NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

PhD THESIS

E-Negotiations for trading Commodities and Services: Predictive Strategies

Marisa I. Masvoula

ATHENS

10 2012

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ηλεκτρονικές Διαπραγματεύσεις για την Προμήθεια Αγαθών και Υπηρεσιών: Στρατηγικές Πρόβλεψης

Μαρίζα Ι. Μάσβουλα

ΑΘΗΝΑ

10 2012

PhD THESIS

E-Negotiations for trading Commodities and Services: Predictive Strategies

Marisa I. Masvoula

SUPERVISOR: Drakoulis Martakos, Associate Professor UoA

THREE-MEMBER ADVISORY COMMITTEE:
Drakoulis Martakos, Associate Professor UoA
Constantine Halatsis, Emeritus Professor UoA
Panagiotis Georgiadis, Professor UoA

SEVEN-MEMBER EXAMINATION COMMITTEE

Drakoulis Martakos,
Associate Professor UoA

Grigoris Mentzas
Professor NTUA

Constantine Halatsis,
Emeritus Professor UoA

Andreas Stafilopatis,
Professor NTUA

Panagiotis Georgiadis,
Professor UoA

G. Giaglis,
Professor AUEB

Panagiotis Stamatopoulos,
Assistant Professor UoA

Examination Date 8/10/2012

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ηλεκτρονικές Διαπραγματεύσεις για την Προμήθεια Αγαθών και Υπηρεσιών: Στρατηγικές Πρόβλεψης

Μαρίζα Ι. Μάσβουλα

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Δρακούλης Μαρτάκος, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Δρακούλης Μαρτάκος, Αναπληρωτής Καθηγητής ΕΚΠΑ
Κωνσταντίνος Χαλάτσης, Ομότιμος Καθηγητής ΕΚΠΑ
Παναγιώτης Γεωργιάδης, Καθηγητής ΕΚΠΑ

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Δρακούλης Μαρτάκος,
Αναπληρωτής Καθηγητής ΕΚΠΑ

Γρηγόρης Μέντζας,
Καθηγητής ΕΜΠ

Κωνσταντίνος Χαλάτσης,
Ομότιμος Καθηγητής ΕΚΠΑ

Ανδρέας Σταφυλοπάτης,
Καθηγητής ΕΜΠ

Παναγιώτης Γεωργιάδης,
Καθηγητής ΕΚΠΑ

Γ. Γιαγλής,
Καθηγητής ΟΠΑ

Παναγιώτης Σταματόπουλος,
Επίκουρος Καθηγητής ΕΚΠΑ

Ημερομηνία εξέτασης 8/10/2012

ABSTRACT

Negotiation can be viewed as an exchange mechanism of two or more parties that search for a mutually acceptable agreement. The art and science of negotiation has attracted the interest of many different scientific fields, therefore different viewpoints and approaches have been developed by psychological and sociological sciences, as well as by economists, mathematicians and computer scientists.

Particularly in the field of computer science the contribution is multifold, as the technological evolution has paved the way and the means to establish negotiations. Electronic markets (e-markets) and the provision of tangible or intangible objects through electronic platforms constitute an example of transferring the negotiation arena to electronic settings. The development of support systems and of automated agents advances the development of socio-technical systems and also contributes to the evolution of negotiation science. During the last decade, the application of learning techniques is very common in negotiation support systems and in automated agents that undertake various stages of the negotiation process. Negotiation processes are complex, as negotiators often seek to maximize their utility (a measure of individual satisfaction).

The current thesis takes into account the advances in the field of electronic bi-lateral negotiations, adopting state-of-the-art protocols, as well as strategies that characterize the behavior of each party. The research objective is the application of strategies that are based on the estimation of the counterpart’s next offer, and give the predictive agent the advantage to establish agreements that are more beneficial. Another issue that is contemplated is that of the risk of employing a predictive strategy. A new strategy that incorporates the adoption of different attitudes towards risk is proposed.

This thesis also focuses on the AI-based models that are used for the purpose of estimating the counterpart’s next offer, as well as on the comparison of these models. The main problem of the majority of related applications is their inability to capture the dynamics of turbulent negotiation environments, and provide accurate estimations also in cases where the data distributions change. For this reason the utilization of models that are based on the data that are acquired from the current negotiation discourse, as well as the utilization of models that adapt their structure in time are examined. More specifically the application of neural networks that adapt their structure on the basis of a genetic algorithm, as well as a simple evolving connectionist structure, eMLP, that does one-pass learning of data are developed and assessed. Numerous experiments that result from simulations of different negotiation environments and justify the proposed solutions are presented. Finally, future research issues that relate to the domain of application of the proposed strategy as well as to other learning models that could be enhanced with the negotiating agents in order to estimate the counterpart’s next offer are also discussed.

SUBJECT AREA: Electronic Commerce

KEYWORDS: electronic negotiations, negotiating agents, neural networks, genetic algorithms, predictive strategies

ΠΕΡΙΛΗΨΗ

Η διαπραγμάτευση αποτελεί έναν από τους βασικότερους μηχανισμούς εύρεσης αμοιβαίας αποδεκτής λύσης ή «συμφωνίας», μεταξύ δύο ή περισσότερων μερών. Ειδικότερα στον τομέα της πληροφορικής η συνεισφορά είναι πολλαπλή, καθώς η εξέλιξη της τεχνολογίας οδηγεί σε εξέλιξη του τρόπου και των μέσων υλοποίησης των διαπραγματεύσεων. Οι ηλεκτρονικές «αγορές» (electronic markets) και η παροχή αγαθών και υπηρεσιών μέσα από ηλεκτρονικές πλατφόρμες, συνιστούν ένα παράδειγμα μετατόπισης της αρένας των διαπραγματεύσεων στον ηλεκτρονικό χώρο. Η ανάπτυξη συστημάτων υποστήριξης και πρακτόρων λογισμικού διαπραγμάτευσης (negotiation software agents) προάγουν τη δημιουργία κοινωνικο-τεχνικών συστημάτων (socio-technical systems) και συντελούν επίσης στην εξέλιξη της επιστήμης της διαπραγμάτευσης. Τα τελευταία χρόνια, η χρήση τεχνικών μηχανικής μάθησης είναι πολύ διαδεδομένη στα συστήματα υποστήριξης αλλά και σε εξελιγμένους πράκτορες λογισμικού, που αναλαμβάνουν να διεκπεραιώσουν συναλλαγές σε πραγματικό χρόνο και με την μεγαλύτερη δυνατή ικανοποίηση των στόχων που έχουν αρχικά τεθεί από τους εντολείς τους. Η παρούσα διατριβή αξιοποιεί τις εξελίξεις στην περιοχή των διμερών ηλεκτρονικών διαπραγματεύσεων, υιοθετώντας θεμελιωμένα πρωτόκολλα και στρατηγικές που χαρακτηρίζουν τη συμπεριφορά του εκάστοτε συμμετέχοντα. Αντικείμενο έρευνας αποτελεί η χρήση στρατηγικών πρόβλεψης μελλοντικών προσφορών του αντιπάλου, προσφέροντας με τον τρόπο αυτό πλεονέκτημα κινήσεων προς συμφωνίες με μεγαλύτερο όφελος. Επίσης μελετάται ο κίνδυνος χρήσης εργαλείων πρόβλεψης και προτείνεται μια νέα στρατηγική που επιτρέπει την υιοθέτηση διαφορετικών συμπεριφορών απέναντι στον κίνδυνο. Γίνεται εκτενής καταγραφή των μοντέλων μηχανικής μάθησης που χρησιμοποιούνται με σκοπό την πρόβλεψη της επόμενης προσφοράς του αντιπάλου, καθώς επίσης συγκριτική αξιολόγηση βάση βιβλιογραφικών αναφορών. Το βασικό μειονέκτημα της πλειοψηφίας των μεθόδων είναι η αδυναμία παροχής έγκυρης πρόβλεψης σε δυναμικά περιβάλλοντα, όταν αλλάζουν οι κατανομές των δεδομένων στα οποία βασίστηκαν τα αρχικά μοντέλα πρόβλεψης. Για το λόγο αυτό ερευνάται η χρήση μοντέλων που εκπαιδεύονται με δεδομένα που εξάγονται από την τρέχουσα διαπραγμάτευση, καθώς επίσης και δομών που μεταβάλλονται με το χρόνο. Στα πλαίσια τις διατριβής μελετάται η χρήση νευρωνικών δικτύων που εξελίσσουν τη δομή τους σε κάθε βήμα πρόβλεψης με χρήση γενετικού αλγορίθμου, καθώς επίσης και η απλή αυτοεξελισσόμενη δομή eMLP, που μαθαίνει με ένα μόνο πέρασμα των δεδομένων, καθιστώντας ταχύτατη τη διαδικασία μάθησης. Παρουσιάζονται εκτενή πειραματικά αποτελέσματα που προκύπτουν από προσομοιώσεις διαφορετικών περιβαλλόντων διαπραγμάτευσης, και αποδεικνύουν την επάρκεια των λύσεων που προτείνονται, αφού οδηγούν στα προσδοκώμενα αποτελέσματα. Τέλος παρατίθενται θέματα για μελλοντική έρευνα που σχετίζονται με το πεδίο εφαρμογής της προτεινόμενης στρατηγικής, αλλά και με άλλα μοντέλα που θα μπορούσαν να συνδυαστούν με τους πράκτορες διαπραγμάτευσης, με σκοπό την εκτίμηση της επόμενης προσφοράς του αντιπάλου.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ηλεκτρονικό Εμπόριο

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ηλεκτρονικές διαπραγματεύσεις, πράκτορες διαπραγμάτευσης, νευρωνικά δίκτυα, γενετικοί αλγόριθμοι, στρατηγικές πρόβλεψης

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα Αναπληρωτή Καθηγητή Δρακούλη Μαρτάκο, για τη στήριξή του και για τη δυνατότητα που μου έδωσε να διεξάγω την έρευνα αυτή στο τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών.

Θερμές ευχαριστίες θα ήθελα να εκφράσω και στον Ομότιμο Καθηγητή Κωνσταντίνο Χαλάτση για την ενθάρυνση, την καθοδήγηση και την υποστήριξη που μου προσέφερε καθόλη τη διάρκεια της έρευνας.

Επίσης ευχαριστώ τον Καθηγητή Παναγιώτη Γεωργιάδη και τον ερευνητή Σταύρο Περαντώνη.

Τέλος θα ήθελα να ευχαριστήσω τα μέλη της οικογένειάς μου, τους γονείς και το σύζυγό μου, για την υποστήριξη και ενθάρρυνση που μου έδειξαν. Είμαι σε όλους ευγνώμων.

LIST OF PUBLICATIONS

1) Masvoula, M., Halatsis, C., Martakos, D. (2012). Predictive Automated Negotiators with different Attitudes Towards Risk. Engineering Intelligent Systems Journal, 20(1):89-105

2) Masvoula, M., Halatsis, C., Martakos, D. (2012). Forecasting negotiation counterpart's offers with the use of neural networks: A focus on session long learning agents. Neurocomputing and Applications. (Submitted for Publication)

3) Masvoula, M., Halatsis, C., Martakos, D. (2011). Integrating Negotiating Agents with Evolving Connectionist Structures to Facilitate the Prediction of Counterpart's Responses. In Proceedings of the International Conference on Neural Networks (ICNN 2011) (July 27-29, 2011, Paris, France).

4) Masvoula, M., Halatsis, C., Martakos, D. (2011). Predictive Automated Negotiators Employing Risk-Seeking and Risk-Averse Strategies. In Proceedings of 12th EANN/7th AIAI Joint Conferences (September 15-18, 2011, Corfu, Greece). EANN/AIAI (1) 2011: 325-334.

5) Masvoula, M., Kanellis, P., Martakos, D. (2010). A Review of Learning Methods Enhanced in Strategies of Negotiating Agents. In Proceedings of 12th International Conference on Enterprise Information Systems (June 8-12,2010,Madeira). ICEIS pp. 212-219.

6) Masvoula, M., Kanellis, P., Martakos, D. (2010). Evolving Structures for Predictive Decision Making in Negotiations. In Proceedings of 12th International Conference on Enterprise Information Systems (June 8-12,2010,Madeira). ICEIS pp. 391-394.

7) Masvoula, M., Kanellis, P., and Martakos, D. (2009). Integrating agents with connectionist systems to extract negotiation routines. In Proceedings of the 11th International Conference on Enterprise Information Systems (6 – 10, May 2009 Milan, Italy). ICEIS (2): 251-256.

8) Masvoula, M., Kontolemakis, G., Kanellis, P., and Martakos, D. (2005). Design and Development of an Anthropocentric Negotiation Model. In Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (July 19-22, 2005, Munich). CEC. IEEE Computer Society, Washington, DC, pp. 383-386.

9) Kontolemakis, G., Masvoula, M., Kanellis, P., Martakos, D. (2005). Transcending Taxonomies with Generic and Agent-Based E-Hub Architectures. In Proceedings of the Seventh International Conference on Enterprise Information Systems (May 24-28, 2005, Miami Florida). ICEIS vol. 4, pp. 297-300

10) Kontolemakis, G., Masvoula, M., Kanellis, P., Martakos, D. (2005). Software Agents and the Quest for a Generic Virtual Marketplace Architecture. In Proceedings of the Sixteenth International Conference on International Resources Management, Managing Modern Organization with Information Technology (May 15-18, 2005, San Diego California). IRMA.

ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΔΙΑΤΡΙΒΗΣ

Αντικείμενο της διατριβής αποτελεί η μελέτη ηλεκτρονικών διαπραγματεύσεων μεταξύ πρακτόρων λογισμικού διαπραγμάτευσης (negotiation software agents) και ειδικότερα η δυνατότητα αυτών να υιοθετήσουν «έξυπνες» στρατηγικές, με στόχο την αύξηση της ατομικής τους ωφέλειας. Στα πλαίσια της διατριβής προτάθηκε και αναπτύχθηκε μια νέα στρατηγική, η οποία βασίζεται στην εκτίμηση της επόμενης προσφοράς του αντιπάλου, και επιτρέπει την αφομοίωση διαφορετικών συμπεριφορών απέναντι στον κίνδυνο. Επιπλέον, εξετάστηκε η απόδοση των υπαρχόντων μοντέλων πρόβλεψης σε δυναμικά περιβάλλοντα, και δοκιμάστηκαν συσχετιστικά μοντέλα που μεταβάλλουν τη δομή τους με το χρόνο. Βασικό απόφθεγμα αποτελεί η ανάγκη για επανεκπαίδευση των μοντέλων πρόβλεψης με δεδομένα που προέρχονται από την τρέχουσα διαπραγμάτευση.

Στα δύο πρώτα κεφάλαια παρουσιάζονται βασικές αρχές, ορολογίες, επιστημονικές προσεγγίσεις, καθώς επίσης και συστήματα που έχουν αναπτυχθεί για την υποστήριξη διαφόρων φάσεων της διαπραγμάτευσης. Το ερευνητικό πεδίο είναι διεπιστημονικό, καθώς ψυχολόγοι, κοινωνιολόγοι, πολιτικοί επιστήμονες, μαθηματικοί και οικονομολόγοι έχουν συνεισφέρει στη διαμόρφωση θεωριών, μοντέλων και μεθόδων. Ο ορισμός που υιοθετείται στην παρούσα διατριβή είναι ο ακόλουθος:

«Η διαπραγμάτευση αποτελεί έναν από τους βασικότερους μηχανισμούς αναζήτησης αμοιβαία αποδεκτής λύσης μεταξύ δύο ή περισσότερων μερών».

Πρόκειται για μια επαναληπτική διαδικασία όπου οι συμμετέχοντες στέλνουν εναλλάξ προσφορές μέχρι να υπάρξει συμφωνία, ή να παρέλθει ο μέγιστος προκαθορισμένος χρόνος. Στο βιβλίο του καθηγητή Howard Raiffa «Τέχνη και Επιστήμη της Διαπραγμάτευσης» [16], αναφέρονται μια σειρά από χαρακτηριστικά που χρησιμοποιούνται συνήθως για τη διαφοροποίηση και κατηγοριοποίηση των διαπραγματεύσεων. Στην τρέχουσα διατριβή γίνεται μελέτη διμερών διαπραγματεύσεων για την προμήθεια αγαθών ή υπηρεσιών που περιγράφονται από μια σειρά ποσοτικών χαρακτηριστικών. Τα συμφέροντα των συμμετεχόντων είναι αντικρουόμενα, και ο στόχος του κάθε διαπραγματευτή είναι να αυξήσει το προσωπικό του κέρδος. Θεωρούμε ότι το περιβάλλον είναι δυναμικό, δηλαδή δεν παρουσιάζεται επαναληψιμότητα στις ενέργειες του αντιπάλου. Επίσης θεωρούμε ότι οι στρατηγικές παράμετροι και οι προτιμήσεις των διαπραγματευτών αποτελούν ιδιωτική πληροφορία.

Την τελευταία δεκαετία η συνεισφορά της πληροφορικής είναι πολύ σημαντική, καθώς έχει προσφέρει το μέσο για τη διεξαγωγή των διαπραγματεύσεων σε ηλεκτρονικές πλατφόρμες. Έχουν δημιουργηθεί ηλεκτρονικά συστήματα διαπραγμάτευσης που ευνοούν την οργάνωση, διευκόλυνση, υποστήριξη ακόμα και αυτοματοποίηση των διαδικασιών. Σε αυτά τα συστήματα ανήκουν και οι πράκτορες λογισμικού, που ενσωματώνουν μοντέλα και διαδικασίες για τη διεκπεραίωση διαφόρων ή και όλων των σταδίων μιας διαπραγμάτευσης.

Ο μηχανισμός της διαπραγμάτευσης είναι ευρύτατα διαδεδομένος στις ηλεκτρονικές αγορές. Ένα παράδειγμα ανταγωνιστικού περιβάλλοντος που μοντελοποιείται συχνά με χρήση πρακτόρων λογισμικού διαπραγμάτευσης είναι και οι προθεσμιακές αγορές ηλεκτρικής ενέργειας. Μεγάλοι πελάτες, κατά βάση αγροτικός και βιομηχανικός τομέας, προμηθεύονται ενέργεια από ενδιάμεσους προμηθευτές και παραγωγούς. Επίσης μικροί πελάτες, κατά βάση νοικοκυριά, προμηθεύονται ενέργεια επίσης από ενδιάμεσους προμηθευτές. Στις προθεσμιακές αγορές, οι συμμετέχοντες οδηγούνται σε συμφωνίες μελλοντικής εκπλήρωσης. Μέσω της διαπραγμάτευσης συμφωνούν τους όρους παροχής της ενέργειας, όπως την ποσότητα και την τιμή των κιλοβατώρων, τη διάρκεια παροχής, καθώς και το ποσοστό επιστροφής χρημάτων, με το οποίο βαρύνεται ο προμηθευτής σε περίπτωση που δεν τηρήσει τη συμφωνία.

Το τρίτο κεφάλαιο επικεντρώνεται στο πρωτόκολλο και τις στρατηγικές που υιοθετούνται από τους πράκτορες λογισμικού διαπραγμάτευσης. Το πρωτόκολλο καθορίζει τους κανόνες, τις επιτρεπόμενες ενέργειες σε κάθε στάδιο. Πιο συγκεκριμένα, το πρωτόκολλο ξεκινά με τη φάση σχεδιασμού. Κατά την φάση αυτή, τα δύο μέρη καλούνται να καθορίσουν τις προτιμήσεις τους. Θέτουν μέγιστη και ελάχιστη τιμή για κάθε ποσοτικό χαρακτηριστικό, καθορίζουν το μέγιστο χρόνο που διατίθενται να διαπραγματευτούν, τη συνάρτηση αποτίμησης προσφοράς που τους επιτρέπει τη σύγκριση μεταξύ διαφορετικών προσφορών, καθώς επίσης τη στρατηγική, που θα καθορίσει τη συνολική συμπεριφορά κατά τη διάρκεια της διαπραγμάτευσης. Αφού ολοκληρωθεί η φάση σχεδιασμού, οι συμμετέχοντες περνούν στο κυρίως μέρος της διαπραγμάτευσης, που αφορά την ανταλλαγή προσφορών. Σε κάθε γύρο, ο διαπραγματευτής χρησιμοποιεί την προκαθορισμένη στρατηγική του για να παράγει την προσφορά που προτίθεται να στείλει. Κάνοντας χρήση της συνάρτησης αποτίμησης προσφοράς, συγκρίνει την προσφορά αυτή με αυτήν που του έστειλε ο αντίπαλος στον προηγούμενο γύρο. Αν η τελευταία είναι πιο συμφέρουσα την αποδέχεται και η διαδικασία τερματίζει με επιτυχία. Σε αντίθετη περίπτωση, η διαδικασία συνεχίζεται μέχρι να παρέλθει ο μέγιστος προκαθορισμένος χρόνος, οπότε και η διαπραγμάτευση λήγει ανεπιτυχώς.

Θεμελιώδους σημασίας για την έκβαση της διαπραγμάτευσης και την επίτευξη των στόχων των διαπραγματευτών αποτελεί η στρατηγική που θα υιοθετήσουν. Μέσω αυτής, οι συμμετέχοντες σχεδιάζουν τις ενέργειές τους σε κάθε γύρο. Ο όρος «στρατηγική» χρησιμοποιείται συχνά ως συνώνυμο της συμπεριφοράς. Οι πράκτορες λογισμικού διαπραγμάτευσης που δεν ενσωματώνουν τεχνικές μηχανικής μάθησης, υιοθετούν αποκριτικούς μηχανισμούς. Οι μηχανισμοί αυτοί βασίζονται σε ένα συνδυασμό τακτικών, δηλαδή ένα συνδυασμό από συναρτήσεις γεννήτριες προσφορών [25]. Οι βασικές κατηγορίες τακτικών που χρησιμοποιούνται ευρέως στις αυτοποιημένες διαπραγματεύσεις είναι αυτές που εξαρτώνται από το διαθέσιμο χρόνο, τους διαθέσιμους πόρους και τη συμπεριφορά του αντιπάλου. Οι δύο πρώτες κατηγορίες μοντελοποιούνται με πολυωνυμικές και εκθετικές συναρτήσεις, ενώ η τρίτη εκφράζει μιμητική συμπεριφορά των αποκρίσεων του αντιπάλου.

Η αβεβαιότητα που διέπει το περιβάλλον διαπραγμάτευσης έχει οδηγήσει στην ανάγκη υιοθέτησης «έξυπνων» στρατηγικών, που ενσωματώνουν τεχνικές μηχανικής μάθησης. Ο βασικός στόχος των διαπραγματευτών είναι να κάνουν αποτελεσματικότερες επιλογές στη φάση σχεδιασμού, να επιλέξουν τους κατάλληλους αντιπάλους, να διαμορφώσουν τη στρατηγική τους ώστε να μεγιστοποιηθεί η ατομική ή η κοινή ωφέλεια, και τέλος να ανιχνεύσουν και να διακόψουν σε πρώιμο στάδιο ατελέσφορες διαπραγματεύσεις. Το τέταρτο κεφάλαιο της διατριβής αφιερώθηκε στην καταγραφή και κατηγοριοποίηση των «έξυπνων» στρατηγικών ανάλογα με το στάδιο εφαρμογής και το είδος μηχανισμού μάθησης που χρησιμοποιείται. Οι τρεις βασικές κατηγορίες είναι οι στρατηγικές εξερεύνησης, οι στρατηγικές επανάληψης και οι στρατηγικές πρόβλεψης.

Στις στρατηγικές εξερεύνησης οι διαπραγματευτές δοκιμάζουν νέες λύσεις, επιλέγουν στρατηγικές που δεν έχουν ξαναχρησιμοποιήσει προκειμένου να επιτύχουν το επιθυμητό αποτέλεσμα. Οι συνήθεις τεχνικές που εφαρμόζονται είναι γενετικοί αλγόριθμοι και Q-Learning. Οι γενετικοί αλγόριθμοι, όταν χρησιμοποιούνται στη φάση σχεδιασμού, επιτρέπουν τη μελέτη της εξέλιξης των στρατηγικών στους διαφορετικούς πληθυσμούς. Από την άλλη, όταν χρησιμοποιούνται κατά τη διάρκεια της διαπραγμάτευσης, επιτρέπουν τη μάθηση των προτιμήσεων του αντιπάλου και την κατάλληλη προσαρμογή της εκάστοτε προσφοράς. Εφαρμογή της τεχνικής Q-Learning μπορεί να οδηγήσει σε αύξηση της ατομικής ωφέλειας των διαπραγματευτών. Στην περίπτωση που το περιβάλλον είναι στατικό, είναι δυνατή η εύρεση της βέλτιστης στρατηγικής μετά από έναν αριθμό διαπραγματεύσεων. Σημαντικό μειονέκτημα και των δύο μεθόδων είναι ότι προϋποθέτουν έναν αρκετά μεγάλο αριθμό επαναλήψεων μέχρι να συγκλίνουν. Επιπλέον η μέθοδος Q-learning προϋποθέτει την αξιολόγηση των κινήσεων του διαπραγματευτή από τον αντίπαλό του, γεγονός που δεν είναι ρεαλιστικό σε κάθε περίπτωση.

Οι στρατηγικές επανάληψης βασίζονται σε τεχνικές επαναχρησιμοποίησης γνώσης, όπου είναι δυνατή η δημιουργία ρουτινών. Η περισσότερο διαδεδομένη τεχνική σε αυτή την κατηγορία είναι η μέθοδος Case-based Reasoning. Η ειδικότερη γνώση που αποκτάται εφαρμόζεται σε παρόμοιες περιπτώσεις με σκοπό την επίτευξη αντίστοιχα καλών αποτελεσμάτων. Κατ'αυτόν τον τρόπο είναι δυνατό να δημιουργηθούν βέλτιστες πρακτικές. Μειονέκτημα της μεθόδου αποτελεί ο κίνδυνος εφαρμογής μη αποτελεσματικών ενεργειών αν το περιβάλλον διαπραγμάτευσης είναι δυναμικό.

Τέλος, στην τρίτη κατηγορία εντάσσονται οι στρατηγικές πρόβλεψης, όπου οι διαπραγματευτές προσαρμόζουν τις ενέργειες και τις προσφορές τους βασιζόμενοι σε εκτιμήσεις εξωτερικών παραγόντων, που αφορούν είτε το περιβάλλον, είτε το στρατηγικό μοντέλο του αντιπάλου τους. Οι τεχνικές μάθησης που χρησιμοποιούνται σε αυτήν την κατηγορία είναι possibilistic case-based reasoning, Bayesian learning, μη γραμμική παλινδρόμιση και νευρωνικά δίκτυα. Οι δύο πρώτες μέθοδοι προϋποθέτουν τη γνώση πολλών πιθανοτήτων, ενώ η μη γραμμική παλινδρόμιση προϋποθέτει τη γνώση της μορφής της συνάρτησης που αποτελεί τη στρατηγική του αντιπάλου.

Η παρούσα διατριβή επικεντρώνεται στην τρίτη κατηγορία και πιο συγκεκριμένα στην προσαρμογή της στρατηγικής κατά τη διάρκεια της ανταλλαγής προσφορών, με σκοπό την αύξηση της ατομικής ωφέλειας. Μελετώνται περιπτώσεις όπου η προσφορά που παράγει ο διαπραγματευτής σε κάθε γύρο βασίζεται στην εκτίμηση της επόμενης προσφοράς του αντιπάλου. Παρουσιάζονται δύο χαρακτηριστικά παραδείγματα: το σύστημα Smart-agent [8] και ο πράκτορας λογισμικού διαπραγμάτευσης Negotiator [9]. Στο σύστημα Smart-agent, ο διαπραγματευτής συγκρίνει σε κάθε γύρο την προσφορά που θα έστελνε βάση της προκαθορισμένης στρατηγικής του, με αυτή που προβλέπει ότι θα στείλει ο αντίπαλός του στον επόμενο γύρο. Αν η τελευταία είναι συμφερότερη, τότε η νέα προσφορά διαμορφώνεται σύμφωνα με την εκτίμηση, όπως ορίζεται από τον κανόνα (eq. 3). Η συμπεριφορά αυτή ευνοεί την ανάπτυξη πρακτόρων λογισμικού που υπερνικούν αντιπάλους που δε διαθέτουν μηχανισμούς μάθησης (Σχήμα 13).

Στο σύστημα Negotiator ο διαπραγματευτής κάνει χρήση του μηχανισμού πρόβλεψης στον προτελευταίο γύρο. Συγκεκριμένα, κάνει το μέγιστο δυνατό συμβιβασμό αποστέλλοντας την τιμή ορίου του αν η πρόβλεψη της προσφοράς του αντιπάλου είναι λιγότερο συμφέρουσα, διαφορετικά στέλνει την ίδια τιμή με την πρόβλεψη. Κατ'αυτόν τον τρόπο επιτυγχάνει αύξηση της ατομικής του ωφέλειας.

Οι δύο κανόνες που περιγράφονται στο [8] και το [9] εκφράζουν δύο ακραίες συμπεριφορές απέναντι στον κίνδυνο. Στο μεν σύστημα Smart-agent ο διαπραγματευτής επιτυγχάνει θεαματική αύξηση της ατομικής του ωφέλειας, παρατείνοντας όμως σημαντικά το χρόνο διαπραγμάτευσης. Ο αντίπαλός του τείνει να ανταποκριθεί με αποστολή αντιπροσφοράς, γεγονός που αυξάνει την πιθανότητα αποχώρησής του και τερματισμού της διαπραγμάτευσης χωρίς συμφωνία. Η συμπεριφορά αυτή εκφράζει ροπή προς τον κίδυνο (risk-seeking). Στο Negotiator, ο διαπραγματευτής επιτυγχάνει πιο περιορισμένη αύξηση της ατομικής του ωφέλειας, χωρίς ωστόσω να υπάρχει κίνδυνος αύξησης των ανεπιτυχών διαπραγματεύσεων, αφού ο αντίπαλος τείνει να ανταποκριθεί με αποδοχή προσφοράς. Υιοθετώντας τη συμπεριφορά αυτή ο διαπραγματευτής αποστρέφεται τον κίνδυνο (risk-averse).

Υπάρχει μια σειρά από μελέτες που συσχετίζει το περιβάλλον διαπραγμάτευσης με τη συμπεριφορά του διαπραγματευτή απέναντι στον κίνδυνο. Πιο συγκεριμένα παρουσιάζεται μεγαλύτερη ροπή προς τον κίνδυνο όταν η διαπραγμάτευση γίνεται για τη μείωση ζημιάς και λιγότερη όταν αφορά την αύξηση κερδών.

Ένας από τους στόχους της διατριβής είναι η δημιουργία μιας στρατηγικής πρόβλεψης που να επιτρέπει την υιοθέτηση πολλών διαφορετικών συμπεριφορών απέναντι στον κίνδυνο, και η επέκταση αυτής ώστε να υποστηρίζονται διαπραγματεύσεις πολλαπλών χαρακτηριστικών.

Στο πέμπτο κεφάλαιο γίνεται συζήτηση για τον κίνδυνο που ελοχεύουν οι στρατηγικές πρόβλεψης και παρουσιάζεται η προτεινόμενη στρατηγική.

Κατά τη φάση σχεδιασμού ο διαπραγματευτής θέτει τις προτιμήσεις και την αρχική στρατηγική του, καθώς επίσης και μια παράμετρο RP, η οποία εκφράζει το ποσοστό του χρόνου που είναι διατεθειμένος να παρατείνει τη διαπραγμάτευση. Κατά τη φάση ανταλλαγής προσφορών χρησιμοποιεί την εκτίμηση της επόμενης προσφοράς του αντιπάλου. Σε κάθε βήμα, στέλνει την προσφορά που παράγεται από την προκαθορισμένη στρατηγική, όσο αυτή είναι συμφερότερη από την πρόβλεψη (σημείο MP). Το MP σηματοδοτεί το σημείο που θα αντιστοιχούσε σε συμφωνία αν οι δύο πράκτορες δεν εφήρμοζαν τεχνικές μηχανικής μάθησης. Όταν αυτό ανιχνευθεί, και για όσο διάστημα καθορίζεται από την παράμετρο RP, υιοθετείται η στρατηγική με ροπή προς τον κίνδυνο. Όταν το RP καταναλωθεί, υιοθετείται η στρατηγική με αποστροφή του κινδύνου μέχρι τη λήξη της διαπραγμάτευσης. Ο κανόνας που εφαρμόζεται στην πρώτη περίπτωση (με ροπή προς τον κίνδυνο) βασίζεται στον κανόνα που χρησιμοποιείται στο σύστημα Smart-agent, ενώ αυτός που εφαρμόζεται στη δεύτερη περίπτωση (αποστροφή κινδύνου), βασίζεται στο Negotiator. Και οι δύο κανόνες έχουν επεκταθεί ώστε να υποστηρίζονται διαπραγματεύσεις πολλαπλών χαρακτηριστικών. Όσο μεγαλύτερη είναι η τιμή του RP, τόσο μεγαλύτερη είναι η αύξηση της ωφέλειας σε περίπτωση συμφωνίας, και ταυτόχρονα τόσο μεγαλώνει ο κίνδυνος αποχώρησης του αντιπάλου και τερματισμού της διαπραγμάτευσης.

Για την αποτίμηση της προτεινόμενης στρατηγικής, δημιουργήσαμε ένα περιβάλλον προσομοίωσης διαπραγματεύσεων. Οι βασικές κλάσεις των διαπραγματευτών είναι υλοποιημένες σε Java, και η χρήση τους επεκτείνεται με την προσαρμογή τεχνικών μηχανικής μάθησης σε κλάσεις στο matlab. Βασικός μας στόχος είναι να μετρηθεί η αύξηση της ωφέλειας, όπως επίσης και η μείωση των συμφωνιών όταν γίνεται χρήση της στρατηγικής πρόβλεψης, για τις διαφορετικές τιμές των RP. Θεωρήσαμε διαφορετικά σενάρια αναφορικά με το μέγιστο διαθέσιμο χρόνο των συμμετεχόντων, το εύρος της ζώνης συμφωνιών και των στρατηγικών που ορίζονται στη φάση σχεδιασμού, δημιουργώντας έτσι 2,352 περιβάλλοντα διαπραγμάτευσης. Για κάθε περιβάλλον διενεργήθηκε μια σειρά πειραμάτων μεταξύ του «έξυπνου» πράκτορα και ενός πράκτορα διαπραγμάτευσης που δεν ενσωματώνει μηχανισμό πρόβλεψης. Ο «έξυπνος» πράκτορας διαθέτει μοντέλο πρόβλεψης με μηδενικό σφάλμα και δοκιμάζεται για 21 διαφορετικές τιμές RP ([0:5:100]). Η τελική αποτίμηση γίνεται συγκρίνοντας τη διαφορά της ατομικής ωφέλειας του διαπραγματευτή, τη διαφορά του χρόνου διαπραγμάτευσης και τη διαφορά του αριθμού των ανεπιτυχών διαπραγματεύσεων, όταν χρησιμοποιείται η στρατηγική πρόβλεψης. Όπως φαίνεται και στο Σχήμα 18, με RP=0% η μέση απόλυτη αύξηση της ατομικής ωφέλειας είναι 0.94% και αυξάνει με την αύξηση του RP μέχρι την τιμή 12.05% για RP=100%. Αντίστοιχα αυξάνεται και ο μέσος χρόνος διαπραγμάτευσης, που για RP=0% είναι 0.96% και για RP=100% είναι 23.07%. Η αύξηση του χρόνου διαπραγμάτευσης είναι η κύρια αιτία μείωσης του αριθμού των επιτυχών διαπραγματεύσεων, καθώς αυξάνεται η πιθανότητα αποχώρησης του αντιπάλου. Όπως φαίνεται και στο Σχήμα 20 για RP=0% δεν υπάρχει μείωση του αριθμού των διαπραγματεύσεων, ενώ για RP=100% η μέση μείωση φτάνει το 20.78%. Για την εφαρμογή της προτεινόμενης στρατηγικής θα πρέπει να γίνει κατάλληλη επιλογή των RP, η οποία μπορεί να επιτευχθεί μέσω συνεκτίμησης της πιθανότητας αποχώρησης του αντιπάλου στον επόμενο γύρο, ή μέσω πρόβλεψης του μέγιστου διαθέσιμου χρόνου του αντιπάλου. Αν είναι διαθέσιμη η πρόβλεψη του μέγιστου χρονικού ορίου, ο διαπραγματευτής μπορεί να υιοθετήσει τον κανόνα Risk Seeking από τη στιγμή που ανιχνεύεται το MP μέχρι ένα βήμα πριν τη λήξη του χρονικού ορίου του αντιπάλου του, οπότε και μπορεί να υιοθετήσει τον κανόνα Risk Averse. Κατ'αυτόν τον τρόπο μπορεί να επιτευχθεί σημαντική αύξηση της ατομικής ωφέλειας και ταυτόχρονα σημαντική μείωση των ανεπιτυχών διαπραγματεύσεων. Για αποτίμηση της στρατηγικής με κατάλληλο καθορισμό των RPs, επαναλάβαμε τα πειράματα και καταγράψαμε μέση απόλυτη αύξηση της ατομικής ωφέλειας 12.017%, (κοντά στην ποσοστό που είχαμε για RP=100%) και μέση μείωση των ανεπιτυχών διαπραγματεύσεων 0.61% (κοντά στο ποσοστό που είχαμε για RP=0%).

Στα πειράματα που διεξήχθησαν, θεωρήθηκαν πράκτορες λογισμικού που διαθέτουν μοντέλο πρόβλεψης με μηδενικό σφάλμα. Κατά τη διάρκεια της διαπραγμάτευσης οι τιμές που συλλέγονται σε διακριτά χρονικά διαστήματα προκύπτουν από την παρατήρηση των προηγούμενων προσφορών του αντιπάλου και του «έξυπνου» πράκτορα. Βασικός στόχος είναι η προσέγγιση μιας άγνωστης συνάρτησης, με χρήση ενός συνόλου τιμών της μορφής (x,y) όπου x είναι ο γύρος διαπραγμάτευσης στον οποίο στοιχειοθετείται και προτείνεται μια τιμή προσφοράς y. Με τον τρόπο αυτό σχηματίζεται ένα ιστορικό τιμών που αποθηκεύεται από τον πράκτορα που εφαρμόζει την απαραίτητη ευφυΐα για την πρόβλεψη της τιμής του αντιπάλου.

