου 2013

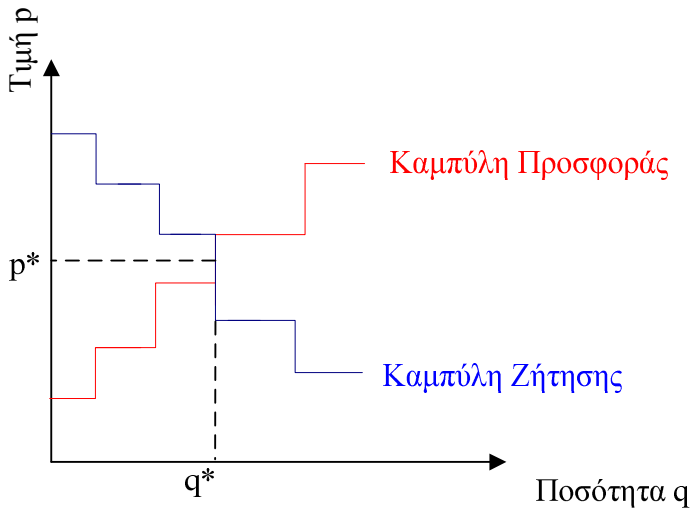
Α/Α	ΕΠΩΝΥΜΙΑ	ΣΥΝΤΟΜΟΓΡΑΦΙΑ
1*	ALPIQ ENERGY SE	ALPIQ_ENERGY
2	COMPAGNIE NATIONALE DU RHONE	CNR
3*	EDELWEISS ENERGIA S.P.A.	EDELWEISS
4*	ELECTRADE S.R.L.	ELECTRADE SPA
5	ELECTRICITY TRADING COMPANY HELLAS A.E.	ETC_HELLAS
6*	ELPEDISON ENERGY A.E.	ELPEDISON_ENERGY
7	ENI S.P.A.	ENI
8	EVN TRADING SOUTH EAST EUROPE EAD	EVN
9*	GREE ENVIRONMENTAL&ENERGY NETWORK ΑΝΩΝΥΜΗ ΕΤΑΙΡΕΙΑ	GREENENV
10*	NECO TRAIDING A.E.	NECO_TRADING
11	PROTERGIA A.E. ΠΡΟΜΗΘΕΥΤΗΣ	PROTERGIA
12	TINMAR - IND S.A.	TINMAR
13*	VOLTERRA A.E.	VOLTERRA
14	WATT AND VOLT A.E.	WATT_AND_VOLT
15*	ΔΗΜΟΣΙΑ ΕΠΙΧΕΙΡΗΣΗ ΗΛΕΚΤΡΙΣΜΟΥ Α.Ε.	PPC
16	ΗΛΕΚΤΡΟΠΑΡΑΓΩΓΗ ΣΟΥΣΑΚΙΟΥ Α.Ε.	SUSAKI_POWER
17*	ΗΡΩΝ ΘΕΡΜΟΗΛΕΚΤΡΙΚΗ Α.Ε.	HERON
18	ΠΡΟΜΗΘΕΥΤΗΣ ΚΑΘΟΛΙΚΗΣ ΥΠΗΡΕΣΙΑΣ	PPC_LRS

* Οι εταιρείες, έχοντας άδεια προμήθειας, δραστηριοποιήθηκαν στον ΗΕΠ ως Έμποροι ηλεκτρικής ενέργειας.

2.3 Οριακή Τιμή Συστήματος (ΟΤΣ)

Η Οριακή Τιμή Συστήματος (ΟΤΣ) είναι η τιμή στην οποία γίνεται η εκκαθάριση της αγοράς ηλεκτρικής ενέργειας και αποτελεί την τιμή με βάση την οποία αμείβονται οι πωλητές και χρεώνονται οι αγοραστές του Συστήματος. Η διαμόρφωσή της γίνεται βάσει του συνδυασμού προσφορών και ζήτησης ηλεκτρικής ενέργειας καθημερινά. Οι προσφορές αφορούν τις προσφορές τιμών και ποσοτήτων του συνόλου των διαθέσιμων παραγωγικών μονάδων και η ζήτηση τους καταναλωτές.

Η ΟΤΣ διαμορφώνεται από το σημείο τομής της αύξουσας καμπύλης προσφοράς και της φθίνουσας καμπύλης ζήτησης και καλείται και *ανταγωνιστικό σημείο ισορροπίας* (competitive equilibrium-CE). Στη γενική της μορφή, η ΟΤΣ αποτελείται από το ζεύγος της τιμής ισορροπίας p^* και της ποσότητας ισορροπίας q^*, ενώ η σημασία της έγκειται στο ότι εάν η ζήτηση είναι μεγαλύτερη από την προσφορά, η τιμή του αγαθού αυξάνεται και κατά συνέπεια η ζήτηση μειώνεται λόγω δυσβάσταχτου κόστους για κάποιους αγοραστές, ενώ η προσφορά αυξάνεται, λόγω συμφέροντος ορισμένων πωλητών. Αντίθετα, αν η προσφορά είναι μεγαλύτερη απ' τη ζήτηση, η τιμή μειώνεται κι έτσι η ζήτηση αυξάνεται, ενώ η προσφορά μειώνεται. Συμπερασματικά, θα λέγαμε ότι η ανταγωνιστική αγορά ηλεκτρικής ενέργειας τείνει την όλη αλληλουχία των δημοπρασιών στο τελικό σημείο ισορροπίας, δηλαδή την ΟΤΣ.



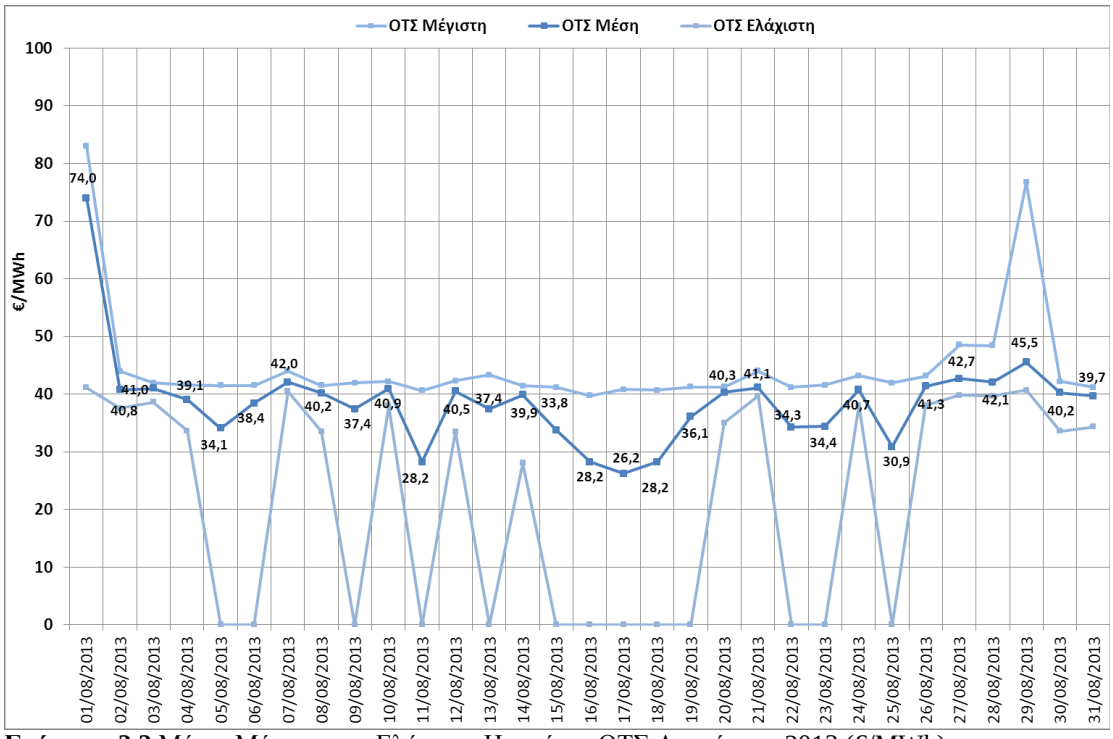
Σχήμα 2.1 Καθορισμός ΟΤΣ μέσω καμπυλών Προσφοράς και Ζήτησης

Για λόγους προστασίας των καταναλωτών και διαμόρφωσης συνθηκών υγιούς ανταγωνισμού, η Ρυθμιστική Αρχή Ενέργειας (ΡΑΕ), έχει θέσει ως ανώτερο όριο προσφερόμενης τιμής τα 150€/MWh και ως κατώτατο επίπεδο προσφορών το μεταβλητό κόστος της εκάστοτε μονάδας, ούτως ώστε οι παραγωγοί να αμείβονται τουλάχιστον το κόστος του καυσίμου τους. [20]

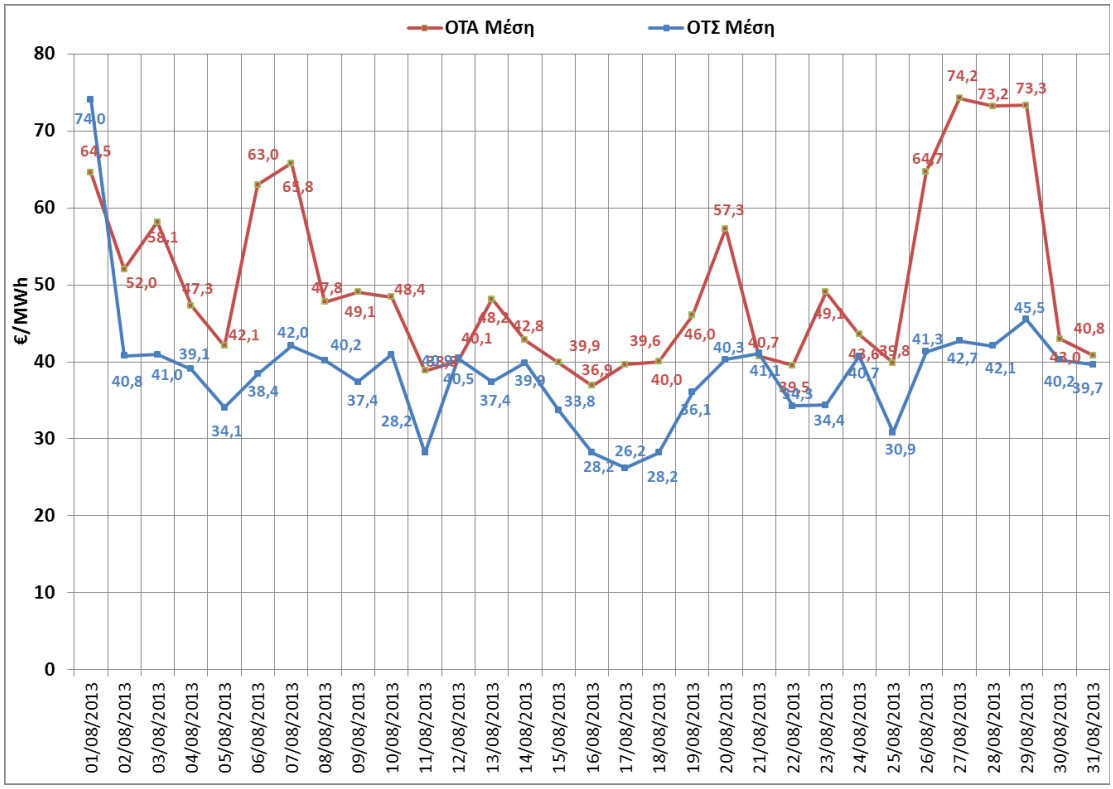
Μέχρι σήμερα, οι προμηθευτές (ουσιαστικά η ΔΕΗ) αγοράζουν το ρεύμα που παράγεται από ΑΠΕ στην Οριακή Τιμή Συστήματος. Οι παραγωγοί ΑΠΕ, όμως αμείβονται σύμφωνα με τη δεδομένη τιμή που έχουν υπογράψει στη συμφωνία τους με τον ΛΑΓΗΕ. Αυτό σημαίνει ότι τη διαφορά μεταξύ ΟΤΣ και συμφωνηθείσας τιμής ΛΑΓΗΕ-παραγωγών ΑΠΕ, πρέπει να την καλύψει ο ειδικός λογαριασμός ΑΠΕ του ΛΑΓΗΕ ο οποίος έχει έσοδα κυρίως από το τέλος ΕΤΜΕΑΡ και δευτερευόντως από τις δημοπρασίες ρύπων, το λιγνιτικό τέλος και μέρος του τέλους της ΕΡΤ. Βεβαίως, η ΡΑΕ δεν μπορεί να επιτρέπει τη συνεχή αύξηση του τέλους ΕΤΜΕΑΡ και δημιουργείται μια “τρύπα” στα ταμεία του ΛΑΓΗΕ η οποία πλέον έχει καταστεί δυσβάσταχτη.

Προκειμένου να επιλύσει το συγκεκριμένο πρόβλημα, το ΥΠΕΚΑ, εξετάζει μεταξύ άλλων και τη λύση της χρησιμοποίησης ως μέγεθος αναφοράς αντί της Οριακής Τιμής Συστήματος (ΟΤΣ), την Οριακή Τιμή Αποκλίσεων (ΟΤΑ). Η ΟΤΑ υπολογίζεται με βάση την εκκαθάριση των αποκλίσεων που αποδεικνύεται ότι σημείωσαν οι μονάδες παραγωγής σε σχέση με τη δήλωση που έκανα εκ των προτέρων οι ιδιοκτήτες τους. Όπως, λοιπόν είναι φυσικό, η ΟΤΑ είναι πάντα μεγαλύτερη της ΟΤΣ.

Στα γραφήματα που ακολουθούν, εμφανίζονται τα στοιχεία για τις ΟΤΣ και ΟΤΑ για το μήνα Αύγουστο του 2013, όπως τα δημοσίευσε ο ΛΑΓΗΕ [21]:

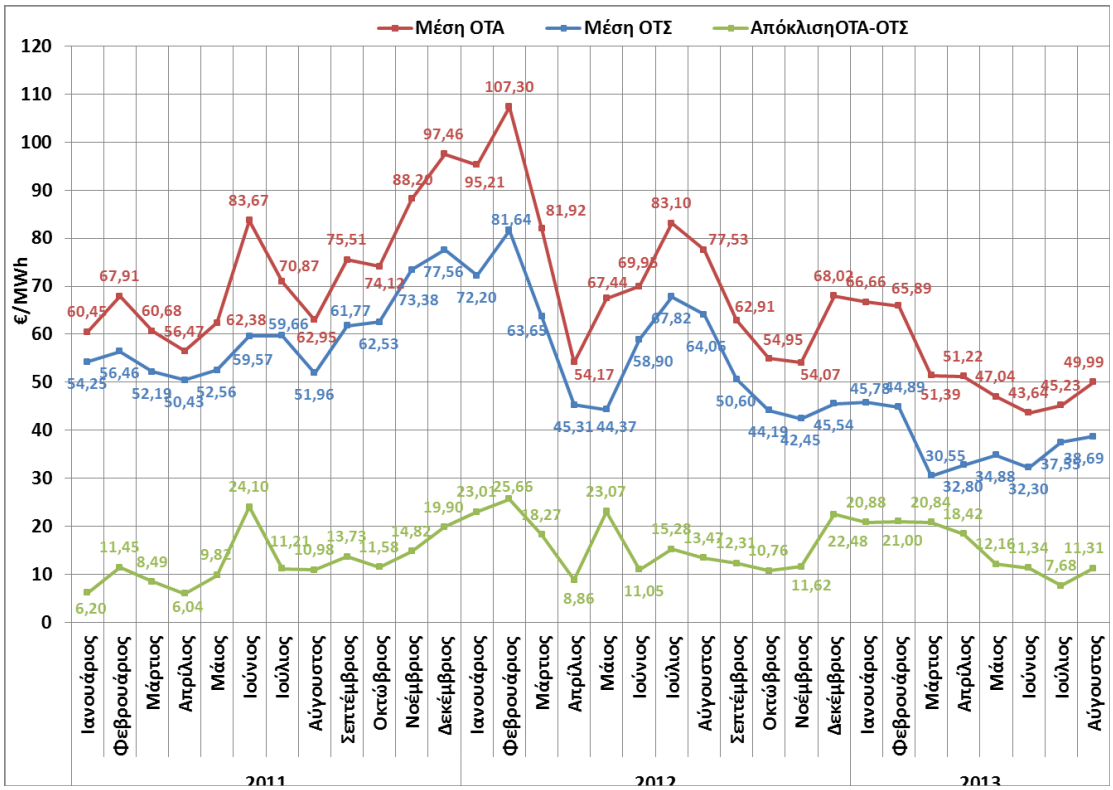


Γράφημα 2.2 Μέση, Μέγιστη και Ελάχιστη Ημερήσια ΟΤΣ Αυγούστου 2013 (€/MWh)



Γράφημα 2.3 Μέση Ημερήσια ΟΤΣ και ΟΤΑ Αυγούστου 2013 (€/MWh)

Στο γράφημα 2.4 [21] μπορεί να δει κανείς την εξέλιξη της μέσης μηνιαίας ΟΤΣ και ΟΤΑ καθώς και της απόκλισής ΟΤΑ-ΟΤΣ που όπως προαναφέρθηκε είναι πάντα θετική:



Γράφημα 2.4 Εξέλιξη της Μέσης Μηνιαίας ΟΤΣ και ΟΤΑ καθώς και της απόκλισής τους, Ιανουαρίου 2011 έως Αυγούστου 2013 (€/MWh)

Κεφάλαιο 3: Μάθηση

Στα πλαίσια του κεφαλαίου αυτού επιχειρείται η ανάλυση της έννοιας της μάθησης σε ένα υπολογιστικό ή ένα αυτοματοποιημένο σύστημα, η μαθηματική αναπαράστασή της καθώς και η συνοπτική παρουσίαση των γενικότερων κατηγοριών της.

3.1 Η έννοια της Μάθησης στις υπολογιστικές μηχανές

Σύμφωνα με τους Narendra και Thathachar (1974) [7], "ως μάθηση ορίζεται οποιαδήποτε σχετικά μόνιμη αλλαγή συμπεριφοράς λόγω παρελθόντων εμπειριών και ένα σύστημα μάθησης χαρακτηρίζεται από την ικανότητά του να βελτιώνεται με την πάροδο του χρόνου, σαν να τείνει προς έναν τελικό του σκοπό." [6]

Η διαδικασία της μάθησης μελετάται από τα τέλη του 19^{ου} αιώνα. Το 1898 ο Thorndike [8] κατέγραψε μια θεωρία στην οποία η σχέση μεταξύ ενός ερεθίσματος και μιας αντίδρασης δυναμώνει ή εξασθενεί ανάλογα με το αποτέλεσμα της αντίδρασης. Αυτό τον τύπο μάθησης τον ονόμασε *συντελεστική μάθηση*. Η *κλασική θεωρία της απόκτησης αντανακλαστικών* που προτάθηκε απ' τον Pavlov το 1899 [9], ασχολείται με την περίπτωση που ένα φυσικό αντανακλαστικό σε ένα συγκεκριμένο ερέθισμα γίνεται απόκριση ενός δεύτερου ερεθίσματος αρκετές φορές. Το 1930 ο Skinner [10] ανέπτυξε την ιδέα του Thorndike, αλλά ισχυρίστηκε ότι η μάθηση αποτελεί περισσότερο αποτέλεσμα "προσπάθειας και επιτυχίας" παρά "προσπάθειας και λάθους". Αυτές οι ιδέες ανήκουν σε ένα ρεύμα της ψυχολογίας που καλείται *συμπεριφορισμός*. Από το 1950 και μετά ο *ορθολογισμός* ή *ρασιοναλισμός* κέρδισε το μεγαλύτερο μέρος του ενδιαφέροντος. Σύμφωνα με αυτή την άποψη τα κίνητρα και ο αφαιρετικός λογισμός παίζουν πιο σημαντικό ρόλο στη μάθηση. Παράδειγμα αυτής αποτελεί η θεωρία του Bandura το 1969 όπου η παρατήρηση είναι σημαντική στη μάθηση. Ωστόσο, στα πλαίσια της διπλωματικής αυτής, λόγω του ότι γίνεται αναφορά σε μεθόδους μάθησης σε υπολογιστικά μοντέλα ασχολούμαστε ελάχιστα με τα κίνητρα ή τον αφαιρετικό λογισμό όπως τον αντιλαμβανόμαστε στην καθημερινότητα. Η ιδέα της *ενισχυτικής μάθησης*, στην οποία στηρίζεται η εργασία και περιγράφεται εκτενώς στο κεφάλαιο 3, έχει πολλά κοινά με τη συντελεστική μάθηση που περιέγραψαν οι Thorndike και Skinner.