Στο έκτο και το έβδομο κεφάλαιο, γίνεται συζήτηση για τα μοντέλα πρόβλεψης που έχουν χρησιμοποιηθεί για την εκτίμηση της επόμενης προσφοράς. Μεταξύ αυτών είναι τα μοντέλα μη γραμμικής παλινδρόμισης, τα νευρωνικά δίκτυα, και οι προσεγγιστές με τη βοήθεια πολυωνύμων όπως η μέθοδος ελαχίστων τετραγώνων και η προσέγγιση με κυβικές splines. Τα μοντέλα μη γραμμικής παλινδρόμισης έχουν εφαρμοστεί με επιτυχία, παρόλα αυτά, επειδή προϋποθέτουν γνώση της μορφής της συνάρτησης, έχουν πιο περιορισμένο εύρος εφαρμογής. Οι προσεγγιστές πολυωνύμων συγκριτικά με τα νευρωνικά δίκτυα εμφανίζονται λιγότερο ακριβείς. Στατιστικά μοντέλα χρονοσειρών, που έχουν εφαρμοστεί για την πρόβλεψη οικονομικών δεδομένων, δεν έχουν χρησιμοποιηθεί για το πρόβλημα της εκτίμησης της επόμενης προσφοράς του αντιπάλου, καθώς απαιτείται μια σειρά από ελέγχους για να εξασφαλιστεί η ορθότητα της εφαρμογής τους. Έτσι καταλήγουμε στην επιλογή των νευρωνικών δικτύων, τα οποία μπορούν να χρησιμοποιηθούν χωρίς να απαιτείται γνώση της μορφής της συνάρτησης των δεδομένων.

Τα περισσότερο διαδεδομένα νευρωνικά δίκτυα που έχουν χρησιμοποιηθεί για το εν λόγω πρόβλημα είναι τα Multi-Layer Perceptrons (MLPs) και τα Radial Basis Function Networks (RBFNs). Επιλέγεται η χρήση των MLPs καθώς παρουσιάζουν καλύτερη δυνατότητα γενίκευσης και είναι μικρότερα από τα RBFNs. Τα MLPs που χρησιμοποιούνται για την πρόβλεψη της επόμενης προσφοράς αποτελούνται από έναν κόμβο κρυφού επιπέδου με νευρώνες που έχουν σιγμοειδή συνάρτηση ενεργοποίησης, και ένα κόμβο εξόδου με νευρώνες που έχουν γραμμική συνάρτηση ενεργοποίησης.

Στα υπάρχοντα συστήματα τα MLPs εκπαιδεύονται μία φορά στη φάση σχεδιασμού, είτε με συνθετικά δεδομένα που προκύπτουν από προσομοιώσεις διαπραγματεύσεων, είτε με δεδομένα που εξάγονται από προηγούμενες διαπραγματεύσεις. Τα εκπαιδευμένα δίκτυα χρησιμοποιούνται στη συνέχεια σε κάθε διαπραγματευτικό γύρο. Είσοδό τους αποτελούν οι προηγούμενες προσφορές των διαπραγματευτών, ενώ η έξοδος αποτελεί την πρόβλεψη της επόμενης τιμής που θα στείλει ο αντίπαλος.

Το πρόβλημα που πραγματεύεται η παρούσα διατριβή αφορά τον τρόπο χρήσης των νευρωνικών δικτύων MLPs στα υπάρχοντα συστήματα. Το βασικό μειονέκτημα είναι ότι η ακρίβεια της πρόβλεψης είναι άμεση εξαρτώμενη από τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση. Έτσι σε δυναμικά περιβάλλοντα, όταν αλλάζουν οι κατανομές των δεδομένων στις οποίες βασίστηκαν τα αρχικά μοντέλα, οι προβλέψεις είναι λιγότερο έγκυρες. Επιπλέον παρατηρείται ανομοιογένεια της αρχιτεκτονικής, αλλά και του αριθμού των προηγούμενων προσφορών που στοιχειοθετούν την είσοδο των MLPs.

Για την αντιμετώπιση των ανωτέρω προβλημάτων ερευνάται η χρήση μοντέλων που εκπαιδεύονται με δεδομένα που εξάγονται από την τρέχουσα διαπραγμάτευση (Session-long Learning Agents). Βασική αρχή των μοντέλων αυτών αποτελεί η επανεκπαίδευσή τους σε κάθε διαπραγματευτικό γύρο. Στην παρούσα διατριβή επιλέγεται η χρήση δικτύων μικρού μεγέθους, έτσι ώστε να επιτύχουμε μικρότερο αποθηκευτικό και υπολογιστικό κόστος [**Σημ.**: η εκπαίδευση γίνεται με τη μέθοδο Levenberg and Marquardt], αλλά και χαμηλότερο σφάλμα πρόβλεψης που είναι ανάλογο του λόγου των ελεύθερων παραμέτρων προς το πλήθος των παρατηρήσεων.

Τα μοντέλα Session-long Learning Agents που αναπτύχθηκαν στα πλαίσια της διατριβής και περιγράφονται στο όγδοο και ένατο κεφάλαιο, είναι είτε στατικά (Static Session-long Learning Agents, SSLAs) είτε δυναμικά με δυνατότητα προσαρμογής (Adaptive Session-long Learning Agents, ASLAs). Τα SSLAs χρησιμοποιούν ένα μικρό δίκτυο σταθερής αρχιτεκτονικής, το οποίο επανεκπαιδεύεται σε κάθε γύρο, δημιουργώντας κάθε φορά νέο εκπαιδευτικό σύνολο από τις διαδοχικές προσφορές του αντιπάλου. Το δίκτυο αυτό χρησιμοποιείται σε κάθε βήμα από τον πράκτορα που υιοθετεί την προτεινόμενη στρατηγική.

Προκειμένου να δειχτεί το πρόβλημα των υπαρχόντων συστημάτων εκτελέστηκαν πειράματα όπου συγκρίθηκε ο διαπραγματευτής που χρησιμοποιεί ένα MLP το οποίο εκπαιδεύεται μία μόνο φορά κατά τη φάση σχεδιασμού (Pre-Trained Agent, PTA), με τον SSLA. Αναπτύχθηκαν 3 διαφορετικοί PTAs βασιζόμενοι σε 3 MLPs που προέκυψαν από 3 διαφορετικές περιοχές διαπραγμάτευσης (negotiation domains) και συγκρίθηκαν με 3 SSLAs με MLPs που διαθέτουν 3 κόμβους εισόδου (τις 3 προηγούμενες προσφορές του αντιπάλου) και 2 κόμβους κρυφού επιπέδου. Για τη σύγκριση διεξήχθησαν διαπραγματεύσεις μεταξύ PTAs και πρακτόρων που δε διαθέτουν υπολογιστική ευφυΐα και μεταξύ SSLAs και πρακτόρων που δε διαθέτουν υπολογιστική ευφυΐα. Τα περιβάλλοντα διαπραγμάτευσης που χρησιμοποιήθηκαν για τη σύγκριση ήταν διαφορετικά από αυτά που χρησιμοποιήθηκαν για τη δημιουργία των PTAs. Στην περίπτωση χρήσης στατικών μοντέλων πρόβλεψης που χρησιμοποιούν δεδομένα της τρέχουσας διαπραγμάτευσης, αποδεικνύεται ότι η μείωση του σφάλματος πρόβλεψης ανέρχεται στο 92.67% σε σχέση με αυτό που προκύπτει από μοντέλα που βασίζονται σε συνθετικά δεδομένα, ή σε δεδομένα προηγούμενων διαπραγματεύσεων.

Στην περίπτωση χρήσης δυναμικών μοντέλων, οι ASLAs αποτελούν την εξέλιξη των SSLAs, αφού βελτιστοποιούν την αρχιτεκτονική και τις παραμέτρους εισόδου των στατικών MLPs. Κατ’αυτόν τον τρόπο αντιμετωπίζεται το θέμα της ανομοιογένειας. Πιο συγκεκριμένα τα νευρωνικά δίκτυα εξελίσσουν τη δομή τους σε κάθε βήμα πρόβλεψης με χρήση γενετικού αλγορίθμου. Το χρωμόσωμα που επιλέγεται ενσωματώνει στις παραμέτρους εισόδου τις προηγούμενες προσφορές του αντιπάλου, αλλά και του διαπραγματευτή. Επιπλέον ενσωματώνει τον αριθμό των νευρώνων κρυφού επιπέδου. Χρησιμοποιείται δυαδική γραμματική, έτσι στο χρωμόσωμα των 9 bit αναζητείται το καλύτερο MLP, επιλέγοντας από 0 έως 7 προηγούμενες προσφορές του αντιπάλου, από 0 έως 7 προηγούμενες προσφορές του διαπραγματευτή και από 2 έως 7 νευρώνες κρυφού επιπέδου. Σε κάθε βήμα της διαπραγμάτευσης ακολουθείται ο παρακάτω γενετικός αλγόριθμος. Αρχικά δημιουργείται ένα τυχαίος πληθυσμός λύσεων, όπου κάθε λύση αποκωδικοποιείται στην αντίστοιχη αρχιτεκτονική MLP. Το ιστορικό των προηγούμενων προσφορών χρησιμοποιείται για τη δημιουργία του εκπαιδευτικού συνόλου στο οποίο βασίζεται η εκπαίδευση των δικτύων. Η αξιολόγηση των διαφορετικών MLPs γίνεται με χρήση αντικειμενικής συνάρτησης ανάλογα με το Mean Squared Error (MSE). Ευνοούνται λύσεις όπου είναι εφικτός ο διαχωρισμός σε training, validation και test set, όπως και λύσεις που ο λόγος των ελεύθερων παραμέτρων του δικτύου προς τον αριθμό των παρατηρήσεων είναι μικρός. Από την αξιολόγηση επιλέγονται οι καλύτερες λύσεις και εφαρμόζονται οι τελεστές crossover και mutation για τη δημιουργία του νέου πληθυσμού. Η διαδικασία επαναλαμβάνεται για 10 γενιές, οπότε και επιλέγεται το βέλτιστο MLP, το οποίο και χρησιμοποιείται για την εκτίμηση της επόμενης προσφοράς του αντιπάλου.

Στο ένατο κεφάλαιο διεξάγεται μια σειρά πειραμάτων για να συγκριθούν οι SSLAs και ASLAs. Λήφθηκαν υπόψη διαφορετικά σενάρια διαπραγμάτευσης αναφορικά με το εύρος της ζώνης συμφωνιών, τα μέγιστα χρονικά περιθώρια και τις στρατηγικές προτιμήσεις των συμμετεχόντων και δημιουργήθηκαν 1,359 διαφορετικά περιβάλλοντα. Σε κάθε περιβάλλον διεξήχθησαν διαπραγματεύσεις μεταξύ SSLAs και πρακτόρων διαπραγμάτευσης που δε διαθέτουν υπολογιστική ευφυΐα, και μεταξύ ASLAs και πρακτόρων που δε διαθέτουν υπολογιστική ευφυΐα. Για τους Session-long learning agents, σε κάθε γύρο υπολογίστηκε το απόλυτο σφάλμα πρόβλεψης και στο τέλος της διαπραγμάτευσης υπολογίστηκαν ο μέσος όρος, η μέγιστη τιμή και η τυπική απόκλιση του απόλυτου σφάλματος. Αποδεικνύεται ότι οι ASLAs παρέχουν πιο σταθερές προβλέψεις μεγαλύτερης ακρίβειας, αφού η μείωση του μέσου σφάλματος ανέρχεται στο 38.34%, η μείωση της μέσης τυπικής απόκλισης ανέρχεται στο 38.03% και η μέση μέγιστη τιμή είναι μειωμένη κατά 44.75% σε σχέση με τα στατικά μοντέλα.

Στο δέκατο κεφάλαιο της διατριβής, ερευνάται η δυνατότητα επέκτασης των μοντέλων πρόβλεψης σε διαπραγματεύσεις πολλαπλών χαρακτηριστικών. Παρουσιάζονται δύο δυνατότητες. Στην πρώτη ο Session-long Learning Agent χρησιμοποιεί ένα MLP για κάθε χαρακτηριστικό, δηλαδή για διαπραγματεύσεις n χαρακτηριστικών διαθέτει n MLPs με ένα κόμβο εξόδου. Στη δεύτερη περίπτωση χρησιμοποιεί ένα MLP για την πρόβλεψη του διανύσματος προσφοράς, δηλαδή για διαπραγματεύσεις n χαρακτηριστικών χρησιμοποιείται ένα δίκτυο με n κόμβους εξόδου. Η βέλτιστη αρχιτεκτονική στην περίπτωση των SSLAs αναζητήθηκε εμπειρικά. Στην πρώτη περίπτωση μικρότερο σφάλμα και τυπική απόκλιση σημειώθηκε στην περίπτωση των 5 κόμβων εισόδου και 4 κόμβων κρυφού επιπέδου, που οδήγησε σε αύξηση της ατομικής ωφέλειας κατά 10.78%. Στη δεύτερη περίπτωση χαμηλότερο σφάλμα σημειώθηκε όταν επιλέχθηκαν 8 κόμβοι εισόδου και 5 κόμβοι κρυφού επιπέδου, οπότε και η αύξηση της ατομικής ωφέλειας έφτασε το 10.5%.

Καθώς η εφαρμογή γενετικών αλγορίθμων έχει μεγαλύτερο χρονικό και υπολογιστικό κόστος για τους διαπραγματευτές, μελετήθηκε επίσης η χρήση της απλής αυτοεξελισσόμενης δομής eMLP που απαιτεί ένα μόνο πέρασμα των δεδομένων, καθιστώντας ταχύτατη τη διαδικασία μάθησης. Τα eMLPs αποτελούνται από 3 επίπεδα: input, evolving και output . Κάθε κόμβος του evolving layer, πραγματοποιεί αντιστοίχιση ενός υποχώρου της εισόδου, σ'έναν υποχώρο της εξόδου, κι έτσι η μάθηση γίνεται τοπικά σε κάθε κόμβο. Στο παράρτημα παρουσιάζονται εκτενή πειραματικά αποτελέσματα που προκύπτουν από προσομοιώσεις σε διαφορετικά περιβάλλοντα διαπραγμάτευσης. Αποδεικνύεται ότι τα eMLPs είναι λιγότερο ακριβή από τα MLPs. Παρόλα αυτά αξίζει να σημειωθεί ότι παρουσιάζουν μεγαλύτερη σταθερότητα, αφού σημειώνεται πολύ χαμηλότερη μέση τυπική απόκλιση. Τελικά με χρήση των eMLPs ο «έξυπνος» διαπραγματευτής επιτυγχάνει μέση αύξηση της ατομικής του ωφέλειας κατά 5.327%.

Στο ενδέκατο κεφάλαιο γίνεται συνοπτική παρουσίαση των συμπερασμάτων και της συνεισφοράς, και γίνεται λόγος για την επάρκεια των λύσεων που προτείνονται, αφού οδηγούν στα προσδοκώμενα αποτελέσματα.

Παρατίθενται, τέλος στο δωδέκατο κεφάλαιο, κατευθύνσεις-προτάσεις για συνέχιση και επέκταση της έρευνας. Μια κατεύθυνση αφορά τη διεύρυνση του πεδίου εφαρμογής. Θα πρέπει να διερευνηθούν σενάρια όπου ο αντίπαλος αναπαράγει τη συμπεριφορά του διαπραγματευτή που χρησιμοποιεί το μοντέλο πρόβλεψης στην στρατηγική του, καθώς σε αυτές τις περιπτώσεις η εφαρμογή της στρατηγικής που ρέπει προς τον κίνδυνο δεν έχει το ίδιο ποσοστό επιτυχίας. Μια πρόταση είναι η συνεκτίμηση του ποσοστού που ο αντίπαλος υιοθετεί στρατηγική εξαρτώμενη από τη συμπεριφορά, έτσι ώστε να αποφασιστεί κατά πόσο η στρατηγική πρόβλεψης μπορεί να οδηγήσει σε αύξηση της ατομικής ωφέλειας. Ένα ακόμη ζήτημα για μελλοντική έρευνα αφορά την εφαρμογή στρατηγικών πρόβλεψης σε περιβάλλοντα συνεργασίας όπου στόχος είναι η αύξηση της κοινής ωφέλειας.

Ολοκληρώνοντας, ένας μελλοντικός ερευνητικός στόχος που μπορεί να έχει ως απαρχή υλοποίησης του αυτή τη διατριβή, είναι η διερεύνηση και άλλων μοντέλων πρόβλεψης που χρησιμοποιούν δεδομένα που εξάγονται από την τρέχουσα διαπραγμάτευση. Προτείνεται η χρήση συστημάτων που εξελίσσουν τη δομή τους στο χρόνο, καθώς οι κατανομές των δεδομένων μεταβάλλονται. Τέτοια παραδείγματα αποτελούν τα Evolving Fuzzy Neural Networks (EFuNNs) και DENFIS που ανήκουν στην κατηγορία των Evolving Connectionist Systems (ECoS).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

There is no shortage of disputes. They appear between siblings, friends, firms, nations and are often settled through traditions, regulations, courts, markets, and negotiations. This thesis is concerned with situations where two parties recognize the differences of interest and values that exist among them and seek a compromise agreement through negotiation. There is an art and a science of negotiation. “Science” refers to the systematic analysis for problem solving and “art” refers to the interpersonal skills and the wisdom to know when and how to use them. My aims here are to examine how these skills are developed when negotiation arenas are transferred to electronic settings. For this research I needed to delve deeper into the field of machine learning and artificial intelligence and investigate how learning models can add value to the field of negotiations.

I would like to thank my advisor, Assistant Professor of the National and Kapodistrian University of Athens, Drakoulis Martakos, and expert in the field of Information Systems, for his encouragement and for providing me the opportunity to conduct this research at the department of Informatics and Telecommunications. I am also grateful to the third member of the Advisory Comitee, Professor Panagiotis Georgiadis, who is also head of the e-Governement Laboratory.

My special thanks go to the second member of the Advisory Comitee, Emeritus Professor Constantinos Halatsis, for his encouragement, guidance and support during this thesis, as well as to researcher Stavros Perantonis, who has high expertise in the field of machine learning.

Finally, I would like to thank all my family members, especially my parents and my husband, for their support and encouragement. I am deeply grateful to all of them.

1. INTRODUCTION

Electronic Marketplaces (E-markets), is an important component of e-business that brings demand and supply of commodities and services into balance. These arenas are the meeting places of producers and consumers that use exchange mechanisms, varying from catalogues where requests and offers are posted, to negotiations where participants bargain over the conditions of the exchange, to auctions where multiple participants compete against each other [1]. The term e-market is used in a broad sense and incorporates the various types and configurations of markets, stores, agoras and other meeting places where transactions take place.

The exchange mechanisms that are embedded in e-markets are models and procedures which control access to and regulate execution of the transactions. While the commonly used catalogue-based exchanges provide one example of an institution, of greater interest are mechanisms which permit richer dynamics and more complex behavior on the part of participants, e.g. negotiations and auctions. This thesis focuses on the negotiation exchange mechanism.

1.1 Negotiation: A brief review of the research field

There is a grand variety of problems drawn from everyday life where negotiations are evident. A typical list would contain economic transactions, distribution of services, management of business processes, labor negotiations, political and juridical disputes etc. Such scenarios do not always relate to conflict as might be assumed. Negotiation is the key decision-making approach that is used to reach consensus whenever a person, organization or another entity cannot achieve its goals unilaterally. It can be defined as an iterative communication and distributed decision-making process, where participants are searching for an agreement. Negotiation is thus a mechanism that can be used for allocating and sharing resources. The term ‘resource’ is used in the broadest possible sense and may involve commodities, services, time, money etc. Yet it is not guaranteed that an agreement always exists or that it will be established.

During the last decades, scientists belonging in various scientific areas such as anthropology, psychology and sociology, law, political science, economics, mathematics and computer science have made efforts to model and study negotiation interactions. These efforts have resulted to different methodologies, architectures and approaches.

Among many significant contributions is the transfer of negotiation encounters in electronic settings. Electronic platforms have been designed to facilitate the conduct of negotiations. Furthermore, computer scientists have contributed to the development of software components that either assist negotiators in various stages of the negotiation process (Negotiation Support Systems, NSSs), or in some cases are capable of undertaking stages or even the whole negotiation process. This thesis focuses on the latter category, where negotiation software agents (NSAs) are used for the representation of market stakeholders.

An agent can be viewed as an encapsulated computer system that is situated in an environment and is capable of flexible, autonomous action in order to meet its design objectives. Negotiation Software Agents are good representatives of human negotiators.

The specific rules of the interaction, which constitute the negotiation protocol, are predefined in negotiations between autonomous agents. Based on the negotiation protocol, agents need to plan their specific actions, their strategy, in order to meet their objectives. The action planning is usually not disclosed to the other participants and takes place before the actual conduct of negotiation (at a pre-negotiation phase). Yet, it is possible that during the interaction an agent reassesses the negotiation problem and adapts his strategy to the responses of his counterpart. State of the art negotiating agents use learning techniques in order to increase their profits or fulfill their objectives. Learning techniques usually assist agents to select optimal or suboptimal strategies and better model their counterparts.

1.2 Problem statement and contribution of the thesis

In order to facilitate comprehension of the domain, in this thesis we provide a categorization of strategies that are enhanced with learning techniques and are used by state of the art automated negotiators. In this respect, we devise agents to those who use explorative, repetitive and predictive strategies, either at the planning phase or during discourse. Explorative strategies imply the search for new solutions and are based on trial and error learning processes, such as Q-learning and Genetic algorithms, until some convergence criteria are met. Repetitive strategies are based on knowledge reuse, and specific knowledge is acquired by repeated execution of actions. Case-based reasoning is one such technique. Finally, in predictive strategies learning is introduced in the form of predictive decision making, where estimations of factors that influence strategy selection or update serve as input to the agents' decision making.

In this thesis we focus on the third category and particularly on agents who update their strategy based on estimations of their counterpart's future responses. Such a technique has proved valuable, as in most cases negotiators manage to increase their profits compared to the non-learning case. However, in some situations agents tend to prolong the negotiation discourse and this increases the risk that negotiation breaks off, as the counterpart may decide to terminate the process. A first issue we investigate is how such strategies affect the establishment of negotiating agreements. In this vein, we propose a negotiation strategy that introduces a risk related parameter, mainly linked to the prolongation of the negotiation discourse. This parameter allows the specification of different attitudes towards risk, where risk measures an agent's willingness to stay in negotiation in order to use the predictive mechanism more extensively and heighten his gains. For example a negotiator with a risk-seeking behavior would decide to exhaust negotiation time in order to increase his profits, while a risk-averse agent would act more conservatively, and would not risk prolonging negotiation time.

Another issue that is studied in this thesis is the type of learning mechanism that is used by predictive negotiators who estimate their counterpart's future offers. When it comes to forecasting the partners' future offers, techniques can be summarized into those based on statistical approaches (particularly non-linear regression) [2] [3], mathematical models based on differences [4] [5], and connectionist approaches, particularly some special types of neural networks, Multi Layer Perceptrons (MLPs) and Radial Basis Function Networks (RBFN) [6] [7] [8] [9] [10] [11] [12]. From the above methods we argue that neural networks are best applicable for the purpose of forecasting the counterpart's future offers. Experiments have shown that mathematical models give poorer results when compared to non-linear regression models [3]. Non-linear regression models are more restrictive than artificial neural networks, since they require specific assumptions regarding the strategy of the other party. On the other hand neural networks are applicable in the general case, without assuming implicit knowledge of the function that maps input to output data. This is particularly desirable for negotiation forecasting situations where data relations are not known.

In current research approaches neural networks are trained at a pre-negotiation phase with data extracted from past negotiations, and are used in the current discourse to provide estimations of the counterpart's future offers. However, the accuracy of the forecasting tool depends heavily on data acquired from previous interactions. We investigate how the forecasting accuracy is affected when data distributions change, and propose building and training neural networks with data extracted from the current interaction. We term agents that exploit data from the actual discourse session-long learning, and prove that a small neural network with few training examples is capable of capturing the negotiation dynamics. In this thesis we introduce two types of session-long learning agents: Static session-long learning agents (SSLAs), who use a neural network with a static structure during the negotiation process, and adaptive session-long learning agents (ASLAs), who use a neural network which evolves its structure and input features based on a genetic algorithm. We also study the use of another adaptive structure eMLP, which is a simple evolving connectionist structure that engages in one-pass, lifelong learning. From the experiments conducted it is empirically proved that ASLAs provide the most promising results (forecasts yield the smallest error), however the combination of neural networks with genetic algorithms require a lot of time and resources which is sometimes restricting in negotiation domains. This result makes SSLAs a good selection for the problem of forecasting the counterpart's future offers.

1.3 Organization of the thesis

This thesis is organized as follows. In the second Chapter we provide some foundations related to the negotiation domain, terminologies and classifications, as well as to the research methodologies and approaches. We also present a number of software platforms, systems and agents that have been developed to support the various stages of the negotiation process and describe an example domain concerning electricity distribution. In the third Chapter attention is focused on the description of the negotiation protocol and particular families of non-learning strategies that are often employed by automated negotiation agents. The fourth Chapter concludes our review of the negotiation field by apposition of agent models that are enhanced with learning techniques to increase their individual gain. In this respect classification to explorative, repetitive and predictive strategies is illustrated, and virtues and weaknesses of the developed models are presented. Special attention is given to the class of predictive strategies, and particularly to those that make use of the estimation of the counterpart's future offers. In the fifth Chapter we discuss how such strategies, often related with prolongation of the negotiation process, pose the risk of negotiation breakdowns, and propose a strategy that incorporates a risk-related parameter, enabling the adoption of different attitudes towards risk. We also discuss how this parameter can be appropriately set to avoid negotiation breakdowns. The remainder of this thesis contemplates the issue of learning models applied by the predictive agents. The sixth Chapter provides a brief overview and comparison of the forecasting tools employed by negotiators, and bibliographical research reveals the superiority of artificial neural networks (particularly MLPs), which are more extensively discussed in the seventh Chapter. In the eighth Chapter we identify two issues that require further examination. The first concerns application of the learning tools in order to capture the dynamics of changing negotiation environments, and the second concerns optimization of the architecture of the employed tools. To address the first issue, we argue that it is crucial to retrain the learning tools during the negotiation discourse and we introduce Static Session-long Learning Agents (SSLAs). In the same chapter SSLAs are compared with current state of the art agents who train their networks only at a pre-negotiation phase (Pre-Trained Agents, PTAs). In chapter 9 we address the second issue by optimizing the architecture of the MLP with the use of a genetic algorithm. The agent that adapts the architecture of the employed learning tool and the subset of input features is termed

Adaptive Session-long Learning Agent (ASLA) and is compared to the SSLA. Other evolving learning structures, such as a simple evolving connectionist system eMLP, is also discussed and illustrated in the appendix of this thesis. Finally, in the tenth Chapter we illustrate extension of the proposed agents to support multi-issued negotiations.

2. FOUNDATIONS

2.1 Negotiation: A multidisciplinary research field

Negotiation is a multidisciplinary research field and its definition has been biased by the different views of the procedure. For this reason, numerous definitions exist in the literature revealing the different objectives that can be approached.

Gulliver [13] defines negotiation as a process in which two parties attempt to reach a joint decision on issues under dispute.

Robinson and Volkov [14] view negotiation as a process in which participants bring their goals to a bargaining table, strategically share information and search for alternatives that are mutually beneficial.

Putnam and Roloff [15] view negotiation as a special form of communication that centers on perceived incompatibilities and focuses on reaching mutually acceptable agreements.

Actually negotiations have attracted the interest of researchers from several scientific fields, including anthropology, psychology and sociology, law, political science, economics, mathematics, and computer science. Raiffa [16] identifies those perspectives that act as reference to the development of any negotiation theory. Particularly he describes the “is” and “ought” of decision making and identifies the perspectives of the “describers” and the “prescribers”.

The describers examine how people actually behave, how they think, how they rationalize their choices to themselves. The main contributors of descriptive studies are anthropologists, psychologists, sociologists and political scientists who are oriented towards studying a negotiator’s perceptions and ways of interaction in particular problem situations. The describers perform analysis to help understand the selection of a choice that has been made. They identify negotiation patterns, reasons for certain decisions and the implications of cultural differences in behavior.

The prescribers are interested in how people should or ought to behave. Their aim is to guide the perplexed decision maker in choosing an action that is consonant with the decision-makers true beliefs and values. The main contributors of prescriptive studies are game theorists -applied mathematicians and economists- who examine what rational, all-knowing, super people should do in competitive, interactive situations. They develop normative models and perform analysis to help in the selection of a choice to be made.

Studies in management science also have a prescriptive orientation with the development of models designed to identify the “goodness” of the procedures. These are based largely on multi-attribute utility theory, optimization, and multiple criteria decision making theories.

In the last decade the contribution of computer science is also very significant as it advances the theoretical development of negotiation and examines its applied nature with the construction of negotiation tables, decision and negotiation support systems, software agents and software platforms [17]. The use of AI-based techniques to support various stages of the negotiation process also advances negotiation theory.

Figure 1, as depicted in [18], illustrates the different views of negotiation, the contribution of the various scientific fields and the interdependencies between negotiation models and procedures.

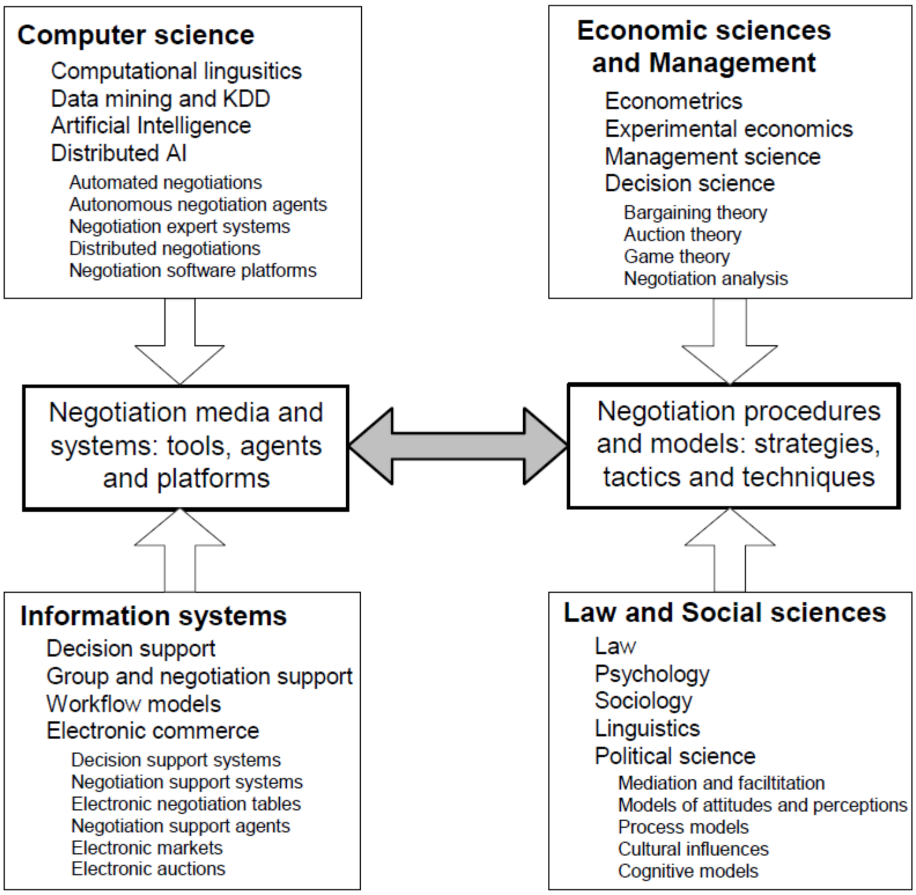


Figure 1: The negotiation landscape [18]

The variety of disciplines and perspectives has created different terminologies and concepts, resulting to inconsistencies and contradictions [13] [14] [15] [16] [17] [18]. For example in the economic literature, the term negotiation is often used synonymously with bargaining. In behavioral studies negotiation is viewed as a social interaction involving the distribution of power, resources and commitments. It is evident that negotiations require an interdisciplinary approach because of their psychological, social and cultural character; economic, political and legal considerations; quantitative and qualitative aspects; strategic, tactical and managerial perspectives.

2.2 Terminology and classification of the negotiation domain

This paragraph introduces some essential terminology and identifies the parameters that are used to classify the different negotiation domains.

Negotiations involve establishment of agreements characterized by a series of attributes (issues or features). Such attributes specify the *negotiation agenda* and may represent tangible characteristics of the commodities (or *objects*) being negotiated, or non tangible characteristics such as contract terms. For each attribute, negotiators specify a range of permissible (*reservation*) values, a minimum and a maximum, which they are not willing to exceed. Participants usually know where to stop; they specify a plan to achieve their goals skipping negotiation, what is termed best alternative to negotiating agreement (BATNA). Additionally participants set a *deadline* indicating the maximum time they can spend in a negotiation encounter. The place where negotiations are conducted constitutes the negotiation *arena* and the negotiation *outcome* (or *result)* can be a compromise or a failure, as agreement is not always guaranteed. The specific rules of communication constitute the negotiation *protocol, which* determines the way messages are exchanged. Based on the protocol each agent adopts a negotiation strategy which consists of the decision making rules that are used to determine, select and analyze the decision alternatives.

At the beginning of a negotiation encounter each participant has a portion of space where he is willing to make agreements. During negotiation each participant's space may expand or contract. The negotiation ends when participants find a mutually acceptable point in the negotiation space, which of course belongs in both participants' region of acceptability (*agreement zone*). Figure 2a illustrates the negotiation space of two participants, A_1 and A_2, where x stands for potential deals and o constitutes the final outcome. The shaded space represents the agreement zone, which exists and is stationary during the search of a solution. Figure 2b illustrates a change of the search space of participant A_2, in order to trace an agreement. Negotiation concerns the distributed search through the space of potential agreements, sometimes invoking the search of spaces that include potential agreements. Alternation of the bounds of the search space (expansion or contraction) is related to alternation of the negotiators' beliefs, often due to changes that take place in the environment, or to persuasive power of his counterparts.

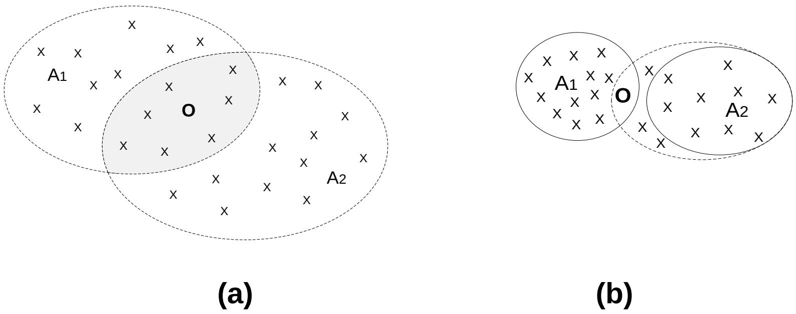


Figure 2: (a): Searching for an agreement in stationary spaces (contracts are here represented in 2D space). (b): A2 changes the space of potential agreements during the discourse

The type of interaction, the participants and the role they play, their social behavior, as well as the commodities (tangible or not) that are being discussed have been used as discriminative entities in an attempt to classify the various negotiation domains. We give a description of some commonly stated parameters and considerations of a negotiation discourse. These focal points have also been discussed by Raiffa [16].

<u>The *Negotiable Object*</u>: As mentioned earlier the negotiable object consists of at least one issue. The number of issues is often used as a discriminative parameter of different negotiation domains, as it gives rise to different negotiation behaviors. If only one issue is negotiated (*single-issued*), its value shifts along one dimension, therefore gain for one negotiator might result loss for the other. Opposing, in cases where negotiation involves multiple issues (*multi-issued*), negotiators are given the opportunity to consider the overall gain, thus adopt a more cooperative behavior. *The type of issues* is also used to discriminate different negotiation domains. It concerns the acceptable values an issue can take, and therefore the space of possible agreements. For example if at least one attribute takes values in the continuous space, the space of possible agreements is infinite. Opposing, if all attribute values are picked from a discrete set, alternatives are quantifiable. Additionally, the value of an issue may be quantitative or qualitative.

The Participating entities: Negotiation can be considered as a two-sided setting. Depending on the number of participants on each side, we can have one-to-one (or bilateral), one-to many, many-to-one or many to many negotiation encounters. In conflict situations where more than two disputants are involved, coalitions may be formed and act in concert against the remaining participants. In some cases an individual or a group may experience internal conflict, in which case we have non-monolithic participants. Another important issue that discriminates the different negotiation domains is related to third party interventions. In some cases decision making is transferred "outside" the negotiation arena, and agreement is suggested by the central decision maker. Issues like trustworthiness and truthfulness arise in such scenarios. Discrimination of a facilitator, mediator, arbitrator or rule manipulator entity depending on the role of the intervener can be found in [16].

The Negotiation Objective: Each negotiator has a subjective measure that indicates his individual satisfaction for each decision alternative, also termed utility. According to Blake and Mouton [19] who have introduced the Dual Concerns model in the mid 1960s, there are five behavioral classifications regarding the level of assertiveness and cooperativeness. Assertiveness reflects the concern to satisfy one's own interests, while cooperativeness reflects the concern to satisfy the other party's interests. These classifications are competing, collaborating, compromising, accommodating and avoiding. The different behavioral classifications also reflect different negotiation objectives. For example in a competing environment, each participant is trying to maximize his individual utility, while in a collaborating environment participants are trying to maximize the joint utility.

Affect of time and resources: In many negotiation scenarios elapsing time and resources that diminish through time play a crucial role. Negotiators in haste are usually at a disadvantage and this is because the penalties incurred in delays may be quite different for the two parties. Additionally the "value" of a resource may change as time elapses.

Knowledge: This parameter relates to the type of knowledge available to each participant, concerning the specific negotiation stance, the determination of his preferences and goals as well as the preferences and goals of his counterpart. In some scenarios participants may have full knowledge, while in others they may be ignorant or fuzzy about their own preferences let alone their counterpart or the dynamics of the environment.

Knowledge about the Environment: Knowing a negotiation domain relates to the experience a negotiator has gained on the domain, and/or to information collected from third parties. The evolving rules of the environment play also a crucial role into determining how knowledgeable one can be of a specific domain.

Knowledge of Individual preferences and goals: Participants who engage in negotiations usually know why they do so. They expect to be benefited from such a choice, thus they are capable of determining preference relations among alternatives. Knowing the alternatives that are most beneficial (individually or socially) implies knowing those that are not. Therefore participants usually know where to stop; they specify a plan to achieve their goals skipping negotiation – best alternative to negotiating agreement (BATNA) and set reservation values - bounds they are not willing to exceed, in quantitative issues.

Knowledge of Partners' behavior: Among a negotiator's considerations lies the expectation of the counterparts' behavior. Different modes of behavior are expected when discussing a point of disagreement with a business partner, from those you expect to occur between firms or countries. Raiffa [16] discriminates between cooperative antagonists, strident antagonists and fully cooperative partners.

Environments: Another discriminative parameter relates to whether the negotiation environment is static or dynamic. In a static environment repetitive encounters may be observed, e.g. repetitive behavior of the participants.

Negotiation Protocol and Strategy: A significant point of each interaction, which is also being investigated in this thesis, concerns the mechanism of the interaction. What are the exact rules of the encounter, the supported or legal actions for each participant, and how does one decide how to guide the discourse and what actions to take?

Agreements: If the negotiating parties cannot establish a mutually acceptable agreement, negotiations are broken-off. Additionally, at any point of the discourse the negotiator may decide to walk away. Negotiators usually specify a best alternative to negotiation (BATNA) and identify the point where the negotiation is no longer “meaningful”. In cases where negotiators are knowledgeable and rational, they are capable of specifying the risk associated with staying at a particular discourse. Finally in cases where agreement zones do not exist, negotiation terminates without establishing an agreement. It should also be noted that there is no way of assuring that the other side will abide by an agreement. For this reason a negotiator may request for ratification, resulting to strengthening his side and stiffening the resolve of the other party.

2.3 Negotiation process model

In order to comprehend the different phases of negotiation, we proceed with the adoption of a process model, which provides a structure for the negotiation process. The search of a solution concerns an interaction of the engaged parties as well as an arrangement of individual beliefs. Braun et al. [20] identify the lack of process models from behavioral science specific to e-negotiations, and adopt a behavioral phase model based on Gulliver’s eight phase model [13]. This model comprises of five phases and is presented in Figure 3, along with the activities the negotiators undertake in each phase.

1. Planning

- Formulate representation of negotiation problem
- Specify issues and options
- Specify objectives and preferences
- Specify constructs and reservation values
- Learn about opponents
- Decide strategy
- Select negotiation location, time and communication modes

2. Agenda Setting and Exploring the Field

- Discuss negotiation Issues and meaning
- Discuss protocol
- Discuss timing of exchanges, deadline
- Discuss priorities and constraints
- Revise strategy after discussion

3. Exchange Offers and Arguments

- Modify Strategies
- Exchange Offers

4. Reaching Agreement

- Develop joint proposals
- Identify a number of compromises

5. Concluding Negotiation

- Evaluate compromise, discuss improvements
- Discuss agreement implementation

Figure 3: The five phase negotiation model [20]

The phases depicted above are subsequently executed. However, the engaged parties may return to one of the previous phases at sometime during the course of negotiation or may bypass a phase. The first two phases consist of pre-negotiation activities. In the Planning phase negotiators collect all relevant information, try to model their counterpart and specify objectives, preferences, reservation values, and their initial strategy. In the Agenda Setting and Exploring the Field phase, they exchange information concerning negotiation issues, protocol and deadline. At this phase the engaged parties may reassess the negotiation problem and change their initial strategy. In some cases many activities of this phase may be skipped, particularly if negotiators do not want to disclose strategic information, such as their deadline. The third phase, Exchange Offers and Arguments, is the phase that negotiation actually takes place. The two sides take alternate turns and exchange offers and counter offers. During the negotiation dance, the engaged parties may modify their strategy. Negotiation continues until deadline is reached, where the process terminates without success, or until an agreement is established. It is at the fourth phase that both parties confirm the agreement. Concluding Negotiation consists of post-negotiation activities, and takes place when negotiators reach an agreement. In this phase they may discuss additional issues which have no impact on the negotiations (e.g. the agreement implementation).

2.4 Research methodology used in the study of negotiations

The main research methods used in the theoretical analysis of negotiations are game theory (a normative approach – how groups of ultra-smart individuals should make separate interactive decisions), decision analysis (a prescriptive approach - how an analytically inclined individual should and could make wise decisions), behavioral decision making (a descriptive approach - the psychology of how ordinary individuals do make decisions), negotiation analysis (mostly prescriptive - how groups of reasonably wise individuals should and could make joint, collaborative decisions) and artificial intelligence (a prescriptive approach).

2.4.1 Game Theory

Game Theory has been extensively used in the study of negotiation interactions. Its analytical approach specifies what each of the rational players should do given a set of tightly defined assumptions.

A central concept in game theory is that of equilibrium. A notable type of equilibrium is the so-called Nash equilibrium, where no player has an incentive to deviate from a particular strategy, given that the other players stick to their strategies. Two strategies S and T are in Nash equilibrium if one player uses strategy S and the other player cannot do better by using some strategy other than T, and vice versa. Another level of equilibrium is that of Perfect equilibrium, which is achieved in games with multiple steps, and given that a player uses strategy S, there is no state in the game where the other player can do better by not sticking to strategy T. Finally, a third level of equilibrium is that of dominant equilibrium, where a player cannot do better than play strategy S, irrelevant to the strategy of the other player.

Another concept used in game theory is that of mechanism design, also known as the implementation problem. Given a group of negotiators with predefined utility functions and preferences over the different social outcomes, the objective is to design a game with a unique solution (equilibrium strategies). If each negotiator acts ‘rationally’ and adopts the equilibrium strategy, the social welfare function, which rates all possible social outcomes, will be maximized.

A seminal work on the use of game theory tools to the study of automated negotiations is that of Rosenschein and Zlotkin [21].

Numerous game theoretic frameworks that provide neat solutions to the negotiation problem can be found in literature [22] [23] [24].

However, the assumptions made in game theory are too restrictive to have wide applicability. Unbounded computational power (resources) of the negotiators, complete knowledge of the outcome space and of the preferences and utilities of the other parties, as well as the assumption that all parties act rationally, are some commonly stated points that restrict the application of game theoretic models in real situations. The third assumption is required in game theory, because each negotiator assumes that his counterparts will adopt the optimal strategy, and searches for the best response to optimal strategies. However, if game theory’s predictions become inaccurate, its prescriptive advice becomes unreliable.

In an attempt to overcome the problems of game theoretic approaches, heuristics are applied to the design of negotiation procedures. In heuristic approaches the assumptions of complete knowledge, rationality and unbounded computational power are relaxed, and negotiators operate in uncertain and dynamic environments, and adjust their behavior with respect to the elapsing time, diminishing resources and counterpart’s moves. Heuristic approaches result in sub-optimal heuristic search in the space of possible agreements [25].

An extension of heuristic approaches is argumentation-based techniques that use communication performatives such as lies, threats, promises and rewards during negotiation. Negotiators increase the likelihood and quality of an agreement by exchanging arguments that influence each others’ states [26].

2.4.2 Decision Analysis

Another research method which has been used in the theoretic analysis of negotiations is that of decision analysis. Decision analysis deals with modeling, optimizing and analyzing decisions, and assists decision makers in complex situations, usually under uncertainty. Decision analysis provides prescriptive orientation and should be evaluated by its ability to help people make better decisions.

According to Simon [27], decision making comprises of the following four steps: recognize the problem, specify the decision-makers' objectives, develop alternatives, evaluate and choose among the alternative decisions. Decision theory provides a wide range of instruments which can help represent, analyze, solve and evaluate a decision problem, as well as uncover existing relationships among data. For the representation and analysis of the problem graphical paradigms, such as decision trees and influence diagrams play an important role. Other tools, mainly based on statistical methods, such as forecasting and regression analysis, have also proved important in the analysis and recognition of relationships between data. The decision maker explores the list of various alternatives and predicts the consequences that would arise from each particular alternative [28]. In order to make a wise decision he assesses his judgments about uncertainty and examines his attitude towards risk. The decision maker performs uncertainty analysis, for example assigns subjective numerical probabilities to the likelihoods of the outcomes. A von Neumann-Morgenstern expected utility criterion [29] typically aggregates subjective probabilities, values, risk and time preferences in ranking possible actions to determine the optimal choice.

2.4.3 Behavioral Decision Making

There also exists a descriptive view of decision-making which focuses on how people actually make decisions. This view, which heavily relies on psychology, is empirical and provides evidence that people process information, assess probabilities, and make decisions in ways not consistent with the rational prescription of decision analysis and game theory. Research in factors, such as social relationships, egocentrism, ethics, emotions, and intuitions was incorporated into the field of negotiations [30]. Descriptive research assists a negotiator into anticipating the likely behavior of the counterparts. In [31] it is stated that negotiators care more about the relative than about the absolute outcome, often preferring Pareto-inefficient solutions in order to avoid being comparatively disadvantaged. For instance people were found to prefer the outcome of seven dollars to each side, than eight dollars for them and ten for the counterpart. Another finding provided by Thompson and Loewenstein [32], states that the more egocentric negotiators are the less likely it is to conduct successful negotiations. There are also studies that focus on the permissibility of common bargaining tactics. A characteristic debate is that of ethics of deception in negotiations [33]. Emotion of the negotiators is another factor that plays an important role in negotiations. It is found that positive mood increases negotiator's tendency to adopt a cooperative strategy and helps avoid the development of hostility and conflict [34]. Additionally, negotiators' frames (positive or negative) also seem to play a crucial role on the risk profile of the disputants. In cases where negotiations are viewed as procedures of gain maximization (positive or gain frame), negotiators are more risk aversive, while in cases where they are viewed as procedures of loss minimization (negative or loss frame), negotiators are more risk prone [35]. Reliance on intuition is another issue studied by the behavioral scientists. It is believed that negotiators trust their intuition and this often leads to irrational behavior, improper weighting of information and sub-optimal outcomes [36].

Good descriptive analysis is highly empirical and can lead to good predictions of actual behaviors.

2.4.4 Negotiation Analysis

The field of negotiation analysis lies between the fields of behavioral decision making, decision analysis and game theory. It tries to fill the gap between prescriptive and descriptive studies, and develop theories that will aid negotiators and third parties. Game theory mostly gives normative advice to all parties, while negotiation analysis concentrates on giving prescriptive advice to one of the negotiators after reflecting on the behavior of other negotiators. Advice to one side does not necessary presume the full (game-theoretic) rationality of the other sides. Instead it tends to de-emphasize the application of game-theoretic solution concepts to find unique equilibrium outcomes. Negotiation analysts generally focus on changes in perceptions of the zone of possible agreement and the (subjective) distribution of possible negotiated outcomes conditional on various actions. Sebenius in [37] identifies the basic elements of negotiation analysis and associates them with the corresponding research disciplines. Mapping the set of potentially relevant parties and their relationships, identifying personal interests, and assessing alternatives lie in the context of decision analysis. Negotiation analysts combine decision analysis with game theoretic concepts when it comes to structure the negotiation outcome. More specifically they use game theoretic techniques to compute the strategies that will result to equilibrium and assess the risk of applying them. The negotiator's subjective distribution of beliefs about the negotiated outcome conditional on using the game-theoretic tactic is compared with his subjective distribution of beliefs about the negotiated outcome conditional on not using them. The tactic is attractive if the former distribution gives the negotiator higher expected utility than the latter.

2.4.5 Artificial Intelligence

Approaches stemming from computer science and particularly artificial intelligence (AI), have also contributed in the design of software agents and negotiation support systems. Negotiation problems are usually ill-defined, information is not equally distributed among participants and negotiators have only partial knowledge about their counterparts. Methods of AI allow negotiators to learn and update their knowledge about their counterparts and the environment. They are also able to make wiser decisions and search for optimal or sub-optimal strategies. In the planning phase, where the negotiator has to select negotiating partners and strategy, and during the conduct of negotiation where he has to update his knowledge and decide his next action, models based on probabilistic decision theory, possibilistic decision theory, Bayesian learning, case based reasoning, Q-learning, genetic algorithms and neural networks are used to support the negotiation activities. This thesis focuses on the contribution of AI in negotiations and these methods will be discussed in detail in the following sections.

2.5 Electronic Negotiation Systems

At this point it is essential to distinguish between face to face (F2F) and electronic negotiation systems (ENS) [38]. In F2F negotiations, the participants, provided with enhanced degree of freedom, are foremost responsible to decide on information, rules, activities etc. This is the case with traditional negotiations which rely on human expertise and little, if at all, on information systems. The advancement of software engineering and internet technologies has given rise to the development of electronic negotiation systems (ENS), defined as software tools for the purpose of organizing, facilitating, supporting and/or automating negotiation processes. Kersten and Lai in [39] identify four kinds of software that have been designed for negotiations; e-negotiation tables (ENT), negotiation support systems (NSS), negotiation software agents (NSA) and negotiation agent assistants (NAA). ENTs provide the participants with a virtual space and tools in order to undertake negotiation activities. They are considered as *arenas*, or negotiation workbenches, and are usually *passive* systems, oriented in facilitating communication of participants. In their simplest form they are virtual spaces where negotiators and third parties post offers and messages.

NSSs are defined as software tools which incorporate communication facilities and implement models and procedures to support participants in negotiation activities.

NSAs and NAAs are based on software agent technologies. The key characteristics of software agents is that they are autonomous, they act on behalf of their human or artificial principles, they are able to be reactive and proactive in deciding on undertaking an action and they exhibit some level of capabilities such as learning, co-operation and mobility. NSAs are designed with the purpose to automate one or more negotiation activities. NAAs are agents designed to provide advice and critique, without engaging directly in the negotiation process. NAAs play the role of analysts and experts and provide negotiators with relevant knowledge about their counterparts, process and problem.

2.5.1 Agent Types

Braun et al. [20] discuss the different types of software agents with respect to their role, and distinguish user profile, information, opponent profiling, proposer, critic, negotiator, and mediator agent. The tasks delegated to each agent type relate to the process model depicted in section 2.3 and each agent may be assigned tasks of different phases.

User Profile Agent: A user profile agent focuses on determining user preferences expressed in terms of reservation values, aspiration levels, as well as best alternative to negotiating agreement (BATNA), objectives, and strategies.

Information Agent: Information agents are engaged in actively seeking, retrieving, filtering and delivering information relevant to the negotiation domain.

Opponent Profiling Agent: Knowing the opponent's profile relates to the identification of objectives, preferences, and strategies of the opponent. Such information yields better strategic decisions and can be delivered by the opponent profiling agent.

Proposer Agent: Proposer agent is concerned with the search and generation of offers to be submitted to the opponent.

Critic Agent: Critic agent is a type of NAA concerned with the evaluation of offers received from and addressed to the opponent. Such agents are capable of providing verbal feedback on drawbacks and benefits of these offers.

Negotiator Agent: NSAs are capable of conducting negotiations on a semi or fully autonomous fashion, based on certainty upon the objectives, preferences and tactics.

Mediator Agents: The purpose of this agent is to coordinate activities and generate mutually beneficial offers. Raiffa [16] discusses the roles of interveners in negotiations and distinguishes between facilitator, mediator and arbitrator.

Increasing level of automation gives rise to interaction of different agent types which update their beliefs about the particular negotiation stance, and carry on with their task. Figure 4 illustrates a generic agent architecture, as depicted in [20].

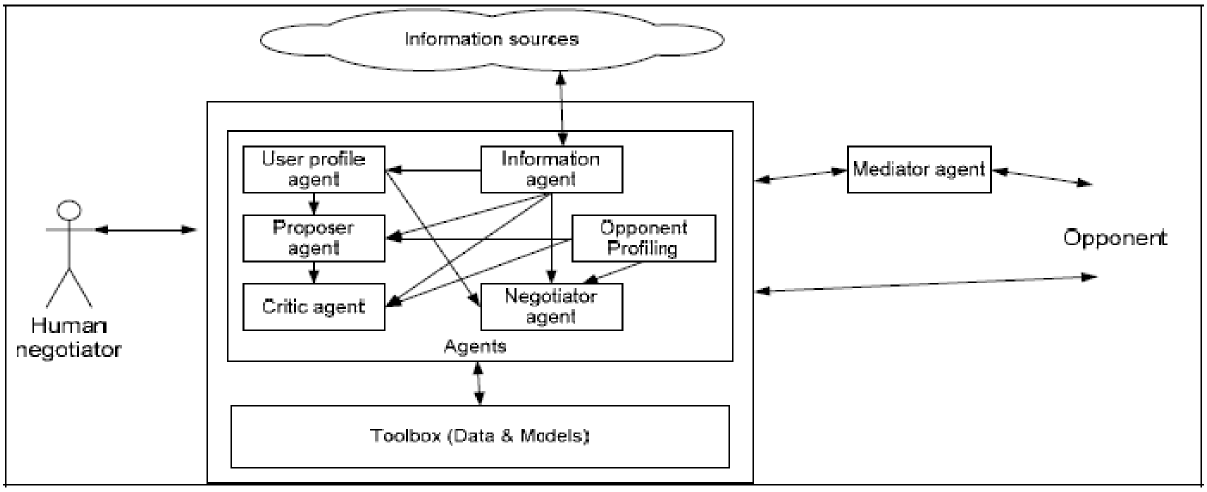


Figure 4: A Generic Agent Architecture

2.5.2 Systems and platforms

Numerous software platforms have been developed to support negotiations in different exemplar domains. WebNS [40] is one such example that focuses on offer preparation and provides support to the exchange of offers and counter-offers with the use of real-time chat and video conferencing. Inspire [41] is another ENS that is based on a three phase process model and is mainly used to investigate cross-cultural negotiations. Inspire has been used to facilitate the bicycle-parts purchasing problem. The InterNeg Support System (INSS) [41] is an extension of Inspire. It provides a workbench of negotiation tools and allows the introduction of new issues at any point during negotiation. A system that is used to generate ENS instances and allows the execution of several negotiation processes is InterNeg virtual integrated transaction environment (Invite) [42]. It is mainly used for training and research purposes.

Other software platforms are based on agent technology. e-Negotiation Agents (eNAs) [43] is an example of an agent-based e-platform demonstrated in a number of test-beds of e-commerce trading. It provides a suite of negotiating agents that act on behalf of their users and can engage in automated negotiations over the Internet. The agents share information about objects and conduct negotiations usually following a predefined protocol. Fuzzy e-Negotiation Agents (FeNAs) [44], is another prototypical system with e-commerce trading agents. FeNAs consists of a number of autonomous agents that conduct concurrent negotiations, and specify fuzzy constraints and preferences. Other agent-based negotiation systems are Intelligence Trading Agency (ITA) [45], a pilot application that uses the Personal Computer trading scenario, and Kasbah [46], a prototype online virtual marketplace where users dynamically configure and describe the items to sell. Similarly, negotiations in market places may include dispute resolution over non tangible items. For instance Cybersettle [47] is an on-line system that supports negotiations of insurance claims. When electronic commerce moves into business-to-business marketplaces or even supply chain management, negotiation over complex, mutually determined contracts describing the terms of the transaction is necessary. Business processes are supported by infrastructures, where control and management of activities is entrusted to autonomous agents. One such example is ADEPT

(Advanced Decision Environment for Process Tasks) [25], which can be viewed as a generic method of structuring the design, development and conduction of business processes, based on a set of autonomous agents which interact when they have interdependencies. Another example is Tete - a –Tete [48] which supports negotiations across multiple terms of a transaction including warranties, delivery times, service contracts, return policies and other value-added services. Similarly, MAGNET (Multi AGent NEgotiation Testbed) is an experimental architecture developed at the University of Minnesota to provide support for complex agent interactions such as in automated multi-agent contracting. Agents in MAGNET negotiate and monitor the execution of contracts among multiple suppliers [49].

Last but not least, eAgora [50] is an e-marketplace that allows buyers and sellers to engage in multi-issue negotiations. Its services include a software agent that generates and critiques offers.

E-market players are often modeled with the use of autonomous software agents. Multi-agent platforms is a preferred mechanism for studying market deregulation, since social aspects are taken into account and reflect with better accuracy the relationships of various market players compared to auction mechanisms. In the following section we discuss the domain and interactions of an electricity market.

2.6 An example domain: electricity markets

An electricity supply system consists of three basic functions: power generation, transmission and distribution. Electricity is produced from a number of energy sources, distinguished in those with high capital cost (such as hydro or nuclear stations) and those with low capital but high operating cost (such as gas turbines), mainly used to meet peak loads. Large power and heat stations are often located at considerable distances from the main areas of electricity demand. For this reason it is essential to have an adequate electrical system to transport electrical power from the large stations to the main load centers. Transmission of very large amounts of power involves high voltage networks (HV). An electricity distribution system is then used to deliver electrical energy from transmission substations or small generating stations to each customer, transforming to a suitable voltage, medium (MV) or low (LV), when necessary. In the 1990's there has been a strengthening trend towards breaking up the vertical integration in the electric power industry by separating the generation, transmission, and distribution of electricity into separate business areas. This trend has increased the demand towards opening up transmission and distribution networks to producers, suppliers and consumers and the appearance of independent power producers. In the power distribution sector deregulation allows consumers to select their electricity suppliers.

2.6.1 Market deregulation: the situation in Greece

All the countries of Western Europe have taken steps to liberalize their electricity industries. Large consumers (selected customers) in every country can choose their electricity suppliers and in some countries this choice is given to every consumer. In Greece an independent administrative authority, the Regulatory Authority for Energy (RAE), has been established to promote harmonization of the Greek law with the directives of the European Community concerning the liberalization of the electricity market. RAE controls and monitors the operations of all sectors in energy market. It is provided with the ability to issue administrative and normative acts, which are later approved by a governmental department (the Ministry of Development). It also publishes the annual electricity transactions and progress reports (RAE). Deregulation of electricity was conducted in 2001. Although the energy business areas have been liberalized, some network activities remain monopolistic. To this extent two organizations, one controlling the transmission system (ΔΕΣΜΗΕ) and one the distribution networks of medium and low voltage (ΔΙΑΧΕΙΡΙΣΤΗΣ ΔΙΚΤΥΟΥ), have been established. Stakeholders who wish to insert energy to the system also need to pay fees for the use of interconnection lines. The main entities that form the electricity market are producers (generators), suppliers and consumers. The latter are further distinguished in selected customers, mainly industries, who are provided with the ability to select among a number of suppliers, and non-selected customers who are supplied by a particular organization (ΔΕΗ). The types of bills of sale among the entities of an electricity market that have been published by RAE actually disclose the different business transactions and are grouped in the following activities:

Electricity Supply to Selected Customer

Contracts are established between suppliers (or producers) and selected customers, such as industries, who mainly connect to high or medium voltage networks. Electricity supply to selected customers is operating under commercial competition.

Electricity Supply to Non-selected Customer

Since 2006, the ability to select one’s own supplier has been opened to households who connect to low voltage networks. Nevertheless, this operation is still monopolistic, since there is only one organization in Greece (ΔΕΗ) which supplies non-selected customers.

Use of the Transmission System

This type of contract is issued by the organization which controls the transmission system (ΔΕΣΜΗΕ) to those who wish to insert energy to the system (producers). It involves fees imposed for the use of the transmission system and is monopolistic.

4. Use of the Medium Voltage Network

5. Use of the Low Voltage Network

The above operations are monopolistic and issued by one organization (ΔΕΗ) to those who wish to use the distribution networks (for instance suppliers who provide energy to consumers who connect to medium or low voltage networks).

2.6.2 Electricity e-market

Restructuring the electricity industry into an open market has created demands for new software tools to meet future challenges and requirements of competitive environments. There exist two types of markets in which energy is traded: the spot market and the forward (over-the-counter) market [51]. In the spot market energy is traded in real-time and transactions are conducted through centralized auction mechanisms that determine how much energy each unit should produce to meet the demand. On the other hand, in the forward market, bilateral contracts concerning future delivery of electricity are established. Electricity supply of selected customers and use of the transition system, are two activities where bilateral negotiations are encountered. In the first case producers (generating companies) trade energy by way of signing bilateral contracts, which are referred to as physical forward contracts, with their counterparts (e.g., selected customers).

The parties communicate with the use of agents who automate the negotiation process, to facilitate computational overhead, be able to analyze larger stacks of data, and reduce human intervention. Supply of electricity is considered as a service according to Directive 2003/54/EC. Specific details such as trading quantity, trading duration, trading price and penalty terms (refunds) are bilaterally negotiated between the engaged parties. Negotiation contracts are discussed in the following paragraph.

2.6.3 Negotiation Contracts

The object being negotiated may also be referred to as a contract. Contracts contain identification and a negotiation part. The identification part includes information to uniquely identify the contract, such as contract id, contract name etc. The negotiation part involves the actual issues included in the offers exchanged between the engaged parties. According to [52] there are six elements (categories) that describe the attributes of the negotiation contracts (Who, What, Where, When, Why and How). These elements are described below:

"Who": This element provides the identities of the parties involved in the targeted negotiation process. It may include primary parties such as stakeholders (producers and consumers) or secondary parties such as supporting institutions, shipping companies etc.

"What": This element provides information about the negotiation subject. If for example negotiation is conducted for the provision of electricity, the number of KWh and its related price are two issues that fall under this category.

"Where": Attributes of this group concern the region where the service will be provisioned, or the object will be delivered. In the case of electricity trade, this element may involve the connection point of the distribution network (HV, MV or LV).

"When": Temporal clauses involve location information of the negotiation processes. An example could be time and duration of service provision.

"Why": This part describes the motivation of negotiation. In electricity provisioning, motivation depends on the electricity customers; therefore it may involve industrial, commercial, agricultural, public or house holding use.

"How": This attribute involves the penalty cost if terms are violated. Penalties are expressed in terms of price deductions as a means of customer insurance. If a supplier proves inconsistent he is obliged to reduce the agreed price by a specified amount.

Contracts that involve the provision of services are defined as Service Level Agreements (SLAs). In the Figure 5, we give an example of an SLA equivalent to an electronic electricity contract, established between a producer and a consumer (selected customer).

Service Name: Electricity Supply to Selected Customers
SLA_ID: Contract 1

Agreement Initiator: National Bank of Greece
Agreement Responder: Energy Global Trading Ltd
Service Provider: Energy Global Trading Ltd
WHO?

Connection Network: MV
Region of Provision: Athens
WHERE?

Motivation: Commercial
WHY?

Expiration Date: January 2013
Start Time: 07.00
End Time: 17.00
Duration: 10 hours
WHEN?

Number of KWh: 100
Price in Euro Cents per KWh: 12,8292
WHAT?

Penalty Cost: 10% of the agreed price
HOW?

Negotiation Part

Figure 5: A Contract between Suppliers and Selected Customers

In the example, negotiations are conducted for the settlement of duration (measured in hours), penalty terms (percentage of the sum which will be returned to the consumer in case of dissatisfaction), quantity (number of KWh) and price (in Euro Cents) per unit of electrical energy. These attributes constitute the Negotiation part. The negotiation environment is competitive and the two agents have opposing interests; the consumer will start from a low price and a low number of Kwh, which he will increase in each round, while he will start from a high percentage of refund and high service duration which he will decrease in each round. At the same time the producer will initially request high price per Kwh, and high number of Kwh which he will lower in each round, and low penalty and duration of service provisioning which he will increase in each round. In the next chapter we discuss the model of bilateral negotiations.

3. MODEL OF BILATERAL NEGOTIATIONS

The type of conflict that governs the interaction, the participants and the role they play, their social behavior, as well as the commodities (tangible or not) that are being discussed are used as discriminative entities in an attempt to classify the various negotiation domains. The domain being investigated in this thesis, concerns bilateral (one-to-one), multi-issued negotiations between competing agents (i.e. consumer and producer), where each party's individual preferences and strategic information are private and not disclosed to the counterpart.

3.1 Requirements of the negotiation domain

As mentioned earlier, the negotiation environment considered is tied to bilateral multi-issue negotiations, where all issues are bundled and discussed together (package deal). Further assumptions concerning the application domain are the following:

1. Negotiators, referred as producers or consumers, have conflicting interests.
2. Issues are quantitative and negotiation amounts to determining a value between delimited ranges.
3. Negotiators have a prior agreement over the set of negotiable issues.
4. Negotiators are autonomous and do not have access to private information of their opponents (assumptions about opponent's strategies and preferences is only considered for the development of experimental settings).
5. Negotiators are subject to time restrictions, and define a deadline, the maximum time they are willing to negotiate. Deadline is not revealed to the counterpart.
6. Negotiators have limited resources.

In the next sections the bilateral negotiation model is described.

3.2 The negotiation protocol

The negotiation activities, usually translated as rules of encounter, indicate permissible actions, content and timing of utterances. These rules constitute the negotiation protocol, and guide an agent to address the challenge of "what he should say and when in a particular negotiation framework". Each interaction is governed by a set of rules that constrain the public behavior of the participants [21]. They are incorporated in the negotiation protocol and guide software processing, decision making, communicational tasks and specification of permissible inputs and actions [53].

The interaction is modeled as a sequence of offers and counter-offers, which terminates with either a commitment by both parties to a mutually agreed solution or unsuccessfully without establishment of an agreement. The possible state transitions of the negotiation protocol are shown in Figure 6.

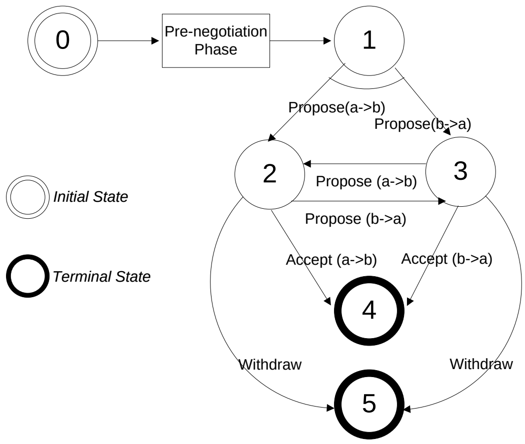


Figure 6: The negotiation protocol

The protocol starts with a dialogue to establish the conditions for the negotiation (transition from state 0 to state 1). In this pre-negotiation phase, the engaged parties must specify the set of issues and decide who will initiate the process. The initiator then makes the first offer (transition from state 1 to state 2 or 3; the 'or' transition from state 1, is represented with an arc joining the two proposals in Figure 6). The counterpart may accept the initial proposal and successfully terminate the process moving to state 4, send a counter-offer moving to state 2 or 3, depending on who was the initiator, or withdraw moving to state 5. If he decides to send a counter-offer, it is the initiators' turn to make a move. He may accept the received offer, withdraw from negotiation or send a counter-offer as well. The engaged parties may iterate between states 2 and 3, until an offer is accepted, or until any of the negotiators reaches his deadline and withdraws. Termination of the negotiation protocol is guaranteed through the presence of deadlines. This protocol is an extension of the Contract Net Protocol [21].

3.3 Formal definition of negotiation entities

In this section we proceed with a more strict definition of the negotiation environment. Let Agents = { α, b }, be the set of autonomous agents that engage to the discourse. At a pre-negotiation stage the agents agree upon the negotiable issues (or attributes) and their meaning. Therefore, we consider a finite set of quantitative issues I = {i_1, i_2, . . ., i_n} which form the negotiation part of the contract. For each issue in I, agent *α* assigns a range of permissible values. This information is not revealed to the counterpart and the domain of reservation values for each issue i is defined as $D_i^a : [\min_i^a, \max_i^a]$.

Agent *α* also specifies a utility function $U_i^a : D_i^a \rightarrow [0,1]$ that scores issue i in the range of its permissible values. For convenience, scores are kept in the interval [0,1].

The relative importance for each issue i is assigned by a weight w_i^a by agent *α*. The weights are normalized, thus for n issues $\sum_n^1 w_i^a = 1$.

At a pre-negotiation phase, agent *α* also needs to specify the deadline $T_{\max}^a$, which indicates the maximal time he is willing to spend during the discourse. In the cases studied time variable *t* is discrete and expresses the interaction step (negotiation round).

Having set the reservation values, weights, utility functions, and deadlines the engaged parties can proceed with the actual conduct of negotiation (state 1).

As shown in Figure 6, the agents take alternate turns proposing offers and counter-offers (they iterate between states 2 and 3) until an agreement is established (state 4) or until any of the involved parties reaches his deadline and negotiation terminates without success (state 5). The offers exchanged during the discourse are represented by vectors in the multi-dimensional space. $X^t_{(a \to b)} = \left(x^t_{1(a \to b)}, x^t_{2(a \to b)}, \ldots, x^t_{n(a \to b)}\right)^T$ represents the negotiation offer sent from agent α to agent b at time t, and each attribute $x^t_{i(a \to b)}$ denotes the offered value of negotiable issue i. It should be noted that agents keep track of the offers exchanged during the discourse and formulate the negotiation thread, formally defined below:

A negotiation thread between agents a,b at time t_n, noted $X^{t_n}_{a \leftrightarrow b}$, is any finite sequence of length n, of the form ($X^{t_1}_{(a \to b)}, X^{t_2}_{(b \to a)}, X^{t_3}_{(a \to b)} \ldots$) where:

$t_{i+1} > t_i$, the sequence is ordered over time

for each issue i, $x^{t_j}_{i(a \to b)} \in D^a_i$ and $x^{t_{j+1}}_{i(b \to a)} \in D^b_i$, where j = 1,3,5 …

Agents can assess the different negotiation offers by computing the overall utility, which is the weighted sum of the utilities attributed to each issue, thus:

$$U^a(X^t_{(a \to b)}) = \sum_{i=1}^{n} w^a_i * U^a_i(x_i) \qquad \text{(eq. 1)}$$

The additive utility function in (eq. 1) allows the consolidation of individual preferences over each issue into a single preference value.

An agreement may be reached only if the attribute values of the proposed offer lies within the acceptable range for both parties, or if any of the two agents receives an offer that incurs higher utility than the offer he is planning to send in the next round. Thus the action A^a_t taken by agent a at time t is formulated as follows:

$$A^a_t = \begin{cases} Quit, \ if \ t > T^a_{\max} \\ Accept \ X^{t-1}_{(b \to a)}, \ if \ U^a(X^{t-1}_{(b \to a)}) \geq U^a(X^t_{(a \to b)}) \\ Send \ X^t_{(a \to b)}, otherwise \end{cases} \qquad \text{(eq. 2)}$$

Where $X^t_{(a \to b)}$ is the offer agent a is planning to send at time t, and $X^{t-1}_{(b \to a)}$ is the offer he received from agent b at time t-1. The construction of offers in each round is based on the agents' negotiation strategy, which is discussed in the following section.

3.4 Negotiation strategies

Strategy involves the decision of an action, given a set of permissible ones (specified by the protocol). The term is often used synonymously with behavior. As mentioned in the second chapter, Blake and Mouton introduced the Dual Concerns model which describes five behavioral classifications regarding the level of assertiveness and cooperativeness [19]. These classifications are competing, collaborating, compromising, accommodating and avoiding. Assertiveness reflects the concern to satisfy one's own interests, while cooperativeness reflects the concern to satisfy the other party's interests. Cooperativeness and assertiveness are viewed as two independent dimensions that run from weak to strong as shown in Figure 7.

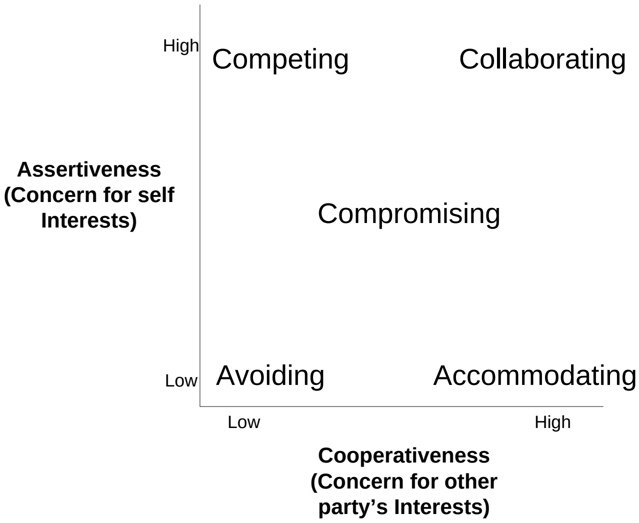


Figure 7: Behavioral Classifications (the Dual Concerns Model)

Avoiding conflict style is characterized by inaction and passivity, and is typically used when an individual has low concern for his own and for the other party's interests. Accommodating conflict style is characterized by high concern for the other party's interests and low self-concern. Negotiators of this type tend to make high concessions in order to maintain stable, positive social relationships. In contrast, competitive conflict style maximizes individual assertiveness (self-concern). Groups consisting of competitive members generally seek domination over others, and typically see conflict as a "win-lose" situation. At the other end, collaborating conflict style is characterized by high concern to satisfy one's own and the other party's interests. Collaborating negotiators see conflict as a creative opportunity, and are willing to invest time and resources into finding a "win-win" solution. Finally, compromising conflict style is typical for negotiators who possess an intermediate-level of concern for both personal and other party's interests. Compromisers anticipate mutual give-and-take interactions.

The different actions taken by the negotiators during discourse can be viewed as moves in the utility space, where utility is computed by equation 1. In bargaining theory, the different shares of the utility space distinguish two different types of bargaining, integrative or "win-win" and distributive or "win-lose". Integrative negotiations are highly cooperative, and can be viewed as non-zero sum games where the values of different issues shift along different and usually independent dimensions [21]. Opposing, distributive negotiations may be viewed as a zero-sum games, where the issue value shifts across one dimension with any gain for one party being the other's loss [21,54].

Faratin et al. [25] has implemented a responsive and a trade-off mechanism that can be used to generate offers and counter-offers covering all types of behaviors. In Figure 9 a scenario of compromising agents using the responsive mechanism is illustrated. The two agents gradually concede, making offers of decreasing utility, until an agreement is established. In the trade-off mechanism, which results to "win-win solutions", contracts that increase the joint utility are searched in the pareto optimal curve (curve illustrated in Figure 8).