Μέχρι στιγμής, όσα έχουμε αναφέρει, σχετίζονται με τη διαδικασία της μάθησης μόνο στους ανθρώπους. Όμως, πώς μπορούμε να περιγράψουμε τη διαδικασία της μάθησης σε μια υπολογιστική μηχανή; Σύμφωνα και με τους προαναφερθέντες ορισμούς, ένα σύστημα (είτε αυτό είναι μια μηχανή, είτε ένας άνθρωπος, είτε και ένα ζώο) λέμε ότι "μαθαίνει" αν τροποποιεί τη συμπεριφορά του βάσει των εμπειριών που συλλέγει. Συνεπώς, η απάντηση εξαρτάται απ' το πώς αναπαριστούμε τη μάθηση ή τη συμπεριφορά. [6]

Προκειμένου να είμαστε σε θέση να χρησιμοποιούμε τη διαδικασία της μάθησης σε μια οποιαδήποτε υπολογιστική διαδικασία, δηλαδή μια διαδικασία που μπορεί αυτόματα να προσομοιώνεται από έναν υπολογιστή ή μηχανή, χρειάζεται να επιχειρήσουμε μια πιο μαθηματική περιγραφή της.

Ας υποθέσουμε ότι υπάρχουν κάποιοι παίχτες που εφεξής θα τους καλούμε **‘πράκτορες’ (agents)**, οι οποίοι συμμετέχουν σε κάποιο επαναλαμβανόμενο παιχνίδι (iterated, multi-stage game), δηλαδή σ’ ένα παιχνίδι που αποτελείται από πολλούς γύρους οι οποίοι λαμβάνουν χώρα ο ένας μετά τον άλλον. Ο κάθε πράκτορας έχει ως σκοπό να κερδίσει σε καθέναν από τους γύρους και προκειμένου να το επιτύχει, εφαρμόζει μια συγκεκριμένη ‘στρατηγική’ (strategy). Η στρατηγική εναλλακτικά μπορεί να καλείται και ‘πολιτική’ (policy).

Οι στρατηγικές στα επαναλαμβανόμενα παιχνίδια διέπονται απ’ την εξής αρχή :

- Η δράση (action) που υπαγορεύεται απ’ τη στρατηγική του πράκτορα σε κάθε δεδομένη στιγμή διαμορφώνεται με βάση την παρούσα κατάστασή του, δηλαδή με βάση τις πληροφορίες που διαθέτει. Όμως η κατάσταση αυτή ορίζεται εκ των προτέρων για κάθε γύρο, εν ολίγοις ο πράκτορας δρα βάσει των πληροφοριών που έχει συλλέξει μέχρι και το τέλος του προηγούμενου γύρου. [3]

Ένα απλό παράδειγμα τέτοιας στρατηγικής είναι το ακόλουθο:

Πίνακας 3.1 Παράδειγμα αλγορίθμου απλής στρατηγικής επαναλαμβανόμενων παιχνιδιών

1. Ξεκίνα να παίζεις
2. Κάνε ότι έκανε ο ανταγωνιστής σου στο προηγούμενο γύρο

Τέτοιες στρατηγικές, με κατάλληλες προσαρμογές μπορούν να εφαρμοστούν και για πιο σύνθετες μορφές επαναλαμβανόμενων παιχνιδιών, όπως αυτά της αγοράς. Πριν αναφερθούμε όμως σε αυτό, θα πρέπει να ερευνήσουμε την έννοια της μάθησης σε τέτοια επαναλαμβανόμενα παιχνίδια αγοράς. Η μάθηση σημαίνει για εμάς ότι θέλουμε να επιτρέπεται σε έναν ή περισσότερους πράκτορες να αλλάξουν εκ βάθους τις στρατηγικές τους κατά τη διάρκεια διαδοχικών επαναλήψεων του παιχνιδιού, βασιζόμενοι σε παρατηρήσεις προηγούμενων γεγονότων.

Έστω ένα επαναλαμβανόμενο παιχνίδι. Ορίζουμε τις εξής έννοιες :
- **πολιτική (policy) ‘π’:** η στρατηγική που ακολουθεί ο κάθε πράκτορας.
- **δράση (action) ‘α’:** η δράση που επιλέγει ο πράκτορας για κάθε γύρο.
- **κατάσταση (state) ‘s’:** το σύνολο των πληροφοριών που λαμβάνονται απ’ το περιβάλλον.
- **ανταμοιβή (reward) ‘r’:** το κέρδος ή η ζημία στο τέλος κάθε γύρου που προκύπτει απ’ την επιλογή κάποιας δράσης.
- **περιβάλλον (environment) ‘e’:** οι ανταγωνιστές αλλά και το γενικότερο περιβάλλον του παιχνιδιού (αγοράς).

Η διαδικασία της μάθησης (learning) μπορεί να περιγραφεί ακολούθως [3] :
Ένας πράκτορας ξεκινά να συμμετέχει σε ένα επαναλαμβανόμενο παιχνίδι με μια αρχική στρατηγική (πολιτική) **π**, η οποία τον οδηγεί στην υλοποίηση μιας δράσης **α** οποτεδήποτε συναντά μια συγκεκριμένη κατάσταση **s**:

Κατάσταση **s**→ Δράση **α**

Στη συνέχεια, βάσει της επιλογής της δράσης α, ο πράκτορας λαμβάνει την ανταμοιβή του (reward) **r**. Αν υποθέσουμε ότι η ανταμοιβή αυτή δεν τον ικανοποιεί και ξέροντας ότι την έχει λάβει εφαρμόζοντας το συσχετισμό **s→ α**, αποφασίζει να αλλάξει το συσχετισμό σε :

Κατάσταση **s**→ Δράση **α***

Η διαδικασία αυτή κατά την οποία ο πράκτορας αποφασίζει να αλλάξει το συσχετισμό **s→ α** σε **s→ α*** με κριτήριο την ανταμοιβή που έλαβε για την αρχική του επιλογή, ονομάζεται **μάθηση (learning)**.

Στο σημείο αυτό αξίζει να δοθεί ιδιαίτερη προσοχή στη διαφορά μεταξύ της έννοιας μάθησης και της έννοιας προσαρμογής. Υποθέτουμε ότι ένας πράκτορας δρα σε συμφωνία με ένα συγκεκριμένο συσχετισμό **s→ α** στο πλαίσιο ενός ευρύτερου περιβάλλοντος **e**. Έστω ότι κάτι αλλάζει στο περιβάλλον το οποίο από **e** μετατρέπεται σε **e***. Τότε ο πράκτορας αποφασίζει να αλλάξει το συσχετισμό που ακολουθεί από **s→ α** σε **s→ α***. Η μεταβολή αυτή δεν καλείται μάθηση αλλά **προσαρμογή**. Κι αυτό διότι αν θεωρήσουμε ως αρχική κατάσταση **S** το σύνολο (**s, e**) το οποίο οδηγεί στη δράση **α**, τότε αφού στη συνέχεια υπάρχει μετάβαση από **e** σε **e*** είναι σα να μετατρέπεται αντιστοίχως και η κατάσταση **S** σε **S***. Συνεπώς, μάθηση ονομάζεται η διαδικασία κατά την οποία η κατάσταση **s** μένει σταθερή ωστόσο εμείς επιλέγουμε να αλλάξουμε τη δράση μας. Όταν η κατάσταση **s** μεταβάλλεται και μεταβάλλεται και η δράση που επιλέγουμε, έχουμε το φαινόμενο της προσαρμογής και όχι της μάθησης.

Πίνακας 3.2 Διαφορά Μάθησης - Προσαρμογής

Μάθηση : Γύρος 1 : **S→ α**
Γύρος 2 : **S→ α***

Προσαρμογή : Γύρος 1 : **S=(s, e)→ α**
Γύρος 2 : **S*=(s, e*)→ α***

3.2 Γενικοί τύποι Μάθησης

Είναι προφανές ότι οι μηχανές μπορούν να μαθαίνουν μέσω εμπειριών μεταβάλλοντας κάποιες παραμέτρους εντός ενός μοντέλου ή ενός πίνακα με δεδομένα. Όμως, τί ακριβώς μαθαίνει το σύστημα; Η απάντηση σε τέτοιου είδους ερωτήσεις εξαρτάται απ' το είδος της μάθησης για το οποίο μιλάμε.

Οι τύποι μάθησης μπορούν να κατηγοριοποιηθούν με κριτήριο τη διαδικασία που επιφέρει την προαναφερθείσα μεταβολή από **S→ α** σε **S→ α***. Εν γένει, μπορούμε να εντοπίσουμε τρεις τύπους μάθησης [3], [4], [6]:

1. Μάθηση χωρίς επιτήρηση (Unsupervised Learning) :

Στην κατηγορία αυτή εμπίπτει κάθε διαδικασία μάθησης κατά την οποία αναθεωρείται ο ακολουθούμενος συσχετισμός βασιζόμενος σε εσωτερικά κίνητρα του πράκτορα. Τέτοια είναι, για παράδειγμα, η περιέργεια, η ικανοποίηση, το αίσθημα της ηθικής υποχρέωσης. Εδώ, δεν υπάρχει κάποιος εξωτερικός παράγοντας

ή επιτηρητής ο οποίος να επιβάλει στο σύστημα πώς να συμπεριφερθεί. Δηλαδή, δεν υπάρχει εξωτερική ανάδραση. Οι εμπειρίες του συστήματος συνίστανται από ένα σύνολο σημάτων και η ποσοτικοποίηση της απόδοσής του γίνεται συνήθως μέσω κάποιων στατιστικών μεγεθών όπως η τυπική απόκλιση, η διασπορά και η συσχέτιση.

2. Μάθηση υπό επιτήρηση (Supervised Learning) :

Αποτελεί την αντίθετη μέθοδο της μάθησης χωρίς επιτήρηση. Ένας εξωτερικός παρατηρητής επιβάλει στο σύστημα την επιθυμητή (σωστή) απόκριση για κάθε είσοδο. Δηλαδή, δεδομένου της κατάστασης **s**, ο επιτηρητής επιβάλει το συσχετισμό **s→ α** που πρέπει να ακολουθηθεί στηριζόμενος στη γνώση ή στην πείρα του. Οι εμπειρίες στην κατηγορία αυτή αποτελούνται από ζεύγη ερεθισμάτων και επιθυμητών αποκρίσεων και ο μόνος τρόπος να βελτιώσει κανείς την απόδοση του συστήματος είναι να ελαχιστοποιήσει κάποια μεγέθη σφαλμάτων (πχ τα τετράγωνα της διαφοράς μεταξύ της εξόδου και της επιθυμητής εξόδου του συστήματος). [6]

3. Ενισχυτική Μάθηση (Reinforcement Learning) :

Η αναθεώρηση του συσχετισμού επιλογής δράσης στην κατηγορία της ενισχυτικής μάθησης είναι απόρροια των διαδοχικών αμοιβών που έχουν επιτευχθεί μέσω των δράσεων. Αν θελήσουμε να παρομοιάσουμε τον πράκτορα με ένα δάσκαλο και το σύστημα με ένα μαθητή, θα λέγαμε ότι σε κάθε γύρο ο δάσκαλος λέει στο μαθητή πόσο καλά ή άσχημα απέδωσε, αλλά δεν του δίνει καμία πληροφορία σχετικά με τα επιθυμητά αποτελέσματα (επιθυμητές δράσεις). Αυτό δε συμβαίνει τυχαία. Σε πολλά συστήματα είναι πολύ δύσκολο ή και ακατόρθωτο να διαπιστώσει κανείς ποια έξοδος είναι η επιθυμητή για κάθε είσοδο (μπορεί επίσης πολλές έξοδοι να είναι ισάξιες). Αυτό που μπορεί όμως να προσδιορίσει είναι αν το σύστημα πέτυχε ή απέτυχε σε μια συγκεκριμένη διαδικασία. Εν προκειμένω, ο πράκτορας κρίνει αν η ανταμοιβή που λαμβάνει είναι ικανοποιητική ή όχι και αποφασίζει ανάλογα για το αν θα επαναλάβει τη δράση που ακολούθησε. Δεν είναι σε θέση όμως να γνωρίζει ποιά ακριβώς ανταμοιβή θέλει να πετύχει. Αυτό καθιστά την ενισχυτική μάθηση γενικότερη από τη μάθηση υπό επιτήρηση καθώς μπορεί να εφαρμοστεί και σε προβλήματα των οποίων τη λύση δε γνωρίζουμε [5]. Στο επόμενο κεφάλαιο αναλύεται λεπτομερώς ο αλγόριθμος της ενισχυτικής μάθησης για τη μοντελοποίηση συστημάτων όπως είναι οι αγορές.

Κεφάλαιο 4: Ενισχυτική Μάθηση (Reinforcement Learning)

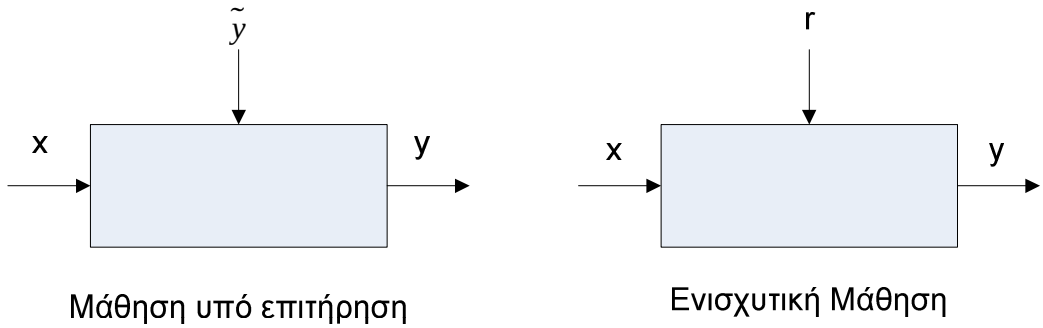
Στο κεφάλαιο αυτό παρουσιάζεται το θεωρητικό αλλά και μαθηματικό υπόβαθρο των αλγόριθμων μάθησης (RL). Σκοπός είναι η απάντηση σε ερωτήσεις όπως το τί είναι η ενισχυτική μάθηση, πώς διαμορφώνονται οι αλγόριθμοί της, πού χρησιμεύουν και πώς εφαρμόζονται σε ένα αυτοματοποιημένο υπολογιστικό περιβάλλον.

4.1 Περιγραφή Ενισχυτικής Μάθησης

Οι Kaelbling, Littman και Moore, το 1996 είχαν επιχειρήσει να απαντήσουν στο τί καλείται ενισχυτική μάθηση (Reinforcement Learning) με τη φράση: «ο τρόπος προγραμματισμού πρακτόρων μέσω ανταμοιβής και τιμωρίας, χωρίς να χρειάζεται ο προσδιορισμός του πώς θα επιτευχθεί η ζητούμενη εργασία». [12]

Ως *ενίσχυση* ή *ενισχυτικό ερέθισμα* ορίζεται ένα ερέθισμα το οποίο ενδυναμώνει τη συμπεριφορά που το παρήγαγε. Για παράδειγμα, ας θεωρήσουμε τη διαδικασία εκπαίδευσης ενός ζώου. Προφανώς δεν είναι δυνατό να εξηγήσουμε στο ζώο πώς να συμπεριφέρεται. Ο μόνος τρόπος είναι να ανταμείβουμε το ζώο κάθε φορά που εμφανίζει την επιθυμητή συμπεριφορά, δίνοντας του κάποια τροφή που του αρέσει. Με αυτό τον τρόπο, μετά από ένα χρονικό διάστημα, θα μάθει να ακολουθεί σχεδόν πάντα τη συμπεριφορά που του επέφερε την ανταμοιβή. Τότε, λέμε ότι η συμπεριφορά αυτή έχει *ενισχυθεί*. Εξαιτίας αυτού, οποιαδήποτε διαδικασία μάθησης στηρίζεται στην αρχή της ενίσχυσης μιας συμπεριφοράς, λόγω της ανταμοιβής που λαμβάνεται, καλείται *ενισχυτική*.

Η ενισχυτική μάθηση (Reinforcement Learning) μπορεί να θεωρηθεί ως ενός είδους υπό επιτήρηση μάθησης με μία βασική διαφορά. Αν επιχειρούσαμε να μοντελοποιήσουμε ως συστήματα την ενισχυτική και την υπό επιτήρηση μάθηση, στη μεν υπό επιτήρηση μάθηση θα εμφανιζόταν ως ανάδραση του συστήματος η επιθυμητή έξοδος ($\tilde{y}$ ή y_{ref}), ενώ στην ενισχυτική μάθηση μια βαθμωτή ανάδραση (έστω r). [6]



Σχήμα 4.1 Η διαφορά μεταξύ μάθησης υπό επιτήρηση και ενισχυτικής μάθησης. Στην πρώτη η ανάδραση αποτελείται από το διάνυσμα της επιθυμητής εξόδου, ενώ στη δεύτερη από ένα βαθμωτό ποιοτικό μέγεθος r.

Ας πάρουμε για παράδειγμα ένα παιδί το οποίο προσπαθεί να μάθει κολύμπι. Οι γονείς του δεν είναι σε θέση να περιγράψουν στο παιδί ακριβώς τις κινήσεις και τη συμπεριφορά που πρέπει να έχει μέσα στο νερό, αλλά μπορούν, παρατηρώντας το να προσπαθεί, να γνωρίζουν πόσο καλά ή άσχημα τα καταφέρνει. Αυτός είναι και ο

κυριότερος λόγος για τον οποίο οι μέθοδοι ενισχυτικής διδασκαλίας έγιναν ιδιαίτερα δημοφιλείς για χρήση σε αυτοματοποιημένα και ρομποτικά συστήματα. Ο προγραμματιστής των συστημάτων δεν είναι υποχρεωμένος να ξέρει εκ των προτέρων τη λύση ενός προβλήματος, είναι όμως σε θέση να γνωρίζει αν και πόσο καλά το σύστημα καταφέρνει να το λύσει. Έτσι, γίνεται αντιληπτό ότι τα συστήματα ενισχυτικής μάθησης χρειάζονται μεν μια ανάδραση, ωστόσο η ανάδραση αυτή είναι εξαιρετικά απλή σε σχέση με αυτή που απαιτείται σε άλλα συστήματα μάθησης. Λόγω αυτού, η δουλειά του επιβλέποντα του συστήματος μπορεί να καταστεί τόσο εύκολη που μπορεί ακόμα και ο ίδιος να ενσωματωθεί στο σύστημα. Λόγου χάρη, σε πολλά συστήματα που υιοθετούν την ενισχυτική μάθηση, η ανάδραση μπορεί να είναι απλά μία απάντηση για το αν ένα μέγεθος (πχ θερμοκρασία) είναι το ζητούμενο ή όχι. Αυτό σημαίνει ότι ο επιβλέπων μπορεί να αντικατασταθεί από ένα σύνολο αισθητήρων κι έτσι το σύστημα να μοιάζει με ένα χωρίς επιτήρηση σύστημα που όμως στην ουσία θα είναι υπό επιτήρηση.

Μέχρι στιγμής, έχουμε εστιάσει σε δυο βασικά πλεονεκτήματα της ενισχυτικής μάθησης : αφενός, στο ότι αποτελεί μια γενικότερη (ευρύτερα εφαρμόσιμη) μέθοδο σε σχέση με την υπό επιτήρηση μάθηση και αφετέρου στο ότι είναι εξαιρετικά απλή. Υπάρχει, όμως, κι ένα ακόμη. Η δυνατότητα του συστήματος να μαθαίνει με βάση τις ανταμοιβές που λαμβάνει του δίνουν τη δυνατότητα να καταστεί ικανότερο και απ' τον ίδιο του το "δάσκαλο". Μπορεί να βελτιώνει τη συμπεριφορά του εκπαιδεύοντας τον εαυτό του όπως συμβαίνει με το πρόγραμμα τάβλι του Tesauro. [11]

Υπάρχει, ωστόσο, κι ένα μειονέκτημα που αφορά στην περιορισμένη σε πληροφορία ανάδραση που λαμβάνει το σύστημα. Στη μάθηση υπό επιτήρηση, η διαφορά μεταξύ της εξόδου και της επιθυμητής εξόδου του συστήματος μπορεί να χρησιμοποιηθεί για να προσδιοριστεί η κατεύθυνση προς την οποία πρέπει αναζητήσουμε μια καλύτερη λύση. Δημιουργείται, έτσι, μια πληροφορία ως προς την επιθυμητή έξοδο που στην ενισχυτική μάθηση δεν υφίσταται. Ο καλύτερος τρόπος να αντιμετωπιστεί το πρόβλημα αυτό είναι η εισαγωγή θορύβου στον αλγόριθμο ούτως ώστε το σύστημα να στρέφεται προς την αναζήτηση καλύτερης λύσης με στοχαστικό τρόπο.