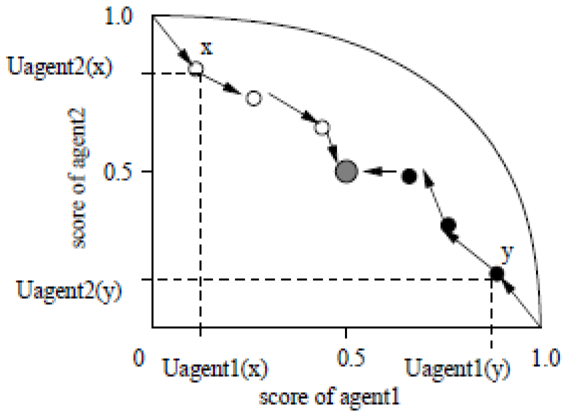


Figure 8: Negotiation example with use of the Responsive Mechanism [25]

This thesis focuses on competing behaviors where negotiators act as individual utility maximizers. The use of trade-offs is out of the scope of this research, and will not be further analyzed. In the following section the responsive mechanism [25], providing a heuristic-based approach to the study of negotiations (section 2.4.1), is discussed.

3.4.1 The responsive mechanism

The Responsive Mechanism is based on the combination of simple functions called *tactics*, to generate an offer or counter-offer of a single issue of the negotiable object. Tactics are classified to Time Dependent (TD), Resource Dependent (RD) and Behavior Dependent (BD), reflecting the agent's behavior with respect to the elapsing time, diminishing resources and counterpart's responses respectively. These criteria are motivated by an agent's computational and informational bounds. For example as time elapses and resources are consumed, offers which incur high utility may be unattainable, and agents may prefer to make higher concessions in order to reach an agreement. Given that the agents may consider different criteria to compute the value of a single issue, the generation of a counter-offer is modeled as a weighted combination of tactics. The tactics of the responsive mechanism are discussed in detail in the following subsections.

3.4.1.1 Time dependent tactics

In this family of tactics rising recessional tendency is modeled as the deadline approaches. Time *t* is the predominant factor to the formulation of the next offer and the concession curve is what differentiates tactics in this set.

The value computed by agent α for issue i varies in the interval $[\min_i^a, \max_i^a]$, and is defined by a function $a_i^a(t)$ as follows:

$$x_{i(a\to b)}^{t} = \begin{cases} \min_i^a + a_i^a(t)(\max_i^a - \min_i^a), \textit{if value increases over time} \\ \min_i^a + (1 - a_i^a(t))(\max_i^a - \min_i^a), \textit{if value decreases over time} \end{cases}$$

A wide range of Time Dependent functions can be defined by varying $a_i^a(t)$. A constant k_i^a is also used to specify the initial value to be offered. It must be ensured that $0 \le a_i^a(t) \le 1$, $0 < k_i^a \le 1$, $a_i^a(0) = k_i^a$ and $a_i^a(T_{\max}^a) = 1$. Two types of functions, polynomial and exponential are mainly used to model the time-dependent function. Both types are parameterized by a real value β which indicates the convexity degree of the curve. Function $a_i^a(t)$ is thus formulated as follows:

Polynomial: $a_i^a(t) = k_i^a + (1 - k_i^a)\left(\frac{\min(t, T_{\max}^a)}{T_{\max}^a}\right)^{\frac{1}{\beta}}$

Exponential: $a_i^a(t) = e^{(1 - \frac{\min(t, T_{\max}^a)}{T_{\max}^a})^{\beta} \ln k_i^a}$

Polynomial functions tend to concede faster than exponential. For the same large value of β, the polynomial function concedes faster at the beginning than the exponential one, and then they behave similarly. Depending on the value of β two extreme sets showing different patterns of behavior are identified:

Boulware Tactics: This set is adopted by "hard" negotiators who stick to their initial offer until time is almost exhausted, whereupon they decide to concede up to the reservation value. This behavior can be realized with values of β<1. Remaining firm in terms of demands is a technique to handle uncertainty: when the counterparts' preferences are unknown, one possible strategy is to stick to the same value during the discourse.

Conceder Tactics: This set is adopted by "soft" negotiators who decide on fast concession and quickly reach their reservation value. This behavior is realized with values of β>1.

The following Figure demonstrates the different behavioral patterns with respect to the value of β.

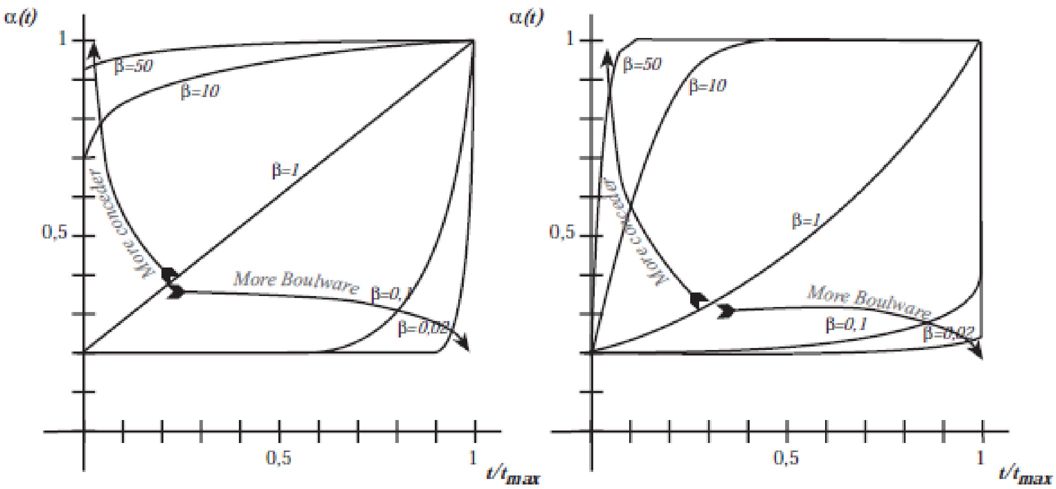


Figure 9: Polynomial (left) and exponential (right) functions for the computation of a (t) [25]

Time Dependent tactics can be viewed as a special case of Resource Dependent tactics, where the resource being consumed is time. Resource Dependent tactics are discussed in the following section.

3.4.1.2 Resource dependent tactics

This family of tactics models the pressure that the limited resources (e.g. number of negotiating agents, remaining time) and the nature of the environment impose on the negotiators. The expected behavior is for the agent to progressively become more conservative as the quantity of a resource diminishes. Formally, function $a_i^a(t)$ is computed as follows:

$a_i^a(t) = k_i^a + (1 - k_i^a)e^{-resource^a(t)}$

where the function resource$^{\alpha}$(t) measures the quantity of the resource for agent α at time t. If the number of agents who negotiate with agent α at time t is the resource, then resource$^{\alpha}$(t)=$\left|N^{a}(t)\right|$. As expected, the more agents negotiate with agent α, the lower the pressure to reach an agreement with any specific individual. If time is the resource then resource$^{\alpha}$(t)=min(0, t-$T^{a}_{\max}$).

3.4.1.3 Behavior dependent tactics

In this family of tactics negotiators imitate the behavior of their counterpart. The responsive action is the result of the observation of the other party's behavior and the degree of imitation forms the fundamental distinctions among the available tactics of this set. Like Boulware tactics, they can also be selected as a technique to handle uncertainty. Whereas Boulware tactics handle uncertainty by ignoring the behavior of the counterpart, these tactics condition their actions on the observed behavior. Three families of tactics Relative Tit-for-Tat, Random Absolute Tit-for-Tat and Averaged Tit-for-Tat are distinguished. Given a negotiation thread: $\left\{ \ldots, X^{t_n-2\delta}_{(b\to a)}, X^{t_n-2\delta+1}_{(a\to b)}, X^{t_n-2\delta+2}_{(b\to a)}, \ldots, X^{t_n-2}_{(b\to a)}, X^{t_n-1}_{(a\to b)}, X^{t_n}_{(b\to a)} \right\}$ with δ≥1 the families of tactics are formally defined as follows:

Relative Tit-For-Tat:

This family of tactics is based on the proportional imitation of the counterpart's behavior. Agent α reproduces, in percentage terms, the behavior (increase or decrease) performed by his opponent δ≥1 steps ago. The condition of applicability of this tactic is n>2δ. The offer produced is derived by the equation:

$$x^{t_{n+1}}_{i(a\to b)} = \min(\max(\frac{x^{t_{n-2\delta}}_{i(b\to a)}}{x^{t_{n-2\delta+2}}_{i(b\to a)}} x^{t_{n-1}}_{i(a\to b)}, \min^{a}_{i}), \max^{a}_{i})$$

Random Absolute Tit-for-Tat:

In this family of tactics agent α produces an offer mimicking the absolute value of increase or decrease performed by his opponent within the last δ steps. This value is increased or decreased by a random value which belongs to the range [0,M]. If R(M) is the function producing these random values, and s is 0 or 1 if the issue has a decreasing or increasing value over time respectively, the equation illustrating the produced offer is the following:

$$x^{t_{n+1}}_{i(a\to b)} = \min(\max(x^{t_{n-1}}_{i(a\to b)} + (x^{t_{n-2\delta}}_{i(b\to a)} - x^{t_{n-2\delta+2}}_{i(b\to a)}) + (-1)^{s} R(M), \min^{a}_{i}), \max^{a}_{i})$$

The condition of applicability of this tactic is again n>2δ.

Averaged Tit-for-Tat:

In this family of tactics agent α computes the average of percentages of changes in a window of size γ≥1 of its counterpart's history. When γ=1 the behavior is similar to the Relative Tit-For-Tat with δ=1. The condition of applicability for this tactic is γ>2. The produced offer is derived by the equation:

$$x^{t_{n+1}}_{i(a\to b)} = \min(\max(\frac{x^{t_{n-2\gamma}}_{i(b\to a)}}{x^{t_{n}}_{i(b\to a)}} x^{t_{n-1}}_{i(a\to b)}, \min^{a}_{i}), \max^{a}_{i})$$

3.4.1.4 Strategy: A linear combination of tactics

Different combination of tactics can be used to determine the agent's strategy. This concept is similar to the concept of mixed strategies used in game theory.

Given a set of m tactics and n issues, the negotiators' strategy is a weighted linear combination of the tactics, formulating the strategy matrix:

$$S^a = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1m} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ \gamma_{n1} & \gamma_{n2} & \cdots & \gamma_{nm} \end{bmatrix}$$

, where each γ_{ij} denotes the weight of tactic j in value generation of issue i, and for each issue i, $\sum_{j=1}^{m} \gamma_{ij} = 1$.

3.4.2 Metastrategy selection

Strategy selection involves the decision of the negotiators actions and this decision might not be crisp. As new knowledge penetrates the negotiation settings, participants might decide to refine or completely alter their initial strategy. In the process model adopted by Braun et al. [20] strategy selection relies among the tasks of the planning phase and is updated during the negotiation analysis phase.

Deciding which actions to take on a specific encounter is a process influenced by a number of factors. Thus an agent must be knowledgeable about the encounter in order to make efficient strategic choices. Knowledge sources stem from the environment, the data of previously concluded negotiations and from offer exchanges of current discourse. The generic agent architecture in [20] indicates possible interactions that help an agent formulate beliefs and estimations of the "world", and guide his actions and behaviors. In frameworks where fully automation is supported by a single NSA, the agent must be skillful enough to collect relevant information and extract knowledge that will result the generation of efficient strategies.

When an agent uses his experience to decide upon strategies prior to negotiation (at the planning phase), he engages to a meta-level decision-making process, usually conducted off-line, termed as meta-strategy decision. The agent takes into account the factors that influence strategy selection, collect all relevant information, and decides an appropriate mechanism for the interaction. An agents' knowledge can be partitioned to self knowledge, concerning personal resources, preferences, goals and risk attitude, and to situation knowledge acquired by the agents' sensors and concerning environmental changes. The latter includes knowledge concerning the counterpart's strategy [55].

Strategy selection can be defined as a search problem in the space of potential strategies. Decision making in the planning phase is equivalent to the initial selection of the interaction mechanism, which may evolve during discourse with the use of trajectory methods (for weight adjustment). Initial selection is guided by the current state of the environment and the initial assumptions of the opponents' preferences and strategy, while the evolution and adaptation of a strategic scheme mainly depends on the observation of the opponent's moves and the modulation of his concessions.

In chapter 4 a review of negotiators enhanced with learning techniques to facilitate strategy selection at planning phase and during discourse is found.

4. AI-BASED NEGOTIATING AGENTS

Different behaviors, reflected through the strategies, result to different negotiation outcomes. Extensive experiments have proved that there does not exist a universal best strategy, rather it depends on the negotiation domains, protocols, participants' goals and attitude towards risk, as well as counterparts' behavior. Negotiators often have to deal with vague data, limited information, uncertainty and time restrictions. Enhancement of their strategic core with AI-Based techniques provides a way to address these issues and adds value to negotiators since it allows them to extend their knowledge and perception of the domain.

This chapter provides an overview of learning methods that form the core of state-of-the art negotiators. The main objective is to facilitate the comprehension of the domain by framing current systems with respect to learning objectives and phases of application (at planning phase or during discourse), as well as to reveal current trends and virtues of the applied methods.

4.1 Classification with respect to the learning technique

Negotiation process model adopted in most frameworks discriminates strategy selection at the planning phase and strategy update during discourse. This has lead to the existence of two schools when it comes to studying negotiation strategies. The first is concerned with the selection of a strategy at a pre-negotiation phase, during formulation of the problem. The second is concerned with strategy update, the change of behavior during discourse, which may be due to changing preferences or environmental parameters. We devise agents to those who intuitively adjust their behavior, and to those who use reasoning skills in the decision-making process. In the former category agents engage in learning methods that differ to the extent of knowledge exploration and exploitation. Specifically, explorative techniques also imply the search for new solutions, while repetitive techniques are based on knowledge reuse. For agents who engage in reasoning processes to decide upon appropriate actions, learning is introduced in the form of predictive decision making, where estimations of factors that influence strategy selection or update serve as input to the agents' decision making. With respect to these factors we discriminate the following three categories: explorative, repetitive and predictive which may be applied either at the planning phase for initial strategy selection or during discourse.

4.1.1 Explorative strategies

Explorative strategies are equivalent to search techniques that follow a trial and error learning process until some convergence condition is satisfied. Such techniques are Q-learning and Genetic Algorithms (GAs).

An agent that uses reinforcement learning techniques is rewarded or punished based on the consequences of the action taken. Each state-action pair is mapped to a value named Q-value. When an action is performed, agent receives a reward, which is used to evaluate the transition to a new state. The Q-value $Q(i,a)$ of state i after taking action α is updated after the following formula:

$$Q(i,a) = Q(i,a) + m\left[r(i) + \gamma \max_{a'} Q(j,a') - Q(i,a)\right]$$

where m is a learning rate, r(i) is the reward gained by performing action a in state i, γ is a discount parameter and j is the state attained. The reward may be positive or negative depending on whether the action had good or bad results. There exist three different policies regarding knowledge exploration and exploitation. These are Greedy, E-Greedy and Boltzman exploration. The Greedy policy favors knowledge exploitation, and the agent tends to pick the action with the highest Q-value. The E-Greedy policy favors knowledge exploration, as the agent picks at each state a random action with probability ε and the action with the highest Q-value with probability (1-ε). Finally Boltzman exploration is another policy favoring knowledge exploration, as the agent picks an action at each state with a probability controlled by a temperature parameter. Q-learning may be applied to learn from previous encounters where trials are the previous negotiations, or from the current encounter, where trials are the previous offers.

Cardoso and Oliveira, [56] implemented a Q-learning agent who acts in a dynamic environment and tries to estimate which combination of tactics to use in each state. Knowledge is acquired from previous encounters, since the state is defined by environmental parameters that relate to the number of agents and available time of the adaptive agent. Actions are defined as combinations of tactics and are assessed at the end of negotiation, as positive rewards if a deal is achieved, or negative rewards (penalties) if negotiation ends without an agreement. The measure of the reward (Q-value) is determined by the utility or benefit that the procedure incurred to the agent.

Application of Q-learning to the current encounter requires feedback from the opponent in order for the agent to compute the reward value used in the learning process. An example of applying Q-learning algorithm for learning from the history of the current negotiation can be found in [57]. The state is defined as the current offer in the form of a sequence of values, and the action specifies how each attribute should change (increase, maintain, or decrease) in order to generate the next offer. If the attribute space is continuous then change is realized by a predefined amount, while if it is ordinal, it moves to the next enumerated value. After sending an offer, the learning agent receives qualitative feedback from the negotiating partner and calculates the reward of its action, which is used to update the Q-value of the corresponding state-action pair. Claus and Boutilier discuss the application of Q-learning in game-theoretic cooperative setting with multiple players [58].

Benefits of the application of Q-Learning summarize to the increase of utility incurred to the agents after a number of negotiation episodes. It is empirically proved that when the counterparts had fixed strategies, the agents managed to adopt the optimal strategy, while when the former changed their strategy the agents re-adapted their strategy. However, an issue that is left open is the ability of Q-Learning technique to deal with large state-action spaces. The major weakness of this procedure is that it requires many iterations. Additionally, when Q-learning is applied during the current discourse, the agent requires his opponents' feedback in order to update the Q-values. It is not guaranteed though that the opponent will agree to engage in such protocol or that he will be truthful.

The second 'family' of explorative strategies consolidates in Genetic algorithms, optimization techniques inspired by evolution. A population of candidate solutions, encoded into chromosomes is generated and evaluated using an objective function, termed the fitness function. The best solutions are assigned the highest fitness and are combined with the use of selection, crossover and mutation techniques, to create new candidate solutions that comprise the next generation. Selection, crossover and mutation are applied with a probability, as shown in Figure 10, which describes a simple Genetic algorithm. The cycle continues until a stopping condition, usually related to a stable average fitness, is met.

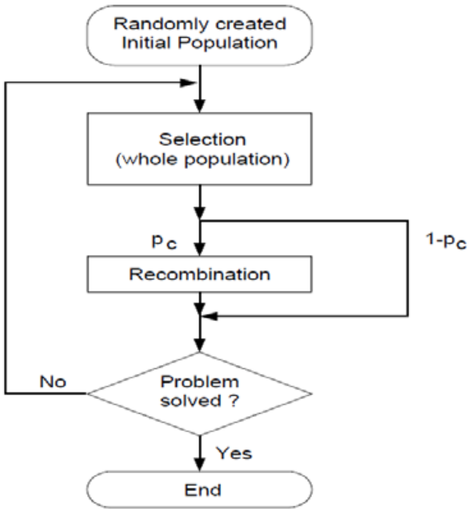


Figure 10:Genetic Algorithm

This technique is adopted by negotiating agents who seek for robust strategies. The major drawback is that it requires many iterations. Application of GAs at the planning phase is a tool that facilitates analysis of the dynamics of the interaction. It is used to search strategies that are best responses to the counterparts' best strategies, starting from random points. Oliver describes a framework where strategies are formed by simple, sequential rules that consist of acceptance thresholds and counterproposals [59]. For each negotiator a random population of strategies is generated. The testing of different strategies is repeated and the fitness of each one is determined by the utility it incurs to the agent. After a number of strategies have been tested, the genetic algorithm is run in order to generate a new population of strategies and this procedure is repeated until an exit condition is satisfied. In [60] we find application of genetic algorithms in domains where strategies are defined as a combination of tactics [25]. In such approaches the chromosomes comprise of specific strategic information such as deadlines, reservation values, weights of tactics and parameters specifying each tactic. Individuals of the population are negotiating agents (buyers and sellers), which are tested against each other (tournament), and those with the highest fitness are selected. Crossover and mutation are applied with some probability to create new individuals to the next generation of strategies. The simulations were repeated until stabilization of populations (95% of the individuals had the same fitness) or until the number of iterations reached a predefined threshold. In this work, as in [59], the concept is to determine a profile of negotiation strategies that constitute equilibrium. The system is searching for a strategy that is the best response to the best strategy of the counterpart. Gerding, van Bragt and La Poutre analyze in [61] the negotiation results achieved by GA-based agents, with respect to fairness and symmetry.

GAs, when applied in the planning phase, help to the analysis of the evolution of strategies through populations. Pair-wise rankings for buyer and seller strategy combinations lead to the most dominant strategies in each situation. Such applications of GA are not particularly interesting when viewed in a single negotiation instance. On the contrary, in cases where GAs are applied during the current discourse, populations of chromosomes are used to represent the population of feasible offers. Such application can be found in [62] where the fitness of each offer is measured with respect to its distance from the most preferred offer, the distance from the opponents' previous offer and the time pressure. In each round the offers considered fit by the agent may change. A threshold controlling the number of evolutions in a cycle, which is triggered after n negotiation rounds, determines the exit criteria. This technique aids the agent to gradually learn and adapt to its opponents' preferences. This approach does not assume knowledge of prior negotiations and it could be also applied in dynamic environments. An obvious limitation is that the algorithmic complexity increases with the increase of alternatives of each negotiable attribute.

4.1.2 Repetitive Strategies

In this category we place strategies which follow a routine-based concept; Substance of routines lays on the specific knowledge acquired by the repeated execution of an act combined with the ability to apply this knowledge to specific situations. It has the potential to substitute deliberate planning and decision making since it is used to determine which operations to implement in order to achieve certain intended state. Routinization techniques force agents to develop 'best practices'. The most commonly used is Case-based reasoning (CBR), where previously solved cases are maintained in a case base and when a new problem is encountered, the system retrieves the most similar case and adapts the solution to fit the new problem as closely as possible. CBR is common in negotiations, particularly in the planning phase supporting the process of strategy or supplier selection, or during discourse in argumentative frameworks. A commonly stated risk posed by routinization is the application of ineffective acts. Routines in dynamic environments have proved to be of degrading efficiency, the so called "acting inside the box situation". As stated by Nelson and Winter, with increasing repetitions decision making prior to the operation tends to decrease [63]. The use of routines entails rigidity and once a solution is established, it is not further questioned. Another weakness accumulates on the requirement to store the case base and the difficulty to collect the information that best discriminates different situations.

When applied at planning phase, CBR technique proceeds as follows. Each case contains information related to the agent profile and the negotiation environment, which is used as search criteria for similar cases. If more than one similar cases are returned, the negotiation outcome is used to select the most similar and preferred case. The agent then uses the strategy of the old case to the current negotiation and after negotiation is completed the case base is maintained (the new case is added to the case base or an old case is replaced).

In [64], PERSUADER, a program that acts as a labor mediator, enters in negotiation with each of the parties, the union and the company, proposing and modifying compromises until a final agreement is reached. The PERSUADER's input is a set of conflicting goals and the output is either a plan or an indication of failure. Additionally the system is capable of persuading the parties to change their evaluation of a compromise. CBR is used to keep track of cases that have worked well in similar circumstances. The most suitable case is retrieved from memory and adapted to fit the current situation. If the parties disagree, PERSUADER appropriately repairs the compromise and updates the case base or generates arguments to change the utilities of the disagreeing parties. The system integrates CBR and Preference Analysis, a decision theoretic method, to construct the initial compromise in the planning phase. If previous similar cases are not available, the PERSUADER uses Preference Analysis to find suitable compromises. Another CBR-based approach, found in [55], describes multi-sensor target tracking in a cooperative domain, where each agent controls one sensor and consumes resources (cpu, time, memory etc.). The agents are motivated to share their knowledge about the problem, based on their viewpoint, in an effort to arrive to a solution. The model uses case-based reasoning to retrieve the most similar case based on the incurred utility, adapts the case to the current situation and uses the cases' strategy to perform negotiations. It is proved that agents who use CBR method perform better compared to those using a static strategy. More specifically, agents with a static strategy had 18% worst accuracy in a cooperative framework, where they negotiated the position of a target.

At the other end, when CBR technique is applied during the current discourse, it proceeds as follows. The current negotiation stance is organized in decision making episodes, where agents propose their offers. The case base is searched to retrieve the most similar case, based on the agent profiles and series of offers and counter offers. The best matched case is then used to generate the counter offer. An application of CBR to the current discourse can be found in [65], where a support system that assists negotiators with agent opponents over used cars is implemented. The system matches current negotiation scenario with previous successful negotiation cases, and provides appropriate counter-offers for the user, based on the best-matched negotiation case. A contextual case organization hierarchy is used as an organization structure for categorizing the negotiation cases and similarity filters are used to select the best-matched case from the retrieved set of cases. Strategic moves, concessions and counter-concessions of a past discourse, are adapted to the current situation. If no case is found based on the organization hierarchy, the buyer uses a default strategy. This approach considers a single negotiable attribute, price, and does not consider learning from failure.

The virtues of repetitive strategies summarize to saving planning and decision making costs by reusing previously applied solutions. The trade-off, often termed the 'routine trap', relates to the increased risk of applying inefficient acts, if dynamics of the negotiation environment change over time.

4.1.3 Predictive strategies

The third group relates to estimating opponents' strategic parameters and preferences, as well as future behaviors, in order to select the most appropriate acts, assessed in terms of individual or joint satisfaction. The learning methods which are used to estimate the counterpart's model, strategic parameters, preferences and future behaviors summarize to possibilistic CBR, Bayesian learning, regression analysis and neural networks.

In possibilistic case-based reasoning (possibilistic CBR), agents follow principles of possibility decision theory, and use the following possibilistic rule: "the more similar the situations are, the more possible the outcomes are similar". This rule is expressed by the following formula:

$$\mu(y) = \max_{(s^i, o^i) \in H} S(s^t, s^i) \otimes P(o^i, y)$$

Where H is the history of situations s and outcomes o of previous negotiations, and S and P are similarity relations, comparing situations and outcomes respectively.

Each possible outcome y, is assigned a level of plausibility (extent to which an event may occur).This forms a possibility distribution $\mu(y)$, which is aggregated with the utility function to determine the optimal decision.

When predictive strategies are encountered in the planning phase, the agent computes the expected utility of a potential interaction and ranks his opponents in order to negotiate only with the most prosperous ones and save time and resources. In [66] the buyer agent uses possibilistic CBR to predict the outcome of a future negotiation, assuming it is in a particular situation. The situation is characterized by the negotiation strategy and the preferences of the buyer. The likelihood of successful negotiation is derived from the history of previous interactions in the form of a possibility distribution function. The expected utility of the future negotiation is an aggregate of the distribution function with the current agents' utility and is used to rank the negotiating partners. Although agents save time and resources by selecting the most prosperous opponents for negotiation, the computational complexity increases as the number of potential outcomes, which are required to acquire the possibility distribution, increases.

When it comes to using predictive strategies during the current discourse, focus lies on the estimation of opponents' strategic parameters, preferences and future offers.

A significant number of applications use Bayesian learning techniques to update beliefs about the opponents' structure. Bayes theorem provides a way to calculate the probability of an hypothesis H_i based on its prior probability $P(H_i)$, the conditional probability of various events given that the hypothesis is true, and the observed data e (new evidence). Every new event is used to update the posterior probability of hypothesis given the event according to Bayes rule:

$$P(H_i|e) = \frac{P(H_i)P(e|H_i)}{\sum_{k=1}^{n} P(e|H_k)P(H_k)}$$

Zeng and Sycara developed Bazaar [67], a negotiating system which uses a Bayesian network to update the knowledge and belief each agent has about the reservation value of his opponent. The agent holds a set of hypothesis, representing reservation prices of his counterpart, and their probabilities in his knowledge base. Domain knowledge is encoded in the form of conditional statements where events (e) are offers of the opponent in previous negotiations. Each offer in the current discourse is then used to update the subjective (posterior) probability of hypotheses, by using the Bayesian updating rule. Estimation of the opponent's reservation value contributes to approximating his payoff function and provides the agent with the ability to propose more attractive offers to his counterpart. The negotiation domain in Bazaar is rather simplified, as the authors assume a finite set of offers. Bazaar, as most systems that apply Bayesian methods, has been critiqued on the requirement of initial knowledge of many probabilities. Probability distributions of hypothesis representing potential reservation prices of the opponents, as well as domain knowledge of previous offers represented as conditional statements, constitute the prior knowledge of the system. These probabilities are estimated based on background knowledge, previously available data and assumptions about the form of the underlying distributions. Nevertheless if the distributions change, the model will no longer produce reliable estimations. To the stated weaknesses we add the fact that illustration was available only for a single attribute (price).

Other approaches based on Bayesian learning can be found in [68] where the authors present a classification method for learning opponents' preference relations during bilateral multi-issue negotiations. Similar candidate preference relations are grouped into classes, and a Bayesian technique is used to determine the likelihood that the opponents' true preference relations lay in a specific class. Negotiations are conducted over subsets of a set of objects and the goal is to increase knowledge upon the counterparts' preferences, so that an effective strategy can be devised. As the authors suggest, building an initial set of classes is a difficult task, depending on the specifics of the problem and additional information about the other party. Another work using a

Bayesian classifier can be found in [69] where agents assign probability distributions about their opponents' preference structure, in order to reduce the overall communication cost in a co-operative framework. The system suffers from the difficulty of collecting prior probabilities as all pre-mentioned Bayesian-based approaches.

Estimating opponents' strategic parameters has also been approached by statistical methods, and particularly non-linear regression.

Non linear regression involves a form of analysis in which observed data y are modeled by a function f which is a nonlinear combination of the model parameters and depends on one or more independent variables as follows:

$$y_i = f(\beta_0, \beta_1, \ldots, \beta_n; t_i) + e_i$$

The calculation of the parameters *β,* which minimize the error term, involves an iterative search process, which can be realized with a variety of function minimization algorithms (i.e. steepest descent, Gauss-Newton method, Marquardt method etc.). When the agent applies non-linear regression the main objective is to identify the counterpart's decision function, estimate strategic parameters as well as the counterpart's next offer. The agent assumes a series of models (decision functions) and selects the one that best fits the counterpart's previous offers.

Hou describes a non-linear regression-based model to predict the opponents' family of tactics and specific parameters [2]. This approach is restrictive in that it relies on the assumption of a known function form that models the concessions of the opponents. The author has assumed two non-linear functions that model time and resource dependant tactics, based on [25]. The objective is to fit the function to the opponents' previous offers, by estimating the vector of parameters that minimizes the distance of the actual offer and the estimated one. The optimization problem is dealt with an iterative method combining grid search and the Marquardt algorithm. Non-linear regression is applied in each negotiating round of the predicting agent and the authors adopt a number of heuristics to fix their prediction upon opponents' deadline and reservation value. Predicting the counterpart's deadline, allows agents to avoid negotiation breakdowns, by offering attractive deals as the deadline approaches. Additionally, agents are able to terminate unprofitable negotiations from an early round, since they can estimate the counterpart's reservation value. The authors assume the seller to be the predicting agent, and estimate the buyers' offer at the expiration of the formers' deadline. If this estimation is less than the reservation value of the seller, the buyer agent withdraws from negotiation and saves communication cost and time. Although this approach adds value to the agents, experiments are only conducted with pure strategies, where extreme behaviors are easier to distinguish.

An application of non-linear regression with mixed strategies can be found in [3]. The purpose is to predict the opponents' future offers, foresee potential negotiation threads and adopt the strategy that will result to the most beneficial discourse. The authors have developed four models to address the issue of mixed strategies that result from a combination of time and behavior dependent tactics with various weights assigned. Prediction of the counterpart's future offers, allows agents to foresee potential negotiation threads and adopt more beneficial strategies. Although this model involves more strategies than the one mentioned earlier, it does not extensively cover the space of possible strategies as discussed in [25]. The complexity is expected to increase as the number of assumed models increases, therefore extending this solution is not an easy task.

Brzostowski and Kowalczyk also take an approach based on the difference method, in order to predict the opponents' future offers [5]. This method has the advantage that the agent does not need to know precisely the opponents' strategic function. The authors assume that the opponent uses a mixture of time and behavior dependent tactics and try to determine to which extent he imitates the predicting agents' behavior and to which extent he responds to a time constraint imposed on the encounter. This is achieved with the use of two criteria combined with time depending and imitation depending predictions, obtained from the previous offers of the opponent, and from a combination of opponents' and predicting agents' offers respectively. Results have proved that the method is not as accurate as the non-linear regression and the accuracy of the weights assessments still needs improvement.

Determining the sequence of counterpart's responses has aided negotiators to identify the optimal sequence of offers. As illustrated in Figure 11, the predictive agent foresees the future offers of the opponent and assumes the same average concession up to a terminal state, in order to meet an agreement.

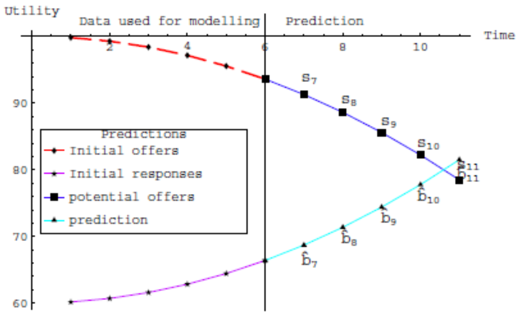


Figure 11: An example of prediction as a response to sequence of potential offers [3]

As Brzostowski and Kowalczyk state, it turns out that various sequences of offers may result to the same final utility, therefore offers are averaged over all the optimal sequences in order to calculate the offer to be proposed. After the counterpart has responded the whole mechanism is reused for further decision making in the next steps of the negotiation. Assessment of this model is provided in terms of comparison with an agent using a random strategy. In Figure 12 we illustrate an example of buyers' gain in utility.

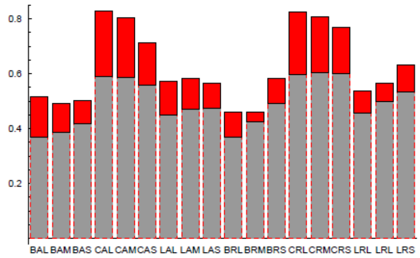


Figure 12: The gain of buyer agent over the random strategic choice is depicted with the red bars [5]

The area of predicting opponents' offers during discourse has attracted much attention, since an agent may refine his strategy and increase individual or overall gain. Other very popular models which are used for this purpose are Neural Networks. These models comprise of similar interconnected processing units (neurons) which receive signals from neighboring units or external sources, and compute an output which is propagated to other units. Connections between neurons are defined by weights which determine the extent to which a signal is amplified or diminished. Learning is realized through adjustment of such weights, in order to minimize some error function. When neural networks are used for forecasting, a window of the d+1 most recent values is used to formulate the training patterns, where the first d values represent the networks' input while the last one represents the desired output. The architecture of Neural Network models is discussed in detail in Chapter 7. When Neural Networks are applied during discourse the main objective is to forecast the counterpart's future offers. Forecasting involves either the estimation of the opponents' next offer (single-lag) or the estimation of the opponent's offers multiple steps ahead (multi-lag).

As far as multi-lag predictions are concerned, estimations have proved valuable in cases where the agents use forecasts to detect unsuccessful negotiations from an early round. Such approaches have been discussed in [11] where the decision of the agents to withdraw or not from the current negotiation is supported by determining the providers' offer before the clients' deadline expires. As the authors claim, the predictive ability has aided agents to detect up to 91.1% of unsuccessful negotiation threads and decrease the mean duration of unsuccessful discourses up to 63.8%.

Moving to the realm of single-lag predictions, previous offers and domain-specific information are used as input to the neural network in order to calculate the next offer of the opponent. This encourages more sophisticated decision making, irrespective of the type of e-market component (negotiation support system or negotiating software agent). In [8] trading scenarios via an internet platform are facilitated with the use of SmartAgent. This work illustrates a way of enhancing an automated agents' strategy with a neural network, with the purpose to predict the counterpart's next offer. The estimation of the counterparts' next move is used at each negotiation round to adjust the agents' proposal and leads to increased individual gain of the final outcome. Particularly the seller agent is enhanced with the predictive ability and its strategic core is formulated as follows:

$\mathrm{If}\ \mathrm{U}\left(\hat{\mathrm{X}}^{\mathrm{t+1}}_{\mathrm{b\to s}}\right) > U\left(\mathrm{X}^{\mathrm{t}}_{\mathrm{s\to b}}\right)$

$\mathrm{Offer} = \hat{\mathrm{X}}^{\mathrm{t+1}}_{\mathrm{b\to s}} + \varepsilon$

Else

$\mathrm{Offer} = \mathrm{X}^{\mathrm{t}}_{\mathrm{s\to b}} - \varepsilon$ (eq. 3)

where:

$\hat{\mathrm{X}}^{\mathrm{t+1}}_{\mathrm{b\to s}}$ is the estimation of the next offer of the buyer (at time t+1)

$\mathrm{X}^{\mathrm{t}}_{\mathrm{s\to b}}$ is the offer the seller would send to the buyer (at time t), based on its default strategy

ε is a domain dependent parameter

and $\mathrm{U}\left(\mathrm{X}^{\mathrm{t}}_{\mathrm{s\to b}}\right)$ is the utility (measure of satisfaction based on preferences and reservation values) of offer $\mathrm{X}^{\mathrm{t}}_{\mathrm{s\to b}}$ from the sellers' viewpoint.

We are particularly interested in the specific application of the predictive model, because it favors adaptive behavior in every step of the process. Since only preliminary results have been illustrated by the authors, we have conducted a number of experiments to highlight the gain in utility of the predictive agent.

If the rule in (eq. 3) is applied until the expiration of the agents' deadline, it is possible to develop a manipulative agent, who takes advantage of non learning agents. Figure 13(a) illustrates an example of offers exchanged between two non-learning agents, while Figure 13(b) illustrates how the predictive agent may “tease” his opponent until the time expires.

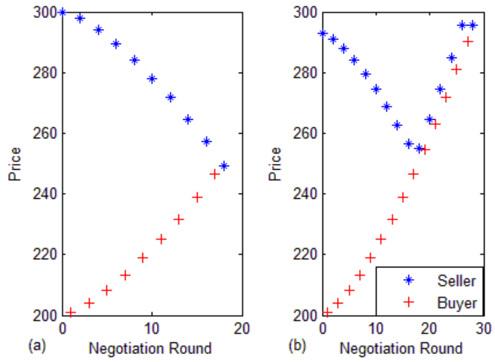


Figure 13: (a) offer exchange of two non-learning agents and (b) seller agent adjusts his price offers knowing the next move of the buyer

A similar approach is followed in [6] where a predictive model based on neural networks, with the purpose to optimize an agents’ current offer is developed. The model is incorporated in a support tool which simulates the possible response to the offer the user is contemplating and assesses offers and counter-offers based on the users’ utility function. This allows the user to test various estimated counter-offers that will result from specific offers in the current situation, without actually submitting them to the counter-part. Optimization is therefore achieved by conducting “What-if” analysis over the set of possible alternatives, and selecting the proposal that will result to the most beneficial response. The authors have tested the support tool in a domain with a small number of issues and options, thus exhaustive search could be performed to the whole set of possible offers. As the authors claim, even small variations in the current offer can have important impact on the expected counter-offer from the opponent. The model has been tested for a particular negotiation case in a static domain and the accuracy of its predictions may be less adequate in the general case. A similar negotiation support tool is applied by Lee and Ou-Yang in a supplier selection auction market, where the demander benefits from the suppliers’ forecasts, by selecting the most appropriate alternative in each round [7]. The input of the support tool comprises of past offer records and environment information such as inventory level, scheduled production plan and surplus capacity of scheduled production plan of suppliers. In addition, order quantity and due date are used to calculate the suppliers’ next bid. The authors provide an illustrative example, where three alternative bids denoting minimum, middle and maximum price are generated. The predictive model of the support tool is used to foresee the likely relationship of the current bid price of the demander and the next bid price of the supplier. This feature assists the demander to select the most appropriate from the generated alternatives.

Finally a different approach, where prediction of opponent’s next offer is carried only once during the discourse, in the pre-final round, can be found in [9]. The authors developed an agent who applies the predictive mechanism at the pre-final step of the process, in order to increase the likelihood of achieving an agreement, and to produce an outcome of maximal utility. More specifically, the authors illustrate a client agent that negotiates with a provider using a behavior-dependent strategy, and makes use of the estimation of his opponents’ next offer one step before the expiration of his deadline. The client makes the highest possible concession if the estimation is higher than his reservation price, or offers the same value as the estimation, if his reservation price is higher. The first decision increases the likelihood of achieving an agreement, while the second suggests an offer that is more beneficial (of higher utility) to the client. Assessment of the predictive agent is provided in terms of comparison with the non-predictive one. In Figure 14, the grey surface illustrates the price agreements of the non-predictive agents, while the blue and red surfaces illustrate the price agreements when predictive decision making was applied. Figure 14 also shows the increase of successful negotiations even in cases where the agreement zone is reduced and the opponents' deadline is significantly higher.

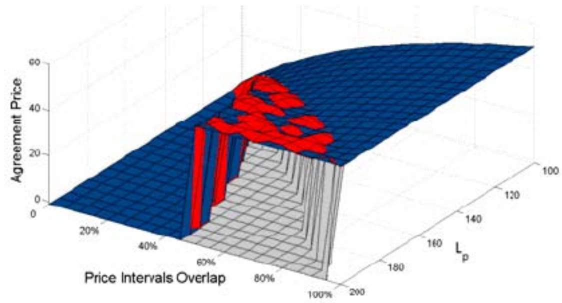


Figure 14: The use of predictive decision making increases the stances of successful negotiations [9]

Overall results for the set of 1239 experiments show that predictive agents increased the number of agreements by 38%. The weakness of current connectionist approaches summarize to the restriction of being tested solely in bounded spaces, where opponents follow static strategies, or where negotiations are conducted over fixed, pre-defined alternatives. An open and challenging issue lays in the application of predictive decision making in environments with changing data distributions.

4.2 Summary of virtues and weaknesses

This chapter provides a review of the learning methods adopted by negotiating agents who either adopt intuitive strategies or engage in predictive decision making. We aimed to provide a categorization with respect to the learning objectives, in order to facilitate comprehension of the domain. Our review has led to the discrimination of explorative, repetitive and predictive strategies applied at a pre-negotiation phase or during discourse. Under this frame we presented various systems that reflect the trends of learning in negotiation strategies, as well as the weaknesses depending on the applied domain. Virtues and weaknesses are summarized in Table 1:

Table 1: Virtues and weaknesses of learning methods

Explorative	Virtues	Weaknesses
GA **(in planning phase)**	1. Reach optimal strategy 2. Analyze negotiation interactions	Increased number of iterations due to large strategy space
GA **(during discourse)**	Adapt to opponent's responses, approach pareto-optimal solutions	Increased complexity as number of alternatives increases
Q-L	Converges in static environments	Increased complexity in dynamic

(in planning phase)		environments, as state-action pairs increase
Q-L **(during discourse)**	Adapt to opponent's responses, approach pareto-optimal solutions	Unrealistic assumption of opponents' feedback after each action, or difficulty in estimating the Q-value
Repetitive		
CBR **(in planning phase)**	Save agents from decision making costs in planning	1. The 'routine trap' 2. Maintain and search large case-base 3. Collect and identify domain-specific information to discriminate situations 4. Accuracy decreases as data distributions change
CBR **(during discourse)**	1. Decision making shortcuts in state transitions, related to concessions 2. Generation of arguments in argumentative negotiations	
Predictive		
Possibilistic CBR (in planning phase)	Estimate expected utility, facilitating supplier selection	1. The 'routine trap' 2. Maintain and search large case-base Collect and identify domain-specific information to discriminate situations 3. Accuracy decreases as data distributions change
Bayesian Learning	Estimate opponents' reservation value 1. Estimate Opponents' preference relations 2. Estimate Opponents' payoff structure	1. A-priori knowledge of many probability distributions 2. Models' accuracy reduces in dynamic environments with changing distributions
Non-Linear Regression	1. Estimate Opponents strategic parameters (reservation value, deadline, concession parameter) 2. Estimate Opponents' future offers 3. Withdraw from unprofitable negotiations	Assumes knowledge of function forms
Difference Method	1.Estimate Opponents' future offers 2.Withdraw from unprofitable negotiations	Weight assessment needs improvement, less accurate compared to non-linear regression and neural networks
Neural Networks	*Estimate Opponents' future offers* Multi-Lag Predictions: Withdraw from unprofitable negotiations Single-Lag Predictions: refine offer in each step and increase final utility	Tested in bounded domains

5. RISK OF PREDICTIVE STRATEGIES

Although Predictive Strategies add value to the field of negotiations, an issue that has not been studied in similar aforementioned work is that of risk encountered in predictive settings. The focus of this chapter is to propose a predictive strategy that takes into account the notion of risk, and allows agents to predefine the level of risk they are willing to take. In this respect we begin with a definition and a short discussion of risk in negotiations, and then proceed to the proposed approach and relevant illustrations.

5.1 Risk in negotiations

Risk is defined as a situation that involves exposure to bad outcomes. It generally increases as bad outcomes are becoming more probable. According to Dyer and Sarin an individual's preference for risky alternatives is influenced by the strength of preference he feels for the consequences (concern for outcomes) and by his attitude towards risk taking (risk tolerance) [70]. There is a line of work that attempts to measure risk as a basic attribute of a lottery (a hypothetical game). The different attitudes towards risk, risk-neutral, risk-averse and risk-seeking, are usually represented by the utility function of the decision makers.

Decision makers who adopt the risk-averse attitude prefer the sure deal to the risky option. Richard characteristically states that such decision makers prefer getting some of the "best" or some of the "worst" to taking a chance on all of the "best" or all of the "worst" [71]. In a lottery that offers different outcomes, for example a 50-50 chance of receiving either 0$ or 100$, or a guaranteed amount of 40$, the risk averse agent would prefer the guaranteed amount of 40$. The utility function in this case is concave, thus the marginal utility of wealth decreases as wealth increases (each additional 1$ contributes less utility than the one before it). The property of risk aversion has its basis on the principle of expected utility maximization, which states that the rational investor will select the alternative that maximizes his expected utility of wealth [72].

Conversely, decision makers who adopt the risk seeking attitude prefer the risky option to the sure deal. The utility function is convex, thus the marginal utility of wealth increases as wealth increases.

Intuitively, a risk-seeking individual is the one who prefers taking chances, while a risk-averse individual behaves conservatively in the face of risk. Last, a risk-neutral decision maker equally prefers the sure deal to the risky option.

In behavioral approaches, there is a line of work apposed in [35] that associate a negotiator's attitude towards risk with his outcome frame, that is his conception of the dispute as positive, involving gains and profits (gain frame) or as negative, involving losses and costs (loss frame). More specifically negotiators with a gain frame demand less, concede more and settle more easily than those with a loss frame. The former tend to adopt a risk-averse attitude, while the latter tend to adopt a risk-seeking attitude. Additionally losses are more aversive than equivalent gains are attractive. As Schneider states 'people have a stronger desire to minimize losses than to maximize gains' [73].

A negotiation strategy that takes into account an agent's attitude towards risk, the Zeuthen Strategy, is also discussed in [21]. In Zeuthen Strategy the notion of risk is related to an agent's willingness to risk conflict. More specifically Risk $R_t^a : A^a \rightarrow [0,1]$, where A^a is the space of possible actions of agent a, is defined as the utility loss of the agent if he accepts his counterpart's offer, divided by his utility loss if they do not agree and negotiation terminates with the conflict deal (a deal specified at a pre-negotiation stage). As R_t^a approaches 1, agent has less to lose from a conflict and is more willing to stay in negotiation and not concede. Conversely as R_t^a approaches 0, agent is more willing to make a concession. Intuitively, as R_t^a grows agent becomes more risk-seeking.

Since negotiating agents may adopt different attitudes towards risk, it is important to take such attitudes into account when generating predictive strategies. In this respect, a risk-related parameter is embodied in the strategy of learning agents, as described in the next section.

5.2 The proposed predictive strategy

If the two agents do not employ any learning technique, and each applies the default strategy, as illustrated in [25], an agreement will be established at a point which we term the “Meeting Point” (MP). The proposed predictive strategy is based on the assumption that in each decision making step, if the negotiators decide to send a counter-offer, the risk-averse agent will adopt a conservative behavior and generate an offer that will be accepted by his counterpart, while the risk-seeking agent will provoke gradual concessions of his counterpart, so as to increase his individual gain. The main objective of the proposed predictive strategy is to prolong negotiation beyond MP and increase the incurred utility of the agent, by taking into account the two extreme attitudes. In this respect, the agent pre-defines a reference point that is related to his willingness to prolong the discourse. He adopts the risk seeking behavior until that reference point and the risk-averse behavior from the reference point until expiration of his deadline or termination of the discourse. This behavior is illustrated in Figure 15.

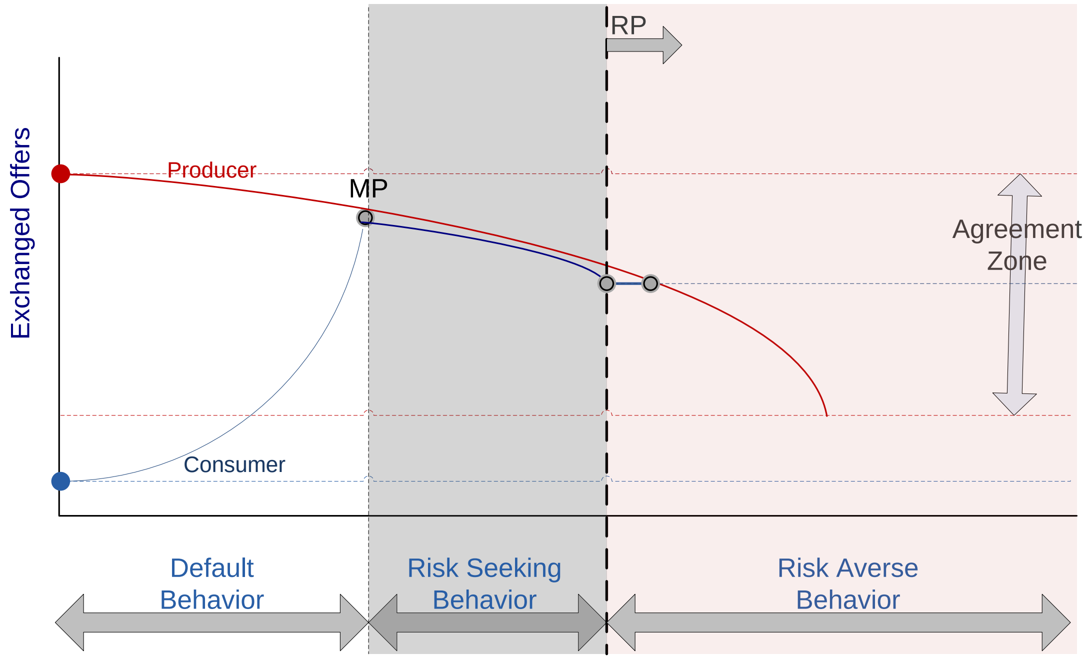


Figure 15: The predictive strategy

At this point it is essential to distinguish the different usages of the predictive skill. When agents employ a predictive strategy as the one described in [8], they run into the danger of prolonging the negotiation process, and increase the risk the other party exhausts his deadline and walks out of the process. Prolongation is due to the fact that the counterpart tends to respond with a counter-offer to the predictive agent, which is the case of a risk-seeking attitude. On the other hand, when agents employ a predictive strategy at the pre-final step of the process, as the one discussed in [9] [10], they manage to increase the number of successful negotiations if the counterpart has equal or higher deadline than the predictive agent. In this second case, the predictive agent sends an offer that is likely to be accepted by the counterpart, which is the case of a risk-averse behavior.

The reformed strategy illustrated in [74] [75], combines the virtues of the two strategies with the introduction of a parameter noted risk portion (RP). The proposed strategy is a predictive strategy that allows agents pre-specify in percentage terms how much they are willing to prolong the negotiation process in order to achieve a more satisfying outcome compared to the outcome they would achieve in the non-learning case.

In subsection 5.2.1 the predictive strategy is described, and in subsection 5.2.3 the RP parameter is used to analyze the different negotiation outcomes, taking into account opponent agents employing various types of behaviors. The objective of this section is to illustrate the trade-off of increasing the utility of agreements with the number of successful negotiations. In subsection 5.2.4 we discuss the issue of setting appropriate values to the RP parameter.

5.2.1 Description of the strategy

At each time step t agent α estimates the next offer of his counterpart, $\hat{X}^{t+1}_{b\to a} = \left(\hat{x}^{t+1}_{1(b\to a)}, \hat{x}^{t+1}_{2(b\to a)}, ..., \hat{x}^{t+1}_{n(b\to a)}\right)^T$. The proposed decision rule makes use of the default strategy (S^a) of the predictive agent to generate offers until the detection of a "meeting point" (MP) with the "opponent". MP is a point which would result an established agreement if the agent was guided solely by his default strategy. When such point is detected, and according to the agent's attitude towards risk, agent risks staying in the negotiation in order to maximize the utility of the final agreement. In this respect two extreme attitudes can be generated: risk-seeking and risk-averse. The risk-seeking agent is willing to spend all the remaining time until expiration of his deadline engaging in an adaptive behavior to turn the estimations of his counterpart's responses to profit. This risk-seeking behavior is based on the decision rule discussed in [8] and is extended to support multiple issues. More specifically:

<u>Risk-seeking Behavior:</u>

For each issue i

If *issue value is increasing with time*

$$x^t_{i(a\to b)} = \hat{x}^{t+1}_{i(b\to a)} - \varepsilon$$

Else

$$x^t_{i(a\to b)} = \hat{x}^{t+1}_{i(b\to a)} + \varepsilon$$

End For

Generate Offer $X^t_{a\to b} = \left(x^t_{1(a\to b)}, x^t_{2(a\to b)}, ..., x^t_{n(a\to b)}\right)^T$

where ε is a domain dependent parameter.

On the other hand risk-averse agents follow a more conservative behavior when they detect an MP. They use the prediction as discussed in [9] [10] and thereafter do not make any further concessions and insist on sending their previous offer, waiting for the opponent to establish an agreement.

Risk-Averse Behavior:

When MP is detected:

For each issue i

$x^t_{i(a\to b)} = \hat{x}^{t+1}_{i(b\to a)}$

End For

If t>MP:

For each issue i

$x^t_{i(a\to b)} = x^{t-2}_{i(a\to b)}$

End For

Generate Offer $X^t_{a\to b} = \left(x^t_{1(a\to b)}, x^t_{2(a\to b)}, \ldots, x^t_{n(a\to b)}\right)^T$

Fusions of the two extreme attitudes have led to the specification of risk portions (RPs) which characterize the predictive agent's behavior after the detection of MP. As shown in Figure 15, RP^a determines the percentage of the distance between MP and deadline $T^a_{\max}$ that agent α is willing to adopt the risk-seeking behavior. After RP^a is consumed agent adopts the risk-averse behavior. For a predictive agent who is not willing to take any risks RP^a is set to 0%, while for an agent who is willing to risk until expiration of his deadline RP^a is set to 100%. The decision making rule repeated in each step is thus formulated as follows:

If $U^a\left(\hat{X}^{t+1}_{b\to a}\right) > U^a\left(X^t_{a\to b}\right)_{default}$ (*detection of MP*)

If RP^a is not consumed

Generate Offer adopting Risk-Seeking Behavior

Else

Generate Offer adopting Risk-Averse Behavior

Else

Generate Offer $\left(X^t_{a\to b}\right)_{default}$

where $\left(X^t_{a\to b}\right)_{default}$ is the offer generated by agent α at time t based on his default strategy.

In the following example we consider negotiations conducted between an electricity provider and a consumer agent over the service terms of an electricity trade. The negotiable object is characterized by four attributes representing the number of Kwh, the Price per Kwh (measured in euro cents), the Penalty term (percentage of the sum which will be returned to the consumer in case of dissatisfaction), and the duration of service provision (measured in hours). The deadline of the consumer is set to 150 rounds and that of the producer is set to 152 rounds. We assume that the two agents have opposing interests; the consumer will start from a low price and a low number of Kwh, which he will increase in each round, while he will start from a high percentage of returns and high service duration which he will decrease in each round. At the same time the producer will initially request high price per Kwh, and high number of Kwh which he will lower in each round, and low penalty and duration of service provisioning which he will increase in each round. In the example we assume that the two agents have set the same reservation (min and max) values as demonstrated in Table 2.

Table 2: The reservation values of the negotiable attributes

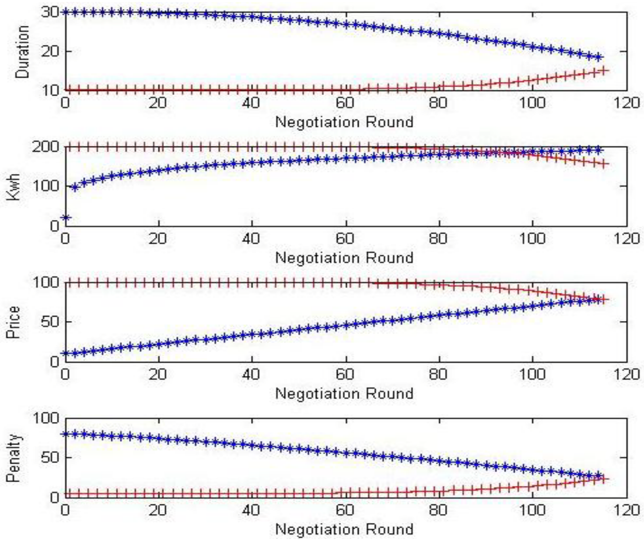
Attribute	Min Value	Max Value	Consumer Role	Provider Role
Duration (in hours)	10	30	Decreasing	Increasing
Kwh	20	200	Increasing	Decreasing
Price (euro cents)	10	100	Increasing	Decreasing
Penalty (% returns)	5	80	Decreasing	Increasing

Figure 16(a) illustrates the negotiation discourse of the two agents when they do not employ any learning techniques. The offered values of each negotiable attribute in each round are depicted with blue for the consumer and with red for the producer. In the non-learning case (Figure 16(a)), negotiation terminates at round 116, where the consumer agent decides to accept his counterpart's offer. The final offer vector (Duration, Kwh, Price, Penalty) is (14.95, 155.36, 77.68, 23.59) and the utility incurred to the consumer agent is 0.248.

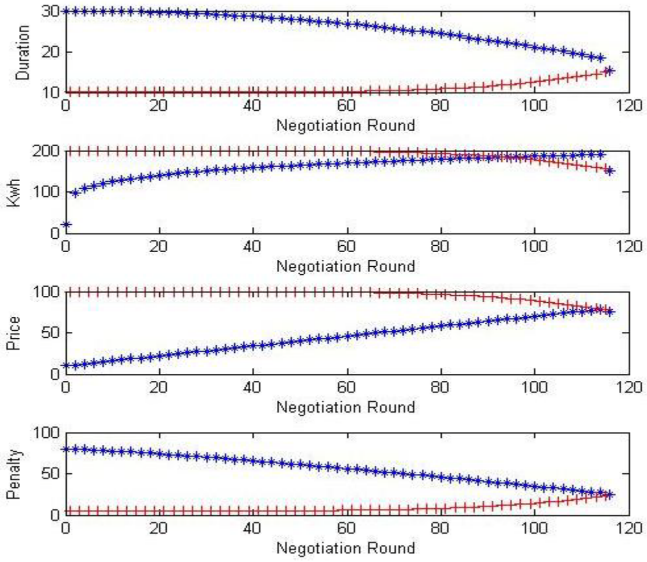
Figure 16(b) illustrates a discourse where the consumer agent applies the proposed strategy with RP=0%. At round 116, the consumer agent detects the meeting point (MP) and initiates the predictive behavior. Since RP is set to 0% (it is consumed upon detection of MP), the agent generates the risk-aversive behavior and sends the predicted offer to his counterpart. Negotiation terminates at round 117, where the provider decides to accept the consumer's offer. The final offer vector (Duration, Kwh, Price, Penalty) is (15.40, 151.34, 75.67, 25.27) and the utility incurred to the consumer agent is 0.2703. Note that in case RP=0%, the predictive strategy incurs 2.23% absolute increase in utility, without much prolongation of the negotiation discourse. The maximum prolongation of the discourse when the risk-averse behavior is applied is by 1 round (the counterpart will accept the predictive agent's proposal in the next round).

In continuance, in Figure 16(c) we illustrate a discourse with a consumer agent who uses the proposed strategy with RP=50%. At round 116, the consumer detects the MP and adopts the risk-seeking behavior, until round 134 where the RP is consumed. The offer sent at round 134 is based on the risk-averse behavior and the offer vector (Duration, Kwh, Price, Penalty) = (20.66, 101.39, 51.30, 45.42) is accepted by the provider at round 135. The utility incurred to the consumer in this case is 0.5403, thus the absolute increase compared to the non-learning case is 29.23%.

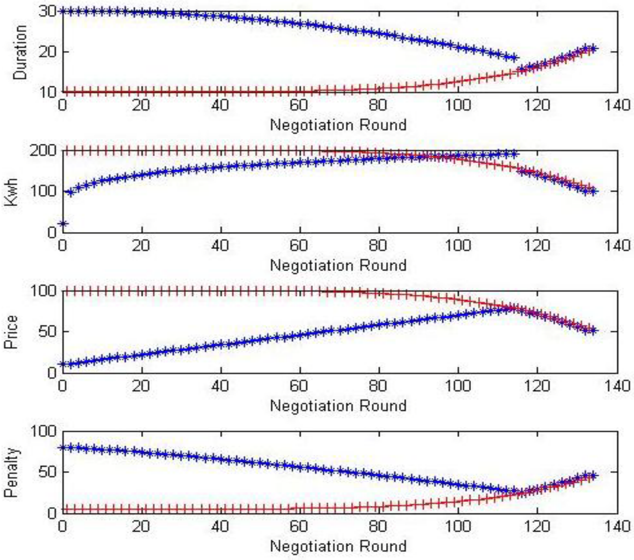
Finally we demonstrate how a predictive agent with RP 100% may “tease” his opponent until an agreement is established. The consumer agent makes use of his default strategy until round 116, where the meeting point MP is detected. The agent risks staying in negotiation after round 116, and makes use of the risk-seeking behavior until exhaustion of his deadline, in order to attain a more eligible deal. At round 150, where RP is consumed, the consumer sends his final offer based on the risk-aversive behavior. The final offer vector (Duration, Kwh, Price, Penalty) = (29.35, 25.84, 12.92, 77.56) is accepted by the provider. The utility incurred to the consumer in this case is 0.9675, yielding 71.95% absolute increase in utility compared to the non-learning case. The discourse when the consumer's RP is set to 100% is illustrated in Figure 16(d).



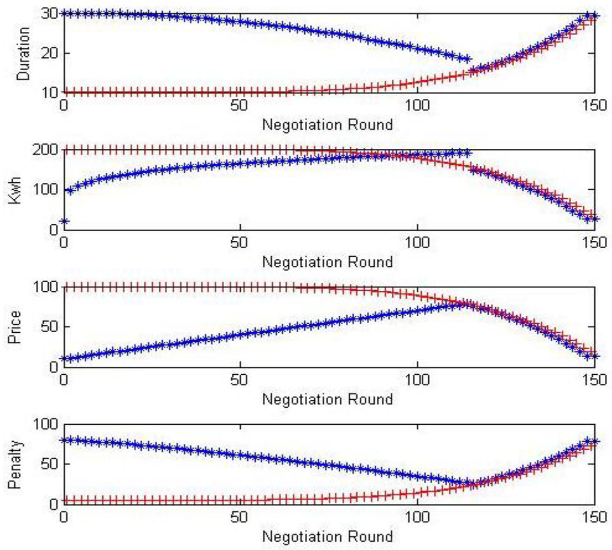
(a)



(b)



(c)



(d)

Figure 16: (a): Negotiation between two non learning agents, (b): Negotiation with a consumer agent employing the proposed strategy with RP=0%, (c): Negotiation with a consumer agent employing the proposed strategy with RP=50%, (d): Negotiation with a consumer agent employing the proposed strategy with RP=100%

5.2.2 Rationality of the two extreme behaviors

In the proposed strategy the risk-averse agent sends an offer that increases the probability his counterpart will accept, while the risk seeking agent sends an offer that increases the probability his counterpart will respond with a counter-offer. In this section the rationality of the two extreme behaviors is examined. By definition, for two agents a and b with opposing interests, the following two rules apply:

Rule 1: *If each attribute* $x_{i(a\to b)}^{t_n}$ *of agent's* a *selected counter-offer* $X_{(a\to b)}^{t_n}$ *lies between the respective attribute* $x_{i(b\to a)}^{t_n-1}$ *of the counterpart's previous offer* $X_{(b\to a)}^{t_n-1}$ *and the respective* *attribute* $\hat{x}^{t+1}_{i(b\to a)}$ *of the estimation* $\hat{X}^{t_{n+1}}_{(b\to a)}$ *, agent* b *will accept* $X^{t_n}_{(a\to b)}$ *if he has not reached his deadline (*$t_n < T^b_{\max}$*).*

Rule 2: *If each attribute* $x^{t_n}_{i(a\to b)}$ *of agent's* a *selected counter-offer* $X^{t_n}_{(a\to b)}$ *surpasses each respective attribute* $\hat{x}^{t+1}_{i(b\to a)}$ *of the estimation* $\hat{X}^{t_{n+1}}_{(b\to a)}$*, agent* b *will not accept the offer, but rather propose counter-offer* $X^{t_{n+1}}_{(b\to a)}$ *if he has not reached his deadline (*$t_n < T^b_{\max}$*).*

As stated in [72], the rational agent should try to maximize his expected utility of wealth. In the general case, if we consider two agents a and b who engage in a negotiation discourse, the negotiation thread at time t_n, formulated by the exchanged offers, is $X^{t_n}_{a\leftrightarrow b} = \{\ldots, X^{t_{n-2}}_{a\to b}, X^{t_{n-1}}_{b\to a}\}$ and it is agent's a turn to make a move. Supposing that agent a is a predictive agent, his knowledge at time t_n consists of the negotiation thread $X^{t_n}_{a\leftrightarrow b}$ and the prediction of his counterpart's next offer $\hat{X}^{t_{n+1}}_{b\to a}$. According to (eq. 2 in section 3.3) the agent may accept his counterpart's proposal, reject the proposal and terminate the process, or send a counter-offer. Since rejection is only selected when the time deadline set by the agent is expired, if $t_n < T^a_{\max}$, agent a must decide either to accept $X^{t_{n-1}}_{b\to a}$, or to send a counter-offer $X^{t_n}_{a\to b}$.

The first option would result a guaranteed deal, and the expected utility of the agent would be $U^a(X^{t_n-1}_{(b\to a)})$. The second option would trigger the following reactions of agent b :

$$C^b_{t_{n+1}} = \begin{cases} Quit, if\ t_n > T^b_{\max} \\ Accept\ X^{t_n}_{(a\to b)}, if\ U^b(X^{t_n}_{(a\to b)}) \ge U^b(X^{t_n}_{(b\to a)}) \\ Send\ X^{t_{n+1}}_{(b\to a)}, otherwise \end{cases} \quad (4)$$

If $P_{Quit}, P_{Accept}, P_{Send}$ are the probabilities of agent b quitting negotiation, accepting $X^{t_n}_{(a\to b)}$ and sending counter-offer $X^{t_{n+1}}_{(b\to a)}$ respectively, agent's a expected utility if he selects to send counter-offer $X^{t_n}_{a\to b}$ is: $P_{Accept} * U^a(X^{t_n}_{(a\to b)}) + P_{Quit} * U^a(BATNA) + P_{Send} * U^a(X^{t_{n+1}}_{(b\to a})$, where BATNA is defined as the best alternative to negotiating agreement, which results to zero utility ($U^a(BATNA) = 0$). The rational agent will prefer to send counteroffer $X^{t_n}_{(a\to b)}$, if the expected utility of sending the counteroffer is higher than the expected utility of the guaranteed deal (which results from accepting his counterpart's proposal), thus the agent will send $X^{t_n}_{(a\to b)}$ if:

$$P_{Accept} * U^a(X^{t_n}_{(a\to b)}) + P_{Send} * U^a(\hat{X}^{t_{n+1}}_{(b\to a)}) > U^a(X^{t_n-1}_{(b\to a)}) \quad (5)$$

The risk-averse behavior discussed in 5.2.1, is consistent with Rule 1, as $X^{t_n}_{(a\to b)} = \hat{X}^{t_{n+1}}_{(b\to a)}$.

In this case, the probabilities $P_{Quit}, P_{Accept}, P_{Send}$ are formulated as follows:

$$P_{Send} = 0$$

$$P_{Accept} = 1 - P_{Quit} \quad (6)$$

From (5) and (6) the condition that must be satisfied so that risk-averse agent a rather sent a counter-offer is the following:

$$P_{Accept} * U^a(X^{t_n}_{(a\to b)}) > U^a(X^{t_n-1}_{(b\to a)}) \quad (7)$$

If $P_{Accept} \to 1$ or $P_{Quit} \to 0$, the inequality (7) is satisfied by definition.

Moving to the other end, the agent with a risk-seeking attitude would rather take the all or nothing deal. This agent makes an offer $X^{t_n}_{(a\to b)}$ that will provoke his opponent to respond with a counter-offer rather than accept the deal. This will result to gradual concessions from the side of the non-learning agent, and the final deal will yield very high utility to the predictive agent. This behavior is consistent with Rule 2, as the predictive agent sends an offer that surpasses the estimation by the constant ε. In this case the probabilities P_{Quit}, P_{Accept}, P_{Send} are formulated as follows:

$$P_{Accept} = 0$$

$$P_{Send} = 1 - P_{Quit} \quad (8)$$

And the condition of applicability is formulated as follows:

$$P_{Send} * U^a(\hat{X}^{t_{n+1}}_{(b\to a)}) > U^a(X^{t_n-1}_{(b\to a)}) \quad (9)$$

If $P_{Send} \to 1$ or $P_{Quit} \to 0$, inequality in (9) is satisfied by definition.

From the above it is proved that both behaviors satisfy the rationality condition if $P_{Quit} \to 0$.

5.2.3 An illustration

This section is attributed to the investigation of the effect the strategy described in 5.2.1 has on the negotiation outcome. Since the objective is to increase the utility that incurs to the predictive agent, focus is set on studying the change of the agent's utility, as well as the change of the number of agreements with variable RP values. For this reason a number of experiments are conducted assuming negotiations between a predictive agent with the perfect forecasting tool (yielding zero error) who makes very accurate estimations and a non-learning counterpart employing many different types of time-dependent behaviors. In the experiments conducted, the strategy of the counterpart is known to the predictive agent, who simply applies the expected values of the counteroffers to the decision rule discussed in 5.2.1. In subsection 5.2.3.1 we give a brief description of the simulator which produces negotiation environments and outcomes and in subsection 5.2.3.2 we illustrate the results.

5.2.3.1 Simulator

For the conduction of the experiments we have developed a simulator that produces negotiator objects in Java (Jdk version 1.6), which are then extended in Matlab (version 2008R) and enhanced with learning techniques. The negotiator objects are capable of conducting bilateral multi-issued negotiations. Experiments involve the generation of different negotiation environments, with provider and consumer agents, described as follows:

$$Negotiation\,Environment = \{T^{\Pr}_{\max}, T^{Con}_{\max}, S^{\Pr}, S^{Con}, \min^{\Pr}, \min^{Con}, \max^{\Pr}, \max^{Con}, W^{\Pr}, W^{Con}\}$$

Where $T_{\max}^{a}$ is the negotiation deadline, S^{a} is the strategy, $\min^{a} = \left(\min_1^a, \min_2^a, \ldots, \min_n^a\right)^T$ and $\max^{a} = \left(\max_1^a, \max_2^a, \ldots, \max_n^a\right)^T$ are the vectors with the minimum and maximum values set for each issue respectively, and $W^{a} = \left(w_1^a, w_2^a, \ldots, w_n^a\right)^T$ is the vector with the preference weights for each issue set by agent a.

A variable Φ is used to describe the degree of intersection between the negotiation intervals of the two agents. More specifically, $\Phi = \left(\Phi_1, \Phi_2, \ldots, \Phi_n\right)^T$, where each Φ_i є [0, 0.99][1] indicates the overlapping degree of ranges specified for issue i by the two agents.

The attributes of the Negotiation Environment as well as the learning parameters when agents employ predictive strategies constitute the simulators' input. The utility functions of the negotiators are linear and are computed with respect to each agent's reservation values. The simulator's output consists of the utilities incurred to each agent, the time and number of iterations of the discourse, as well as the analytic negotiation thread. Simulators' input and output data is illustrated in Figure 17.

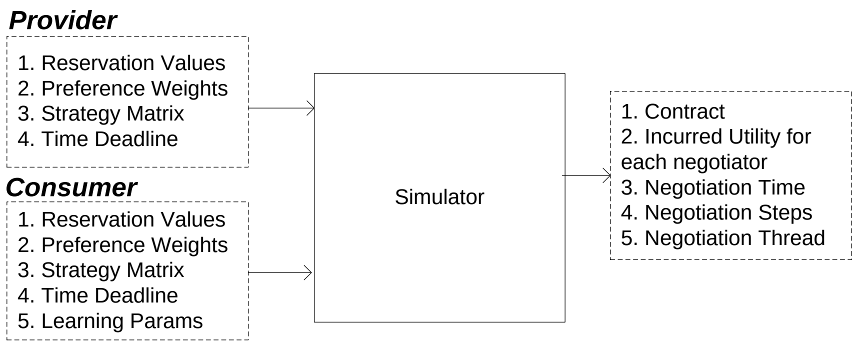


Figure 17: Simulator's input and output data (Consumer employs a predictive strategy)

The negotiator objects implemented for the simulator can perform learning tasks, and may engage in negotiations following the predictive strategy discussed in section 5.2.1. In the following subsection we provide information concerning the parameter values used in the experiments, as well as the results of the negotiations with respect to the different risk attitudes.

5.2.3.2 Results

The proposed strategy is tested with consumer agents assumed to have perfect predicting skills and providers following TD strategies. The experimental workbench issues nine different scenarios with respect to deadline, and overlap of agreement zones of the two negotiators ($\{T_{\max}^{Con} = T_{\max}^{\Pr}, T_{\max}^{Con} < T_{\max}^{\Pr}, T_{\max}^{Con} > T_{\max}^{\Pr}\}$ × {Φ=0, Φ=0.33, Φ=0.66}), where $T_{\max}^{a} \in$ [50:100:350], α={Con,Pr}, and Φ is the parameter that indicates the overlap of the agreement zones. In each scenario a variety of concession curves, defined by parameter β = {0.2, 0.5, 0.8, 1, 3, 5, 7}, is considered in order to build the default strategies of the opposing agents. For each of the 2352 generated negotiation environments different RPs are set to the predictive agent (consumer) (

[1] Full overlap is equivalent to Φ=0, while almost non-overlapping regions are equivalent to Φ=0.99

RP^{Con} $\in$ [0:5:100]), leading to an overall of 49392 experiments. The objective is to measure the gain of consumer agent with respect to the RP parameter, and highlight the value of forecasting counterpart's next offer in multi-issued negotiations. The above settings are illustrated in Table 3.

Table 3: Negotiation Settings

Overlap :	**Φ=0**		**Φ=0.33**		**Φ=0.66**	
Parameters	**Consumer**	**Provider**	**Consumer**	**Provider**	**Consumer**	**Provider**
Kwh(min)	20	20	20	79.4	20	138.8
Kwh(max)	200	200	200	259.4	200	318.8
Price(min)	10	10	10	39.7	10	69.4
Price(max)	100	100	100	129.7	100	159.4
Penalty(min)	5	5	5	29.75	5	54.5
Penalty(max)	80	80	80	104.75	80	129.5
Duration(min)	10	10	10	16.6	10	23.2
Duration(max)	30	30	30	36.6	30	43.2
$T^{\mho}_{max}$	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]
$S^{\mho}$	TD β=[0.2,0.5,0.8, 1,3,5,7]	TD β=[0.2,0.5,0.8, 1,3,5,7]	TD β=[0.2,0.5, 0.8, 1,3,5,7]	TD β=[0.2,0.5,0.8, 1,3,5,7]	TD β=[0.2,0.5, 0.8, 1,3,5,7]	TD β=[0.2,0.5,0.8,1,3,5, 7]
RP^{Con}	[0:5:100]		[0:5:100]		[0:5:100]	

From a total of 2352 negotiation environments, average utilities of negotiations conducted between non-learning agents (AvgUtil_NL) and between predictive and non-learning agents employing different RP values (AvgUtil_L$_{(RP)}$) are computed. Since utilities are specified in range [0,1], the average absolute increase in utility incurred to the agent who employs the proposed strategy is computed as follows:

$$\text{AvgAbsInc}_{(\text{RP})} = (\text{AvgUtil\_L}_{(\text{RP})} - \text{AvgUtil\_NL}) * 100\%$$

Figure 18 depicts the average absolute increase for each RP value.