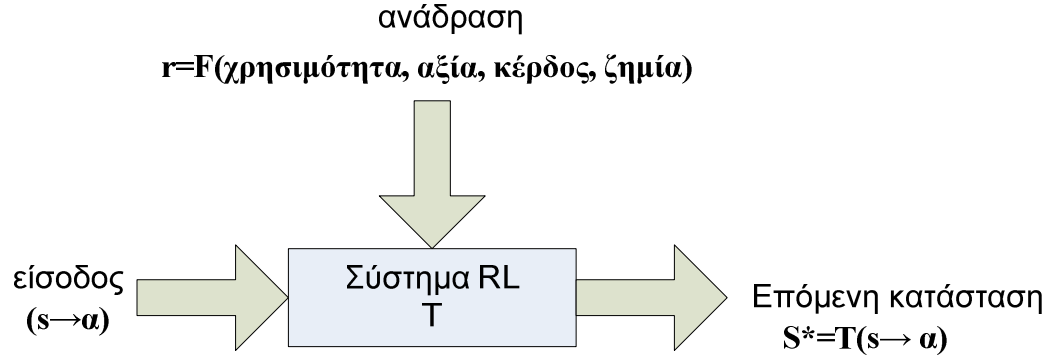
4.2 Βασικό Μοντέλο Ενισχυτικής Μάθησης (RL)

Οι έννοιες που χρειάζονται για την περιγραφή του βασικού μοντέλου της ενισχυτικής μάθησης που στη συνέχεια θα συμβολίζουμε και με RL (Reinforcement Learning) είναι οι ίδιες που είχαμε ορίσει και στην παράγραφο 2.1 στο πλαίσιο της απλής μάθησης [3]. Συγκεκριμένα:

- **πράκτορας (agent):** καλείται αυτός που λαμβάνει τις αποφάσεις ή αλλιώς ο 'μαθητής'.
- **δράση (action) 'α':** η δράση που επιλέγει ο πράκτορας για κάθε γύρο.
- **κατάσταση (state) 's':** το σύνολο των πληροφοριών που λαμβάνονται απ' το περιβάλλον.
- **πολιτική (policy) 'π':** η πολιτική "μεταφράζει" κάθε κατάσταση **s** σε μια επιλογή δράσης **α** (**π(s)=α**).
- **ανταμοιβή (reward) 'r':** είναι η άμεση τιμή της συσχέτισης κατάστασης-δράσης (**s→ α**)

- **περιβάλλον (environment) 'e':** ονομάζεται ο,τιδήποτε αλληλεπιδρά με τον πράκτορα.
- **Μεταβατικό μοντέλο S*=T(s→ α):** μετασχηματίζει την υπάρχουσα συσχέτιση **s→ α** σε μια νέα κατάσταση **S***.

Ας υποθέσουμε ότι η RL μάθηση αποτελεί ένα σύστημα με είσοδο την ισχύουσα κατάσταση **S=(s, e)** όπως την αντιλαμβάνεται ο πράκτορας. Μέσω του μεταβατικού μοντέλου **T,** το σύστημα θα οδηγήσει στην έξοδό του στην επόμενη κατάσταση, έστω **S***. Τέλος, υπάρχει και μια ανάδραση η οποία δεν είναι άλλη από την ανταμοιβή **r**, η οποία αποτελεί συνάρτηση πολλών παραγόντων όπως της χρησιμότητας της κατάστασης στην έξοδο του συστήματος, της τιμής στην οποία τη μεταφράζει ο πράκτορας, του κέρδους και της ζημίας, δηλαδή **r=F(χρησιμότητα, αξία, κέρδος, ζημία)**.



Σχήμα 4.2 Στοιχεία κλασικής RL

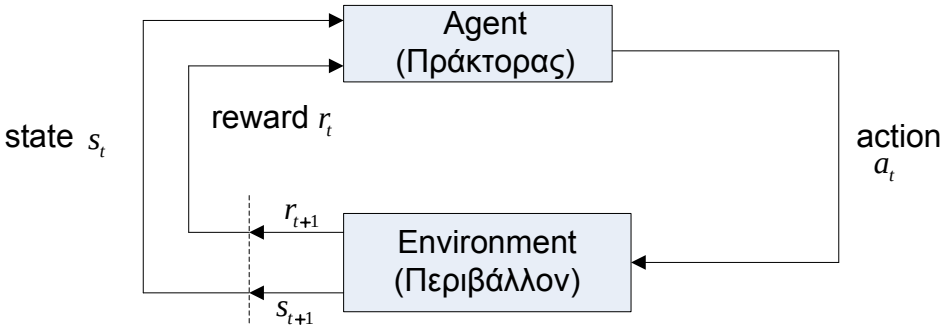
Πλέον, γίνεται ξεκάθαρη η βασική αρχή της RL μάθησης που αναφέραμε στην προηγούμενη παράγραφο (4.1):

- **Βασική Αρχή :** Η τάση να λαμβάνεται μια δράση **α** σε μια κατάσταση **S**, θα έπρεπε να *ενισχύεται* αν παράγει επιθυμητά αποτελέσματα και να *αποδυναμώνεται* αν παράγει ανεπιθύμητα.

Ο πράκτορας και το περιβάλλον του αλληλεπιδρούν σε κάθε αλληλουχία βημάτων διακριτού χρόνου $t = 0,1,2,\ldots$. Σε κάθε βήμα t, ο πράκτορας λαμβάνει κάποια εικόνα της κατάστασης του περιβάλλοντος, έστω $s_t \in S$, όπου S είναι το σύνολο των δυνατών καταστάσεων και με βάση αυτή επιλέγει μια δράση $a_t \in A(s_t)$, όπου $A(s_t)$ είναι το σύνολο των δυνατών καταστάσεων s_t. Στο επόμενο βήμα, ως συνέπεια της επιλογής του, ο πράκτορας δέχεται μια ανταμοιβή, έστω $r_{t+1} \in R$, όπου R το σύνολο των δυνατών ανταμοιβών και μεταβαίνει σε μια καινούργια κατάσταση s_{t+1}.[14]

Σε κάθε χρονικό βήμα, η ισχύουσα κατάσταση **s**, "μεταφράζεται" σε μια πιθανότητα για κάθε μια δυνατή επιλογή. Δηλαδή το γεγονός ότι ισχύει $s_t = s$ για το χρονικό βήμα t μας οδηγεί στην επιλογή της δράσης που συγκεντρώνει τις περισσότερες πιθανότητες να επιλεγεί για τη δεδομένη κατάσταση s_t και που είναι, έστω, η $a_t = a$. Η αντιστοιχία αυτή της δεδομένης κατάστασης σε μια συγκεκριμένη δράση (**s→ α**) λαμβάνει χώρα μέσω μιας διαδικασίας αντιστοίχησης μεταξύ των δύο συνόλων S

και $A(s_t)$ που δεν είναι άλλη από την έννοια που παραπάνω ορίσαμε ως *πολιτική* του πράκτορα και που συμβολίζεται με π_t για τη χρονική στιγμή t. Είναι, λοιπόν σαν η συνάρτηση $\pi_t(s_t, a_t)$ να είναι η πιθανότητα να προχωρήσει στην πράξη a_t αν απ' το περιβάλλον λαμβάνει σήμα ότι βρίσκεται στην κατάσταση s_t.



Σχήμα 4.3 Αλληλεπίδραση πράκτορα-περιβάλλοντος στην ενισχυτική μάθηση

Συνοψίζοντας τη διαδικασία που ακολουθεί ο πράκτορας κατά τη διαδικασία της RL:

Πίνακας 4.1 Αλγόριθμος πράκτορα, για διακριτά χρονικά βήματα : t=0,1,2,…

Ο πράκτορας :

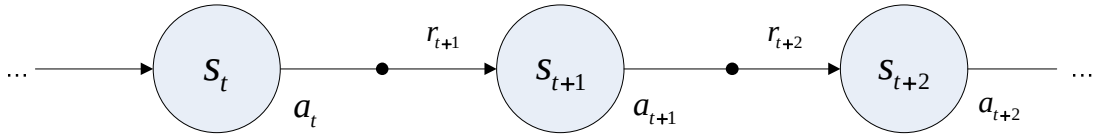
1. Παρατηρεί την ισχύουσα κατάσταση $s_t \in S$ τη χρονική στιγμή t

2. Προχωρά στην υλοποίηση δράσης $a_t \in A(s_t)$ τη χρονική στιγμή t βάσει της s_t

3. Λαμβάνει μια ανταμοιβή $r_{t+1} \in R$ τη χρονική στιγμή $t+1$

4. Μεταβαίνει σε επόμενη κατάσταση $s_{t+1} \in S$ τη χρονική στιγμή $t+1$ και η διαδικασία ξεκινάει και πάλι απ' την αρχή (βήμα 1)

Μια αναπαράσταση του αλγορίθμου του πίνακα 4.1 είναι η εξής:



Σχήμα 4.4 Αναπαράσταση Αλγορίθμου RL μεθόδου για κάθε πράκτορα

Αυτό που αξίζει να επισημάνουμε στο σχήμα 4.4 είναι το γεγονός ότι η ανταμοιβή που προέρχεται από την επιλογή μιας δράσης, ναι μεν έρχεται αμέσως μετά, ωστόσο θεωρούμε για τις ανάγκες του μοντέλου ότι ο πράκτορας τη λαμβάνει στις αρχές του επόμενου γύρου προκειμένου να μπορέσει να επηρεάσει τη διαμόρφωση της νέας κατάστασης και εν τέλει τη λήψη της νέας απόφασης.

Η δομή της RL όπως φαίνεται στο σχήμα 4.3 είναι αρκετά ευέλικτη και μπορεί να εφαρμοστεί σε πολλά διαφορετικά προβλήματα και με παραπάνω από έναν τρόπους. Για παράδειγμα, τα διακριτά βήματα t δεν είναι ανάγκη να αναφέρονται σε πραγματικό χρόνο, αλλά μπορεί να αναφέρονται σε διαδοχικές φάσεις αποφάσεων ή δράσεων, αυθαίρετα επιλεγμένες. Επίσης, οι δράσεις μπορεί να απαιτούν είτε χαμηλού επιπέδου έλεγχο, όπως οι τάσεις που εφαρμόζονται σε έναν κινητήρα, είτε

υψηλού επιπέδου αποφάσεις, όπως η απάντηση στο ερώτημα για το αν θα υλοποιηθεί μια δράση ή όχι. Ομοίως, οι καταστάσεις $s_t \in S$ μπορούν να είναι διαφόρων τύπων. Αρκετά στοιχεία που καθορίζουν την κατάσταση s_t μπορεί να βασίζονται σε εμπειρίες, παλιά γεγονότα ή και να είναι τελείως υποκειμενικά. Λόγου χάρη, ένας πράκτορας θα μπορούσε να είναι στην κατάσταση της αβεβαιότητας ως προς κάτι, ή να είναι και τελείως ξαφνιασμένος από αυτό. Αντίστοιχα, κάποιες δράσεις ίσως είναι απόρροια υπολογιστικών ή νοητικών διεργασιών. Παραδείγματος χάρη, άλλες μπορεί να ελέγχουν τι ένας πράκτορας επιλέγει να σκεφτεί και άλλες το που επικεντρώνει την προσοχή του. Εν γένει, θα λέγαμε ότι δράση είναι οποιαδήποτε απόφαση λαμβάνουμε και κατάσταση ο,τιδήποτε μπορούμε να γνωρίζουμε που θα μας είναι χρήσιμο στο να τη λάβουμε.

Το επόμενο στοιχείο της δομής της RL που πρέπει να διασαφηνίσουμε είναι το περιβάλλον. Γενικά, ακολουθούμε τον κανόνα που ορίζει ως περιβάλλον ο,τιδήποτε δεν μπορεί να αλλάξει αυθαίρετα απ' την πλευρά του πράκτορα. Ωστόσο δε θεωρούμε πως οποιοδήποτε στοιχείο ανήκει στο περιβάλλον είναι άγνωστο στον πράκτορα. Για παράδειγμα, ο πράκτορας συχνά γνωρίζει κάτι ως προς τον τρόπο υπολογισμού της ανταμοιβής του συναρτήσει των δράσεων του και των καταστάσεων. Αλλά πάντα θεωρούμε την υπολογιστική ανταμοιβή ως στοιχείο του περιβάλλοντος του πράκτορα διότι δε δύναται να αλλάξει αυθαίρετα από εκείνον. Στην πραγματικότητα, σε μερικές περιπτώσεις ο πράκτορας ξέρει τα πάντα για τον τρόπο που λειτουργεί το περιβάλλον του και εξακολουθεί να αντιμετωπίζει ένα δύσκολο πρόβλημα προς επίλυση, εντός του πλαισίου της RL. Στην πραγματικότητα, λοιπόν, το σύνορο μεταξύ πράκτορα και περιβάλλοντος αντιπροσωπεύει το όριο του *απόλυτου ελέγχου* του πράκτορα και όχι των γνώσεών του. [14]

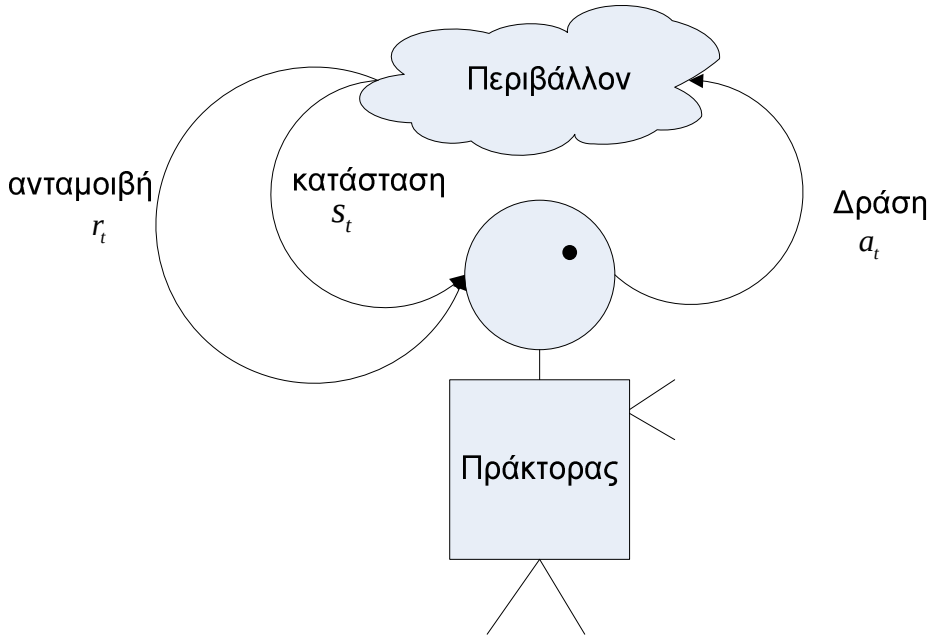
Το σύνορο μεταξύ πράκτορα και περιβάλλοντος μπορεί να τοποθετηθεί σε διαφορετικό σημείο για διαφορετικούς σκοπούς. Σε ένα πολύπλοκο ρομποτικό σύστημα, πολλοί διαφορετικοί πράκτορες μπορεί να λειτουργούν παράλληλα, ο καθένας με το δικό του 'σύνορο'. Λόγου χάρη, ένας πράκτορας μπορεί να λαμβάνει υψηλού επιπέδου αποφάσεις οι οποίες σχηματίζουν τις καταστάσεις για τις αποφάσεις που θα κληθεί να πάρει ένας κατώτερου επιπέδου πράκτορας. Στην πράξη, το όριο μεταξύ πράκτορα και περιβάλλοντος καθορίζεται μόλις ένας απ' τους πράκτορες επιλέξει μια κατάσταση, προχωρήσει σε μια δράση και λάβει την αντίστοιχη ανταμοιβή. Κι αυτό γιατί τότε τίθενται τα όρια το πρώτου προβλήματος και με βάση αυτά μπορούν να τεθούν και τα υπόλοιπα.

Το μοντέλο της RL βασίζεται στη γενίκευση ότι ανεξάρτητα των λεπτομερειών των συστημάτων αντίληψης, μνήμης και ελέγχου, καθώς και του στόχου που κάποιος προσπαθεί να πετύχει, οποιοδήποτε πρόβλημα μαθησιακής συμπεριφοράς με σκοπό την επίτευξη ενός στόχου μπορεί να αναχθεί σε τρία σήματα που κινούνται μπρος και πίσω μεταξύ ενός πράκτορα και του περιβάλλοντός του: ένα σήμα που αναπαριστά τις επιλογές του πράκτορα (δράσεις), ένα που αναπαριστά τη βάση πάνω στην οποία στηρίζονται οι επιλογές (καταστάσεις) και ένα σήμα που προσδιορίζει το στόχο του πράκτορα (ανταμοιβή). Το μοντέλο αυτό, μπορεί να μην αναπαριστά όλα τα προβλήματα αποφάσεων, αποδεικνύεται όμως ευρέως χρήσιμο και εφαρμόσιμο.

Φυσικά οι καταστάσεις και οι δράσεις διαφέρουν πολύ από εφαρμογή σε εφαρμογή και ο τρόπος με τον οποίο αναπαρίστανται μπορεί να επηρεάσει σημαντικά την

απόδοση του μοντέλου. Στην ενισχυτική μάθηση, όπως συμβαίνει και με άλλα είδη μάθησης, η αναπαράσταση τέτοιων επιλογών φαντάζει, προς το παρόν, πιο πολύ τέχνη παρά επιστήμη.

Τέλος, οι καταστάσεις s_t και οι ανταμοιβές r μοντελοποιούνται ως εξωτερικές δυνάμεις οι οποίες καθορίζουν τις επιλογές δράσης ενός πράκτορα. Το τελικό μοντέλο RL για έναν πράκτορα θα μπορούσαμε να το αναπαραστήσουμε ως εξής:



Σχήμα 4.5 RL Μοντέλο: οι καταστάσεις s και οι ανταμοιβές r μοντελοποιούνται ως εξωτερικές δυνάμεις που καθορίζουν τη δράση a του πράκτορα.

4.3 Στόχοι και Ανταμοιβές

Στην ενισχυτική μάθηση, η επιδίωξη ή ο στόχος του πράκτορα παρουσιάζονται στο μοντέλο ως ένα ειδικό σήμα ανταμοιβής το οποίο ξεκινά απ' το περιβάλλον και καταλήγει στον πράκτορα. Σε κάθε βήμα t, η ανταμοιβή είναι απλώς ένας αριθμός $r_{t+1} \in R$, όπου R το σύνολο των δυνατών ανταμοιβών. Ουσιαστικά, ο στόχος του πράκτορα είναι να μεγιστοποιήσει το συνολικό ποσό ανταμοιβών που λαμβάνει, δηλαδή όχι απαραίτητα να μεγιστοποιήσει την άμεση ανταμοιβή του, αλλά την αθροιστική που θα λάβει μακροπρόθεσμα. [14]

Η χρησιμοποίηση του σήματος ανταμοιβής για να υλοποιήσει στο μοντέλο της την έννοια του στόχου, αποτελεί ένα απ' τα πιο ιδιαίτερα χαρακτηριστικά της RL. Παρόλο που ο τρόπος αυτός αναπαράστασης μπορεί να φαντάζει, σε πρώτη φάση, περιοριστικός, στην πράξη αποδεικνύεται ευέλικτος και εύκολα εφαρμόσιμος. Παραδείγματος χάρη, για να κάνουν ένα ρομπότ να περπατήσει, οι ερευνητές παρέχουν μια ανταμοιβή ανάλογη της μπροστινής κίνησης του ρομπότ. Για να οδηγήσουν το ρομπότ να αποφύγει ένα εμπόδιο, η ανταμοιβή είναι 0 έως ότου ξεφύγει, οπότε και γίνεται 1. Μια άλλη συνήθης τακτική είναι το να δίνουν ανταμοιβή -1 για κάθε χρονικό βήμα που παρέρχεται χωρίς να ξεφεύγει απ' το εμπόδιο. Αυτό ενθαρρύνει το ρομπότ να ψάξει νέες κινήσεις με τις οποίες θα ξεφύγει

το γρηγορότερο δυνατό. Αντίστοιχα, για έναν πράκτορα που μαθαίνει να παίζει σκάκι, οι ανταμοιβές θα ήταν +1 για τη νίκη, -1 για την ήττα και 0 για την ισοπαλία.

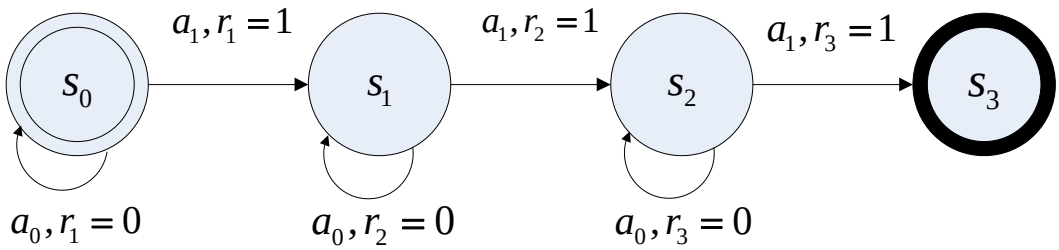
Όπως γίνεται αντιληπτό απ' τα παραπάνω παραδείγματα, ο πράκτορας πάντα επιδιώκει να μεγιστοποιήσει την ανταμοιβή του. Αν λοιπόν θέλουμε να κάνει κάτι για εμάς, πρέπει να παρέχουμε στον πράκτορα ανταμοιβές με τέτοιο τρόπο ώστε να τον ενθαρρύνουμε να κινηθεί προς την κατεύθυνση που επιθυμούμε και πετυχαίνοντας το στόχο του να πετύχουμε κι εμείς το δικό μας. Υπάρχει ένα σημείο όμως που χρήζει προσοχής : το σήμα ανταμοιβής δεν πρέπει να επιβραβεύει παρά μόνο την επίτευξη του τελικού στόχου που επιθυμούμε να πετύχει ο πράκτορας. Αν επιβραβεύει και όλους τους μικρότερους στόχους που επιτυγχάνονται, υπάρχει κίνδυνος ο πράκτορας να επιδιώξει να μεγιστοποιήσει την ανταμοιβή του μέσω αυτών και όχι να στραφεί προς τον έναν τελικό στόχο.