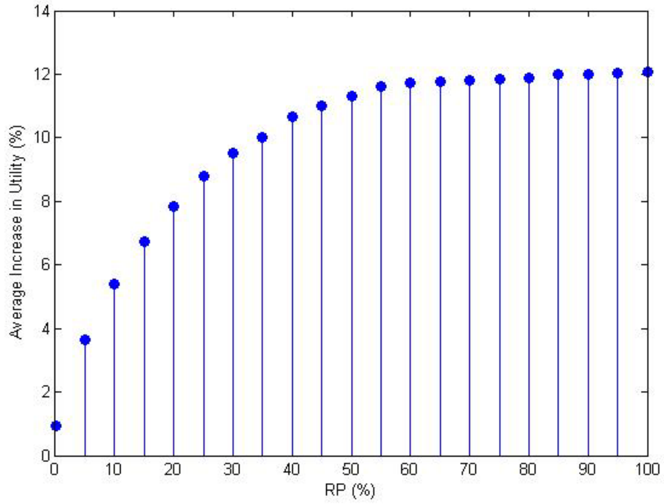


Figure 18: (%) Average gain in Utility with respect to RP

As it is shown in Figure 18 an agent with RP 0% incurs an average increase of 0.94% in utility, while an agent with RP 100% incurs an average increase of 12.05% in utility. Additionally, the concavity of the curve shows that the marginal increase of the utility decreases as RP grows. For different values of RP>60%, the increase in utility does not change significantly.

However employment of the predictive strategy increases the time (number of negotiation rounds) of the discourse, as the learning agent continues the negotiation after detection of the meeting point (MP). The time consumed, TCons, in each negotiation round is normalized thus: $TCons = \frac{NegotiationRounds}{\min(T_{\max}^{Con}, T_{\max}^{\Pr})}$. For the total of 2352 negotiations between non-learning agents the average consumed time (AvgCons_NL) is computed. For each RP value the average consumed time $\text{AvgCons\_L}_{(RP)}$ is also computed. Finally the average absolute increase of the time consumed by the agent who employs the proposed strategy is derived as follows:

$$\text{AvgAbsCons}_{(RP)} = (\text{AvgCons\_L}_{(RP)} - \text{AvgCons\_NL}) * 100\%$$

Figure 19 illustrates the increase of negotiation time with respect to the risk portion.

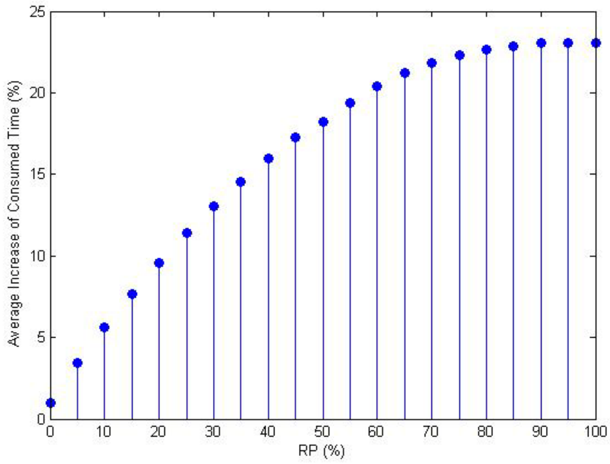


Figure 19: (%) Average Increase of Negotiation Time with respect to RP

As it is shown in Figure 19 the average increase of negotiation time is 0.96% when RP is set to 0% and 23.07% when RP is set to 100%.

Prolonging the negotiation procedure is the main cause of decrease of negotiating agreements, as it likely that the counterpart reaches his deadline and terminates the process. From a total of 2352 negotiations conducted between non-learning agents the number of agreements (Num_NL) is computed. The number of agreements of negotiations conducted between predictive and non-learning agents for each RP value ($Num\_L_{(RP)}$) is also computed. The average relative decrease of the number of established agreements by the agent who employs the proposed strategy is derived as follows:

$$AvgDecAgreement_{(RP)} = \frac{Num\_NL - Num\_L_{(RP)}}{Num\_NL} * 100\%$$

Figure 20 illustrates how the number of agreements is affected by the adoption of the predictive strategy.

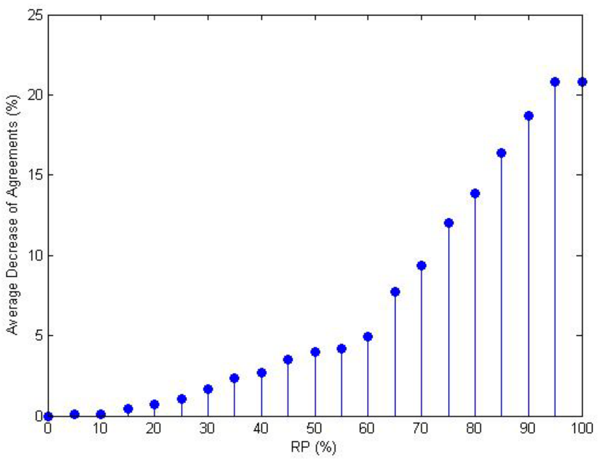


Figure 20: (%) Average Decrease of the number of agreements with respect to RP

More specifically, when RP is set to 0% the number of agreements is not decreased, while when RP is set to 100%, the number of agreements shows an average decrease of 20.78%.

From the experiments conducted, it is proved that as RP value increases the agent may increase his individual utility. For each negotiation environment experiments were conducted for the 21 predefined RP values. However, some of the selected RPs did not necessarily satisfy the rationality condition ($P_{Quit} \to 0$), and this led to negotiation breakdowns. As it is observed, failing to establish an agreement increases with increasing RP value, as the counterpart reaches his deadline before the predictive agent adopts the risk-averse behavior. Setting the appropriate RP values is crucial for a predictive agent who wishes to attain the maximum possible utility gain and avoid negotiation failures, as we discuss in the following subsection.

5.2.4 Setting appropriate RP values

Selecting appropriate RP values can help adopt rational behavior and therefore avoid risks of failures. Appropriate selections can be made if estimations of the opponent's deadline are also available. In case opponent's deadline is shorter than player's deadline, RP can be set just before the expiration of the opponent's deadline. The predictive agent can make use of the risk-seeking behavior until that point and then employ the risk-averse behavior, which will result in acceptance from the counterpart, before the expiration of the latter's deadline.

In order to set appropriate RPs, if an estimation of the counterpart's deadline $\hat{T}^b_{\max}$ is available, predictive agent a should distinguish the two cases: If $\hat{T}^b_{\max}$ falls in agent's a turn, then he should adopt the risk-averse behavior at round $\hat{T}^b_{\max}$ - 2. On the contrary, if it falls in agent's b turn, then agent should adopt the risk-averse behavior at round $\hat{T}^b_{\max}$ - 1. In the first case RP can be set upon detection of MP as follows:

$$RP = \frac{(\hat{T}^b_{\max} - 2) - MP}{T^a_{\max} - MP} * 100\% \quad (10)$$

While in the second case RP can be set upon detection of MP as follows:

$$RP = \frac{(\hat{T}^b_{\max} - 1) - MP}{T^a_{\max} - MP} * 100\% \quad (11)$$

Equations 10 and 11 apply if opponent's deadline is shorter than the player's deadline. Contrarily, the predictive agent can set RP to 100%. In this respect the predictive agent can attain the maximal possible increase of his individual utility by adopting the risk seeking behavior after detection of MP, and the risk-averse behavior only in the final round. Estimating the counterpart's deadline is beyond the scope of this thesis. However, to assess the extension with the appropriate RP values, we have assumed knowledge of the opponent's deadline by the predictive agent. The proposed strategy with the appropriate RP values when applied to the experimental settings discussed in 5.2.3.2 (2352 cases), yields very satisfying results as the average absolute increase in utility is 12.017%, the average absolute increase of the consumed time is 22.53%, and the average relative decrease of the number of agreements is reduced to 0.61%.

Another issue that needs to be clarified is that the *RP value does not measure risk in absolute terms*. Agents who negotiate under different negotiation settings (with different preferences, deadlines or opponents) and who have set the same RP value do not necessarily take the same risks. To make this issue clearer, suppose agent a with a deadline at 500 and RP of 50%. If the agent reaches MP at round 450, he will continue with the risk seeking behavior for 25 more steps and then adopt the risk-averse behavior. If the same agent negotiates with a different opponent and reaches MP at round 100, he will adopt the risk-seeking behavior at round 200 and then the risk-averse behavior. The risk taken when employing the risk-seeking behavior after round 450 is not the same with the risk taken when employing the risk-seeking behavior after round 100, since the likelihood that the counterpart reaches his deadline is greater in the first case. On the other hand *when agents negotiate under the same settings* RP can be used to compare their risk attitudes (i.e. we can say that agent a is more risk-averse than agent b if he has a lower RP value).

6. FORECASTING TOOLS

In the previous chapters we have provided an overview of negotiation strategies enhanced with learning techniques, and we have presented a strategy taking into account the different attitudes towards risk, considering agents with perfect predictive skills. In this chapter focus is set on the type of learning model employed by the agents, for the purpose of predicting the counterpart's next offer. In this respect, we provide a classification of forecasting tools, and provide a comparison over those that have been used in negotiations.

6.1 A Classification

Forecasting is a similar but rather less general concept compared to predicting future events, and involves the estimation of the expected value of a variable in a future time. Forecasting is widely applied in business and economics, with the scope to estimate outcomes, trends, cycles, seasonality, randomness and autocorrelation of economic and business data.

An early classification of forecasting methods makes a distinction between qualitative and quantitative models. Qualitative models are judgmental approaches that base forecasts on the observation of current trends, and identify systematic changes more quickly. Among these are Executive Opinions, Delphi, Sales-Force polling, Consumer Surveys, Sales Forecast, Grass Roots, Market Research and Historical Analogy. Quantitative models have their basis to statistics and mathematics and are further categorized to simulation, cause and effect and time-series models.

Simulation methods involve the use of analogs to model complex systems. These analogs can take on several forms. In the case of negotiations game analogs are used, and the interactions of the players are symbolic of social interactions.

Time-series models include all models in which future values are predicted on the basis of analysis of past series of data. Methods used in the complexity of past data analysis are classified to univariate methods, in which a variable is predicted solely from its past values and multivariate methods, in which other variables are also accounted. Another discriminative issue is related to stationarity of data. Estimation to modeling stationary and non-stationary time-series is discussed by Box and Jenkins and involves iterative use of the three stage process of identification, estimation and diagnostic checking [76]. Autoregressive (AR), Moving Average (MA) and mixed Autoregressive Moving Average (ARMA) are very popular univariate time-series models. ARIMA (Autoregressive Integrated Moving Average) methodology, popularized by Box and Jenkins, attempts to describe the movements of a stationary time series as a function of autoregressive and moving average parameters. Other techniques applied in trend stationary series (eg. weighted moving average, exponentially weighted moving average series), rise from the need to deal with trend and seasonal patterns [77].

Nevertheless these models lie on the assumption of homoscedasticity (constant error variance), which is considered unrealistic in many areas of economics and finance. For this purpose models which allow error variance to vary over time, such as the Autoregressive Conditional Heteroscedasticity model (ARCH) [78] and the Generalized Autoregressive Conditional Heteroskedasticity model (GARCH) [79], have been proposed for studying economic time-series.

Finally, cause-and-effect models capture relationships of the predicting variable with other variables, and these relationships are also accounted to the prediction. Popular cause and effect methodologies have their origin to econometrics, regression (e.g. least squares, max likelihood or stepwise) and Neural Networks [80].

In the following section the separating line between negotiation and other routine forecasting problems is drawn to identify distinctions and challenges.

6.2 Forecasting in negotiations

When physical negotiations are transferred to electronic settings the agents need to represent their owners as closely as possible and acquire their owner's interests, strategies and preferences in a given domain. Forecasting the counterpart's next offer is equivalent to forecasting his behavior at each subsequent step of the interaction. But what are the possible behaviors a negotiating partner may adopt during the discourse?

In the third chapter we discussed the Dual Concerns model, which describes five behavioral classifications regarding cooperativeness and assertiveness [19]. Thomas and Kilmann developed the five behavioral classifications to elicit the different bargaining styles [81], and Ludwig, Kersten and Huang have suggested the use of Thomas-Kilmann Conflict Mode Instrument to measure the conflict mode styles of negotiators [82]. In their work the authors identify three types of concession curves which reflect the recessional tendency of negotiators with respect to the conflict mode styles. These are concave, linear and convex curves. Concave concession curves are used to model high concessions at the beginning of negotiations and small concessions at the end. Linear concession curves yield equal concessions at each time step, while convex curves characterize small concessions at the beginning and bigger concessions at the end. Results showed that negotiators who adopt compromising behaviors have concave concession graphs, negotiators who adopt accommodating behaviors have linear concession graphs, while negotiators who adopt competing behaviors have convex concession graphs.

Additionally, as presented in the third Chapter, Faratin et al. suggest a more generic and domain independent view of modeling negotiator behaviors with the use of formal decision functions [25]. Time and resource dependent strategies are modeled through different types of polynomial and exponential functions, and incorporate various types of concession curves (concave, linear and convex). On the other hand many types of behavior dependent tactics which mimic the relevant recessional tendency of the counterpart cannot be explicitly defined by a polynomial or an exponential function. Lastly, hybrid strategies which stem from mixtures of the aforementioned strategies may emerge, increasing the domain of possible behaviors.

We therefore conclude that the distinctive feature of forecasting the counterpart's future offers to other routine forecasting problems relies on the dynamic nature of the process. The evolving rules of the subsequent offers depend on the negotiator's behavior, which is modeled through the different concession curves. Yet the behavior or strategy of negotiators, which is motivated by their goals and preferences, is not always disclosed, therefore the function form is not known a priori. Additionally, it is not always feasible to describe the behavior of an agent by a well known function, as is the case with agents adopting random behaviors.

From a general perspective, traditional forecasting methods presented in section 6.1, with the exception of Neural Networks, require specific assumptions over the underlying data distributions. Most time-series models lie on the assumption of homoscedasticity, which is guaranteed through the Breusch-Pagan test [83]. ARCH and GARCH models, which allow variable error variance, are appropriate only if the residuals follow specific function forms [79]. Regression, in order to be applied, requires that the residuals are not correlated, in which case autoregressive models are more appropriate [84].

On the other hand, Neural Networks are applicable in the general case and for this reason they are big challengers to conventional forecasting methods. Hornik, Stinchombe and White have proved the universality of Neural Networks which are applicable in the general case, without assuming implicit knowledge of the function that maps input to output data [85].

Methodologies that have been used for the purpose of forecasting the counterpart's future offers can be summarized into those based on statistical approaches (particularly non-linear regression) [2] [3] mathematical models based on differences [4] [5], and connectionist approaches, particularly some special types of neural networks [6] [7] [8] [9] [10] [11] [12].

Experiments have shown that mathematical models give poorer results when compared to non-linear regression models [3]. The authors characteristically state that "Compared to the approach based on the difference method, the regression-based prediction is more precise and results in higher utility gains of the adaptive negotiation agent". The objective of regression is to estimate the parameter vector of the generation functions, so as to minimize some error function and appropriately fit the counterpart's previous offers. Non-linear regression is applied in each negotiating round of the predicting agent over the whole dataset. However, such models are more restrictive than artificial neural networks, since they require the assumption of a known function form of the counterpart's behavior. As applied in [3] they are tied to specific offer generation functions. Additionally it is argued that applying non-linear regression is not appropriate in the general case, since it is not guaranteed that the residuals are not correlated.

Setting focus on other mathematical models, in [86] the superiority of forecasting the counterpart's next offer using neural network models is empirically demonstrated. More specifically, polynomial approximation using a 7th order polynomial, which was proven to be the most appropriate for this purpose, as well as approximation with the use of cubic splines were employed to estimate the counterpart's next offer one step before the expiration of his deadline. Numerous experiments were conducted covering many different negotiation scenarios, and results showed that the agent who applied the forecasting strategy which was based on neural networks, yielded higher utility gains and smaller estimation error. The author suggested that polynomial approximation was not as accurate due to polynomial oscillations, and cubic splines were difficult to extrapolate.

Summarizing, it can be concluded that the current trend in forecasting the negotiation counterpart's next offer lies on neural networks, which are reviewed in the following section.

7. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) represent a multidisciplinary subject with roots in neuro-sciences, mathematics, statistics, physics, computer science and engineering. Numerous books have been written concerning ANNs. In almost every introductory section we encounter the biological analog of the artificial neuron, and the corresponding functions of the interconnected neurons in the human brain. The design of Artificial Neural Networks is indeed inspired by neurobiological elements; nevertheless it is important to recognize that artificial neurons are truly primitive and that the structural organization of levels in the human brain cannot be re-created with artificial networks. The interested reader may refer to [87]. As stated by Krose and van der Smagt, “an artificial neural network consists of a pool of simple processing units (nodes/neurons) which communicate by sending signals to each other over a large number of weighted connections” [88]. Each processing unit receives input from neighbor or external sources and computes an output signal which is propagated to other units. It is conceived as a primitive function and artificial neural networks may also be defined as networks of primitive functions [89]. Each processing unit is split to two functional parts: an integration function $g : R^n \to R$ which reduces the n input arguments to a single numerical value, and an activation function $f : R \to R$, which produces the nodes’ output. Figure 21 depicts the general structure of a processing unit.

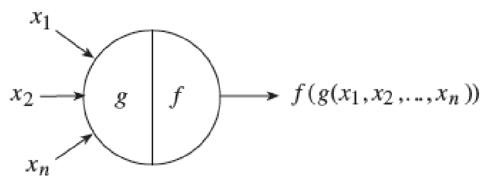


Figure 21: General structure of a processing unit

Knowledge of the neural network is highly distributed among the interconnection weights. Primitive functions along with interconnection weights are combined to produce the network function $F : R^N \to R^M$. Therefore a neural network provides mapping from input to output space. This mapping is learned by adjusting the interconnected weights with respect to some learning rule.

The main differences of neural network models lie in the primitive functions used by each processing unit, the interconnection patterns and the timing of the transmission of information. In general two fundamentally different classes of network architectures are identified: feedforward and networks with feedback connections. In Feedforward Neural Networks (FFNN), neurons are organized in the form of layers. In the simplest form, an input layer of source nodes projects to an output layer of neurons. In the general case intermediate layers of neurons (hidden layers) intervene between input and output layer, enabling the network to extract higher order statistics. This class of networks is strictly acyclic and every node of each layer is connected to nodes of the adjacent layer. Networks with feedback connections on the contrary, allow connections of nodes with other nodes of the same or preceding layers. If the network contains cycles, computation is not straightforward and computing units need to be synchronized.

Another distinctive criterion relates to the learning method which is used for network training. Learning algorithms are divided to supervised and unsupervised. In supervised learning, knowledge of the environment is represented by a set of input-output examples. This class of learning techniques is further divided to reinforcement and error-correction learning. In reinforcement learning, reward signals are assigned after the presentation of each example, and weight correction is based on the input vector. In error-correction learning, examples are used to formulate the training set and instruct the neural network with the desired output of each particular input. The magnitude of the error of the produced output together with the input vector is used to adjust the network weights in a step-wise manner. After the network is trained it can be applied to perform calculations with unknown input.

On the contrary unsupervised learning does not assume the existence of input-output pairs, rather it focuses on the extraction of relevant information within the redundant training samples. Popular feed-forward networks using supervised learning are Perceptron, Adaline, Multi-Layer Perceptron (MLP), Radial Basis Function Networks (RBFNs), functional link networks. Self organizing maps are feedforward networks using unsupervised learning. Networks with feedback connections using supervised learning are recurrent networks, (time-delay networks), Boltzman machines, cellular neural networks, competitive learning and Kohonen networks. Finally networks with feedback connections using unsupervised learning are Hopfield network, and ART.

In general, the learning tasks of neural networks are classification, clustering, pattern association, function approximation, forecasting, time series analysis, feature extraction, signal processing and control [87]. In the next subsection we provide a historical review.

7.1 A historical review of artificial neural networks

The first formal mathematical description of an artificial neuron was provided by Mc Culloch and Pitts [157]. In their work the authors bridged principles of neurophysiology and mathematical logic, and showed that a sufficient number of simple neurons could be used to compute any function. The next important development in neural networks came in 1949, with Hebbs' Organization of Behavior [90]. In this book Hebb states the synaptic modification of a learning organism and the creation of neural assemblies as a result of such modification. Hebbs' book has been inspiring for the development of learning and adaptive systems [91]. Two classical models the 'Perceptron', proposed by Rosenblatt [92], and Adaline, proposed by Widrow and Hoff [93], constitute the first functional artificial neural networks. The goal of the perceptron was to learn the association d: $\{-1,1\}^N \rightarrow \{-1,1\}$, given a number of input,output samples. The difference between Perceptron and Adaline lies in the training procedure. The publication of Minsky and Papert's Perceptrons [94], discouraged research in the area of ANN, since severe restrictions on the representational power of perceptrons were detected and presented. With single layer perceptrons only linear classifiers could be constructed or in the case of function approximations, only linear relationships could be represented. This problem was solved with the introduction of multi-layer perceptrons (MLPs), and the adjustment of weights using the backpropagation algorithm [95] [96]. Another important activity in the field, was the introduction of self-organizing maps using competitive learning by von der Malsburg and Willshaw [97] [98]. Grossberg established a new principle of self-organization known as Adaptive Reasonance Theory (ART) [99] [100] [101], which introduced a bottom-up recognition layer and a top-down generative layer. In 1982, Hopfield introduced Hopfield networks with feedback connections [102], bringing current research efforts to a common mathematical frame. The same year, Kohonens' publication on self-organizing maps, [103], received great attention and still constitutes a benchmark for testing and evaluation of innovative models. An alternative to multi-layer perceptrons (MLPs) was introduced by Broomhead and Lowe, with the design of feedforward networks using Radial Basis Functions (RBFs). Poggio and

Girosi, enriched the theory of RBFs by applying Tikhonovs' regularization theory [104]. Finally in the early 1990s, Vapnik invented support vector machines (SVM) for pattern recognition, regression and density estimation problems [105] [106] [107] [108]. In the next subsection we focus on ANNs that are used for prediction.

7.2 Artificial neural networks for prediction

Artificial Neural Networks (ANNs) are promising tools for predicting future values of business data. Their success is attributed to their unique features and powerful pattern recognition capability. ANNs have global and local function approximation ability, which allows them to capture complex data relations. Classical Neural Networks which have been used for predictions are Multi-Layer Perceptrons (MLPs), Radial Basis Functions (RBFs)(function approximation), Functional Link Networks (function approximation), Recurrent Networks (function approximation, interpolation, forecasting), Time Delay Networks (forecasting and time-series analysis). In the next subsections we provide a review of MLPs and RBFs which are the most popular ANNs that are used in negotiation forecasting.

7.2.1 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP), is a feedforward fully connected network (every node of each layer is connected with every other node of the adjacent layer) and is considered one of the most classical models of ANN. Figure 22 shows the layered structure of an MLP with one hidden layer.

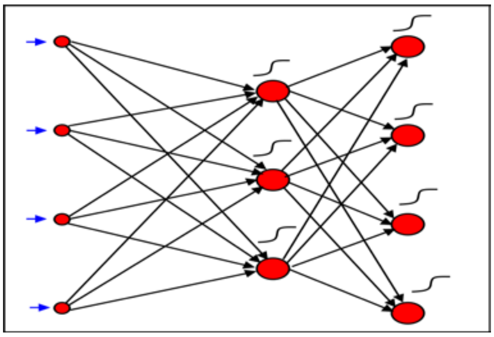


Figure 22: Structure of an MLP network with one hidden layer

Nodes in the input layer only transmit information from external sources and do not perform any computations, while in the hidden and output layers, they follow the general structure depicted in Figure 21. Typically the input signal propagates through the network in a forward direction, on a layer-by-layer basis. The resulting outputs of each layer are in turn applied to the next layer in a sequential manner, until the networks' output is computed. Design issues considering the number of hidden layers and nodes in each layer, as well as the activation functions for each node are related with the task the network is designated. In the case of forecasting and function approximation, MLP nodes use additive aggregation functions. In the hidden layer, nodes have smooth non linear activation functions, following sigmoidal nonlinearity (most commonly used are logistic and tangent hyperbolic functions), while in the output layer they have linear activation functions. Another issue concerning the networks' free parameters relates to the interconnection weights. In most cases initial weights are selected to follow a uniform distribution (-a,a). The network learns the association from input to output space and accordingly adjusts the weights through the training procedure which is discussed in detail in the following section.

7.2.1.1 MLP Training

As stated earlier the network is an implementation of a composite function from input to output space, and the learning problem consists of finding the optimal combination of weights so that the network function **Φ** approximates a given function f. However, function f is not given explicitly, but implicitly through input-output examples. We consider a network that needs to adjust its free parameters in order to provide an accurate mapping of m input to k output space. If we have N input vectors $y(n) = [y_1(n), y_2(n),..., y_m(n)]^T$ and the desired output vectors $d(n) = [d_1(n), d_2(n),..., d_k(n)]^T$ with n є [1,N], the error signal of output node j after presenting input pattern n is defined as the distance of the computed output $y_j(n)$ with the desired output for that particular pattern $d_j(n)$. This relation is given by:

$$e_j(n) = d_j(n) - y_j(n) \qquad \text{(eq.12)}$$

Instantaneous value of the total error energy $E_{(n)}$ is obtained by summing $\frac{1}{2}e_j^2$ for all neurons in the output layer, thus:

$$E_{(n)} = \frac{1}{2}\sum_{j=1}^{k} e_j^2 \qquad \text{(eq. 13)}$$

The average squared error energy is then obtained by summing the error energies of all patterns and normalizing with respect to the set size N:

$$E_{avg} = \frac{1}{N}\sum_{n=1}^{N} E_{(n)} \qquad \text{(eq. 14)}$$

The objective of the training procedure is to adjust the interconnected weights, so as to minimize the error function given by eq. 14. A common practice is to divide the original data patterns to training, validation and test set. The training algorithm is applied to the training set. Validation set is needed for the application of an early stopping method to guarantee better generalization, and the test set is used for overall assessment, since the error of the test set is a common indicative measure. Training procedure is defined as incremental or online if weight updates take place after the presentation of each input pattern, and batch or offline if weight updates are performed after the presentation of all training patterns. Training procedures are classified to first order methods (i.e. Error Back Propagation), where the updates of the interconnection weights are based on the direction of the gradient and second order methods (i.e. Newtons'method, Gauss-Newton and Levenberg and Marquardt method, one step secant (OSS), scaled Conjugate Gradient and Conjugate Gradient (CG)) which account more information about the shape of the error function [109]. Second order methods are applicable only in batch training mode and result to faster convergence. Amongst the most popular of the batch training mode procedures is the Levenberg and Marquardt (LM) method [110]. In the following subsections we provide a detailed description of Error Back Propagation, Newton, Gauss-Newton and Levenberg and Marquardt method.

7.2.1.2 First Order Learning Methods - Back propagation

Back-propagation consists of two passes through the different layers of the network: a forward pass, where the input vector is propagated and the vectors' output is computed and a backward pass, where the error signal is propagated backward through the network. The interconnection weights are adjusted in order to 'move' the networks' response closer to the desired output. To delve into the functional details of the algorithm, we borrow the notation from Haykin [89].

Each pattern n is sequencially presented to the network. In the forward pass each node j at layer l+1 computes the output signal aggregating the signals of the preceding layer l and passing the result to its activation function. Particularly, if the preceding layer has m nodes, then node j will compute $u_j(n) = \sum_{i-0}^{m} w_{ji} y_i(n)$, where w_{ji} is the inter-connection weight between neuron i in layer l and neuron j in layer l+1, and $y_i(n)$ is the signal produced at the ith node of layer l. This value will form the input to the node's activation function ϕ_j , thus the computed output for node j will be: $y_j(n) = \phi_j(u_j(n))$. Minimization of the error energy $E_{(n)}$ using the back-propagation algorithm is similar in rationale to the Least Mean Square Algorithm (LMS). The partial derivative is expressed according to the chain rule as follows:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial u_j(n)} \frac{\partial u_j(n)}{\partial w_{ji}(n)} \qquad \text{(eq. 15)}$$

The distinct partial derivatives are:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j \qquad \text{(eq. 16)}$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \qquad \text{(eq. 17)}$$

$$\frac{\partial y_j(n)}{\partial u_j(n)} = \phi'(u_j(n)) \quad \text{(eq. 18)}$$

$$\frac{\partial u_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad \text{(eq. 19)}$$

Thus applying eq. 16-19 to eq. 15 gives us:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j \phi'(u_j(n)) y_i(n) \qquad \text{(eq. 20)}$$

There are two training modes using the back-propagation algorithm. Online or incremental and offline or batch-mode. In incremental learning weight adjustment takes place after the presentation of each example. The correction of the interconnection weight w_{ji} is defined by the delta rule:

$$\Delta w_{ji} = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \quad \text{(eq. 21)}$$

where η is the learning rate parameter of the back-propagation algorithm. The minus (-) sign in eq. 20, accounts for gradient descent in the weight space (towards the direction which reduces the value $E(n)$). Applying eq. 20 to eq.21 gives:

$$\Delta w_{ji} = \eta \delta_j(n) y_i(n) \qquad \text{(eq. 22)}$$

where $\delta_j(n)$ is the local gradient of node j given by the relation

$$\delta_j(n) = -\frac{\partial E(n)}{\partial u_j(n)} = e_j \phi'(u_j(n)) \quad \text{(eq. 23).}$$

If j is a neuron in the output layer the local gradient is the product of the error signal e_j and the derivative $\phi'(u_j(n))$, both terms associated with neuron j. However, if j is a neuron in the hidden layer, the local gradient is the product of the derivative $\phi'(u_j(n))$ and the weighted sum of the local gradients δ, computed for the neurons of the next layer that are connected to neuron j.

In batch-mode or offline training, weight update is performed after the presentation of all patterns as shown in eq. 24:

$$\Delta w_{ji} = \eta \sum_{n=1}^{N} \delta_j(n) y_i(n) \quad \text{(eq. 24)}$$

In both training modes, examples are presented multiple times during the training procedure. A full presentation of the entire set of examples is termed an epoch, thus learning takes place on an epoch-by-epoch basis, until the network weights stabilize and some convergence criteria, often related to reaching a minimum value of the averaged squared error are met.

Batch-mode training is based on the computation of the true gradient of the error surface and convergence is guaranteed (the proof is based on Kolmogorov's theorem). Incremental training is stochastic in nature, since it approximates the true gradient. However, incremental training requires less storage capacity.

7.2.1.3 Accelerating the back-propagation Process

Back-propagation is guaranteed to converge to a local minimum of mean squared error (MSE) after a number of iterations. Nevertheless, the error surface (MSE with respect to the network weights) may be uneven or highly jagged, resulting to a number of local minima, different from the global minimum being searched. In order to avoid being trapped to a local minimum, it is desirable to compute the average direction of MSE in a small region, rather than the precise gradient at one point. In [95], a heuristic to approximate the average gradient and allow weight adjustment towards the general direction of MSE decrease is proposed. This heuristic is based on relating the weight changes of iteration (n), with the weight changes of the preceding iteration (n-1). Implementation of this rule is accomplished through the generalization of the delta rule, with addition of a momentum term as shown in eq. 25.

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad \text{(eq. 25)}$$

where α is the momentum constant.

Another heuristic for accelerating back-propagation learning algorithm stems from adjustment of the learning rate parameter η. A small value of η signifies small changes of the weights at each iteration step, thus smoother trajectories in weight space with the cost of a slower rate of learning. High values of η speed up the rate of learning, but result to high oscillations, turning the network unstable. A simple heuristic is to increase the learning rate at every iteration that improves performance by a significant amount and to decrease it at every iteration that worsens performance.

7.2.1.4 Second Order methods – Newton's method

Second order methods are based on optimization techniques which account more information about the shape of the error function than the direction of the gradient. These methods are applicable only in batch training mode and result to faster convergence. In second order methods, a Taylor series approximation is used to expand the error surface E_{avg} about the weight vector of the nth iteration $w(n)$ (eq. 26)

$$E_{avg}(w_{(n)} + \Delta w_{(n)}) \approx E_{avg}(w_{(n)}) + g^T_{(n)}\Delta w_{(n)} + \frac{1}{2}\Delta w^T_{(n)} H_{(n)} \Delta w_{(n)} \quad \text{(eq. 26)}$$

where $g_{(n)}$ is the local gradient vector, and $H_{(n)}$ is the local Hessian matrix. The optimal value of the adjustment $\Delta w^*_{(n)}$ is obtained by differentiating eq.26 and setting to zero, in which case we get:

$$\Delta w^*_{(n)} = -H^{-1}_{(n)} g_{(n)} \qquad \text{(eq.27)}$$

Eq.27 constitutes the essence of Newtons' method, where the problem is solved in one iteration. However, inversion of the Hessian matrix is computationally expensive and requires the Hessian to be non singular and positive definite, which is not guaranteed.

To overcome the aforementioned problems, approximation of the Hessian matrix H is attempted (Quasi-Newton and Conjugate-Gradient method). Among the most popular second order optimization techniques lies the Levenberg and Marquardt method. It is a method which combines Gauss-Newton and gradient descent.

7.2.1.5 Gauss-Newton and the Levenberg and Marquardt method

According to Gauss-Newton method, in regions where the error is small, the Hessian matrix can be approximated by the Jacobian matrix J as follows:

$$H = J^T J \quad \text{(eq. 28)}$$

Combination of Gauss-Newton and gradient descent was proposed by Levenberg [111], resulting to the following weight update formula:

$$\Delta w_{(n)} = -(H + \lambda I)^{-1} \nabla E \Rightarrow (J^T J + \lambda I)\Delta w_{(n)} = J^T E \quad \text{(eq. 29)}$$

Small values of λ reduce the affect of gradient descent, and the update rule reduces to the Newton step. On the contrary, with large values of λ second order information is ignored and the rule reduces to gradient descent. λ is Levenberg's damping factor and its value is adjusted at each iteration with respect to the error at the produced weight vector. It is increased if the error is reduced otherwise it is decreased. Intuitively the algorithm follows the gradient descent until it approaches the region of the minimum error, where it gradually switches to Newton's step (using the quadratic approximation). Marquardt improved the algorithm by replacing the identity matrix with the diagonal of the Hessian resulting to equation 29:

$$(J^T J + \lambda diag[H])\Delta w_{(n)} = J^T E \qquad \text{(eq. 30)}$$

With Marquardt's formula [112], even in cases of high values of λ where the algorithm is performing a gradient descent, second order information is accounted. Particularly, each component of the gradient is scaled according to the curvature of the error surface which is proportional to the Hessian. Larger steps are taken in the direction with low curvature (flat terrains) and smaller in the direction with high curvature (steep inclines), addressing the classical problem of the “error valleys”.

LM method has been noted as the fastest and most efficient method for training small and moderate-sized neural networks [113].

7.2.2 Radial Basis Function Networks (RBFN)

Radial Basis Function Networks (RBFN), a popular alternative to the MLPs, are two layered feedforward networks which perform exact interpolation of a set of data points in a multidimensional space. Each unit in the hidden layer implements a radial basis function, enabling a non linear transformation of input to hidden space, while each node in the output layer performs linear combination of the hidden layers' output, enabling a linear transformation of hidden to output space. The rationale behind the nonlinear transformation, followed by the linear transformation is tracked in Covers' theorem on the separability of patterns [114], which states that "a complex pattern classification problem cast in a high dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space" – hence the reason for making the dimension of the hidden layer of an RBFN high. Radial basis functions were introduced by Powel [115] to solve the real multivariate interpolation problem, and were first used in neural networks by Broomhead and Lowe [116]. Figure 23 illustrates the structure of an RBFN.

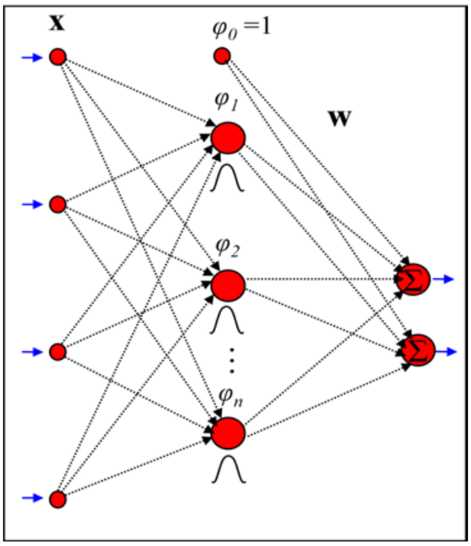


Figure 23: Structure of a RBFN

For an input pattern **X**, the output F(**x**) of the RBFN is given by eq. 31:

$$F(x) = \sum_{k=1}^{n} w_k \phi(\|x - x_\kappa\|) \qquad \text{(eq. 31)}$$

where $\mathbf{w_k}$ is the weight vector which connects the k^th hidden unit with the units of the output layer, $\phi(\|x - x_\kappa\|)$ is a set of n nonlinear functions, known as radial basis functions, $\|.\|$ denotes a norm that is usually Euclidean, and x_k are known data points that comprise the centers of the Radial Basis Functions. For a set of n patterns $\{(\mathbf{x_p},\mathbf{t_p})\}$, p ={1,2,…n}, the interpolation condition which needs to be satisfied is given by eq. 32:

$$F(x_p) = t_p \quad \text{(eq. 32)}$$

In simple RBFNs each input vector is set as an RBF center, therefore the hidden layer comprises of as many units as the available patterns. Inserting equation 31 to 32 we obtain the following set of linear equations

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} \rightarrow \Phi W = T \quad \text{(eq. 33)}$$

where $\phi_{ij} = \phi(\|x_i - x_j\|)$,(i,j) = 1,2,..,n

If matrix Φ is nonsingular and therefore invertible, the weight vector can be determined by:

$$W = \Phi^{-1}T$$

There is a large class of Radial Basis Functions that is covered by Michelli's theorem [117], and for which matrix Φ is invertible for distinct patterns (Multiquadrics, Inverse multiquadrics and Gaussian Functions).