Ίσως προκαλεί εντύπωση το γεγονός ότι οι ανταμοιβές, οι οποίες καθορίζονται απ' το στόχο της διαδικασίας μάθησης, υπολογίζονται απ' το περιβάλλον και όχι απ' τον πράκτορα. Στην παράγραφο 4.2 δώσαμε εκτενώς μια περιγραφή του περιβάλλοντος στο μοντέλο της RL και τονίσαμε ότι ως περιβάλλον ορίζουμε ο,τιδήποτε ο πράκτορας δεν μπορεί να ελέγξει. Στο παράδειγμα της ενισχυτικής μάθησης με το ζώο που επιχειρούμε να το στρέψουμε προς μια συγκεκριμένη συμπεριφορά, είχαμε πει ότι η ανταμοιβή στην ουσία δεν έγκειται στο φαΐ που του δίνουμε αλλά στην οργανική διαδικασία που μετατρέπει την τροφή σε αίσθηση ικανοποίησης στο ζώο, μέσω των αισθητήρων του. Θα μπορούσε, λοιπόν, να επιχειρηματολογήσει κανείς ότι η εσωτερική αυτή διεργασία δεν ανήκει στο περιβάλλον αλλά στο ίδιο το ζώο-πράκτορα. Σύμφωνα όμως με τον ορισμό του περιβάλλοντος, ακόμα και τα μέλη του σώματος του ζώου λογίζονται ως περιβάλλον διότι δεν μπορεί να τα ελέγξει άμεσα. Πρέπει να τονιστεί, άρα, ότι για κάθε πράκτορα, το σύνορο μεταξύ του ιδίου και του περιβάλλοντός του δεν αποτελείται από τα όρια του σώματός του, αλλά από τα όρια του ελέγχου του.

Ο λόγος που συμβαίνει αυτό είναι γιατί ο πράκτορας θα πρέπει να θεωρεί ως τελικό στόχο του κάτι πάνω στο οποίο δεν έχει απόλυτο έλεγχο. Έτσι, τοποθετούμε την πηγή της ανταμοιβής εκτός του πράκτορα. Αυτό ωστόσο δεν τον αποτρέπει από το να αναπτύξει το δικό του, εσωτερικό, σύστημα ανταμοιβών ή μια σειρά από ανταμοιβές. Πράγματι, όπως θα δούμε, οι μέθοδοι ενισχυτικής μάθησης ακολουθούν ακριβώς αυτή την αρχή.

Ως παράδειγμα επίδειξης του τρόπου με τον οποίο μπορούν να λειτουργήσουν οι ανταμοιβές, αλλά και αυτού με τον οποίο εμείς μπορούμε να τις χρησιμοποιήσουμε για να κατευθύνουμε τον πράκτορα, υποθέτουμε τα εξής: έστω μια αγορά, μέσα στην οποία θέλουμε να εντάξουμε έναν υπολογιστικό πράκτορα. Σκοπός μας είναι να κατευθύνουμε τον πράκτορα προς μια τελική κατάσταση, έστω s_3, η οποία μπορεί να αντιπροσωπεύει μια οποιαδήποτε κατάσταση της αγοράς που στα πλαίσια του παραδείγματος αυτού μας αφήνει αδιάφορους. Ο πράκτορας θα πρέπει να ξεκινήσει από μια κατάσταση s_0 και μέσω ενός συστήματος ανταμοιβών που θα προέρχεται απ' το περιβάλλον του να κατευθυνθεί προς την κατάσταση s_3. Προκειμένου να πετύχει το στόχο του, που δεν είναι άλλος απ' το να μεγιστοποιήσει τη μακροπρόθεσμη ανταμοιβή του, ο πράκτορας κάθε φορά που λαμβάνει ανταμοιβή 0 θα επιχειρεί να

κατευθυνθεί προς άλλη κατεύθυνση-δράση που θα του επιφέρει ανταμοιβή 1.Το πρόβλημα αυτό μπορεί να περιγραφεί απ' το ακόλουθο διάγραμμα καταστάσεων :



Σχήμα 4.6 Διάγραμμα καταστάσεων για το προαναφερθέν παράδειγμα: μέσω του συστήματος ανταμοιβών ο πράκτορας οδηγείται τελικά στην επιθυμητή κατάσταση.

Κάθε φορά που ο πράκτορας πραγματοποιεί την δράση a_0, δηλαδή παραμένει στην προηγούμενη επιλογή του λαμβάνει ανταμοιβή 0, ενώ όταν πραγματοποιεί την a_1 και μεταβαίνει στην επόμενη λαμβάνει ανταμοιβή 1. Η επιλογή να γυρίσει σε μια απ' τις προηγούμενες καταστάσεις απ' αυτή που ήδη βρίσκεται θεωρούμε ότι δεν υπάρχει λόγω του ότι πρόκειται για χρονικά βήματα, δηλαδή δεν μπορεί να γυρίσει το χρόνο πίσω και να επιλέξει κάτι διαφορετικό. Φυσικά, στην πραγματικότητα, μετά από κάθε κατάσταση s_t υπάρχει πληθώρα καταστάσεων και όχι μόνο μια s_{t+1}, αλλά πρόκειται για απλούστευση χάριν παραδείγματος.

4.4 Αποκρίσεις

Μέχρι στιγμής, λέγοντας ότι σκοπός του πράκτορα είναι να μεγιστοποιήσει τη μακροπρόθεσμη ανταμοιβή του, ήμαστε ανακριβείς. Ας δώσουμε έναν πιο τυπικό ορισμό του σκοπού του. Αν η αλληλουχία ανταμοιβών που λαμβάνονται έπειτα από ένα χρονικό βήμα t είναι $r_{t+1}, r_{t+2}, \ldots$, τότε αυτό που επί της ουσίας προσπαθεί να μεγιστοποιήσει ο πράκτορας είναι η *αναμενόμενη απόκριση*, όπου η απόκριση R_t ορίζεται ως μια συνάρτηση της αλληλουχίας ανταμοιβών. Στην απλούστερη των περιπτώσεων δίνεται απ' τον τύπο:

$$R_t = r_{t+1} + r_{t+2} + \ldots + r_T, \quad (4.1)$$

όπου T το τελευταίο χρονικά βήμα. Αυτή η προσέγγιση έχει νόημα για εφαρμογές όπου έχει νόημα να μιλάμε για τελικό χρονικό βήμα, δηλαδή όταν η αλληλεπίδραση πράκτορα-περιβάλλοντος μπορεί να 'σπάσει' σε μεμονωμένα κομμάτια, όπως οι γύροι ενός παιχνιδιού ή οποιαδήποτε επαναλαμβανόμενη διαδικασία. Τα κομμάτια αυτά λέγονται *επεισόδια* και κάθε επεισόδιο τελειώνει σε μια ιδιαίτερη κατάσταση, που την καλούμε τερματική κατάσταση, η οποία έπεται από επανεκκίνηση από μια συγκεκριμένη αρχική κατάσταση. Τέτοιες εργασίες με επεισόδια λέγονται *επεισοδιακές*. Σε επεισοδιακά προβλήματα, καμία φορά χρειάζεται να διαχωρίσουμε το σύνολο των μη τερματικών καταστάσεων S, από το σύνολο των καταστάσεων συν την τερματική, που συμβολίζουμε με S^+.

Σε πολλές, όμως, περιπτώσεις, η αλληλεπίδραση πράκτορα-περιβάλλοντος δεν μπορεί να σπάσει με φυσικό τρόπο σε αναγνωρίσιμα επεισόδια, αλλά συνεχίζεται χωρίς όριο. Τέτοια προβλήματα τα λέμε *συνεχή*. Για συνεχείς εργασίες, η μοντελοποίηση της

απόκρισης είναι προβληματική διότι το τερματικό χρονικό βήμα θα ήταν το $T = \infty$ και η απόκριση, την οποία θέλουμε να μεγιστοποιήσουμε, θα ήταν άπειρη. Έτσι, είμαστε αναγκασμένοι να υιοθετήσουμε μια προσέγγιση που την καλούμε *έκπτωση*. Σύμφωνα με αυτή, ο πράκτορας προσπαθεί να επιλέξει δράσεις με σκοπό να μεγιστοποιήσει της επί έκπτωση ανταμοιβές που λαμβάνει. Συγκεκριμένα, επιλέγει μια δράση a_t για να μεγιστοποιήσει την αναμενόμενη *επί έκπτωση απόκριση*:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad , \quad (4.2)$$

όπου γ είναι μια παράμετρος , $0 \le \gamma \le 1$, και λέγεται *εκπτωτική παράμετρος*.

Η παράμετρος αυτή καθορίζει την παρούσα αξία των μελλοντικών ανταμοιβών: μια ανταμοιβή που θα ληφθεί σε k χρονικά βήματα στο μέλλον, θα αξίζει μόνο γ^{k-1} φορές ότι θα άξιζε στο παρόν. Αν $\gamma < 1$, το άπειρο άθροισμα έχει μια πεπερασμένη τιμή όσο η αλληλουχία ανταμοιβών $\{r_k\}$ είναι φραγμένη. Αν $\gamma = 0$, ο πράκτορας είναι 'μυωπικός' στο να βλέπει μόνο τις άμεσες ανταμοιβές: ο σκοπός του στην περίπτωση αυτή είναι να μάθει πώς να επιλέξει μια a_t ώστε να μεγιστοποιήσει μόνο την ανταμοιβή r_{t+1}. Η λογική αυτή, της επιδίωξης μόνο της μεγιστοποίησης της επόμενης ανταμοιβής r_{t+1} μέσω της δράσης a_t, θα μας επέφερε μέγιστη συνολική απόκριση μόνο με την προϋπόθεση ότι η δράση a_t δεν επηρεάζει καμία απ' τις μελλοντικές ανταμοιβές, παρά μόνο αυτή που έγκειται σε εκείνη. Γενικά, ωστόσο, η μεγιστοποίηση της αμέσως επόμενης ανταμοιβής, συνήθως μειώνει τις επόμενες ανταμοιβές κι έτσι δεν είναι καθόλου βέβαιο ότι και η τελική απόκριση, δηλαδή το άθροισμά τους, θα είναι το μέγιστο. Αν, τέλος, $\gamma = 1$, ο πράκτορας δίνει μεγαλύτερη έμφαση στο μέλλον μιας και οι μελλοντικές ανταμοιβές έχουν το μεγαλύτερο βάρος. [14]

4.5 Εγγενή-Εξωγενή Κίνητρα Πράκτορα για Επιλογή Δράσης

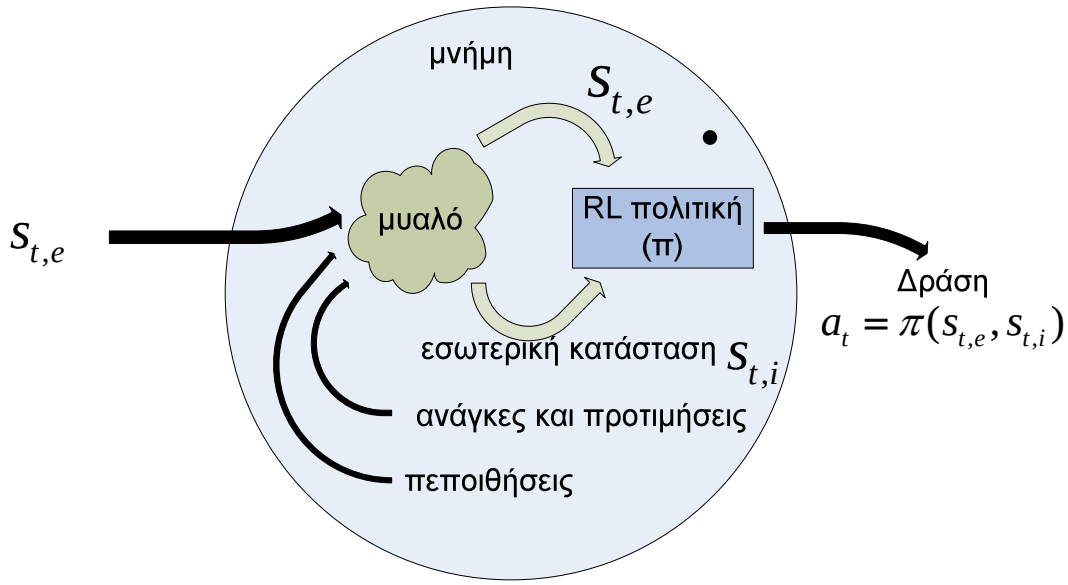
Μέχρι στιγμής, κάναμε λόγο για ανταμοιβές στα συστήματα ενισχυτικής μάθησης, αλλά και για τα όρια μεταξύ πράκτορα και περιβάλλοντος. Ωστόσο δεν μπορούμε να αρκεστούμε στο πώς δίνουμε κίνητρο σε ένα ζώο ή σε ένα ρομπότ. Συχνά, για να κατασκευάσουμε ένα υπολογιστικό σύστημα ενισχυτικής μάθησης, χρειάζεται να χρησιμοποιήσουμε πράκτορες με ανθρώπινη σκέψη και συμπεριφορά κάτι που καθιστά το όλο εγχείρημα πιο περίπλοκο. Είναι, συνεπώς, απαραίτητο να εξετάσουμε όλα τα προαναφερθέντα στοιχεία σε συμφωνία με αυτά που δίνουν κίνητρο στον άνθρωπο.

Οι παράγοντες που δίνουν κίνητρο σ' έναν άνθρωπο να δράσει, μπορούν να κατηγοριοποιηθούν σε εξωτερικά και εσωτερικά : [3]

- **Εξωτερικά κίνητρα:** καλούνται εκείνα που παρακινούν τον άνθρωπο να δράσει με την ελπίδα να λάβει κάποια ανταμοιβή απ' το περιβάλλον του. Τέτοια κίνητρα θεωρούνται η χρηματική ανταμοιβή, η επιβράβευση, κάποιο βραβείο κλπ.

- **Εσωτερικά κίνητρα:** καλούνται εκείνα που παρακινούν τον άνθρωπο να δράσει από έμφυτη επιθυμία. Τέτοια κίνητρα είναι εσωτερική ικανοποίηση, η απόλαυση, η περιέργεια, η τάση για εξερεύνηση, το αίσθημα ηθικής ευθύνης κλπ.

Με βάση αυτή την πολύ απλή κατηγοριοποίηση των ανθρώπινων κινήτρων μπορούμε να παρουσιάσουμε μια λίγο πιο σύγχρονη και πιο σύνθετη διαδικασία για τη λήψη οποιασδήποτε απόφασης που αντιστοιχεί σε μια συγκεκριμένη δράση. Αν μέχρι τώρα θεωρούσαμε την κάθε δράση αποτέλεσμα της επεξεργασίας των στοιχείων του περιβάλλοντος μέσω της πολιτικής του πράκτορα, δηλαδή $a_t = \pi_t(s_t)$, τώρα θεωρούμε ότι υπάρχουν δύο είδη καταστάσεων: ένα που αναφέρεται στα εξωτερικά κίνητρα $s_{t,e}$ κι ένα που αναφέρεται στα εσωτερικά κίνητρα $s_{t,i}$. Έτσι, ο πράκτορας λαμβάνει δυο ερεθίσματα καταστάσεων και μέσω της πολιτικής που υιοθετεί καταλήγει στην απόφασή του για δράση τη a_t, συνεπώς θα ισχύει $a_t = \pi_t(s_{t,e}, s_{t,i})$. Όσον αφορά στο πώς διαμορφώνεται η εσωτερική κατάσταση $s_{t,i}$, μπορούμε να θεωρήσουμε οτι αποτελεί τη συνιστώσα των εσωτερικών πεποιθήσεων, αναγκών, βιωμάτων και αξιών του πράκτορα. Αντίθετα, ως εξωτερική κατάσταση $s_{t,e}$ θεωρούμε ό,τι συμβολίζαμε μέχρι τώρα ως κατάσταση s_t, δηλαδή ο,τιδήποτε προέρχεται απ' το περιβάλλον του πράκτορα. Το σχήμα που ακολουθεί είναι κατατοπιστικό για να αντιληφθούμε αυτή τη διαδικασία λήψης αποφάσεων.



Σχήμα 4.7 Μια πιο σύγχρονη οπτική της διαδικασίας λήψης αποφάσεων του πράκτορα με βάση τα εσωτερικά-εξωτερικά κίνητρα.

Το ερώτημα βέβαια που τίθεται στο σημείο αυτό και που απασχολεί αρκετά τους επιστήμονες που ασχολούνται κυρίως με τον τομέα της τεχνητής νοημοσύνης, είναι το αν μπορεί ένα υπολογιστικό σύστημα εκμάθησης να είναι υποκινούμενο εσωτερικά. Πιο συγκεκριμένα, αν και κατά πόσο μπορεί να καταστεί ένας RL πράκτορας κατασκευασμένος στον υπολογιστή να υιοθετήσει τον τρόπο λήψης αποφάσεων μέσω των εσωτερικών του κινήτρων. Μια ιδιαίτερα ενδιαφέρουσα μελέτη έχουν δημοσιοποιήσει οι Satinder Singh, Richard Lewis, Andrew Barto και Jonathan Sorg το 2010 [15]. Ωστόσο, στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας, το ερώτημα αυτό δεν θα μας απασχολήσει άλλο. Τελικός σκοπός μας είναι να κατασκευάσουμε πράκτορες που να ενσωματώνονται σε ένα υπολογιστικό

μοντέλο ηλεκτρικής αγοράς ενέργειας και συνεπώς οποιαδήποτε πληροφορία παρουσιάζεται στο κεφάλαιο αυτό είναι υπέρ αρκετή για το σκοπό μας.

4.6 Ιδιότητα Markov

Στο βασικό μοντέλο της ενισχυτικής μάθησης (RL), είδαμε πως ο πράκτορας λαμβάνει τις αποφάσεις του ως συνάρτηση του σήματος που δέχεται απ' το περιβάλλον του και που ονομάσαμε κατάσταση s_t. Μάλιστα στην προηγούμενη παράγραφο 4.5, διαχωρίσαμε την κατάσταση αυτή σε δυο συνιστώσες της, τις $s_{t,e}$ και $s_{t,i}$. Στην παράγραφο αυτή θα εξετάσουμε το ποιες απαιτήσεις πρέπει να ικανοποιεί το σήμα κατάστασης και τί είδους πληροφορία πρέπει να αναμένουμε ή όχι να μας παρέχει. Συγκεκριμένα, πρέπει να δώσουμε ακριβή ορισμό σε μια ιδιότητα του περιβάλλοντος και του σήματος κατάστασής του που έχει ιδιαίτερο ενδιαφέρον και καλείται ιδιότητα Markov.

Στο πλαίσιο της παρούσης εργασίας, λέγοντας 'κατάσταση' εννοούμε οποιαδήποτε πληροφορία είναι διαθέσιμη στον πράκτορα. Υποθέτουμε ότι η κατάσταση είναι ένα δεδομένο που προέρχεται από κάποια διαδικασία που αποτελεί μέρος του περιβάλλοντος. Ακολουθούμε αυτή την προσέγγιση, όχι γιατί θεωρούμε την αναπαράσταση της κατάστασης ασήμαντη, αλλά για να μπορούμε να εστιάσουμε πλήρως σε ζητήματα καθαρά λήψης αποφάσεων. Με άλλα λόγια, το βασικό μας μέλημα δεν είναι να σχεδιάσουμε το σήμα κατάστασης, αλλά να λαμβάνουμε αποφάσεις ανάλογα με το σήμα κατάστασης που λαμβάνουμε.

Ιδανικά, θα θέλαμε να διατηρείται όλη η περασμένη πληροφορία που μας ενδιαφέρει και να εμπεριέχεται στο σήμα κατάστασης. Φυσικά, η αντίληψη και οι αισθήσεις του πράκτορα εκείνη τη στιγμή που λαμβάνει το σήμα δεν αρκεί για να έχει όλη αυτή την πληροφορία, αλλά απαιτείται ολόκληρο το ιστορικό τους. Ένα σήμα κατάστασης που επιτυγχάνει να διατηρεί όλη τη σχετική με αυτό πληροφορία λέγεται πως έχει την **ιδιότητα του Markov**.

Ας υποθέσουμε πως σ' ένα πρόβλημα ενισχυτικής μάθησης υπάρχει ένας πεπερασμένος αριθμός καταστάσεων και τιμών ανταμοιβής. Αυτό μας επιτρέπει να δουλέψουμε με όρους αθροισμάτων και πιθανοτήτων αντί ολοκληρωμάτων και πυκνοτήτων πιθανότητας, αλλά εν συνεχεία, τα συμπεράσματά μας μπορούν να γενικευθούν για συνεχείς χώρους καταστάσεων και ανταμοιβών. Θεωρούμε πως ένα γενικό περιβάλλον ανταποκρίνεται σε χρόνο $t+1$σε μια δράση που λήφθηκε σε χρόνο t. Στην πιο γενική και απλή περίπτωση, αυτή η απόκριση μπορεί να εξαρτάται από οτιδήποτε έχει συμβεί προηγουμένως. Σε αυτή την περίπτωση, η ολική κατανομή πιθανότητας είναι των δυναμικών στοιχείων είναι:

$$\Pr\left\{s_{t+1}=s', r_{t+1} \middle| s_t, a_t, r_t, s_{t-1}, a_{t-1}, ..., r_1, s_0, a_0\right\}, \quad (4.3)$$

για όλα τα s' και r και όλες τις δυνατές τιμές των παλαιών γεγονότων: $s_t, a_t, r_t, ...r_1, s_0, a_0$. Αν το σήμα κατάστασης έχει την ιδιότητα Markov, τότε η απόκριση του περιβάλλοντος τη χρονική στιγμή $t+1$ εξαρτάται μόνο από την κατάστασης και τη δράση τη χρονική στιγμή t, κατά την οποία τα δυναμικά των στοιχείων του περιβάλλοντος μπορούν να περιγραφούν μόνο απ' την:

$$\Pr\left\{ s_{t+1} = s', r_{t+1} = r \middle| s_t, a_t \right\}, \quad (4.4)$$

για όλα τα s', r, s_t και a_t. Με άλλα λόγια, ένα σήμα κατάστασης έχει την ιδιότητα Markov και λέγεται **μαρκοβιανό** αν και μόνο αν η (4.3) είναι ίση με την (4.4) για όλα τα s', r και τα ιστορικά δεδομένα $s_t, a_t, r_t, ...r_1, s_0, a_0$. Σε αυτή την περίπτωση λέμε πως το περιβάλλον καθώς και όλο το πρόβλημα έχει την ιδιότητα Markov.

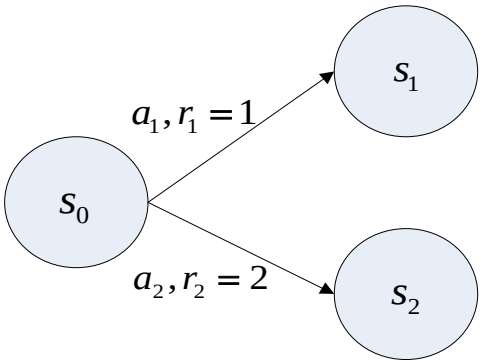
Αν ένα περιβάλλον έχει την ιδιότητα Markov, τότε η εξίσωση (4.4) μας επιτρέπει να προβλέψουμε την κατάσταση και την αναμενόμενη ανταμοιβή στο αμέσως επόμενο βήμα, δεδομένου της τωρινής κατάστασης και δράσης. Μπορεί να δειχθεί ότι επαναλαμβάνοντας της εξίσωση (4.4) είναι δυνατόν να προβλέψει κανείς όλες τις μελλοντικές καταστάσεις και τις ανταμοιβές γνωρίζοντας μόνο την τωρινή κατάσταση ή το σύνολο του ιστορικού των δράσεων, των ανταμοιβών και των καταστάσεων μέχρι και το χρονικό βήμα που βρίσκεται. Είναι, λοιπόν, επακόλουθο ότι οι καταστάσεις Markov αποτελούν την καλύτερη δυνατή βάση για την επιλογή δράσεων. Συνεπώς, *η καλύτερη πολιτική για επιλογή δράσεων ως συνάρτηση της κατάστασης Markov είναι ακριβώς ισοδύναμη με την καλύτερη πολιτική για επιλογή δράσεων ως συνάρτηση του συνόλου των ιστορικών δεδομένων*. [14]

Ακόμη, όμως, κι όταν η κατάσταση δεν είναι μαρκοβιανή, είναι βολικό να σκεφτόμαστε την κατάσταση στην ενισχυτική μάθηση ως προσεγγιστικά μαρκοβιανή. Πάντα θέλουμε η κατάσταση να αποτελεί μια καλή βάση για την πρόβλεψη των μελλοντικών ανταμοιβών και για την επιλογή των επόμενων δράσεων. Σε περιπτώσεις που αντικείμενο μάθησης είναι το μοντέλο του περιβάλλοντος και πάλι η κατάσταση θέλουμε να είναι η βάση για την πρόβλεψη των επερχόμενων καταστάσεων. Οι καταστάσεις Markov αποτελούν μια αδιαμφισβήτητη βάση για να πετύχουμε όλα τα παραπάνω. Στο σημείο που μια κατάσταση προσεγγίζει την κατάσταση Markov, τα συστήματα ενισχυτικής μάθησης είναι σε θέση αποδίδουν καλύτερα. Για όλους αυτούς τους λόγους, στη συνέχεια, θα θεωρούμε τις καταστάσεις σε κάθε χρονικό βήμα t προσεγγιστικά μαρκοβιανές, παρ' ότι θα έχουμε πάντα στο μυαλό μας πως δεν μπορούν να ικανοποιούν πλήρως την ιδιότητα Markov.

Η σημασία της ιδιότητας Markov στην ενισχυτική μάθηση είναι ακόμη μεγαλύτερη αν αναλογιστεί κανείς ότι οι αποφάσεις και οι τιμές θεωρούνται συνάρτηση μόνο της παρούσης κατάστασης. Προκειμένου αυτές οι τιμές και οι αποφάσεις να είναι αποτελεσματικές και χρήσιμες για το μοντέλο μας, θα πρέπει να οι καταστάσεις να εμπεριέχουν πληροφορία. Επίσης, η θεωρία Markov δεν χρησιμεύει μόνο στην περίπτωση που ικανοποιούνται όλες οι απαιτήσεις της, αλλά τα συμπεράσματα που προκύπτουν μέσω αυτής μπορούν να γενικευθούν και να εφαρμοστούν και σε πιο ρεαλιστικές και πιο σύνθετες καταστάσεις. Τέλος, η αναπαράσταση των καταστάσεων ως μαρκοβιανές, δεν εφαρμόζεται μόνο στην RL, αλλά και σε άλλες προσεγγίσεις και αλγορίθμους του κλάδου της τεχνητής νοημοσύνης.

4.7 Μαρκοβιανές Διαδικασίες Αποφάσεων

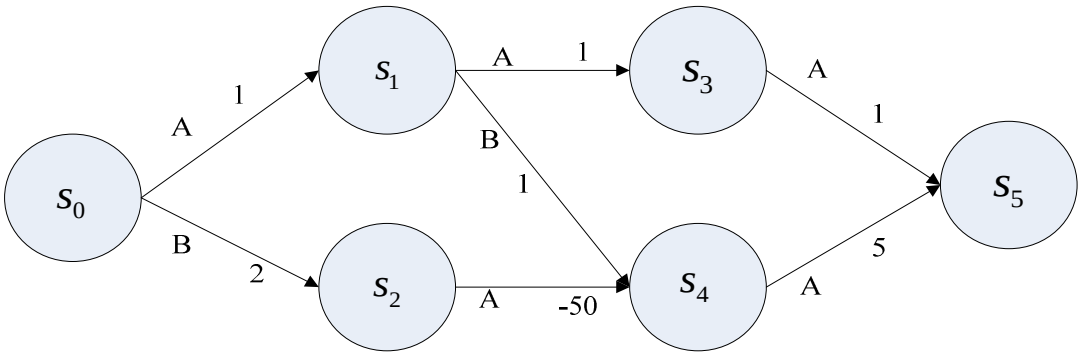
Έστω, όμοια με το παράδειγμα του σχήματος 3.5, ότι ένας πράκτορας βρίσκεται σε μια συγκεκριμένη κατάσταση, s_0 και έχει να επιλέξει μεταξύ, ας υποθέσουμε, 2 δρόμων που οδηγούν σε δυο νέες καταστάσεις s_1 και s_2. Κάθε κατάσταση έχει εξ ορισμού μια ανταμοιβή που αντιστοιχεί στην επιλογή της. Έστω, ότι η κατάσταση s_1 συνοδεύεται από ανταμοιβή 1 και η s_2 από ανταμοιβή 2:



Σχήμα 4.8 Λήψη απλών αποφάσεων.

Υπάρχουν αρκετές θεωρίες για τη λήψη αποφάσεων, ωστόσο, εμείς θα θεωρήσουμε την πιο απλή και τη συνηθέστερη όλων. Έτσι, θα θεωρούμε πάντα ότι μεταξύ όλων των επιλογών ο πράκτορας θα επιλέγει αυτή ή μια εξ αυτών με τη μεγαλύτερη για εκείνον ανταμοιβή. Η θεωρία αυτή καλείται *κριτήριο μέγιστου κέρδους*. Επίσης, ορίζουμε ως *αξία* μιας κατάστασης το άθροισμα των μέγιστων ανταμοιβών απ' την κατάσταση αυτή και μετά. Για παράδειγμα, στο σχήμα 4.8 η αξία της κατάστασης s_0 είναι 2. [13]

Αν, τώρα, γενικεύσουμε το προηγούμενο παράδειγμα κατασκευάζοντας ένα διάγραμμα καταστάσεων με πολλαπλές διαδοχικές καταστάσεις, όπου κάθε κατάσταση επηρεάζει τις επακόλουθες αποφάσεις (δηλαδή έχει την ιδιότητα Markov), θα κατασκευάσουμε ένα μοντέλο που λέγεται **Μαρκοβιανή διαδικασία αποφάσεων** (*Markov Decision Process* ή **MDP**):



Σχήμα 4.9 μια Μαρκοβιανή διαδικασία αποφάσεων (MDP).

Τυπικά μια MDP αποτελείται από :

- **Ένα σύνολο καταστάσεων** $S = \{s_1, s_2, ..., s_n\}$
- **Ένα σύνολο δράσεων** $A = \{a_1, a_2, ..., a_m\}$

- **Μια συνάρτηση ανταμοιβής** $R: S \times A \times S \to \Re$
- **Μια συνάρτηση μεταφοράς** $P_{ij}^{a} = P(s_{t+1} = j \,|\, s_t = i, a_t = a)$ ή μερικές φορές $T: S \times A \to S$

Ένα πεπερασμένο MDP ορίζεται απ' τα σύνολα καταστάσεων και δράσεών του καθώς και από τα δυναμικά χαρακτηριστικά του περιβάλλοντος που μεταβάλλονται από βήμα σε βήμα. Δεδομένης οποιασδήποτε κατάστασης s_t και δράσης a_t, η πιθανότητα να είναι επόμενη κατάσταση η s' δίνεται απ' τον τύπο:

$$\mathrm{P}^{\mathrm{a}}{}_{ss'} = \Pr\left\{s_{t+1} = s' \,|\, s_t = s, a_t = a\right\}, \quad (4.5)$$

Αυτές οι ποσότητες λέγονται *πιθανότητες μετάβασης* (εννοώντας μετάβαση απ' την s στην s'). Ομοίως, δεδομένης οποιασδήποτε ισχύουσας κατάστασης s_t και δράσης a_t, μαζί με την επόμενη κατάσταση s', η αναμενόμενη ανταμοιβή στον επόμενο γύρο είναι η:

$$\mathrm{R}^{\mathrm{a}}{}_{ss'} = E\left\{r_{t+1} \,|\, s_t = s, a_t = a, s_{t+1} = s'\right\}, \quad (4.6)$$

Οι ποσότητες $P^{a}{}_{ss'}$ και $R^{a}{}_{ss'}$ προσδιορίζουν πλήρως τα πιο σημαντικά χαρακτηριστικά των δυναμικών στοιχείων ενός πεπερασμένου MDP (χάνεται μονάχα η πληροφορία περί κατανομής των ανταμοιβών γύρω απ' την αναμενόμενη τιμή). Εφεξής, θα θεωρούμε το περιβάλλον ως πεπερασμένο MDP.

Το ζητούμενο είναι να μάθουμε μια **πολιτική** $\pi: S \to A$ η οποία να μεγιστοποιεί το άθροισμα των ανταμοιβών που λαμβάνουμε στο τέλος της διαδρομής μας.
Για το MDP του σχήματος 3.8 υπάρχουν 3 πολιτικές για να φτάσει κανείς στην τελική κατάσταση 5:

1. $s_0 \to s_1 \to s_3 \to s_5$
2. $s_0 \to s_1 \to s_4 \to s_5$
3. $s_0 \to s_2 \to s_4 \to s_5$

Το ερώτημα που πρέπει να απαντήσουμε είναι ποια απ' τις 3 πολιτικές είναι η 'καλύτερη';
Σύμφωνα με το κριτήριο που αναφέραμε παραπάνω, ως βέλτιστη πολιτική θεωρείται εκείνη με το μέγιστο άθροισμα ανταμοιβών απ' την αρχή μέχρι το τέλος της διαδρομής. Έτσι, έχουμε :

1. $s_0 \to s_1 \to s_3 \to s_5$ =1+1+1=3
2. $s_0 \to s_1 \to s_4 \to s_5$ =1+1+5=7
3. $s_0 \to s_2 \to s_4 \to s_5$ =2-50+5=-43

Συνεπώς η πολιτική που πρέπει να ακολουθήσουμε είναι η 2, δηλαδή η διαδρομή $s_0 \to s_1 \to s_4 \to s_5$.

4.8 Συναρτήσεις Αξίας Κατάστασης

Σχεδόν όλοι οι αλγόριθμοι ενισχυτικής μάθησης βασίζονται στον υπολογισμό *συναρτήσεων αξίας*, που είναι είτε συναρτήσεις κατάστασης είτε συναρτήσεις ζεύγους κατάστασης-δράσης και εκτιμούν πόσο επικερδής είναι μια επιλογή του πράκτορα δεδομένου της ισχύουσας κατάστασης.

Έστω μια πολιτική π που αντιστοιχεί κάθε κατάσταση $s \in S$ και δράση $a \in A(s)$ στην πιθανότητα $\pi(s,a)$, που είναι η πιθανότητα λήψης της a δεδομένης της s. Αξία της κατάστασης s υπό πολιτική π, που συμβολίζεται με $V^{\pi}(s)$ είναι η αναμενόμενη απόκριση όταν ξεκινάμε απ' την s και ακολουθούμε την π από εκεί και πέρα. Για ένα MDP, μπορούμε να ορίσουμε την $V^{\pi}(s)$ ως εξής:

$$\mathrm{V}^{\mathrm{p}}(\mathrm{s}) = E_{\pi}\left\{R_t \middle| s_t = s\right\} = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \middle| s_t = s\right\}, \quad (3.7)$$

όπου $E\{\}$ υποδηλώνει την αναμενόμενη αξία δεδομένης της πολιτικής π που ακολουθεί ο πράκτορας και t κάποιο χρονικό βήμα και λέγεται **συνάρτηση αξίας κατάστασης**.

Αντίστοιχα ορίζουμε την αξία της λήψης μιας δράσης a υπό πολιτική π που τη συμβολίζουμε $Q^{\pi}(s,a)$ και δηλώνει την αναμενόμενη αξία της απόκρισης ξεκινώντας από κατάσταση s, υλοποιώντας δράση a και ακολουθώντας πολιτική π.

$$\mathrm{Q}^{\mathrm{p}}(\mathrm{s},\mathrm{a}) = E_{\pi}\left\{R_t \middle| s_t = s, a_t = a\right\} = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \middle| s_t = s, a_t = a\right\}, \quad (4.8)$$

Η Q^{π} λέγεται **συνάρτηση αξίας κατάστασης-δράσης**.

Χρειαζόμαστε όμως μια θεωρία με την οποία να μπορούμε να ορίζουμε με βεβαιότητα τη βέλτιστη πολιτική για κάθε MDP. Μια πολιτική π θεωρείται καλύτερη ή ισάξια μιας άλλης π' αν το αναμενόμενο αποτέλεσμά της είναι μεγαλύτερο ή ίσο αυτού της π' για όλες τις καταστάσεις. Με άλλα λόγια: $\pi \geq \pi'$ αν και μόνο αν $V^{\pi}(s) \geq V^{\pi'}(s)$ για κάθε $s \in S$. Πάντα υπάρχει κάποια πολιτική η οποία να είναι καλύτερη ή ισάξια των υπολοίπων. Αυτή θα καλούμε *βέλτιστη πολιτική*. Επειδή όμως μπορεί να περισσότερες από μια, συμβολίζουμε όλες τις βέλτιστες πολιτικές με π^*. Αυτές μοιράζονται την ίδια συνάρτηση αξίας κατάστασης που συμβολίζουμε με V^* και δίνεται απ' τον τύπο: [14]

$$\mathrm{V}^{*}(\mathrm{s}) = \max_{\pi} \mathrm{V}^{\mathrm{p}}(\mathrm{s}), \quad (4.9)$$

για κάθε $s \in S$.

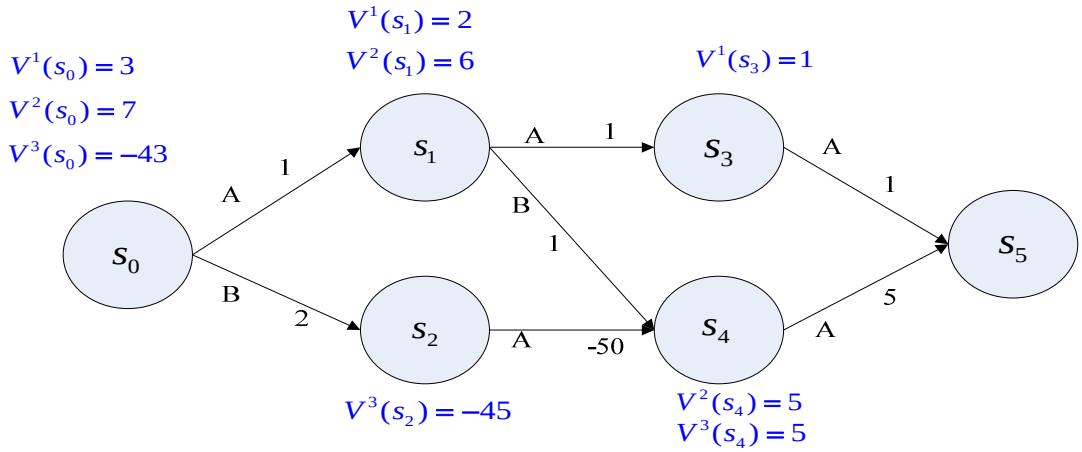
Ορίζουμε επίσης τη **βέλτιστη συνάρτηση αξίας-κατάστασης** Q^{*} που δίνεται απ' τον τύπο:

$$Q^*(s,a) = \max_{\pi} Q^p(s,a), \quad (4.10)$$

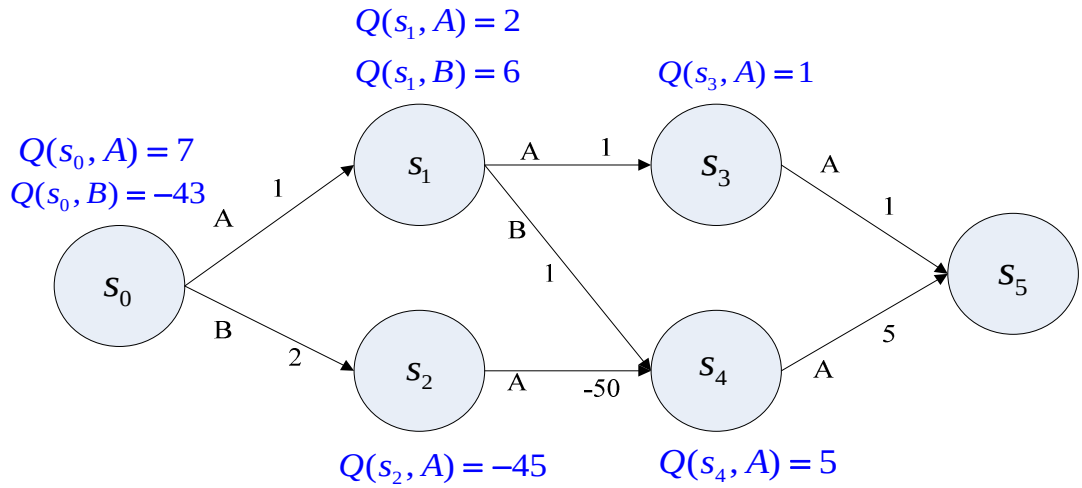
για κάθε $s \in S$ και $a \in A(s)$. Για το ζεύγος (s, α), η εν λόγω συνάρτηση δίνει την αναμενόμενη τιμή της δράσης a που λαμβάνεται στην κατάσταση s και ακολουθώντας τη βέλτιστη πολιτική. Έτσι, μπορούμε να συνδέσουμε τις V^* και Q^* μέσω της σχέσης:

$$Q^*(s,a) = E\left\{ r_{t+1} + \gamma V^*(s_{t+1}) \middle| s_t = s, a_t = a \right\}, \quad (4.11)$$

Συνολικά, για το παράδειγμά μας, οι συναρτήσεις αξίας κατάστασης έχουν ως εξής : [13]



Σχήμα 4.10 Συναρτήσεις Αξίας Κατάστασης V του MDP του σχήματος 3.8



Σχήμα 4.11 Συναρτήσεις Αξίας Κατάστασης Q του MDP του σχήματος 3.8

4.9 Σύνοψη Ενισχυτικής Μάθησης

Ας συνοψίσουμε τα στοιχεία της ενισχυτικής μάθησης (RL) όπως τα παρουσιάσαμε στο κεφάλαιο αυτό. Η RL αφορά στη μάθηση μέσω αλληλεπίδρασης για το πώς πρέπει να συμπεριφερθεί κανείς ούτως ώστε να πετύχει ένα στόχο. Ο *πράκτορας* και το *περιβάλλον* του αλληλεπιδρούν για μια σειρά διακριτών χρονικών βημάτων. Οι επιλογές του πράκτορα καλούνται *δράσεις* και οι *καταστάσεις* αποτελούν τη βάση για την επιλογή τους. Στο τέλος κάθε γύρου, ο πράκτορας λαμβάνει μια *ανταμοιβή*, η οποία αποτελεί το κριτήριο αξιολόγησης των επιλογών του. Οτιδήποτε βρίσκεται στο εσωτερικό του πράκτορα, του είναι απολύτως γνωστό και μπορεί να το ελέγξει πλήρως. Ό,τι βρίσκεται εκτός πράκτορα, δεν είναι απόλυτα γνωστό ούτε ελέγξιμο. Η *πολιτική* είναι μια στοχαστική διαδικασία του πράκτορα μέσω της οποίας επιλέγει τη δράση του, ως συνάρτηση των καταστάσεων. Ο αντικειμενικός σκοπός του πράκτορα είναι να μεγιστοποιήσει το ποσό των ανταμοιβών που λαμβάνει με το χρόνο.