Nevertheless, RBFNs are prone to overfitting and result in degraded generalization performance [116]. Regularization theory proposed by Tikhonov [118] provides a solution to the so called bias-variance problem. Poggio and Girosi [104] explain the use of regularization theory to Radial Basis Function Networks as a method for improved generalization to new data. Regularization theory leads to the design of a network with the same structure as the simple RBFN, termed regularization network. This network has a number of desirable properties; it is universal approximator, as it approximates arbitrarily well any multivariate continuous function given a large number of hidden units and it has best approximation property, as there exists a choice of coefficients that provides the best approximation of an unknown function f.

The one-to-one correspondence between the hidden units and the training input data x_i turns the network prohibitively expensive to implement for large training sets (inversion of a nxn matrix grows polynomially with n, $O(n^3)$). To address this issue, a suboptimal solution which approximates the regularized solution is searched in a lower dimensional space. This approach emerges the so called generalized radial basis function networks, which have the same architecture as the regularized RBFNs, but use less basis functions in the hidden layer. The output of each hidden unit is defined by a green function, with the difference that the center c_i does not necessarily coincide with any of the available input vectors x_i.

7.3 Selection of MLPs to estimate the next offer

At this point it is essential to underline the particular requirements of a negotiation interaction and justify the selection of MLPs with one hidden layer, as an enhancement tool to the predictive agents.

In automated negotiations with incomplete knowledge, no prior information concerning agent's preferences, goals, strategies and deadlines is exchanged between the underlying parties. Enhancing agents with predictive tools has proved significant, since they adopt strategies that assist them in formulating the offers to be proposed, and yield more satisfying outcomes. Each proposed offer plays a crucial role in the formation of the negotiation outcome and in the progress of the overall procedure. As Hindriks states in [119], "In the analysis of negotiation strategies, not only the outcome of a negotiation is relevant, but also the bidding process itself is important. Mistakes made during the bidding can have an enormous impact on both players. Examples from human negotiations are of the form: a wrong offer can upset relationships, even causing the other party to walk away”.

The same issue has also been pointed out by Carbonneau et al. in [6], where the authors claim that even small variations in the current offer can have important impact on the expected counter-offer from the opponent. Since the exchanged offers depend on the prediction of the counterpart’s future offers, accuracy is a key factor to the forecasting tool. Furthermore, negotiating agents have limited capacity and resources. In many negotiation domains, agents need to keep their size low, and abide with severe time restrictions.

As far as accuracy is concerned, it is noted that FFNNs such as MLPs and RBFNs are universal approximators, capable of approximating any continuous function.

More particularly, an MLP with a single hidden layer having sigmoidal activation functions and an output layer using linear activation functions can approximate well any continuous multivariate function to any accuracy [85] [120] [121] [122]. Cybenko’s proof [120] is based on the Hahn–Banach theorem and is concise. The proof of Hornik et al. [85] is based on the Stone–Weierstrass theorem, while Funahashi [121] proved the same problem using an integral formula. Xiang’s proof [122] is derived from a piecewise-linear approximation of the sigmoidal activation function.

RBFN is a popular alternative to the MLP, which has universal approximation and regularization capabilities. Theoretically, the RBFN can also approximate any continuous function arbitrarily well, if the RBF is suitably chosen [104 [123] [124].

Although MLPs and RBFNs are considered equivalent, MLPs are global approximators, have greater generalization ability and are good candidates for extrapolation. On the contrary RBFNs are local approximators and the extension of a localized RBF to its neighborhood is determined by its variance, which restricts the RBFN from extrapolation beyond the training data. (page 334 of [109]). Other types of networks such as Recurrent Neural Networks (RNNs), which have at least one feedback loop, are also universal approximators of dynamical systems. However, due to the difficulty of applying backpropagation and due to higher training times, they have not been used for the purpose of forecasting the counterpart’s next offer.

Considering that the size of agents should be kept small, localized RBFNs have an additional shortcoming. In order to achieve accuracy similar to that of MLPs, they require more data and more hidden units. More specifically, to approximate a wide class of smooth functions, the number of hidden units required for the MLP with one hidden layer is polynomial with respect to the input dimensions, while the number for the localized RBFN is exponential [125].

MLPs require fewer resources due to their small size (compared to RBFNs). For this reason, and because they have been selected by the majority of the aforementioned applications, this research focuses on the class of MLPs with one hidden layer. In the next section we outline the main characteristics of the MLPs that have been applied for forecasting the counterpart’s next offer.

7.4 Architecture of MLPs

In the third chapter the strategy of state of the art agents who conduct single-lag predictions to improve their individual utility was discussed. In this section we will delve into the architectural details of the MLPs that are used for such purpose.

In SmartAgent [8] the provider agent predicts the next offer of the consumer agent by using an MLP with one hidden layer. The negotiations conducted are single-issued, and the values of the three preceding offers of the opponent constitute the network's input. The number of nodes in the hidden layer is not clearly specified; rather it is left an open issue. As the authors state the number of nodes in the hidden layer is set during network training, through trial and error. The output layer consists of one node which represents the value of the counterpart's next offer. The architecture of the 3xnx1 MLP is illustrated in Figure 24:

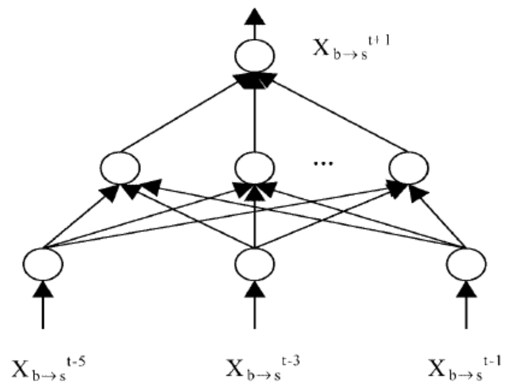


Figure 24: Architecture of the employed MLP [8]

The training set is extracted online, during negotiation. At the beginning of each discourse there is a time window required by the MLP to adapt to the negotiation context, therefore the MLP's learning capability cannot be exploited before the first p proposals ($p \leq 5$). The MLP is trained with the use of Back-propagation with adaptive learning rate. In Oprea's paper only preliminary results are illustrated, where simple and small 3x3x1 networks are used. Weights are initialized in the range of [-0.1, 0.1]. As a pre-training step, 20 runs with 2 different weight initializations and 10 different training sets were used, and the weight vectors that resulted to the best performance (smallest errors) were selected.

Another agent-based application that exploits the learning capability of neural networks is the one described in [9]. The authors compare the results of application of an MLP and an RBFN in forecasting the next offer. In their work the Consumer agent makes use of the neural network to predict the Provider's next offer in single-issued bilateral negotiations. As far as the MLP is concerned, the network's input is formulated by the 9 past offers of the opponent, and the network's output consists of a single linear node which represents the estimated offer. The MLP has one hidden layer with three non-linear nodes. Training is conducted offline, with the use of Levenberg and Marquardt method. It should be noted that the MLP is trained at a pre-negotiation stage, with data from past interactions and is then applied during the discourse. Since the network is trained only once before initiation of the process, the authors have tried to include various training patterns from different negotiation domains.

Application of MLPs for the purpose of estimating the counterpart's next offer is also described in [6], where an NSS that assists providers of bicycle parts in bilateral negotiations with bicycle producers is developed.

The negotiable object has four issues (Price, Delivery, Payment and Returns), that take discrete values. The neural network employed has thirty nine inputs, resulting from past offers (last sellers' and buyers' and the first offer), the current offer, and statistical information (maximum, minimum, standard deviation and average value of each issue). It also has ten hidden nodes and four output nodes, one for each predicted attribute of the estimated offer. The transfer function in the hidden layer is a tan-sigmoid function, while in the output layer is a linear function. The 39x10x4 MLP is illustrated in Figure 25.

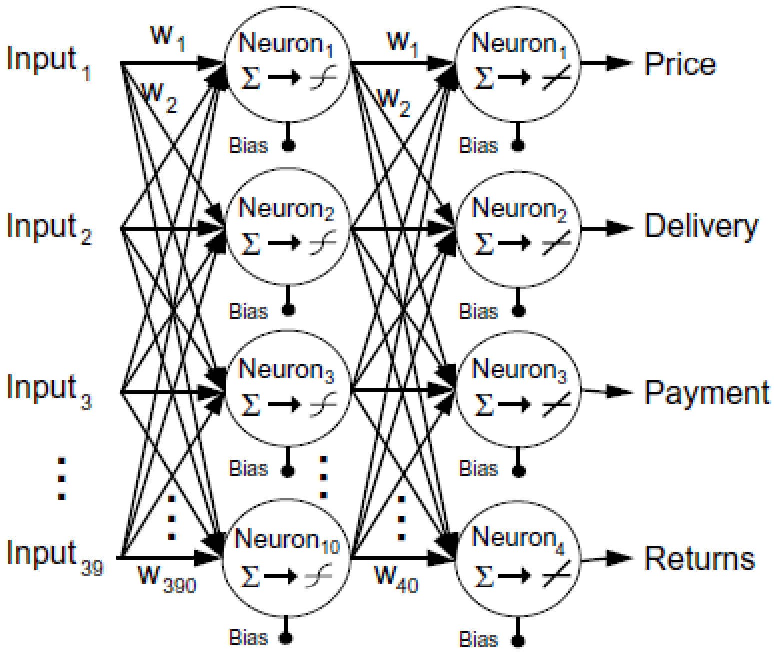


Figure 25: Architecture of the employed MLP [6]

Network training is conducted offline using past negotiation data with the Levenberg-Marquardt algorithm. The approach is tested with data obtained from bilateral negotiations conducted with the use of Inspire negotiation system. The Inspire dataset considered 6310 offers.

A similar work, where an NSS is enhanced with an MLP, is provided by Lee and Ou-Yang [7]. An artificial neural network is applied to a supplier selection auction market, as a negotiation support tool of the demander. In particular, the network is used to forecast the suppliers' next bid price, and allow the demander to appropriately choose among a list of alternatives. The network consists of 9 inputs resulting from combination of environment-specific information (quantity, due date, inventory level, scheduled production plan, surplus capacity, current time step) and offer-specific information (providers' last offer and consumers' last and current offers). It also has a single hidden layer with twelve neurons (selected by means of trial-and-error experiments), and one output neuron that reflects the predicted bid price. Training is conducted at a pre-negotiation stage with data collected from simulations of the negotiation process, by using the online back-propagation algorithm (with momentum term). Extensive experiments are illustrated resulting from 247 negotiation sessions (5982 training patterns and 1386 test patterns).

Table 4 summarizes the characteristics of agent systems employing ANNs with the purpose of predicting their counterpart's next offer. Categorization is made with respect to the agent system, model or platform, the number of negotiable attributes, the side of the agent employing the prediction (consumer or producer/provider), the mechanism used for offer generation (based on particular functions or generated by a human counterpart), the phase of application during a negotiation discourse, technical characteristics of the ANNs employed, and extensiveness of the discussed experiments.

Table 4: Technical characteristics of agents and negotiation support systems using neural network models for single-lag predictions

Agent or Negotiation Support System	**Automated Negotiator SmartAgent platform [8]**	**NSS (Carbonneau, Kersten and Vahidov, [6])**	**NSS in a supplier selection auction market (Lee and Ou-Yang, [7])**	**Automated Negotiator (Papaioannou, Roussaki, and Anagnostou, [9])**	**Automated Negotiator (Papaioannou, Roussaki, and Anagnostou, [9])**
# of Negotiable Issues	1 issue	4 issues	1 issue	1 issue	1 issue
Agent employing prediction	Provider agent predicts consumers' next offer	Provider (Itex manufacturing, a producer of bicycle gears) predicts consumers' next offer (Cypress Cycles, a bicycle producer)	Consumer predicts provider's next bid	Consumer predicts provider's next bid	Consumer predicts provider's next bid
Offer generation mechanism	Based on Faratin et al., 1998	Human (acquired from INSPIRE negotiation system)	Based on Lee's and Ou-Yang's, 2009 iterative strategy	Based on Faratin et al., 1998	Based on Faratin et al., 1998
ANN model	MLP	MLP	MLP	MLP	RBF
Input Features	The 3 previous offers of the opponent	39 inputs resulting from past offers (last provider's and consumer's and the first offer), the current offer, and statistical information (maximum, minimum, standard deviation and average value of each issue)	9 inputs resulting from combination of environment-specific information (quantity, due date, inventory level, scheduled production plan, surplus capacity,current time step) and offer-specific information (providers' last offer and	The 9 last offers of the opponent	The 9 last offers of the opponent

			consumers' last and current offers)		
Number of Hidden Nodes	Left open (3 used for illustrative purposes)	10	12	3	3
Number of Output Nodes	1	4	1	1	1
Training Function	Back propagation with adaptive learning rate	Levenberg and Marquardt method with regularization parameter	Backpropagation with momentum term	Levenberg and Marquardt method	Orthogonal Least Squares (OLS) for the selection of RBF unit centers and Linear Least Squares to train the networks' weights
Training Mode	On-line trained with data extracted from different negotiation contexts (then use network with test data)	Initially trained in Batch mode	Online trained with data extracted from different negotiation contexts (then use network with test data)	Initially trained in Batch mode	Initially trained in Batch mode
Experimental results	Preliminary	Limited to the specific domain	Extensive (5.982 training patterns/ 1.386 test patterns)	Extensive (3 families of tactics resulting to 1.239 data patterns)	Extensive (3 families of tactics resulting to 1.239 data patterns)
Application of Predictive mechanism	In every decision making step	In every decision making step	In every decision making step	Once at the pre-final step of the negotiation	Once at the pre-final step of the negotiation

8. PROBLEM STATEMENT – INTRODUCING SESSION-LONG LEARNING AGENTS

In the previous chapter we discussed why MLPs are preferred for predicting the counterpart's next offer. However, literature review reveals high diversity of the models applied in terms of the selected architecture (input, and hidden nodes), the training procedure and application of the networks, as well as the data considered to formulate the training sets. In this respect we try to address the following issues:

How should training and application of the models be performed in order to capture the dynamics of changing negotiation environments?

How can the architecture of such models be optimized?

As far as the first issue concerned, it is evident that ANNs which are employed by current state of the art negotiators are particularly tied to bound domains, since in the majority they are trained and applied to environments with data of the same underlying distributions. The networks are trained before the initiation of the current negotiation instance with data from previous interactions, and are then set to operate in the current discourse. In some cases, 'synthetic data', produced from simulations of different negotiation environments, is used to acquire the training patterns. In order to handle multiple scenarios, large sized training sets are generated, and even more complex models are designed to accurately fit the data. As a consequence, the predictors' accuracy depends heavily on 'synthetic data' or on data acquired from previous negotiations. Although these models yield very satisfying results when data distributions do not change, we argue that they cannot capture negotiation dynamics in changing environments.

As far as the second issue is concerned, literature review revealed lack of a commonly stated approach on what should constitute the input of the predictive model. In some cases the network's input was formulated by the counterpart's responses, while in others the past offers of both partners were considered. Moreover, there exists a line of work where outside options, related to demand curves, or other statistical parameters were introduced along with the negotiators' previous offers.

In this chapter we argue that retraining the MLPs is crucial to increase accuracy of the forecasting tool and yield significant gains to the predictive agents. In this respect we introduce Session-long Learning Agents and compare them with agents who train their networks only at a pre-negotiation phase (Pre-Trained Agents, PTAs).

8.1 Retraining MLPs with data acquired from the current thread

Evidently, a negotiator may periodically change his strategy and/or preferences due to changes that occur in the environment. This is illustrated in [126] who discuss the impact of "outside options in automated negotiations". Van Bragt and La Poutre also discuss how an agent may be programmed to constantly change his strategy for defensive purposes (to avoid being exploited by learning agents) [127]. As stated in section 7.3 it is important for a learning model to provide accurate predictions even in dynamic environments. When agents act in turbulent settings, it is not rational to expect exhaustion of all possible scenarios in order to formulate the training set. The novel aspect of this research lies in the field of application of Neural Networks in negotiations and not in the algorithmic design of Neural Networks. More specifically, it highlights the need to use MLPs trained with the "real" data, which are the data acquired from the current negotiation thread, rather than train MLPs with past or synthetic data before initiation of the process, and only apply them during the discourse. In the proposed approach the network is both trained and applied in each negotiation round by the predictive agent. By definition, using the “real” data to formulate the training set is expected to increase the model’s accuracy, and consequently lower the error values given by (eq.12) and (eq.13) in Chapter 7. In page 208 of [87] it is stated that using a training set representative of the environment of interest is a factor that significantly affects generalization error.

Moreover, the size of the network can be small, since only a few examples will be available for training (page 208 of [87]).

As far as training with the “real” data is concerned, two options are available. The first is to consider retraining with the “real” data MLPs that have been initially trained with data from previous interactions, and the second is to consider small MLPs with random initial weights, that are periodically retrained during the discourse.

The first option is not considered, since the size of such networks would have to be large to accommodate all training samples. Even if it were realistic to construct an MLP that exhausts all possible interactions, generation of very large training sets would be required, and as a consequence the resulting network would have more hidden neurons and parameters (weights and biases) to be estimated. This could have an effect not only on memory requirements for the agents, but also on training time. More specifically, the LM method requires storage of the Jacobian matrix which is defined as a ($|Dataset|$xO)xP matrix, where $|Dataset|$ is the size of the training set, O is the number of output nodes and P is the number of parameters (weights and biases). As stated in [128] there is memory limitation for large sized patterns. Furthermore, in [129] it is shown that the LM method is also very “expensive” in terms of number of operations for networks that have a significant number of parameters. This is due to the fact that the number of computational steps required for matrix inversion at each iteration is $O(P^3)$.

For the aforementioned reasons we investigate the use of small MLPs, with random initial weights, that are trained with the “real” data during the negotiation interaction. The term Session-Long Learning Agents is hereafter attributed to those agents that exploit the “real” data.

8.2 Introducing static session-long learning agents (SSLAs)

In this section we describe a Static Session-long Learning Agent (SSLA), which is defined as a session-long learning agent with a fixed MLP architecture during the discourse. Without loss of generality, the predictive agent is assumed to be the consumer who initiates the negotiation process at time t_1=0. The two agents take alternate turns until an agreement is established or until any of the two agents decides to terminate the procedure. At time *t,* the series of offers sent by the provider is the following: $\{ X^1_{Pr\rightarrow Con}, X^3_{Pr\rightarrow Con}, \dots, X^{t-1}_{Pr\rightarrow Con}\}$.

In the general case, the forecasting tool of the SSLA makes use of the *n* previous counterpart’s offers to estimate the next offer (at time *t*+1), as is illustrated in Figure 26.

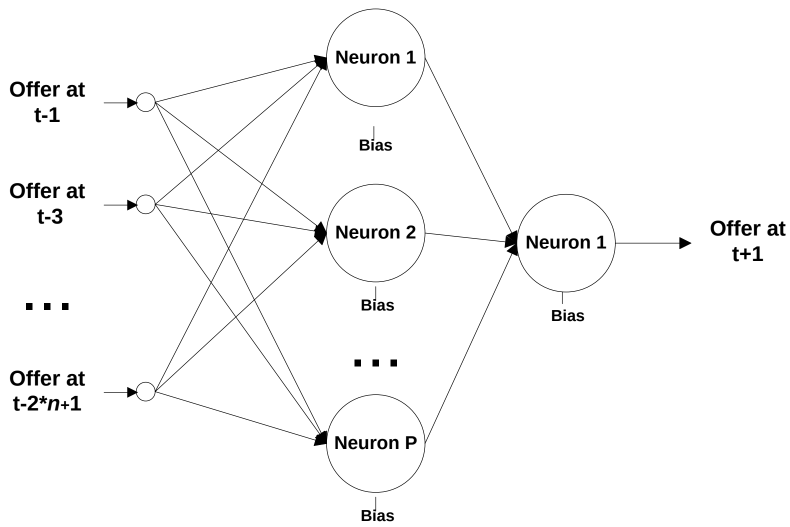


Figure 26: Forecasting tool of the negotiator

At time *t* the consumer formulates a new training set which is constructed from the series of the counterpart's offers. It should be noted that in order to apply the LM method, at least two training patterns are required, therefore the MLP is initially trained at round t_{init} = 2*n+4. The size of the dataset $|Dataset|$ at time $t \geq t_{init}$ is given by

$$|Dataset| = \frac{t}{2} - n \quad \text{(eq. 34)}$$

$|Dataset|$ is initially 2 in order to apply the LM method, and increases by 1 in each turn of the predictive agent. After training the MLP, SSLA makes use of the network to estimate his counterpart's next offer.

More specifically the actions an SSLA undertakes at each predictive round *t* are the following:

Step 1. Receive Opponent's Offer, $X^{t-1}_{Pr\to Con}$

Step 2. Update Negotiation Thread by storing the received offer

Step 3. Formulate training set:

Consider a time series of the opponent's past offers: $\{X^{1}_{(Pr->Con)}, X^{3}_{(Pr->Con)}, ..., X^{t-1}_{(Pr->Con)}\}$

Formulate the set of input-output patterns with respect to the number of input nodes

Step 4. Use the patterns yielded in Step 3 to train the network with the LM method

Step 5. Formulate current input pattern $\{X^{t-2*n+1}_{(Pr->Con)}, ..., X^{t-1}_{(Pr->Con)}\}$

Step 6. Apply input to the trained network

Step 7. Obtain forecast of opponent's next offer, $\hat{X}^{t+1}_{Pr\to Con}$

Step 8. Generate the offer the consumer would send based on its default strategy $X^{t}_{Con\to Pr(Default)}$

Step 9. Evaluate offers produced at steps 7 and 8, with respect to the consumer's utility function (Compute U($\hat{X}^{t+1}_{Pr\to Con}$) and U($X^{t}_{Con\to Pr(Default)}$))

Step 10. Generate next offer based on the decision rule described in 5.2.1

The forecasting tool of the SSLA was selected to be very small and consist of three inputs (n=3), representing the three previous offers of the counterpart (as in [8]), and two hidden nodes (P=2). This architecture is even simpler than the one proposed in [8], since it uses one hidden neuron less.

Although the optimal network architecture cannot be extracted from theoretical findings, it is rather empirically found that the ratio of learning parameters with respect to the size of the training data should be kept small. As stated in ([130], [131]) the generalization error can be decomposed into an approximation error due to the number of parameters and to an estimation error due to the finite number of data available. A bound for the generalization error E is given by

$$E \le O\left(\frac{1}{P}\right) + O\left(\left[\frac{Pn\ln(P|Dataset|) - \ln\delta}{|Dataset|}\right]^{1/2}\right) \qquad (35)$$

where n is the number of input units, P is the number of hidden nodes, δ is a confidence parameter, $\delta \in (0,1)$, and $|Dataset|$ is the size of the dataset. Since in each subsequent step $|Dataset|$ increases, the bound of the generalization error E given in (35) is expected to decrease.

Applying in (eq. 35) n=3 and $|Dataset|$= 2 (minimum value required by the LM method), yields that the agent can initially train and use the MLP at the tenth round.

As far as complexity is concerned, storage of the Jacobian matrix ($|Dataset|$xP), as well as computations for matrix inversion that are of order O(P^3), are required at each iterative step of the LM method. The LM is considered efficient since it can be defined as a polynomial time algorithm (an algorithm that has time complexity that is bounded by a polynomial in the length of the input) [132].

In the following section we focus on comparing the SSLA with Pre-Trained Agents (PTAs) to highlight that the significant increase in the accuracy of the forecasting tool when retraining is performed.

8.3 Comparative Illustration with current State of the Art

In subsection 8.3.1 we describe the Pre-Trained Agent (PTA) and discuss how such an agent may be generated with the use of “synthetic data”. In subsection 8.3.2 we conduct a number of experiments to compare SSLAs and PTAs under the same negotiation settings.

8.3.1 Current state of the art: pre-trained agents (PTAs)

Unlike SSLAs, PTAs use MLPs initially trained with data from previous interactions. More particularly the actions undertook by a PTA are the following:

At a pre-Negotiation Stage:

Step 1. Formulate training set:

Consider numerous negotiation threads(from past or simulated interactions) and extract a time-series of opponent’s offers from each thread

Formulate sets of input-output patterns with respect to the number of input nodes. The patterns will account all scenarios of Step1.a

Step 2. Train the network with input-output patterns yielded from Step1.

__During Negotiation, at each negotiation round t:__

Step 1. Receive Opponent's Offer $X^{t-1}_{Pr\rightarrow Con}$

Step 2. Update Negotiation Thread by storing the received offer

Step 3. Formulate current input

Step 4. Feed current input to the pre-trained network

Step 5. Obtain forecast of opponent's next offer $\hat{X}^{t+1}_{Pr\rightarrow Con}$

Step 6. Generate the offer PTA would send based on its default strategy $X^{t}_{Con\rightarrow Pr\ (Default)}$

Step 7. Evaluate offers yielded in steps 5 and 6, with respect to consumer's utility function (Compute $U(\hat{X}^{t+1}_{Pr\rightarrow Con})$ and $U(X^{t}_{Con\rightarrow Pr(Default)})$

Step 8. Generate next offer based on the decision rule described in section 5.2.1

The data used for the formation of the training set is acquired by conducting negotiations between non-learning agents, based on the scenarios discussed in [9], as they are described analytically and can be easily reproduced. Three experimental sets are presented, each leading to the construction of a different MLP. Consequently three MLPs (MLP1, MLP2, MLP3), fitting the data acquired from the respective experimental set, are used for the generation of three instances of pre-trained agents. For each experimental set, the negotiation parameters concerning the reservation values [$Price_{min}$,$Price_{max}$]), the deadline ('T_{max}'), as well as each agent's strategy ('Strategy') and level of concession (*β)*, are cited in Table 5. The first set consists of cases where the providers have significantly longer time to negotiate and deal with consumers of varying reservation values, resulting to various overlaps of agreement zones (0% - 100%). 100% overlap of the agreement zone is attained when the agents have common reservation values. Providers are selected to follow a linear TD strategy (β=1) and consumers a BD strategy, responding to the offers of their opponent. The second experimental set comprises of scenarios where there is 100% overlap of the agreement zone, and the consumers negotiate with providers of various strategies and deadlines. Finally, the third experimental set consists of cases where consumers deal with providers who adopt a variety of concession strategies and reservation values, resulting to various overlaps of agreement zones. 10.201 different negotiation scenarios yield from the first experimental set, 90 from the second and 2827 from the third.

Table 5: Values of negotiating parameters of the 3 experimental sets

Exp. / Param.	Set 1		Set 2		Set 3	
	Consumer	**Provider**	**Consumer**	**Provider**	**Consumer**	**Provider**
Price$_{(min)}$	0	0	0	0	0	0
Price$_{(max)}$	[0:1:100]	100	100	100	[0:1:100]	100
T_{max}	100	[100:1:200]	100	[100:20:200]	100	200
Strategy	BD	TD (β=1)	BD	TD β=[0.1:0.1:0.9, 1:2:11]	BD	TD β=[0.1:0.05:1, 2:1:10]

Negotiations are conducted between non-learning agents, following the protocol discussed in [25]. Data collected from each negotiation instance consists of the alternate offers exchanged by the two agents (negotiation thread). The threads are separately collected for each experimental set and are used for the formation of training, validation and test set. The number of counterpart's past offers which will be accounted for prediction, rations the number of neurons in the input layer. Determining the network architecture therefore lies on determining the number of input features as well as the number of hidden neurons (one hidden layer is used as it is also assumed in existing systems). One common practice for such decision is through empirical search. Data from each experimental set is split to training, validation and test sets in a proportion of 70:15:15. The optimal number of neurons in the hidden layer as well as the number of opponents' past offers which constitute the networks' input, is such that minimizes the MSE (eq. 14) of the test set. Three up to twenty five past offers were tested for the networks' input, and one up to twenty five neurons were tested for the construction of the hidden layer. For each input-hidden node combination, ten different runs of MLP training were conducted using the LM method and the average MSE of the test set was computed.

From the experiments we conclude that the simplest networks which yield very low MSE of the test set, consist of three hidden and four input nodes (MLP1) when data from the first experimental set is used, four hidden and three input nodes (MLP2) when data from the second experimental set is used and two hidden and three input nodes (MLP3) when data from the third experimental set is used. PTA1, PTA2 and PTA3 are the generated PTAs that use MLP1, MLP2 and MLP3 respectively.

8.3.2 Comparison of PTAs and SSLAs

This section focuses on the comparison of SSLA and the three PTAs, in settings different from those accounted for the generation of the training data. Each agent negotiates with a non-learning counterpart, following the risk-seeking predictive mechanism (RP=100%) discussed in section 5.2.1.

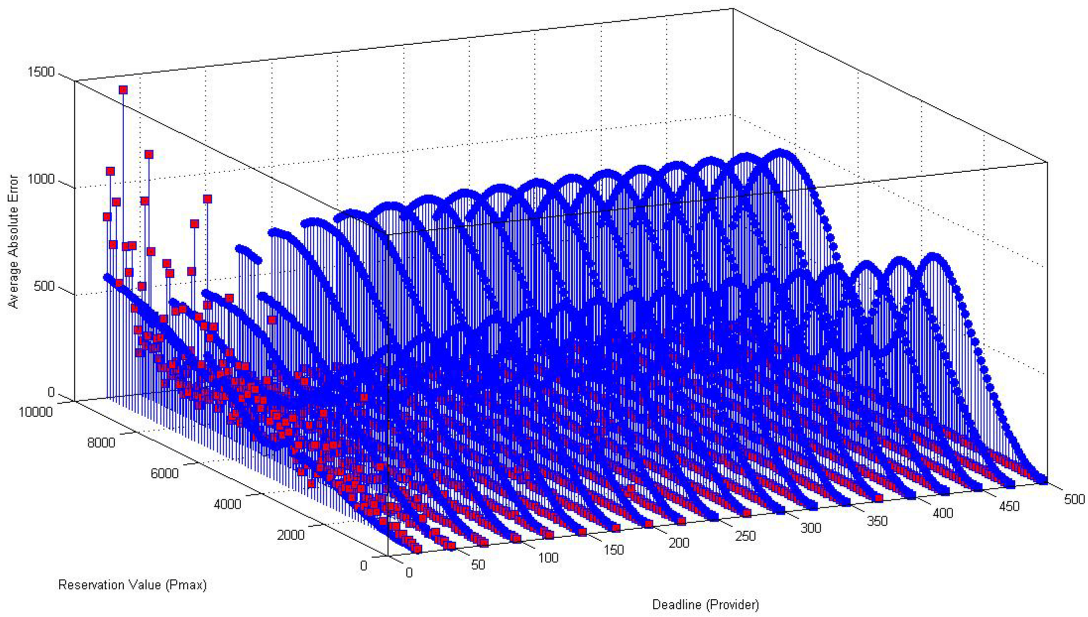
The forecasting method employed by negotiating agents should be highly accurate, fast and with low memory requirements. Since MLPs in PTAs are only trained once before initiation of the process, while MLPs in SSLAs are trained at each negotiation step, PTAs are faster than SSLAs. However, the time required for training the MLP of an

SSLA with the LM method is a few milliseconds, and it is still considered efficient. Furthermore, the SSLA has a small MLP and has lower storage requirements than the PTA. Focus is set on measuring the accuracy of intermediate predictions, which will be used as the evaluation measure. In each negotiation round the absolute error defined as the difference between the prediction and the actual value (eq. 12) is saved. Assessment is provided through the computation of the mean of the absolute errors and other statistical information (standard deviation and maximum value) in each negotiation instance. The purpose of the comparison is to illustrate the deviation of the error as the agents negotiate in new settings and highlight the ability of SSLA to provide more accurate predictions. Three sets of experiments are conducted in order to test the performance of each PTA and compare with the SSLA. Parameters of the negotiation environments used in the competitions are illustrated in Table 6.

Table 6: Experimental settings to test the behavior of PTAs as distributions of real data deviate those used for training their ANNs

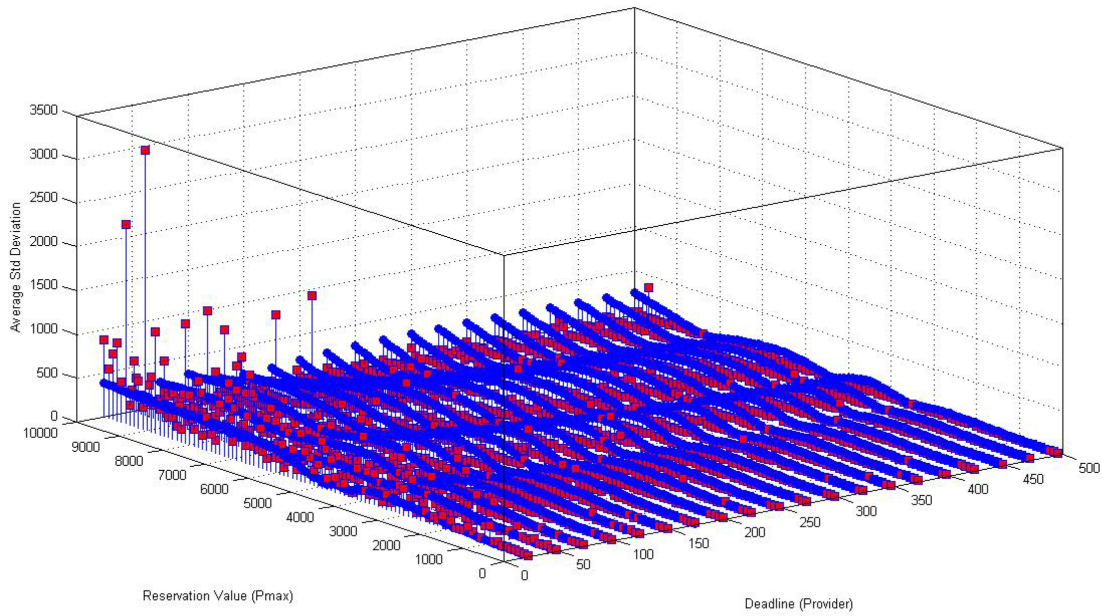
Exp. / Param.	Set 1		Set 2		Set 3	
	Consumer	**Provider**	**Consumer**	**Provider**	**Consumer**	**Provider**
Price$_{min}$	0	0	0	0	0	0
Price$_{max}$	[100:100:10000]	[100:100:10000]	[100:100:10000]	[100:100:10000]	[100:100:100000]	[100:100:100000]
t_{max}	100	[25:25:500]	100	100	100	200
Strategy	BD	TD (β=1)	BD	TD (β=[0.1:0.1:0.9, 1:1:11])	BD	TD (β=[0.1:0.4:0.9, 1:1:25])

As expected, SSLA outperformed the three PTAs, emphasizing the need to develop agents who extract the training set during the negotiation discourse. Figure 27 illustrates a comparison of the SSLA with each PTA depicting the stem plots of the mean, the standard deviation, and the maximum value of the absolute errors in each negotiation instance, with respect to the providers' deadline and reservation value. Statistical measures of the PTAs are depicted with blue circles and of the SSLA with red squares.