Η *απόκριση* είναι συνάρτηση των μελλοντικών ανταμοιβών που ο πράκτορας επιδιώκει να μεγιστοποιήσει. Έχει διαφορετική μορφή ανάλογα με τη φύση του προβλήματος και ανάλογα με το αν κάποιος επιθυμεί να μειώσει την επίδραση των μελλοντικών ανταμοιβών. Το μη εκπτωτικό μοντέλο είναι κατάλληλο για *επεισοδιακά προβλήματα*, ενώ το μοντέλο με την *εκπτωτική παράμετρο* είναι κατάλληλο για συνεχή προβλήματα που δεν μπορούν να σπάσουν σε μικρότερα.

Ένα περιβάλλον ικανοποιεί την *ιδιότητα Markov* αν το σήμα κατάστασης ενσωματώνει το παρελθόν χωρίς να χάνει την ικανότητά του να προβλέπει το μέλλον. Μπορεί αυτή η συνθήκη σπάνια να ικανοποιείται πλήρως, γι' αυτό το σήμα κατάστασης θα πρέπει να κατασκευάζεται ή να επιλέγεται με κριτήριο να είναι όσο το δυνατόν κοντύτερα στη συνθήκη αυτή. Αν η ιδιότητα Markov ισχύει, το περιβάλλον καλείται διαδικασία αποφάσεων Markov (MDP). Ένα πεπερασμένο MDP είναι ένα MDP με πεπερασμένα σύνολα καταστάσεων και δράσεων.

Οι *συναρτήσεις αξίας κατάστασης* μιας πολιτικής, αναλογούν σε κάθε κατάσταση ή σε κάθε ζεύγος κατάστασης-δράσης, δεδομένου ότι ο πράκτορας χρησιμοποιεί την εν λόγω πολιτική. Οι *βέλτιστες συναρτήσεις αξίας* αντιστοιχούν στην κατάσταση ή στο ζεύγος κατάστασης-δράσης με τη μεγαλύτερη αναμενόμενη απόκριση που επιτυγχάνεται μέσω μιας πολιτικής. Μια πολιτική της οποίας η συνάρτηση αξίας είναι βέλτιστη λέγεται *βέλτιστη πολιτική*. Παρ' ότι η βέλτιστη συνάρτηση αξίας για τις καταστάσεις και τα ζεύγη καταστάσεων-δράσεων είναι μοναδική για ένα δεδομένο MDP, μπορούν να υπάρχουν πολλές βέλτιστες πολιτικές.

Κεφάλαιο 5: Αλγόριθμος Roth-Erev

Σκοπός του κεφαλαίου αυτού είναι η περιγραφή του αλγορίθμου των Roth και Erev, του τροποποιημένου αλγορίθμου Roth-Erev, καθώς και η επίδειξη της λειτουργικότητάς τους στην ανταγωνιστική αγορά της ηλεκτρικής ενέργειας.

5.1 Βασική ιδέα Roth & Erev

Σε μια σειρά από μελέτες, οι Roth και Erev [16], [19] προσπάθησαν να κατανοήσουν το πώς οι άνθρωποι μαθαίνουν να συμπεριφέρονται σε παιχνίδια με πολλούς παίχτες, οι οποίοι αλληλεπιδρούν στρατηγικά μεταξύ τους, όπως οι διμερείς δημοπρασίες. Θέλησαν παράλληλα να δείξουν πως τα απλά μοντέλα μάθησης είναι ικανά να αναπαραστήσουν ικανοποιητικά την αλληλεπίδραση αυτή. Έτσι ανέπτυξαν έναν αλγόριθμο ενισχυτικής ατομικής μάθησης με τρεις παραμέτρους, που ονομάζεται *RE αλγόριθμος* (Roth-Erev algorithm).

Η βασική ιδέα πίσω απ' τον αλγόριθμο αυτόν δε διαφέρει καθόλου απ' αυτή της κλασικής ενισχυτικής μάθησης και είναι η εξής : η τάση να εφαρμόζεται μια δράση ενισχύεται όταν παράγει επιθυμητά αποτελέσματα και αποδυναμώνεται όταν παράγει ανεπιθύμητα. Οι Roth και Erev εκλαμβάνουν αυτή την *«αρχή της επίδρασης»* - που είναι ευρέως αποδεκτή και στον κλάδο της ψυχολογίας – ως τη βασική τους αρχή στην αναζήτηση ενός ισχυρού μοντέλου ατομικής μάθησης. Επιπρόσθετα, αναφέρονται σε μια ακόμη αρχή, εξίσου διαδεδομένη στον κλάδο της ψυχολογίας, την οποία αποκαλούν *«νόμο της ισχύος του αποτελέσματος»*. Αυτός ο τελευταίος νόμος παρουσιάζει τις καμπύλες μάθησης να είναι αρχικά απότομες και εν συνεχεία να γίνονται επίπεδες.

Οι ψυχολόγοι, εν γένει, έχουν εστιάσει στην ατομική μάθηση κυρίως σε 'παιχνίδια εναντίον της φύσης' στα οποία όμως υπάρχει μόνο ένα άτομο που λαμβάνει τις αποφάσεις. Αντίθετα, οι Roth και Erev ενδιαφέρονται για περιβάλλοντα όπου υπάρχουν πολλοί παίχτες-πράκτορες. Επισημαίνουν ότι οι δυο βασικές αρχές (της επίδρασης και του νόμου της ισχύος του αποτελέσματος) αποτυγχάνουν να συνυπολογίσουν επαρκώς την επίδραση που έχουν στις αποφάσεις των παιχτών οι αποφάσεις των υπολοίπων.

Βασιζόμενοι στις εκτενείς παρατηρήσεις ατομικής μάθησης σε παιχνίδια με πολλούς παίχτες, οι Roth και Erev παραθέτουν δυο ακόμη αρχές που επιτρέπουν να συνυπολογιστεί η όποια αλληλεπίδραση : «το *φαινόμενο του πειραματισμού*» και «το *φαινόμενο του πρόσφατου ή λησμονηθέντος γεγονότος*». Το πρώτο υποδεικνύει οτι παλαιές πρακτικές που είχαν επιτυχή αποτελέσματα όχι μόνο είναι πολύ πιθανό να εφαρμοστούν ξανά, αλλά είναι πολύ πιθανό να εφαρμοστούν και μέθοδοι που είναι πολύ κοντά σε αυτήν. Η δεύτερη αρχή υποστηρίζει ότι οι πρόσφατες εμπειρίες παίζουν πολύ σημαντικότερο ρόλο στον καθορισμό της συμπεριφοράς των παιχτών απ' ότι οι παρελθοντικές.

Ο *RE αλγόριθμος* ενσωματώνει και τις τέσσερις προαναφερθείσες αρχές σε κάποιο βαθμό. Οι Roth και Erev αποδεικνύουν ότι ο αλγόριθμός τους είναι ικανός να ακολουθήσει επιτυχώς τις ενδιάμεσες συμπεριφορές των παιχτών και μάλιστα για μια μεγάλη ποικιλία παιχνιδιών πολλαπλών παιχτών-πρακτόρων με επαναλαμβανόμενα βήματα. [16], [18]

Τέλος, ο RE αλγόριθμος αποτελεί μέλος των αποκαλούμενων ‘αυτό-προσαρμοστικών στρατηγικών πλειοδοσίας’, καθώς ο πράκτορας αναπροσαρμόζει τη στρατηγική του στηριζόμενος στα δικά του συμπεράσματα και τη δική του βούληση, κρίνοντας πάντα από την απόκριση-ανταμοιβή που λαμβάνει.

5.2 Αλγόριθμος Roth-Erev (RE)

Οι τρεις παράμετροι που χαρακτηρίζουν το RE αλγόριθμο είναι οι εξής :

- Η παράμετρος διαβάθμισης s(1) (scaling parameter)
- Η παράμετρος αμεσότητας r (recency parameter)
- Η παράμετρος πειραματισμού e (experimentation parameter)

Ο βασικός αλγόριθμος Roth-Erev μπορεί να περιγραφεί ως εξής: Πρώτα, αρχικοποιούμε τις ροπές q_j οι οποίες αναφέρονται στην επιλογή κάθε δράσης a_t. Αυτές που αποκαλούμε ροπές q_j αποτελούν την παράμετρο εκείνη που στην ουσία προσπαθεί να ενσωματώσει στον αλγόριθμό μας την έννοια της αρχικής τάσης του κάθε πράκτορα. Δημιουργήθηκαν, λοιπόν, σε μια προσπάθεια να εισάγουμε στο μαθηματικό μοντέλο τον τρόπο με τον οποίο ο πράκτορας επηρεάζεται στην λήψη αποφάσεων απ’ την προδιάθεση που έχει. Εν συνεχεία, με βάση αυτές τις ροπές παράγουμε τις πιθανότητες που αντιστοιχούν σε κάθε δράση, $\mathrm{Prob}_j(t)$ και επιλέγουμε τη δράση η οποία υπερισχύει σε πιθανότητα. Τέλος, ανανεώνουμε τις τιμές των ροπών δράσεων χρησιμοποιώντας την ανταμοιβή που λάβαμε ως απόκριση της δράσης a_t και ξεκινάμε απ’ την αρχή για το νέο γύρο. [3], [18]

Πίνακας 5.1 Περιγραφή αλγορίθμου Roth-Erev

1. Αρχικοποίηση ροπών $q(t)$ για επιλογή δράσεων a_t.

2. Παραγωγή πιθανοτήτων $\mathrm{Prob}_j(t)$ επιλογής δράσεων απ’ τις ισχύουσες ροπές $q(t)$.

3. Επιλογή μιας δράσης a_t σε συμφωνία με τις $\mathrm{Prob}_j(t)$.

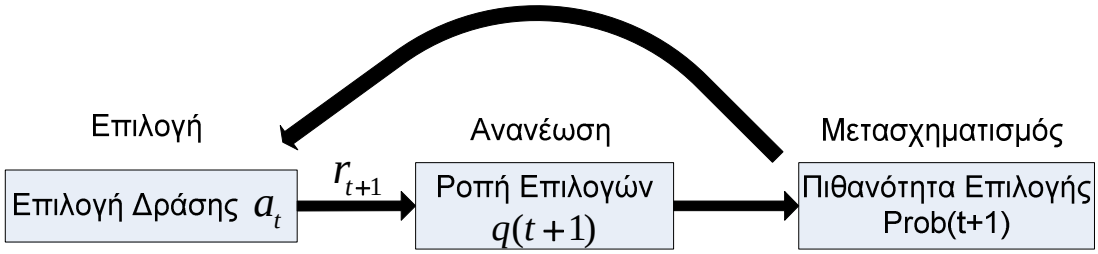
4. Ανανέωση των τιμών $q(t+1)$ με βάση την r_{t+1} που προκύπτει απ’ την επιλογή της a_t.

5. Επανάληψη απ’ το βήμα 2.

Η δομή του RE αλγορίθμου συνοψίζεται σε τρεις βασικές δραστηριότητες: [3]

- **Επιλογή** δράσης a
- **Ανανέωση** ροπών επιλογής δράσης q
- **Μετασχηματισμός** ροπών σε πιθανότητες Prob

Η επιλογή δράσης a οδηγεί σε ανταμοιβή r, ακολουθούμενη από ανανέωση όλων των ροπών επιλογής δράσης q με βάση την ανταμοιβή. Στη συνέχεια, οι ροπές αυτές μετασχηματίζονται σε πιθανότητες επιλογής δράσεων Prob.



Σχήμα 5.1 Δομή αλγορίθμου RE.

Παράμετροι RE αλγορίθμου:

- $q_j(0)$: Αρχική Ροπή (Initial Propensity)
- e: Πειραματισμός (Experimentation parameter)
- r: Παράμετρος Αμεσότητας (Forgetting/Recency parameter)

Μεταβλητές RE αλγορίθμου:

- a_j : Επιλεγμένη Δράση
- q_j : Ροπή Δράσης a_j
- a_k : Τελευταία Επιλεγμένη Δράση
- R_k : Ανταμοιβή Δράσης a_k
- t : Τρέχον Χρονικό Βήμα
- N: Σύνολο Δράσεων

Η ανανέωση των ροπών δράσης γίνεται με βάση τους τύπους:

$$\mathrm{q}_j(\mathrm{t}+1) = [1-r]\cdot q_j(t) + E_j(e,N,k,t), \quad (5.1)$$

$$\mathrm{E}(\mathrm{e},N,k,t) = \begin{cases} R(j,k',t)\cdot(1-e), k=k' \\ R(j,k',t)\cdot\dfrac{e}{K-1}, k\neq k \end{cases}, \quad (5.2)$$

όπου $E(\bullet)$ είναι η συνάρτηση πείρας (experience function) ή αλλιώς συνάρτηση ανανέωσης.

Όσον αφορά στη συγκεκριμενοποίηση του αρχικού επιπέδου ροπών $q_j(0)$ για τις εφικτές επιλογές δράσεων a_j, $j=1,...,N$, οφείλουμε να επισημάνουμε ότι το αρχικό επίπεδο ροπών δρα στην ουσία ως *'επίπεδο φιλοδοξιών'*. Αυτό σημαίνει ότι αν το αρχικό επίπεδο ροπών είναι υψηλό, τότε ο πράκτορας θα είναι απογοητευμένος απ' την ανταμοιβή που θα λάβει με τις πρώτες επιλογές του κι έτσι θα ενθαρρυνθεί να πειραματιστεί. Αντίθετα, με χαμηλό επίπεδο ροπών, ο πράκτορας θα είναι πιο εύκολα ευχαριστημένος απ' την ανταμοιβή των επιλογών του, γεγονός που θα ενθαρρύνει την καταστάλαξή του σε κάποια απ' αυτές.

Μια δεύτερη παρατήρηση έχει να κάνει με το *φαινόμενο του πρόσφατου ή λησμονηθέντος γεγονότος* που προαναφέρθηκε και υποδηλώνει ότι μπορεί να

χρειαστεί να ‘ξεχάσουμε’ τις ανταμοιβές R που ελήφθησαν στο μακρινό παρελθόν, σε περιβάλλοντα που μεταβάλλονται με το χρόνο. Αυτό ελέγχεται από την παράμετρο αμεσότητας r, η οποία κυμαίνεται μεταξύ 0 και 1 ($0 \le r \le 1$). Όταν το r πλησιάζει το 1, το μεγαλύτερο βάρος πέφτει στις πιο πρόσφατες ανταμοιβές R. Όσο, όμως, πλησιάζει στο 0, τόσο το βάρος τίθεται σχεδόν εξίσου σε όλες τις ανταμοιβές R που έχουν ληφθεί. Εν ολίγοις, όταν ισχύουν $r = 0$ και $e = 0$, τότε όλες οι R έχουν ακριβώς το ίδιο βάρος.

Επίσης, υπάρχει μια ανάγκη για ‘μοίρασμα’ των τιμών ανταμοιβής σε όλες τις δράσεις στην αρχή του παιχνιδιού προκειμένου να ενθαρρυνθεί ο πειραματισμός και να αποφευχθεί η πρώιμη καταστάλαξη σε μια μη βέλτιστη δράση a_k. Το ‘μοίρασμα’ αυτό δύναται να ελεγχθεί μέσω της παραμέτρου πειραματισμού e που κυμαίνεται μεταξύ 0 και 1 ($0 \le e \le 1$). Όσο το e αυξάνεται, τόσο διαμοιράζονται οι ανταμοιβές, κάτι που οδηγεί τον πράκτορα να μεταβεί απ’ την επιλεγμένη δράση a_k σε καινούργιες δράσεις a_j, καθώς οι διαφοροποιήσεις μεταξύ των ροπών επιλογής q_k και q_j γίνονται μικρότερες. Όταν, δε, το e πλησιάζει στο 0, η ανταμοιβή που λαμβάνεται επαφίεται μόνο στη δράση a_k από την οποία και προήλθε, κάτι που σημαίνει ότι θα ανανεωθεί μόνο η ροπή q_k.

Πίνακας 5.2 Παράμετροι $q_j(0)$, r και e

- Αρχική ροπή $q_j(0)$: $q_j(0) \uparrow \Rightarrow$ ενθάρρυνση πειραματισμού
  $q_j(0) \downarrow \Rightarrow$ πιο εύκολη καταστάλαξη

- Παράμετρος αμεσότητας r: $r \simeq 1 \Rightarrow$ βάρος σε πιο πρόσφατες R
  $r \simeq 0 \Rightarrow$ ίδιο βάρος σε όλες τις R

- Παράμετρος πειραματισμού e: $e \simeq 1 \Rightarrow$ πολύ πιθανή η μετάβαση από a_k σε a_j
  $e \simeq 0 \Rightarrow$ πιο πιθανή η προσκόλληση στην a_k

Αυτό που έπεται της αρχικοποίησης των ροπών και της ανανέωσής τους με βάση τους τύπους (5.1) και (5.2) είναι ο μετασχηματισμός τους σε πιθανότητες που είναι και το τελικό κριτήριο για την επιλογή δράσης. Ο μετασχηματισμός αυτός δεν είναι υποχρεωτικό να ακολουθεί μόνο έναν τύπο-μοντέλο, αλλά μπορεί και παραπάνω. Ωστόσο, στη βιβλιογραφία συναντάει κανείς κυρίως δυο περιπτώσεις.

Περίπτωση 1: Η πιθανότητα να επιλεγεί η δράση j στο χρονικό βήμα t είναι ίση με τη σχετική ροπή της δράσης j:

$$\mathrm{Prob_j(t)} = \mathrm{p_j(t)} = \frac{q_j(t)}{\sum_{n=1}^{N}[q_n(t)]}, \quad (5.3)$$

Ο τύπος (4.3) αποτελεί και την πιο απλοϊκή αντιστοιχία ροπών-πιθανοτήτων.

Περίπτωση 2: Η πιθανότητα να επιλεγεί η δράση j δίνεται απ' την πιθανότητα Gibbs-Boltzmann:

$$\text{Prob}_j(t) = \text{p}_j(\text{t}) = \frac{e^{q_j(t)/T}}{\sum_{n=1}^{N} e^{q_n(t)/T}}, \quad (5.4)$$

όπου Τ η παράμετρος 'ψύξης', η οποία επηρεάζει το δυναμικό σχηματισμό των κατανομών των πιθανοτήτων $\text{Prob}_j(\text{t})$. Το μεγάλο πλεονέκτημα του τύπου (5.4) έναντι του (5.3) είναι ότι μπορεί να χειρίζεται και αρνητικές τιμές ροπών q_j. [3], [17]

Τέλος, αν θεωρήσουμε έστω λ_i την οριακή τιμή του πράκτορα (αγοραστή ή πωλητή), ορίζονται οι εκτιμήσεις τιμής για έναν αγοραστή και για έναν πωλητή και οι οποίες είναι:

$p_i(n) = \lambda_i + \delta_i(n)$, (5.5) η εκτίμηση τιμής για έναν πωλητή και

$p_i(n) = \lambda_i - \delta_i(n)$, (5.6) η εκτίμηση τιμής για έναν αγοραστή,

όπου δ_i θεωρούμε μια μεταβλητή τέτοια ώστε $P(\delta_i(n) = j) = p_{i,j}(n)$.

Η εφαρμογή του RE αλγορίθμου θα δειχθεί, τώρα, για μια ομάδα πωλητών-αγοραστών που συμμετέχουν σε μια διμερή δημοπρασία. Για λόγους απλότητας κάθε αγοραστής και πωλητής υποτίθεται ότι μαθαίνει με βάση έναν RE αλγόριθμο που χαρακτηρίζεται από τις ίδιες τρεις παραμέτρους.

Η τιμή προσφοράς για κάθε αγοραστή και πωλητή προσεγγίζεται από ένα διακριτό δίκτυο Κ εφικτών δράσεων k (προσφορά ή τιμή αγοράς), όπου Κ είναι το ίδιο για κάθε έμπορο. Στην αρχή του πρώτου γύρου της πρώτης δημοπρασίας, κάθε έμπορος j αναθέτει από μια ίση ροπή $q_{jk}(1)$ σε κάθε μια απ' τις εφικτές δράσεις του, που δίνεται απ' τον τύπο $q_{jk}(1) = s(1) \cdot \frac{X}{K}$, όπου Χ είναι το μέσο κέρδος που μπορούν να πετύχουν οι αγοραστές και οι πωλητές σε ένα δεδομένο γύρο.

Επιπλέον, κάθε έμπορος j αναθέτει από μια ίση πιθανότητα επιλογής $p_{jk}(1)$ σε κάθε εφικτή δράση του που δίνεται απ' τον τύπο $p_{jk}(1) = \frac{1}{K}$. Έπειτα, κάθε πράκτορας j πιθανοτικά επιλέγει μια εφικτή δράση k' που καταθέτει στο κέντρο σε συμφωνία με τις πιθανότητες της τρέχουσας δράσης του. Αφού συλλέξει όλες τις προσφορές και τις ζητήσεις, το κέντρο συναλλαγών καθορίζει την αντιστοιχία πωλητών-αγοραστών. Εν συνεχεία, ανακοινώνει αυτή την αντιστοιχία στους εμπόρους μαζί με την ποσότητα και το μέσο επίπεδο τιμής κάθε αντιστοιχίας. Στη συνέχεια, κάθε έμπορος j υλοποιεί τη δραστηριότητα (πώληση) που του ανατέθηκε και καταγράφει το κέρδος R(j,k',1) που έβγαλε απ' τη συγκεκριμένη δραστηριότητα.

Τώρα, υποθέτουμε ότι ο έμπορος j βρίσκεται στο πέρας του t-οστού γύρου δημοπρασιών, για τυχαίο θετικό αριθμό t. Στο t-οστό γύρο κατέθεσε μια πιθανή

δράση k' στο κέντρο συναλλαγών αποκομίζοντας κέρδος R(j,k',t) από αυτήν. Ο έμπορος j, στη συνέχεια, ανανεώνει την ισχύουσα ροπή δραστηριότητας $q_{jk}(t)$ με βάση το πιο πρόσφατο κέρδος του όπως περιγράφεται ακολούθως. Δεδομένης μιας πιθανής δραστηριότητας k, η ροπή $q_{jk}(t+1)$ για την επιλογή του k στο γύρο συναλλαγών t+1 καθορίζεται απ' τον τύπο (5.1) :

$$q_{jk}(t+1) = (1-r)\cdot q_{jk}(t) + E(j,k,k',t,K,e),$$

όπου το r υποδηλώνει την αξία της πρόσφατης παραμέτρου, το e την αξία της πειραματικής παραμέτρου και το $E(\bullet)$ αποτελεί τη συνάρτηση ανανέωσης με βάση την πείρα που έχει αποκομιστεί από τις παρελθούσες δραστηριότητες.

Η παράμετρος αμεσότητας r μειώνει σταδιακά τη σημασία των παλαιών εμπειριών και εισάγει την επίδραση των πιο πρόσφατων. Η συνάρτηση ανανέωσης $E(\bullet)$ παίρνει τη μορφή (5.2) :

$$\mathrm{E(j,k,k',t,K,e)} = \begin{cases} R(j,k',t)\cdot(1-e), k = k' \\ R(j,k',t)\cdot\dfrac{e}{K-1}, k \neq k \end{cases}$$

Η επιλεγμένη δράση k' ενισχύεται ή αποθαρρύνεται βάσει του κέρδους R(j,k',t) που αποκομίστηκε από τη δράση αυτή, αλλά διατηρείται και κάποια ροπή που οφείλεται στην πείρα από τις άλλες δράσεις. Έτσι, η συνάρτηση $E(\bullet)$ εισάγει την έννοια του φαινομένου του πειραματισμού.

Δεδομένων των ανανεωμένων ροπών $q_{jk}(t+1)$ για το γύρο t+1, η νέα πιθανότητα επιλογής της δραστηριότητας k $p_{jk}(t+1)$ στο γύρο t+1 για τον έμπορο j παίρνει τη μορφή $p_{jk}(t+1) = \dfrac{q_{jk}(t+1)}{\sum_{m=1}^{K} q_{jm}(t+1)}$ σύμφωνα με τον τύπο (5.3).

Συνοψίζοντας, ο αλγόριθμος των Roth και Erev δίνει απαντήσεις στο ερώτημα: «Δεδομένου του αποτελέσματος κέρδους, ποια τιμή πρέπει να επιλέξω;». Δεν υπεισέρχεται σε καμία λογική πρόβλεψης όπως για παράδειγμα: «Αν επιλέξω αυτή την τιμή τώρα, πώς ενδέχεται να επηρεαστούν οι επιλογές τιμής των ανταγωνιστών μου στο μέλλον;». [18]

5.3 Τροποποιημένος Αλγόριθμος Roth-Erev (MRE)

Ο απλός RE αλγόριθμος έχει δυο μειονεκτήματα : τον εκφυλισμό των παραμέτρων και τη μη ανανέωση των πιθανοτήτων ως αντίδραση στα μηδενικά κέρδη.

Πρώτον, η ανανέωση της πιθανότητας επιλογής είναι αργή αν το e είναι κοντά στο [K-1]/K και σταματά εντελώς εάν το e είναι ίσο με [K-1]/K. Συνεπώς πρέπει να λαμβάνονται με την παραπάνω πρόβλεψη οι τιμές των e και K. Δεύτερον, μια πολύ πιο ουσιαστική δυσκολία σε ένα περιβάλλον διμερών δημοπρασιών είναι ότι κάθε

έμπορος ανανεώνει τις πιθανότητες επιλογής του για μια δράση με βάση μόνο τα μη μηδενικά κέρδη. Σε περίπτωση μηδενικών κερδών οι πιθανότητές επιλογής του παραμένουν αμετάβλητες διότι οι τιμές ροπών για κάθε έμπορο συρρικνώνονται κατά τον ίδιο βαθμό. Σε μια διπλή συναλλαγή, οι έμποροι πρέπει να μάθουν να κάνουν προσφορές σε τιμές για τις οποίες η συνολική προσφορά ξεπερνά τη ζήτηση ούτως ώστε να υπάρχει η δυνατότητα αντιστοίχησης και άρα να προκύπτουν θετικά κέρδη. Η απουσία ανανέωσης των πιθανοτήτων σε περίπτωση μηδενικών κερδών μπορεί, λοιπόν, να συνεπάγεται με σημαντική απώλεια της επάρκειας της αγοράς καθώς οι έμποροι αδυνατούν να μάθουν πώς να κάνουν προσφορές σε κερδοφόρες γι' αυτούς τιμές.

Το 2001, οι Petrov, Nicolaisen και Tesfatsion [18] έκαναν μια απλή τροποποίηση στον RE αλγόριθμο, η οποία επιλύει και τα δυο αυτά προβλήματα ενώ παράλληλα παραμένει συνεπής στις αρχές που ενσωματώνει ο κλασικός αλγόριθμος. Συγκεκριμένα, τροποποίησαν τη συνάρτηση ανανέωσης ακολούθως:

$$\mathrm{ME(j,k,k',n,K,e)} = \begin{cases} R(j,k',n)\cdot(1-e), k = k' \\ q_{jk}(n)\cdot\dfrac{e}{K-1}, k \neq k \end{cases}, \quad (5.7)$$

Αυτή η τροποποίηση ουσιαστικά εισάγει μια διαφορική τιμή για την παράμετρο αμεσότητας r για επιλεγμένες έναντι μη επιλεγμένων δράσεων ενώ συγχρόνως παραλείπει τον όρο κέρδους για ροπές που αντιστοιχούν σε μη επιλεγμένες δράσεις. Λεπτομερέστερα, μειώνεται το πλάτος της παραμέτρου αμεσότητας για μη επιλεγμένες δραστηριότητες από r σε $r^* = (r - \frac{e}{K-1})$. Προφανώς, ο εκφυλισμός δεν υφίσταται πλέον για τιμές e=[K-1]/K, αλλά πώς αυτή η τροποποίηση επιλύει επίσης και το πρόβλημα που ανακύπτει για μηδενικά κέρδη;

Σημειώνεται ότι η συρρίκνωση που προκαλείται απ' τον παράγοντα [1-r] στη τιμή της ροπής είναι τώρα μεγαλύτερη κατά [1-r^*] στις ροπές των μη επιλεγμένων δράσεων. Ας δούμε λοιπόν το τί συμβαίνει όταν από μια επιλεγμένη δράση k' προκύπτει μηδενικό κέρδος. Όλες οι ροπές έχουν συρρικνωθεί ωστόσο η ροπή που αντιστοιχεί στη δράση k' υφίσταται τη μεγαλύτερη συρρίκνωση απ' όλες. Επομένως, στον επόμενο γύρο οι πιθανότητες των επιλογών για τις μη επιλεγμένες δραστηριότητες θα αυξηθούν αντίστοιχα με την πιθανότητα της επιλογής k', ενθαρρύνοντας τον έμπορο να απομακρυνθεί απ' τη δράση που κατέληξε σε μηδενικά κέρδη.

Απ' την άλλη, ας υποθέσουμε ότι η επιλεγμένη δράση k' έχει ως αποτέλεσμα θετικό κέρδος. Τότε, η θετικού κέρδους ενίσχυση της k' στη συνάρτηση ανανέωσης ροπών θα τείνει να ξεπεράσει τη μεγαλύτερη συρρίκνωση και τελικά να προκαλέσει μια σχετική αύξηση στη συνάρτηση ανανέωσης πιθανοτήτων για την επιλογή αυτή στον επόμενο γύρο.

Συνεπώς, όταν η συνάρτηση ανανέωσης $E(\bullet)$ στον κλασικό RE αλγόριθμο αντικατασταθεί από την τροποποιημένη $ME(\bullet)$, το πρόβλημα μηδενικού κέρδους εξαλείφεται. Οι πιθανότητες επιλογής δράσεων που αντιστοιχούν σε αποτέλεσμα

μηδενικών κερδών τείνουν να μειώνονται, ενώ παράλληλα οι πιθανότητες που αντιστοιχούν σε επιλογές δράσεων με θετικά κέρδη τείνουν να αυξάνονται.

Σημειωτέον, ότι στο [18], οι υπολογιστικοί πράκτορες (πωλητές-αγοραστές) που ακολούθησαν το μοντέλο του τροποποιημένου RE αλγορίθμου των Nicolaisen, Petrov και Tesfatsion, πέτυχαν βαθμό απόδοσης κοντά στο 90% έναντι 20% αυτών που υιοθέτησαν τον κλασικό RE αλγόριθμο.

Ο αλγόριθμος που εμείς θα υιοθετήσουμε είναι ο τροποποιημένος κι έτσι θα υποθέτουμε ότι οι αγοραστές και οι πωλητές (οι πράκτορες εν γένει) αναπροσαρμόζουν τις προσφορές τους με βάση τη συνάρτηση (5.5), δηλαδή με βάση τον τροποποιημένο RE αλγόριθμο που εφεξής θα αναφέρεται ως *MRE αλγόριθμος* (Modified RE algorithm).

5.4 Κώδικας RE-MRE Αλγορίθμου σε Γλώσσα Προγραμματισμού Python

Το 2010, ο Richard W. Lincoln [22] δημοσίευσε έναν κώδικα γραμμένο σε γλώσσα προγραμματισμού Python για τον αλγόριθμο Roth-Erev (RE) αλλά και για την τροποποιημένο αλγόριθμο MRE. Ο κώδικας αυτός φιλοξενείται στην ανοικτή βιβλιοθήκη πακέτων για τη γλώσσα Python, PyDoc.net [23] και στηρίζεται στην αντίστοιχη δουλειά για τη γλώσσα προγραμματισμού Java που είχε παρουσιάσει ο Charles Gieseler [24] το 2006.

Ο κώδικας είναι ο παρακάτω:

Πίνακας 5.3 Κώδικας RE-MRE αλγορίθμων σε Python

```
import random

from pybrain.rl.learners.rllearner import RLLearner
from pybrain.structure.modules.module import Module

#class Action:
#    """Για 'classes' που αναπαριστούν τις δράσεις που μπορεί να κάνει ένας
#       πράκτορας σε μια προσομοίωση. Μπορεί απλά να συνιστά μια δράση
#       ή να εμπεριέχει δεδομένα (data) και μεθόδους που χρησιμοποιήθηκαν
#       κατά τη λειτουργία.
#    """
#    def getID(self):
#        """Ανακτά το αναγνωριστικό για την παραπάνω δράση ('Action').
#        """
#
#class ActionDomain:
#    """ Αναπαράσταση του χώρου των δυνατών δράσεων που ένας πράκτορας
#        μπορεί να υλοποιήσει σε ένα συγκεκριμένο περιβάλλον.
#
#        Ο τύπος της Δράσης ('Action') όπως και το αναγνωριστικό της μπορούν
#        να παραμετροποιηθούν.
```

```
#    """
#
#    def getAction(self, ID):
#        """ Ανακτά τη δράση (‘Action’) που υποδηλώνεται από το ID. Αν το ID
#            δεν αντιστοιχεί σε καμία δράση, τότε επιστρέφει null(κενό).
#        """
#
#    def getIDList(self):
#        """ Ανακτά μια λίστα από αναγνωριστικά για όλες τις δράσεις στην κατη-
#            γορία(‘Domain’) αυτήν.
#        """
#
#    def size(self):
#        """ Δίνει τον αριθμό των δράσεων στην εν λόγω κατηγορία (‘Domain’).
#        """

#class SimpleEventGenerator:
#    """Παράγει τυχαία διακριτά γεγονότα από μια δεδομένη κατανομή.
#    """
#
#    def __init__(self, distrib):
#        # Συνάρτηση Κατανομής Πιθανότητας.
#        distrib = []
#
#        engine = Random()
#
#
#    def nextEvent(self):
#        eventIndex = 0
#        randValue = self.engine.nextDouble()
#
#        while (randValue > 0.0) and (eventIndex < len(self.distrib)):
#            randValue -= self.distrib[eventIndex]
#            eventIndex += 1
#
#        return eventIndex - 1

def eventGenerator(distrib):
    eventIndex = 0
    randValue = random.random()

    while (randValue > 0.0) and (eventIndex < len(distrib)):
        randValue -= distrib[eventIndex]
        eventIndex += 1

    yield eventIndex - 1

class PropensityTable(Module):
    """Χτίζει μια πολιτική ενισχυτικής μάθησης χωρίς καταστάσεις. Αυτού του τύπου η
        πολιτική απλά ακολουθεί μια κατανομή επιλογών δράσεων ανεξάρτητη από την
```

```
        τωρινή κατάσταση. Αυτό σημαίνει ότι απλά ακολουθεί την πιθανότητα επιλογής
        για κάθε δράση για όλες τις πιθανές καταστάσεις.

        Αποτελεί επί της ουσίας μια διακριτή κατανομή πιθανότητας με βάση την οποία
        Επιλέγεται η Δράση ('Action') από μια δεδομένη κατηγορία ('ActionDomain'),
        ανεξαρτήτως των καταστάσεων του περιβάλλοντος.
    """
    def __init__(self, numActions, actionDomain, initProbs=None, name=None):
        Module.__init__(self, 1, 1, name)
        self.numStates = numStates
        self.numActions = numActions

        # Κάθε πιθανή δράση έχει μια ροπή που συνδέεται με αυτήν. Ουσιαστικά κα-
        # θορίζει την πιθανότητα να επιλεγεί κάθε δράση.

        self.propensities = zeros(self.numActions)

        # Εδώ, η κατανομή πιθανότητας είναι ένας πίνακας από τιμές πιθανοτήτων.
        # Όταν χρησιμοποιείται από κοινού με την eventGenerator, μια τιμή υποδηλώνει
        # την πιθανότητα να επιλεγεί ο δείκτης της.
        #
        # Κάθε Action έχει μια ID και κάθε Action ID έχει ένα δείκτη στη λίστα των IDs
        # που διατηρείται απ' την ActionDomain. Ο δείκτης που αντιστοιχεί σ' αυτή τη
        # συνάρτηση κατανομής πιθανότητας περιέχει μια τιμή πιθανότητας για το
        # συγκεκριμένο ID. Έτσι, υπάρχει μια αντιστοίχηση από πιθανότητες σε
        # ActionIDs και από ActionIDs σε Actions. Αυτό επιτρέπει στην eventGenerator
        # να χρησιμοποιεί τη συνάρτηση κατανομής πιθανότητας για να επιλέγει Actions
        # από την ActionsDomain ανάλογα με την καθορισμένη κατανομή πιθανότητας.
        #
        # Οι τιμές πιθανοτήτων τροποποιούνται από έναν RLLearner ανάλογα με τον
        # αλγόριθμο μάθησης που εφαρμόζεται.

# self.probDistFunction = zeros(numActions)

        # Αρχικοποίηση σε ομοιογενή κατανομή:

# for i in range(numActions):
#     self.probDistFunction[i] = 1.0 / numActions

        self.probDistFunction = array(1.0 / numActions, (numActions, 1))

        # Απαιτείται ένας τυχαίος αριθμός eventGenerator για την randomEngine.

        self.randomEngine = random.Random()


        # Παράγει randomEngine γεγονότα (επιλογές δράσης) ανάλογα με τις
        # πιθανότητες όλων των δράσεων στην ActionDomain. Οι πιθανότητες
        # διατηρούνται στην probDistFunction.
```

```
        self.eventGenerator = eventGenerator

        # Το σύνολο των δυνατών δράσεων (Actions) που ένας πράκτορας μπορεί
        # να υλοποιήσει. domain = [0, 1, 2, 3] # N S E W

        self.domain = actionDomain

        # Λίστα των ActionIDs στον domain. Επιτρέπει να αντιστοιχίσουμε από
        # ακέραιες τιμές επιλεγμένες από την eventGenerator (γεννήτρια τυχαίων
        # γεγονότων) σε actions (δράσεις) στη domain.

        self.actionIDList = []

        # Καταγράφει την τελευταία action που επιλέγει με την ακολουθού-
        # μενη πολιτική.

        self.lastAction = None

        self.init()

    def _forwardImplementation(self, inbuf, outbuf):
        """Συνάρτηση μετασχηματισμού forward
        """
#       outbuf[0] = self.getMaxAction(inbuf[0])
        outbuf[0] = self.generateAction()

    def init(self):
        self.actionIDList = self.domain.keys()
#       self.eventGenerator = SimpleEventGenerator(self.probDistFunction,
#                                                  self.randomEngine)

        self.eventGenerator = eventGenerator

        # Χρειαζόμαστε να αρχικοποιήσουμε την lastAction με κάτι. Επιλέγουμε,
        # λοιπόν, μια τυχαία δράση.
        self.lastAction = self.generateAction()

        self.propensities = zeros(self.numActions)

    def getPropensity(self, ID):
        index = self.actionIDList.index(ID)
        return self.propensities[index]

    def setPropensity(self, ID, prop):
        index = self.actionIDList.index(ID)
        self.propensities[index] = prop

    def generateAction(self):
        """Επιλογή μιας Action ανάλογα με την ισχύουσα συνάρτηση
           κατανομής πιθανότητας.
```

```
        """
        # Διαλέγουμε το δείκτη μιας action. Σημείωση: οι δείκτες ξεκινάν από 0.
        chosenIndex = self.eventGenerator.next(self.probDistFunction)
        chosenID = self.actionIDList.get(chosenIndex)
        chosenAction = self.domain[chosenID]

        self.lastAction = chosenAction

        return chosenAction

    def reset(self):
        numActions = self.numActions
        for i in range(numActions):
            self.probDistFunction[i] = 1.0 / numActions

    def setDistribution(self, distrib):
        """ Θέτει την κατανομή πιθανότητας που χρησιμοποιήθηκε στην επιλογή
            δράσεων από την action domain. Η κατανομή δίνεται από έναν πίνακα
            αριθμών με δεκαδικά ψηφία (floats).
        """
        self.probDistFunction = distrib

    def getProbability(self, actionID):
        """Παίρνει την τρέχουσα πιθανότητα επιλογής δράσης. Η παράμετρος
           actionIndex υποδηλώνει ποια δράση να κοιτάξουμε στη domain της
           ισχύουσας πολιτικής.
         """
        index = self.actionIDList.index(actionID)
        return self.probDistFunction[index]

    def setProbability(self, actionID, value):
        """ Ανανεώνει την πιθανότητα να επιλεγεί η συγκεκριμένη Action.
        """
        index = self.actionIDList.index(actionID)
        self.probDistFunction[index] = value
#       self.eventGenerator.setState(self.probDistFunction)

class RothErev(RLLearner):
    """Για classes που εφαρμόζουν αλγορίθμους ενισχυτικής μάθησης. Οι classes
       που ακολουθούν αυτή τη λογική, είναι υπεύθυνες να 'καθοδηγούν' τη
       διαδικασία μάθησης συγκεκριμένων αλγορίθμων.

       Οι αλγόριθμοι ενισχυτικής μάθησης χρησιμοποιούν μια πολιτική που ανα-
       παριστά τη γνώση που έχει αποκτηθεί. Οι ίδιες οι πολιτικές απαιτούν
       πρόσβαση σε χώρο πιθανών δράσεων, που αναπαρίστανται από
       ActionDomains.
       Απαιτούνται οι παράμετροι πειραματισμού (experimentation), αρχικής ροπής
       q_j(0) (initial Propensity) και αμεσότητας (recency).

       Πηγές: A. E. Roth, I. Erev, D. Fudenberg, J. Kagel, J. Emilie and R. X. Xing,
```

```
        "Learning in Extensive-Form Games: Experimental Data and Simple Dynamic
        Models in the Intermediate Term," Games and Economic Behavior,
        Special Issue: Nobel Symposium, vol. 8, January 1995, 164-212.[19],

        A. E. Roth, I. Erev, "Predicting How People Play Games with Unique
        Mixed-Strategy Equilibria," American Economics Review, Volume 88,
        1998, 848-881. [16]
    """
    def __init__(self, boltzmannTemp=10.0, useBoltz=False, experimentation=0.5,
                 initialPropensity=100.0, recency=0.5):

        # Παράμετρος ψύξης (Cooling Parameter) Gibbs-Boltzmann
        # Για τη μέθοδο πιθανότητας Gibbs-Boltzmann που χρησιμοποιείται στην
        # VRELearner.
        self.boltzmannTemp = boltzmannTemp
        self.useBoltz = useBoltz

        # Η τάση για πειραματισμό μεταξύ των επιλογών δράσης. Ο αλγόριθμος
        # συχνά επιλέγει μη βέλτριστες Actions για χάρη του πειραματισμού της
        # domain. Αυτό επιτρέπει στον αλγόριθμο να έχει μια πιο ακριβή εικόνα της
        # domain και να βρίσκει Actions που δίνουν καλύτερη αμοιβή από όσες έχουν
        # ήδη επιλεγεί.
        self.experimentation = experimentation

        # Η τιμή αρχικής ροπής  q_j(0)  ανατίθεται σε όλες τις δράσεις.
        # Χρησιμοποιείται αντί της παραμέτρου διαβάθμισης s(0).
        self.initialPropensity = initialPropensity

        # Η παράμετρος αμεσότητας r εκφράζει το βαθμό στον οποίο οι
        # παλιές δράσεις (actions) 'ξεχνιούνται'. Χρησιμοποιείται για να καθιστά τις
        # πρόσφατες εμπειρίες πιο 'ισχυρές' από τις παλαιότερες.
        self.recency = recency

        # Ο αριθμός δυνατών δράσεων (actions).
        self.domainSize = -1

        # Αναπαριστά την αποκτημένη γνώση.
#       self.policy = REPolicy()

        # Η τελευταία δράση που επιλέχτηκε με βάση την πολιτική.
        self.lastSelectedAction = None

        # Λίστα των IDs των actions στη domain. Επιτρέπει την αντιστοίχηση από
        # ροπές σε actions στη domain.
        self.actionIDList = []
        # Πόσες φορές ανανεώθηκε η διαδικασία μάθησης.
        self.period = 0

        # Αρχικοποιεί τις πιθανότητες επιλογής.
        if policy is not None:
```

```
            self.updateProbabilities()

        self.init()

    def setModule(self, module):
        super(RothErev, self).setModule(module)
        self.domainSize = len(module.actionDomain)
        self.actionIDList = module.actionDomain.keys()

    def init(self):
     """ Τελειώνει την αρχικοποίηση της μάθησης.
     """
        self.domainSize = len(self.module.actionDomain)
        self.actionIDList = self.module.actionDomain.keys()

        initProp = self.initialPropensity

        for ID in self.actionIDList:
            self.module.setPropensity(ID, initProp)

        self.lastSelectedAction = self.chooseAction()

    def updatePropensities(self, reward):
      """Ανανεώνει τις ροπές για όλες τις δράσεις. Η ροπή για την τελευταία επιλε-
         γμένη δράση θα ανανεωθεί χρησιμοποιώντας την τιμή ανάδρασης που
         προέκυψε από την εφαρμογή της δράσης.

         Αν j είναι ο δείκτης της τελευταία επιλεγμένης δράσης,  r_j είναι η ανταμοι-
         βη που λήφθηκε από αυτήν, i είναι η τωρινή δράση, q_i είναι η ροπή της i και
         phi η παράμετρος αμεσότητας, τότε η συνάρτηση ανανέωσης δίνεται απ' τον
         τύπο:
                 q_i = (1 - phi) * q_i + E(i, r_j)
      """
        phi = self.recency

        for i in range(self.domainSize):
            carryOver = (1 - phi) * self.module.getPropensity(i)
            experience = self.experience(i, reward)
            self.module.setPropensity(i, carryOver + experience)
#           propensities[i] = (1 - r) * propensities[i] + experience(i, reward)

    def experience(self, actionIndex, reward):
    """Αυτή είναι η συνάρτηση εμπειρίας για τον κλασικό RE αλγόριθμο.
       Οι ροπές για όλες τις δράσεις ανανεώνονται. Αν ο δείκτης actionIndex
       δείχνει σε δράση με την οποία σχετίζεται η ανταμοιβή (συνήθως η
       τελευταία επιλεγμένη δράση) τότε απλώς προσαρμόζει το βάρος ανάλογα με την
       παράμετρο πειραματισμού. Αλλιώς, προσαρμόζει το βάρος ανάλογα με μια
       μικρότερη ποσότητα από αυτή της ανταμοιβής.

       Αν j είναι ο δείκτης της τελευταία επιλεγμένης δράσης, r_j η αντίστοιχη αντα-
```

μοιβή της, i η τωρινή δράση, n το μέγεθος της domain δράσεων και e η παράμετρος πειραματισμού, τότε η συνάρτηση πείρας είναι:

$$\mathrm{E}(i, r\_j) = \begin{cases} r\_j * (1-e), i = j \\ r\_j * \dfrac{e}{n-1}, i \neq j \end{cases}$$

```
"""
    e = self.experimentation
    rewardedIndex = self.actionIDList.index(self.lastSelectedAction)

    if actionIndex == rewardedIndex:
        experience = reward * (1 - e)
    else:
        experience = reward * (e / (self.domainSize - 1))

    return experience

def updateProbabilities(self):

    if self.parameters.useBoltz:
        self.generateBoltzmanProbs()
    else:
        # Μέθοδος αναλογικής πιθανότητας.
        propensities = self.module.propensities

        summedProps = 0.0
        for prop in propensities:
            summedProps += prop

        for index, actionID in enumerate(self.actionIDList):
            newProb = propensities[index] / summedProps
            self.module.setProbability(actionID, newProb)

def generateBoltzmanProbs(self):
"""Παράγει τις πιθανότητες των δράσεων χρησιμοποιώντας την κατανομή
    Boltzmann με σταθερά θερμοκρασίας.
"""
    propensities = self.module.propensities
    coolingParam = self.boltzmannTemp

    summedExps = 0.0
    for prop in propensities:
        summedExps += math.exp(prop / coolingParam)


    # Για κάθε δράση υπολογίζει τη σχετική πιθανότητα επιλογής.
    #  p_i = e^(q_i / T) / sum_{j=1}^{N} e^(q_j / T)
```

```
        for index, actionID in enumerate(self.actionIDList):
            newProb = math.exp(propensities[index] / coolingParam) / summedExps
            self.module.setProbability(actionID, newProb)

    def learn(self):
    """Ενεργοποιεί τη διαδικασία μάθησης σύμφωνα με τον τροποποιημένο αλγόριθμο
        (MRE). Η ανάδραση εκλαμβάνεται ως ανταμοιβή για την τελευταία δράση που
        επιλέγεται απ' τη μηχανή αυτή. Οι είσοδοι στην πολιτική που σχετίζονται με τη
        δράση ανανεώνονται αντίστοιχα.

        Η ανάδραση χρησιμοποιείται για να ανανεώνει την πιθανότητα επιλογής δράσης
        ανάλογα με τον αλγόριθμο μάθησης και παραμετροποιείται, καθώς η
        απαιτούμενη είσοδος ποικίλει δεδομένου του επιλεγμένου αλγορίθμου και του
        περιβάλλοντος της προσομοίωσης.

        Σημείωση: Συνηθέστερα, η ανάδραση είναι για την τελευταία επιλεγμένη δράση
        κι έτσι η δεδομένη ActionID θα δείχνει σ' αυτή την Action.
    """
        rewards = self.ds['reward']
        self.updatePropensities(rewards[-1])
        self.updateProbabilities()
        self.period += 1

#   def getAction(self):
     """ Ενεργοποιεί τη δομή με βάση την τελευταία παρατήρηση και
         αποθηκεύει το αποτέλεσμα ως τελευταία δράση (lastAction).

         Εκμαιεύει μια νέα επιλογή δράσης. Η δράση θα επιλεγεί σύμφωνα με
         τον κανόνα επιλογής της SimpleStatelessPolicy. Οι δράσεις επιλέγονται
         από την DiscreteFiniteDomain.
     """
#       nextAction = self.policy.generateAction()
#       nextAction = super(RothErev, self).getAction()
#       self.lastSelectedAction = nextAction
#       return nextAction

    def reset():
    """Καθαρίζει όλη τη γνώση που έχει ληφθεί. Οι ροπές δράσεων τίθενται
        ίσες με την ισχύουσα αρχική τιμή.
    """
        super(RothErev, self).reset()
        self.init()


#   def makeParameters(self):
     """ Δημιουργεί ένα προκαθορισμένο σύνολο παραμέτρων που μπορεί
         να χρησιμοποιηθεί από τον χρήστη.
     """
#       raise NotImplementedError
```

```
    def validateParameters(self):
    """Ελέγχει εάν οι τιμές για όλες τις παραμέτρους είναι έγκυρες.
    """
        valid = True
        if self.boltzmannTemp < 0.0:
            raise ValueError, "Παράμετρος ψύξης για Gibbs-Boltzmann "
            "η παραγωγή πιθανότητας πρέπει να δίνει θετική τιμή"
            valid = False
        if not 0.0 <= self.experimentation <= 1.0:
            raise ValueError, "Πρέπει 0<e<1"
            valid = False
        if self.initialPropensity < 0.0:
            raise ValueError, "Η αρχική τιμή ροπής πρέπει να είναι μη αρνητική"
            valid = False
        if not 0.0 <= self.recency <= 1.0:
            raise ValueError, "Η παράμετρος αμεσότητας πρέπει να είναι 0<r<1 "
            valid = False
        return valid

class VRELearner(RELearner):
    """Τροποποιημένη Μέθοδος Roth-Erev (MRE).

        Πηγή: James Nicolaisen, Valentin Petrov, and Leigh Tesfatsion, "Market
          Power and Efficiency in a Computational Electricity Market with
          Discriminatory Double-Auction Pricing," IEEE Transactions on
          Evolutionary Computation, Volume 5, Number 5, 2001, 504-523.[18]
    """
    def updateProbabilities(self):
    """Ανανεώνει την πιθανότητα για κάθε δράση που επιλέγεται με βάση την πολιτική
    """
        self.generateBoltzmanProbs()

    def experience(self, actionIndex, reward):
     """Είναι η συνάρτηση ανανέωσης του τροποποιημένου αλγορίθμου Roth και
        Erev (MRE).

        Αν j είναι ο δείκτης της τελευταία επιλεγμένης δράσης, r_j η αντίστοιχη αντα-
        μοιβή της, i η τωρινή δράση, q_i είναι η ροπή για τη δράση i, n το μέγεθος
        της domain δράσεων και e η παράμετρος πειραματισμού, τότε η
        συνάρτηση πείρας είναι:

                                ME(i, r_j) = { r_j * (1 - e), i = j
                                             { q_i * e/(n-1), i ≠ j
     """
        e = self.parameters.experimentation
        rewardedIndex = self.actionIDList.index(self.lastSelectedAction)

        if actionIndex == rewardedIndex:
            experience = reward * (1 - e)
```

```
        else:
            propensity = self.policy.getPropensity(actionIndex)
            experience = propensity * (e / (self.domainSize - 1))

        return experience

class AREParameters(REParameters):
""" Παράμετροι που απαιτούνται για μια εξελιγμένη εκδοχή του RE
    Αλγορίθμου. Επιπρόσθετα χαρακτηριστικά:

    Εναλλακτικός τρόπος για την παραγωγή πιθανοτήτων δράσεων
    Εναλλακτικός τρόπος για 'μοίρασμα' ανταμοιβών μεταξύ παρόμοιων δράσεων
    Μια τιμή που καθορίζει το πόσο όμοιες είναι δυο ή περισσότερες δράσεις
"""
    def __init__(self):
        # Ποια μέθοδος 'μοιράσματος' ανταμοιβών να χρησιμοποιηθεί
        selectedSpillover = None

        # Λίστα των διαθέσιμων μεθόδων διαμοιράσματος.
        spilloverList = []


    def init(self):
        if self.spilloverList is None:
            self.spilloverList = []

        self.buildSpilloverSelector()

    def buildSpilloverSelector(self):
        # Προσθέτει την επιλογή NoSpillover αν δεν υπάρχει ήδη.
        if NoSpillover not in self.spilloverList:
            noSpill = NoSpillover()
            self.spilloverList.append(noSpill)

        # Προσθέτει την επιλογή StandardSpillover αν δεν υπάρχει ήδη.
        if StandardSpillover not in self.spilloverList:
            standard = StandardSpillover(self.getExperimentation(),
                                         self.getNumberOfActions())
            self.spilloverList.append(standard)

class ARELearner(RELearner):
"""Επέκταση της VRELearner. Αυτή η μηχανή εφαρμόζει την ίδια τροποποιημένη
    μέθοδο Roth-Erev με επιπρόσθετα χαρακτηριστικά.
"""
    def __init__(self):
# Σημαία για χρήση σχετικών ροπών για την παραγωγή πιθανοτήτων δράσεων.
# Η πιθανότητα για κάθε δράση είναι η ροπή της δράσης προς το σύνολο των ροπών
#
```

```
# $p_i = \frac{q_i}{\sum_{j=1}^{N} q_j}$

        USE_RELATIVE_PROPENSITY_PROBABILITY = 10101
        # Σημαία για χρήση κατανομής Boltzmann για την παραγωγή πιθανοτήτων
        # δράσεων
        # $p_i = \frac{e^{q_i / T}}{\sum_{j=1}^{N} e^{q_j / T}}$
        USE_BOLTZMAN_PROBABILITY = 11212
        # Σημαία που υποδηλώνει ποιά μέθοδος απόδοσης βάρους να χρησιμοποιηθεί
        # κατά τη διάρκεια της ανανέωσης πιθανοτήτων δράσεων από μια δεδομένη
        # ανταμοιβή. Αυτό καθορίζει το κατά πόσο η ανταμοιβή διαμοιράζεται μεταξύ
        # των δράσεων που είναι όμοιες με τη δράση από την οποία προήλθε η
        # ανταμοιβή. Η ομοιότητα μεταξύ των δράσεων καθορίζεται απ' την αντίστοιχη
        # τιμή που αναφέρθηκε παραπάνω.
        spilloverType = -1

        spillover = SpilloverWeightGenerator()

        EXPONENTIAL_SPILLOVER = 20101
        LOGARITHMIC_SPILLOVER = 21212
        LINEAR_SPILLOVER = 22323
        STANDARD_SPILLOVER = 23434
         NO_SPILLOVER = 29999

    def init(self):
        super(ARELearner, self).init()
        self.spillover = None

    def experience(self, actionIndex, reward):
        if self.spillover is None:
            return super(ARELearner, self).experience(actionIndex, reward)

        responseValue = 0
        domain = self.policy.getActionDomain()

        action = domain.getAction(actionIDList.get(actionIndex))

        responseValue = reward * self.spillover.generateWeight(action)

        return responseValue

class AdvancedRothErevLearner(RELearner):

def experience(self, actionIndex, reward):
        if self.spillover is None:
```

```
            return super(ARELearner, self).experience(actionIndex, reward)

        responseValue = 0.0
        weight = 0.0
        similarity = 0.0

        domain = self.policy.getActionDomain()

        action = domain.getAction(actionIDList.get(actionIndex))

        responseValue = reward * self.spillover.generateWeight(action)

        return responseValue

    def updateProbabilities(self):
        """Ανανεώνει την πιθανότητα για κάθε δράση που επιλέγεται με βάση
            την ακολουθούμενη πολιτική.
        """
        method = self.parameters.getProbabilityMethod()

        if method == self.USE_RELATIVE_PROPENSITY_PROBABILITY:
            self.generateProportionalProbs()
        elif method == self.USE_BOLTZMAN_PROBABILITY:
            self.generateBoltzmanProbs()
        else:
            self.generateBoltzmanProbs()

    def generateProportionalProbs(self):
        """Παράγει τις πιθανότητες των δράσεων χρησιμοποιώντας πιθανοτική
            κατανομή.
            Προσοχή: Σε περίπτωση που λάβει αρνητική τιμή για ανταμοιβή, θα
            προκύψει αρνητική τιμή πιθανότητας.
         """
        summedProps = 0.0
        newProb = 0.0

        for actID in self.actionIDList:
            summedProps += self.policy.getPropensity(actID)

        # Για κάθε δράση, διαιρεί τη ροπή της με το σύνολο όλων των ροπών
        # Έπειτα παράγει την πιθανότητα ως αποτέλεσμα
        for actID in self.actionIDList:
            newProb = self.policy.getPropensity(actID) / summedProps
            self.policy.setProbability(actID, newProb)
```

Με '#' και πράσινο χρώμα εμφανίζονται σχόλια ή εντολές που είναι προαιρετικές. Μετά και πριν από '"""'εμφανίζονται επίσης επεξηγηματικά σχόλια.

Βιβλιογραφία

[1] International Energy Agency (IEA). “Electricity/Heat in World in 2009”, 2009. Retrieved from http://www.iea.org/stats/electricitydata.asp?COUNTRY_CODE=29 (Last Access: 17-09-2013).

[2] Paolo Giabardo and Marco Zugno. Competitive Bidding and Stability Analysis in Electricity Markets Using Control Theory. Master’s Thesis, The Technical University of Denmark, July 2008.

[3] Leigh Tesfatsion. Learning Algorithms : Illustrative Examples. Notes for Economics 308 (Agent-Based Computational Economics), Lectures. Iowa State Universtity.

[4] Andrew Barto. Searching in the Right Space, Perspectives on Computational Reinforcement Learning. Okinawa Computational Neuroscience Course Lectures, Autonomous Learning Laboratory – Department of Computer Science, July 2005.

[5] Magnus Borga. Hierarchical Reinforcement Learning. In S Gielen, B Kappen, eds., ICANN'93. Amsterdam: Springer-Verlag, 1993.

[6] Magnus Borga. Reinforcement Learning Using Local Adaptive Models, 1995. Thesis No. 507, ISBN 91-7871-590-3.

[7] K. S. Narendra and M. A. L. Thathachar. Learning automata – a survey. *IEEE Trans. on Systems, Man, and Cybernetics*, 4(4): pages 323-334,1974.

[8] E. L. Thorndike. Animal Intelligence: An experimental study of the associative process in animals. Psychological Review, 2(8), 1998. Monogr. Suppl.

[9] I. P. Pavlov. Selected Works. Foreign Languages Publishing House, Moscow, 1955.

[10] B. F. Skinner. The Behavior of Organisms: An Experimental Analysis. Prentice-Hall, Englewood Cliffs, N.J., 1938.

[11] G. Tresauro. Neurogammon: a neural network backgammon playing program. In *IJCNN Proceedings III*, pages 33-39, 1990.

[12] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore, Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, Volume 4, pages 237-285, 1996.

[13] Bill Smart. Reinforcement Learning: A User's Guide, Lectures. Department of Computer Science and Engineering, Washington University in St. Louis.

[14] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introducrion. A Bradford Book. MIT Press, Cambridge, MA, 1998.

[15] Satinder Singh, Richard Lewis, Andrew Barto και Jonathan Sorg. Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective. *IEEE Trans. on Autonomous Mental Development*, Vol 2, No 2, 2010.

[16] Ido Erev and Alvin E. Roth. Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria. *The American Economic Review*, Volume 88, No 4, pages 848-881, 1998.

[17] Mridul Pentapalli. A Comparative study of Roth-Erev and Modified Roth-Erev reinforcement learning algorithms for uniform-price double auctions. Iowa State University, Ames, IA, Thesis Talk, March 2008.

[18] James Nicolaisen, Valentin Petrov and Leigh Tesfatsion. Market Power and Efficiency in a Computational Electricity Market With Discriminatory Double-Auction Pricing. *ISU Economic Repor*, No 52, pages 504-523, August 27, 2000; revised August 24, 2001.

[19] Ido Erev and Alvin E. Roth. Learning in extensive form games: Experimental data and simple dynamic models in the intermediate term. *Games Economic Behaviour Journal*, Vol 8, pages 164-212, 1995.

[20] Ρυθμιστική Αρχή Ενέργειας. Retrieved from www.rae.gr/ (Last access: 07-10-2013).

[21] Λειτουργός Αγοράς Ηλεκτρικής Ενέργειας. Μηνιαίο Δελτίο Συστήματος Συναλλαγών ΗΕΠ, Αύγουστος 2013.
Retrieved from http://www.lagie.gr/ (Last access: 09-10-2013).

[22] Richard W. Lincoln. Retrieved from
http://pydoc.net/Python/Pylon/0.3.2/pylon.pyreto.roth_erev/
(Last access: 10-10-2013).

[23] Python Documentation Network (PyDoc.net). Retrieved from http://pydoc.net/ (Last access: 10-10-2013).

[24] Charles Gieseler. A Java Reinforcement Learning Module for the Recursive Porous Agent Simulation Toolkit: facilitating study and experimentation with reinforcement learning in social science multi-agent simulations. Master of Science Thesis, Iowa State University, 2005.