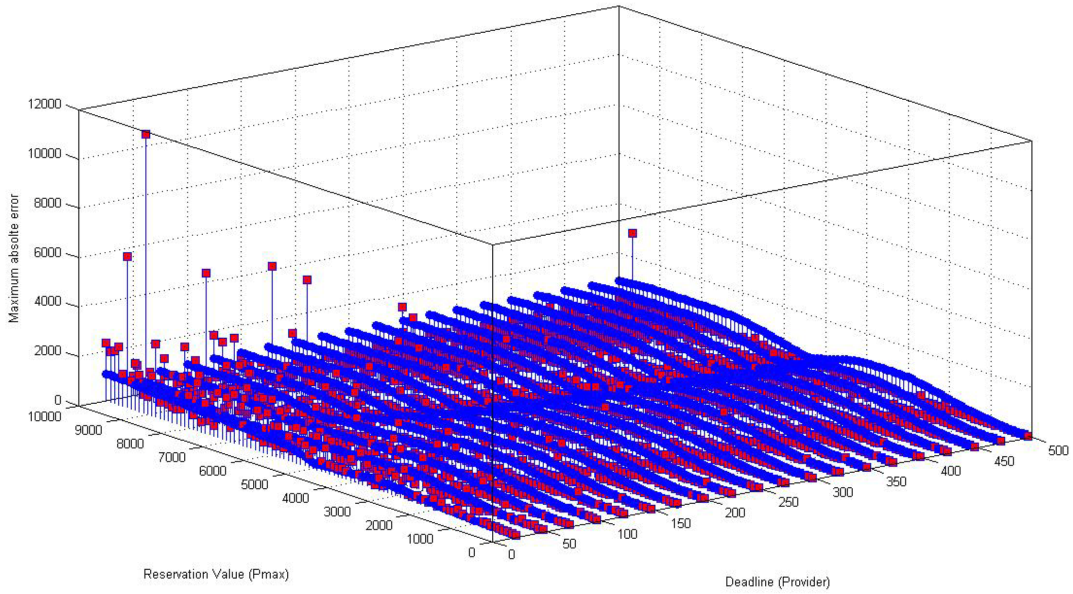
1500
1000
500
0
Average Absolute Error
10000
8000
6000
4000
2000
0
Reservation Value (Pmax)
0
50
100
150
200
250
300
350
400
450
500
Deadline (Provider)

(a)

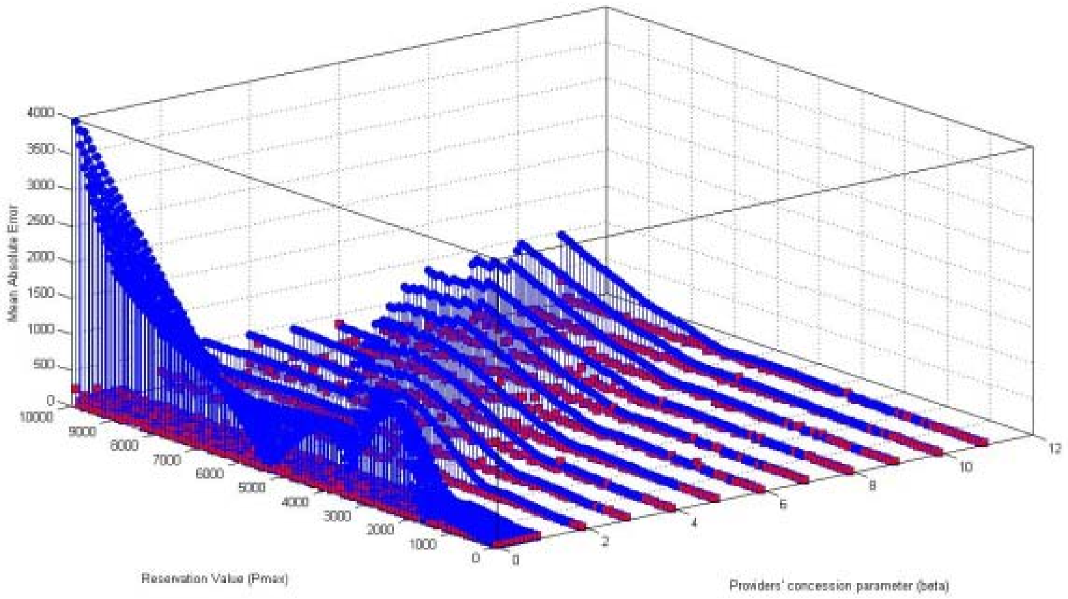


3500
3000
2500
2000
1500
1000
500
0
Average Std Deviation
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Reservation Value (Pmax)
0
50
100
150
200
250
300
350
400
450
500
Deadline (Provider)

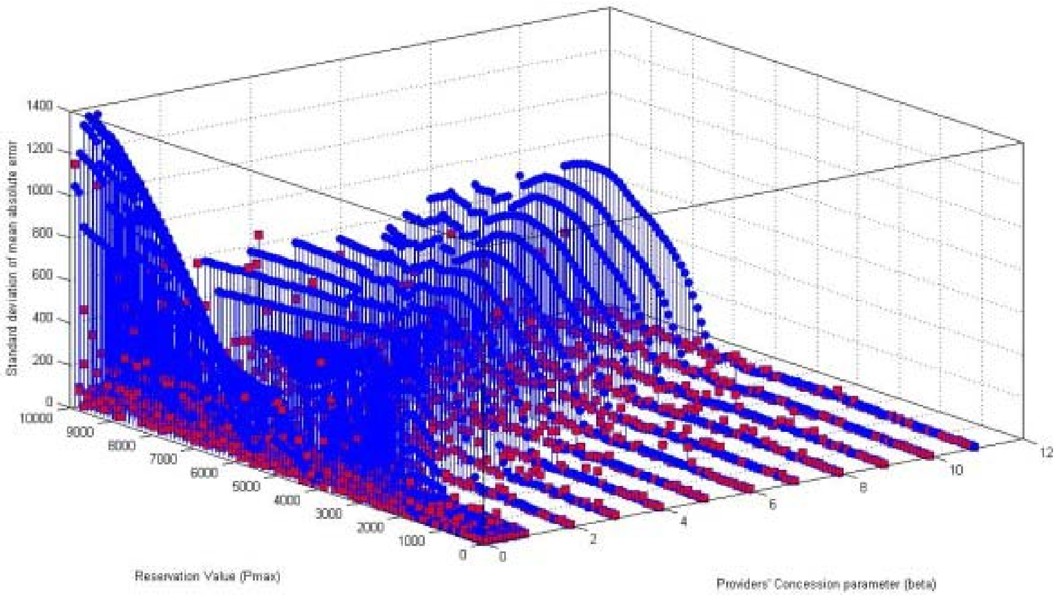
(b)


Maximum absolte error
Reservation Value (Pmax)
Deadline (Provider)

(c)

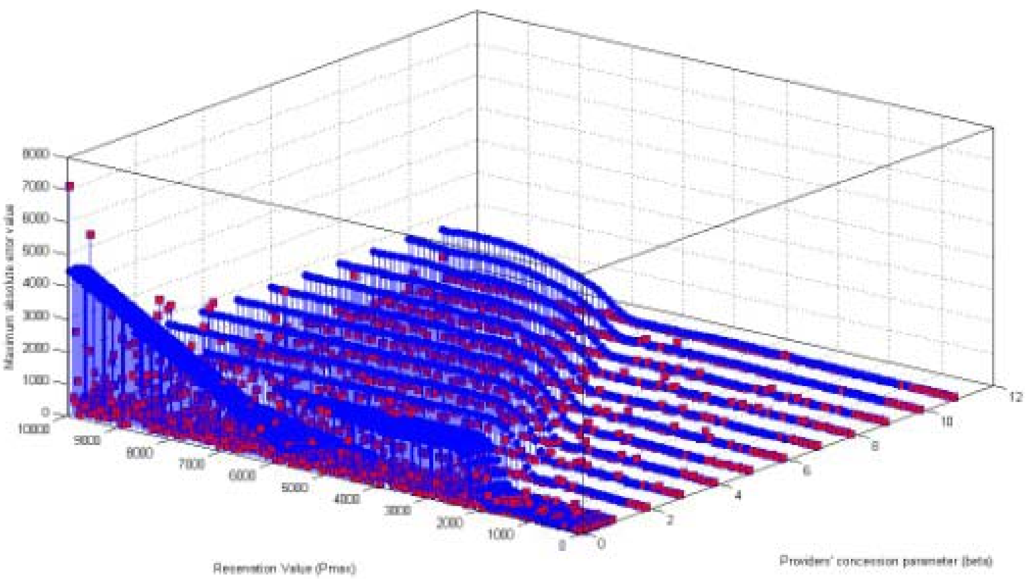

Mean Absolute Error
Reservation Value (Pmax)
Providers' concession parameter (beta)

(d)



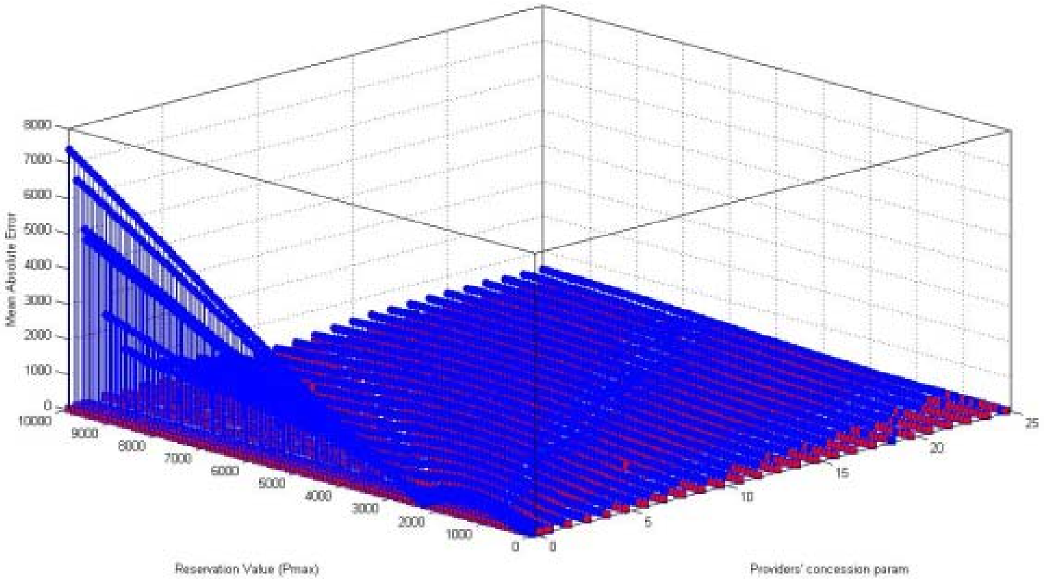
Standard deviation of mean absolute error
1400
1200
1000
800
600
400
200
0
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10
12
Reservation Value (Pmax)
Providers' Concession parameter (beta)

(e)

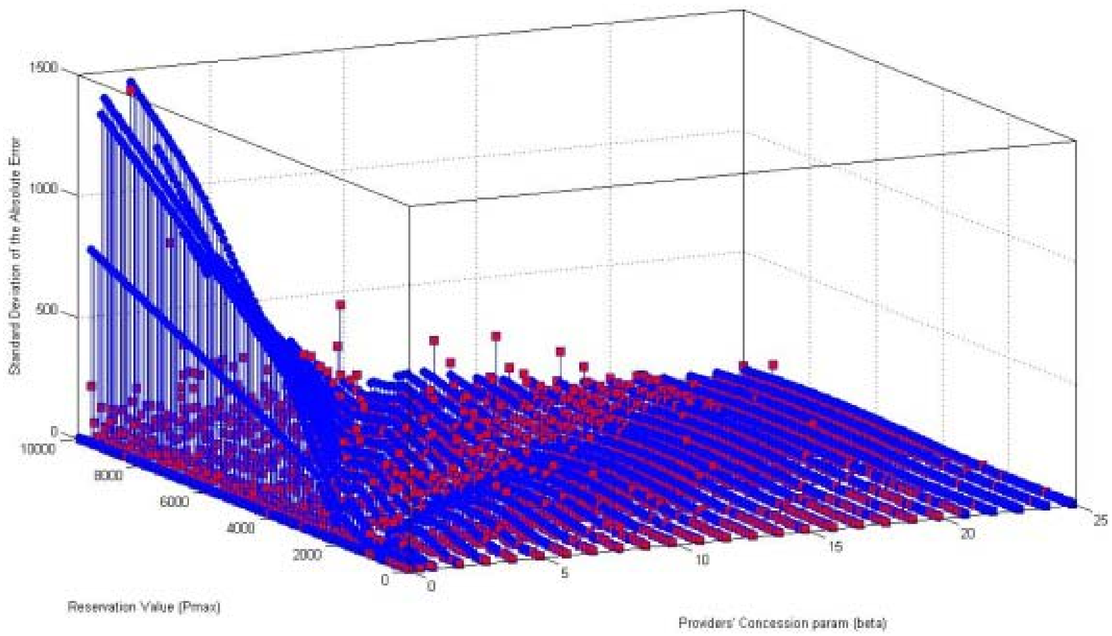


Maximum absolute error value
8000
7000
6000
5000
4000
3000
2000
1000
0
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
2
4
6
8
10
12
Reservation Value (Pmax)
Providers' concession parameter (beta)

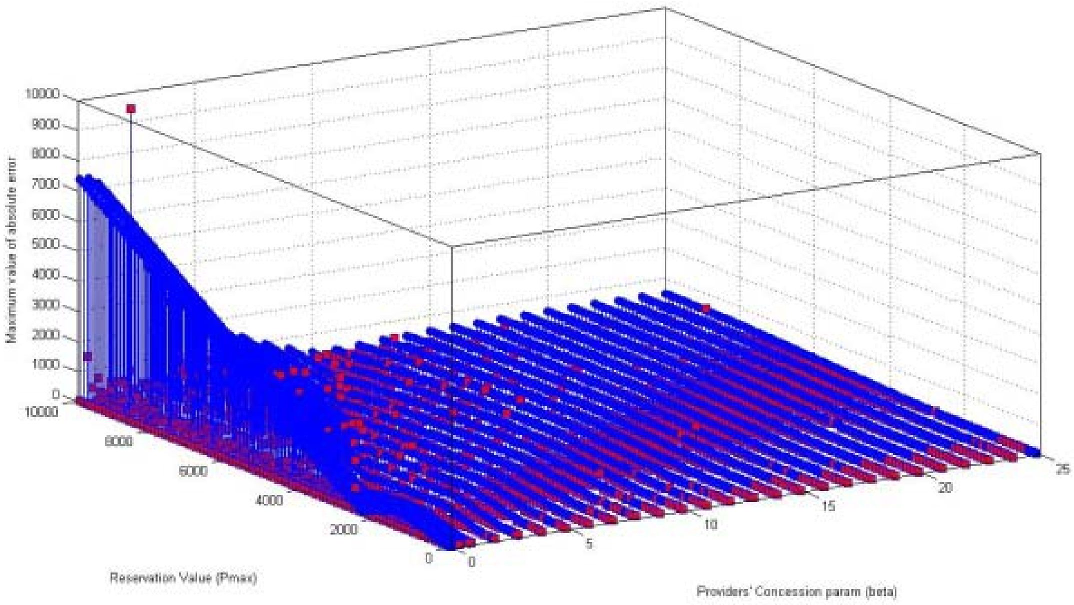
(f)


Mean Absolute Error
Reservation Value (Pmax)
Providers' concession param

(g)


Standard Deviation of the Absolute Error
Reservation Value (Pmax)
Providers' Concession param (beta)

(h)



(i)

Figure 27: Comparison between SSLA and the three PTAs (blue circles illustrate results of the PTA and red squares results of the SSLAs). Stem plots of the mean of the absolute errors when PTA uses ANN1(a), ANN2 (d), and ANN3 (g). Stem plots of standard deviation of the absolute errors when PTA uses ANN1(b), ANN2 (e), and ANN3 (h). Stem plots of the maximum of absolute errors when PTA uses ANN1(c), ANN2(f), and ANN3(i)

The continuous variables MeanSSLA and MeanPTA1, indicating the mean value of the absolute errors in each negotiation instance of SSLA and PTA1 respectively, were tested if they follow the normal distribution using one-sample Kolmogorov-Smirnov non-parametric test. In both cases the normality assumption is violated at a significance level of a=0.05. A way to compare the means of MeanSSLA and MeanPTA1 is by using Kolmogorov-Smirnov non-parametric tests for two independent samples. In both tests p-value<0.05, therefore there is a statistically significant difference for the error means between SSLA and PTA1 agents (MeanSSLA is significantly less than MeanPTA1). The same procedure was followed with PTA2 and PTA3. In both cases it is proved that there is a statistically significant difference between SSLA and the two PTAs (MeanSSLA is significantly less than MeanPTA2 and than MeanPTA3).

Although the SSLA is generally more accurate than the PTAs in all experimental sets, as the mean of the absolute errors is reduced by 92.67%, it does not yield satisfying results in negotiations with short deadline. This is due to the small size of the training data set compared to the number of parameters of the neural network that need to be learned. The incorporation of an optimization technique, genetic algorithm, for the selection of the networks' architecture to address this issue is discussed in Chapter 9, with the introduction of an Adaptive Session-long Learning Agent (ASLA).

9. OPTIMIZING MLP ARCHITECTURE

Since the size of the dataset and the error bound given in (35) is changing in each round, it is desirable to let off the static and investigate more dynamic structures. Hereby we address the second issue discussed in Chapter 8, concerning optimization of the MLP applied by the SSLA. The novel aspect of this research lies again in the field of applying Neural Networks in negotiations. We introduce the Adaptive Session-Long Learning Agent (ASLA) who optimizes its structure and subset of input features during the negotiation discourse with the use of a genetic algorithm. This contradicts the case of the SSLA where only a fixed number of the opponent's previous offers are considered. In section 9.1 we discuss how genetic algorithms can be combined with Neural Networks and in section 9.2 we introduce the ASLAs. Finally in section 9.3 we compare ASLAs with SSLAs introduced in the previous chapter.

9.1 Genetic algorithms to optimize MLP architectures

As mentioned in Chapter 4, Genetic algorithms, are stochastic, population-based search and optimization techniques based on principles of evolution. The decision variables are coded into solution strings of finite length over an alphabet of certain cardinality. These strings are termed individuals or chromosomes, and the characters that comprise them are termed genes. For each solution, a method that assigns a fitness level is applied in order to distinguish preferred from bad solutions. The essence of genetic algorithms is to evolve the solutions of each population with the use of genetic operators. In a simple genetic algorithm, an initial population of solutions is randomly generated. Selection methods are applied and the most promising individuals of the population are placed in a mating pool. Genetic operators, such as crossover, inversion and mutation, are further applied and the evolved solutions comprise the next generation. The same procedure is repeated until some convergence criteria, usually related with the establishment of equilibrium among the solutions, are met [133].

Genetic algorithms have been widely combined with neural networks as a means of optimizing the networks' structure, parameters, or input features. The interested reader may refer to [134] for applications of GAs for feature subset selection, to [135] [136] [137] [138] [139] [140] [141] for specification of optimal parameters such as interconnection weights or training parameters, and to [142] [143] [144] for applications where genetic algorithms search the optimal topology of a Neural Network. Individuals may encode information from all of the above problem spaces, and therefore GAs may yield solutions which combine optimization of networks' parameters, subset of features and/or structure [145] [146]. Since we are most concerned with defining the subset of input features as well as the network topology of a session-long learning agent, we continue with a brief description of utilizing GAs to solve the problems at hand. The main issues under consideration are the coding scheme and the factors which are related to the fitness function.

In the problem of subset selection, candidate solutions are represented by binary vectors in m dimensional space, where m is the number of potential features. Each gene, represented by a bit 1 or 0, indicates the existence or non-existence of a particular input feature, and the fitness assigned to the individual is related to the networks' accuracy if the corresponding subset is used [134]. Alternatively, in cases where the search involves finding the number of past values which will constitute the input of a time series predictor, genes represent the binary values of this number. Moving to the realm of evolving the network's architecture, one of the key issues is to decide how much information will be encoded in a chromosome [147]. On one hand, each individual must contain detailed information about interconnections between nodes, number of intermediate layers and neurons. This representation scheme is also termed direct encoding and may result to infinitely large search spaces, if we consider the number of potential network topologies and sizes. On the other hand, in indirect encoding, only some characteristics of the network's architecture are included. One way of performing indirect encoding is through parametric representation, where a set of network parameters, such as number of hidden neurons or layers, are included in the individual. Input feature selection and optimization of the networks' topology through parametric representation are often combined in the solution string, enabling simultaneous evolution of the input space and the networks' structure. In [148] [149] [150] [151] [152] a neural network with one hidden layer is assumed and a genetic algorithm to search optimal subset of input features and number of hidden neurons is used.

In all aforementioned systems, the mean square error (MSE) is a fitness-related factor and for this reason evaluating each individual presupposes construction of the corresponding neural network and computation of the MSE. In this work, a pseudocode for evolving the network's architecture is considered based on the implementation of a simple genetic algorithm [153] and the typical cycle of evolution of architectures [154]:

Step 1. Randomly generate the initial population P

Step 2. Decode each individual (chromosome) into an architecture

Step 3. Evaluate individuals:

Train each network with a predefined training algorithm and parameters

Define the fitness of each individual according to the training result and other performance criteria, such as the complexity of the architecture

Repeat

Step 4. Select a set of promising individuals and place them in the mating pool

Step 5. Apply crossover to generate offspring individuals

Step 6. Apply mutation to perturb offspring individuals

Step 7. Replace P with the new population

Step 8. Evaluate all individuals in P (as in step 3)

Until certain termination criteria are met

In the next section this algorithm is applied to the case of Session-Long Learning Agents.

9.2 Introducing adaptive session-long learning agents (ASLAs)

As mentioned earlier, the available information to the predictive agent (*Con*) at time *t* is formed by the negotiation thread $\{X^{0}_{(Con\to \mathrm{Pr})}, X^{1}_{(\mathrm{Pr}\to Con)}, ..., X^{t-1}_{(\mathrm{Pr}\to Con)}\}$, which consists of subsequent offers exchanged by the two agents *Con* and *Pr* up to that time. Unlike SSLA, the ASLA considers not only the series of his counterpart's past offers, but also the series of his own past offers, to formulate the subset of input features. Particularly, in order to find the optimal subset which will guide the prediction, two time series are taken into account: one resulting from past offers of the predicting agent $\{X^{0}_{(Con\to \mathrm{Pr})}, X^{2}_{(Con\to \mathrm{Pr})}, ..., X^{t-2}_{(Con\to \mathrm{Pr})}\}$, and one resulting from the past offers of the opponent $\{X^{1}_{(\Pr \to Con)}, X^{3}_{(\Pr \to Con)}, ..., X^{t-1}_{(\Pr \to Con)}\}$. The encoded information represents the number of previously offered values of each agent. Using a binary grammar, three bits are sufficient to encode up to seven past offers for each agent. Consequently a 6-bit length string represents the subset of input features. Since it has been proved that an MLP with one hidden layer can conduct function approximation, and since it has been widely used by existing predicting agents, the architecture of a two layered MLP is assumed, and focus is set on searching the optimal number of hidden units. In an attempt to keep the network small, three bits are used for the representation of the hidden units, resulting to a chromosome of nine bits which simultaneously evolves the subset of input features and architecture of the network (Figure 28).

$$\underbrace{010}_{\substack{number.of\\ predicting\\ agents.past\\ offers}} \quad - \quad \underbrace{101}_{\substack{Number.of\\ opponents'\\ past.offers}} \quad - \quad \underbrace{001}_{\substack{Numberof\\ hidden.neurons}}$$

Figure 28: A chromosome consisting of 9 bits is used to evolve the input subset and the number of hidden neurons of the neural network

The ASLA applies the algorithm illustrated in 9.1 and appropriately adjusts the architecture of the employed MLP. Every time the genetic algorithm is run, the agent selects the MLP with the lowest fitness function. He then applies the MLP to forecast his counterpart's response in a similar way to that of the SSLA.

More specifically, the ASLA initially generates a random population of individuals (Step 1). Each individual is translated to the respective MLP (Step 2) which is then trained and evaluated (Step 3).

The training patterns are extracted from the current negotiation thread. If the available number of previous predicting agent's offers at decision making time *t* is m, and for opponent's offers is n, where m,n є {0,1,…,t/2} and m+n>0, the first input-output example is extracted at time $t' = \begin{cases} 2m+2, if\ 2m-2n-1>0 \\ 2n+2, if\ 2m-2n-1<0 \end{cases}$,

and the size of the available dataset at time *t* is $|Dataset| = 1 + \frac{t-t'}{2}$.

As far as the objective (fitness) function is concerned, since $|Dataset|$ must be at least 2 to apply the LM method, the ASLA favors solutions that result to $|Dataset|$ ≥2. Furthermore, in cases where it is possible to divide the available data in three sets (training, validation and test set), the objective (fitness) function, which is minimized through the GA solver, is proportional to the MSE of the test set. Preference is given to solutions which result to more data patterns, in order to apply an early stopping learning method, which guarantees better generalization.

After evaluation, the most promising individuals are placed in the mating pool (Step 4), and GA operators are applied (Steps 5 and 6) to formulate the new population (Step 7). The new individuals are in turn evaluated (Step 8) and the process is repeated for 10 generations. The trained MLP that yields from the most promising individual is applied for the purpose of forecasting the counterpart's next offer.

It is important to note that implementation of ASLA advances the state of the art in the field of applying Neural Networks in negotiations to predict the counterpart's responses.

It is based on an optimization technique and illustrates a pathway of finding a sub-optimal structure and subset of input features for the network. It could be used as a reference point in the development of other forecasting tools that assist negotiators. Additionally, it is a way of addressing the issue of heterogeneity of existing systems when it comes to selecting the offers of the negotiation thread which will constitute the input of the forecasting tool. Lack of uniformity in the considered subset of input features is evident in Table 4 of Section 7.4.

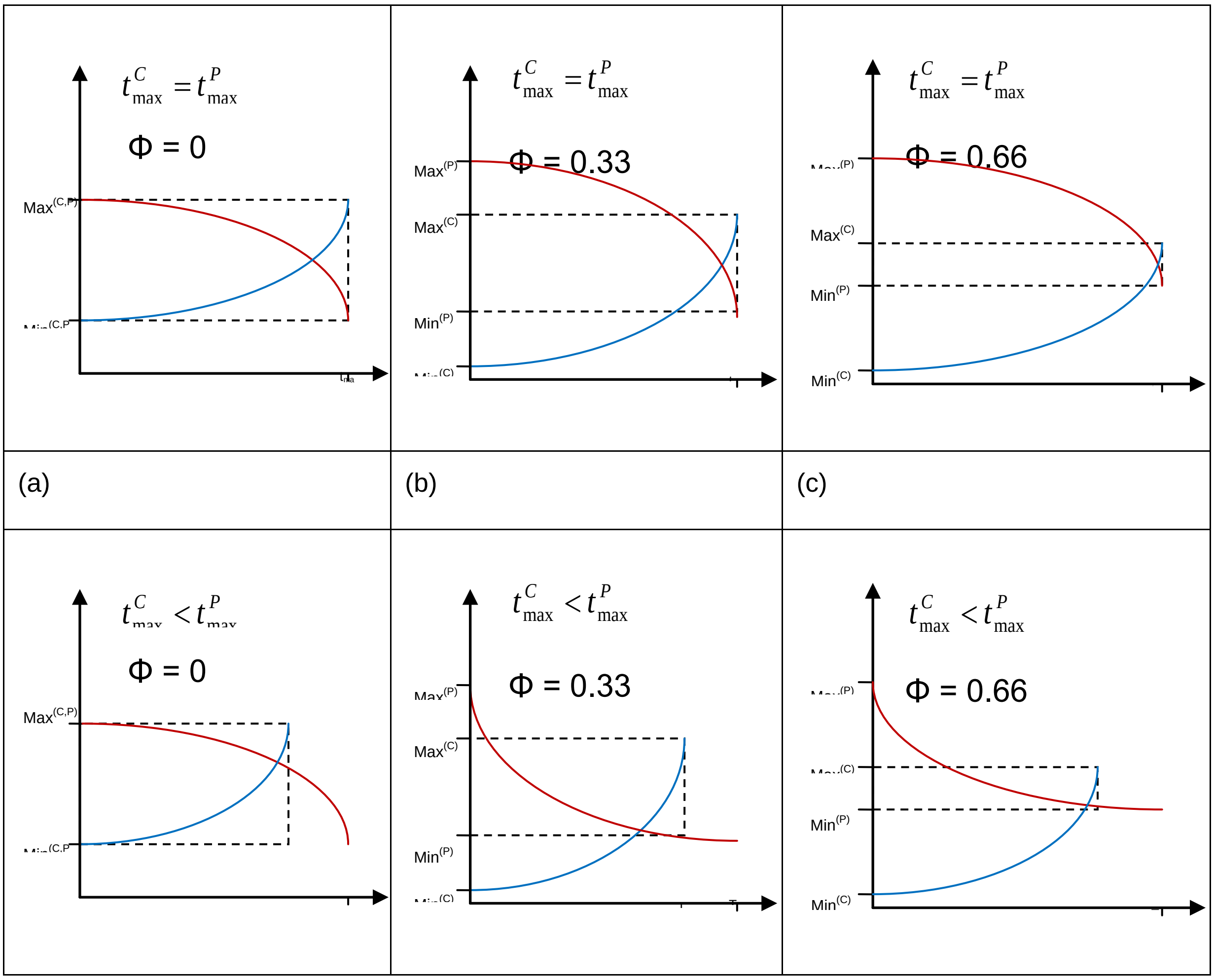
Optimization is expected to reduce oscillations around the mean error and not yield very high error values, which may mislead the involved agent. In the following section SSLAs and ASLAs are compared.

9.3 Comparison of SSLAs and ASLAs

A variety of negotiation scenarios are considered for the comparison of SSLAs and ASLAs. In the first sub-section details concerning the negotiation settings are illustrated, while in the next sub-section results are presented and discussed.

9.3.1 The negotiation settings

For the generation of negotiation environments nine different negotiation scenarios are considered with respect to the overlap in agreement zones and available time to each negotiator, as in [155]. The scenarios involve single-issued negotiations between provider and consumer agents and are depicted in Figure 29:



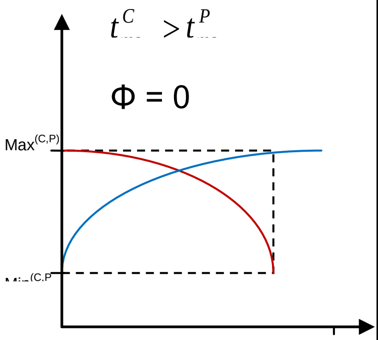
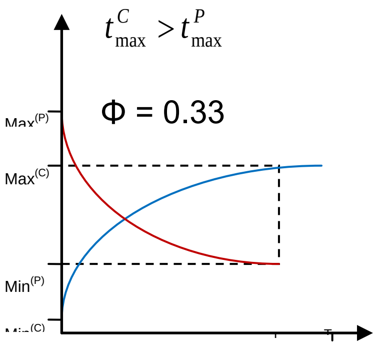
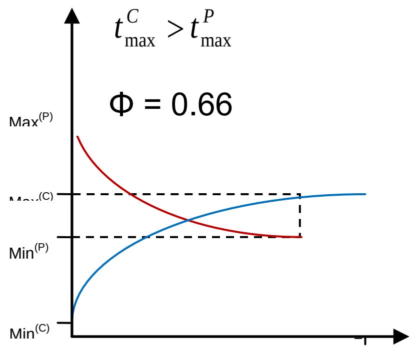
(d)	(e)	(f)
		
(g)	(h)	(i)

Figure 29: Negotiating scenarios used to test the session-long learning agents (a) Scenario 1: Equal Deadline and full overlap, (b) Scenario 2: Equal Deadline and partial overlap (Φ=0.33), (c) Scenario 3: Equal Deadline and partial overlap (Φ=0.66), (d) Scenario 4: Pr. with higher deadline and full overlap, (e) Scenario 5: Pr. with higher deadline and partial overlap (Φ=0.33), (f) Scenario 6: Pr. with higher deadline and partial overlap (Φ=0.66), (g) Scenario 7: Cons. with higher deadline and full overlap, (h) Scenario 8: Cons. with higher deadline and partial overlap (Φ=0.33), (i) Scenario 9: Cons. with higher deadline and partial overlap (Φ=0.66)

For each scenario TD and BD producer's strategies are considered. As mentioned earlier, TD strategies represent many different types of concession curves with respect to reseeding time, and BD strategies represent counterparts following imitative tactics. The Relative Tit-For-Tat family measuring the average concession of the opponent agent the last Window steps is used in the BD strategies. Experimental parameter values are outlined in Table 7.

Table 7: Values of parameters covering the nine negotiation scenarios

Scenarios / Parameters		**Scenario 1**		**Scenario 2**		**Scenario 3**	
		Consumer	**Producer**	**Consumer**	**Producer**	**Consumer**	**Producer**
Price$_{min}$		0	0	0	33	0	66
Price$_{max}$		100	100	100	133	100	166
t$_{max}$	**TD**	[50:50:350]	Equal to Cons.	[50:50:350]	Equal to Cons.	[50:50:350]	Equal to Cons.
	BD	[50:100:350]	Equal to Cons.	[50:100:350]	Equal to Cons.	[50:100:350]	Equal to Cons.
Strategy	**TD**	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])

	BD	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])
Scenarios / Parameters		**Scenario 4**		**Scenario 5**		**Scenario 6**	
		Consumer	**Producer**	**Consumer**	**Producer**	**Consumer**	**Producer**
$Price_{min}$		0	0	0	33	0	66
$Price_{max}$		100	100	100	133	100	166
t_{max}	**TD**	[50:50:350]	375	[50:50:350]	375	[50:50:350]	375
	BD	[50:100:350]	375	[50:100:350]	375	[50:100:350]	375
Strategy	**TD**	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])
	BD	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])
Scenarios / Parameters		**Scenario 7**		**Scenario 8**		**Scenario 9**	
		Consumer	**Producer**	**Consumer**	**Producer**	**Consumer**	**Producer**
$Price_{min}$		0	0	0	33	0	66
$Price_{max}$		100	100	100	133	100	166
t_{max}	**TD**	375	[50:50:350]	375	[50:50:350]	375	[50:50:350]
	BD	375	[50:100:350]	375	[50:100:350]	375	[50:100:350]
Strategy	**TD**	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])	(β = 0.1)	(β =[0.1:0.2:0.9, 1:4:29])
	BD	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])	TD (β = [0.1,1,20])	BD (Window = [1,3,5,7,10])

The SSLA and the ASLA negotiate under the different scenarios, following the decision rule described in 5.2.1. Apposition of the results is provided in the following paragraph.

9.3.2 Experimental results

Performance of the forecasting method is evaluated in terms of accuracy of intermediate predictions. ASLAs are not as fast as SSLAs and have higher storage requirements. However, the main concern here is to investigate the optimization of the MLP structure so as to increase the accuracy of the forecasting tool and study the effect it has on the negotiation outcome. The ASLA is expected to yield better results and can be used as a reference point when testing the accuracy of other forecasting tools in negotiations.

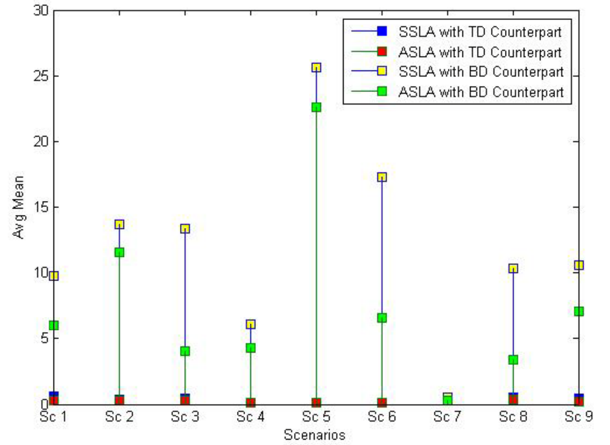
It is a fact that the ASLA is a smoother predictive model as it proves more accurate with decreased standard deviation and maximum error values. A number of experiments are conducted to cover the scenarios described in 9.3.1. For each step of the consumer involving estimation of the counterpart's next value, the absolute difference between the actual offer of the provider and the prediction is measured (absolute error given in eq. 12 of Chapter 7). At the end of each negotiation, the mean, the standard deviation and the maximum value of the absolute errors are computed. Summarized statistics for each scenario are further acquired with the computation of average and maximum values. More specifically, AvgMean in Table 8 refers to the mean of the mean errors computed in each negotiation of a particular scenario and Max Mean refers to the maximum of the mean errors. Accordingly, AvgStd and MaxStd refer to the average and maximum standard deviation observed, and finally AvgMax and Highest Max stand for the average and maximum of the highest error values acquired in negotiations of each scenario. Detailed results with respect to the negotiation scenario and strategy of the opponent are illustrated in Table 8. In the same table "Inst." indicates number of negotiation instances that result from each particular scenario and "Pr. Used" indicates the number of negotiation instances where the predictive agent (SSLA or ASLA) made use of the forecasting tool.

Table 8: Average and Maximum values of statistic measures acquired at each scenario

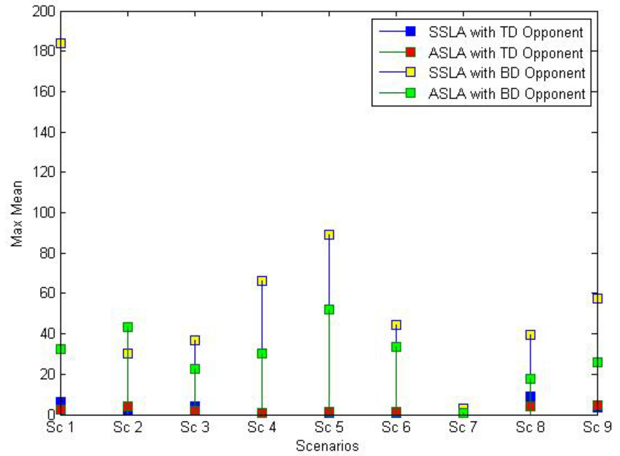
Measures / Scenario/ Strat.		Inst.	Pr. Used		Avg Mean		Max Mean		Avg Max		Highest Max		Avg Std		Max Std	
			SSLA	ASLA	SSLA	ASLA	SSLA	ASLA	SSLA	ASLA	SSLA	ASLA	SSLA	ASLA	SSLA	ASLA
Sc 1	TD	91	82	89	0.58	0.25	6.22	2.57	6.66	1.17	102.42	7.72	1.21	0.37	22.66	2.54
	BD	60	36	36	9.73	5.99	183.92	32.25	110.85	38.46	1029.5	195.36	21.28	9.57	346.99	62.14
Sc 2	TD	91	91	91	0.35	0.28	2.11	3.83	5.22	4.29	36.28	34.93	0.74	0.71	4.45	7.90
	BD	60	36	36	13.68	11.53	30.36	43.28	78.3	66.37	213.3	169.25	19.15	16.22	51.02	34.68
Sc 3	TD	91	91	91	0.48	0.28	4.12	2.16	10.95	7.29	223.14	127.14	1.49	0.97	26.77	15.19
	BD	60	56	56	13.38	4.08	36.81	22.68	92.17	56.66	411.51	131.08	20.17	10.63	91.78	32.65
Sc 4	TD	91	91	91	0.16	0.12	0.58	0.63	2.69	3.11	17.44	14.02	0.38	0.42	2.37	1.73
	BD	60	36	36	6.07	4.31	66.08	30.22	46.69	20.32	415.53	106.04	11.41	5.69	132.59	38.65
Sc 5	TD	91	91	91	0.16	0.12	0.66	1.13	3.26	2.94	39.79	22.64	0.42	0.42	4.38	4.93
	BD	60	56	56	25.64	22.64	89.18	52.16	115.48	81.43	517.43	264.78	36.65	28.64	128.93	68.11

Sc 6	TD	91	91	91	0.16	0.12	0.76	1.22	4.18	3.64	100.59	51.73	0.47	0.49	7.69	7.61
	BD	60	56	56	17.26	6.61	44.38	33.26	132.9	62.15	442.7	188.92	26.26	13.66	71.23	40.96
Sc 7	TD	91	91	91	0.4	0.28	3.15	2.57	5.63	4.64	68.81	26.4	0.81	0.72	8.2	5.14
	BD	60	56	56	0.53	0.31	3.19	0.7	10.84	4.80	107.89	13.39	1.48	0.71	13.87	1.92
Sc 8	TD	91	91	91	0.51	0.34	8.9	3.98	10.28	6.82	334.81	47.96	1.5	0.98	49.85	7.27
	BD	60	56	56	10.33	3.39	39.4	17.73	69.09	46.09	677.07	98.5	15.83	9.06	90.69	32.36
Sc 9	TD	91	91	91	0.43	0.24	3.62	4.67	11.10	3.74	476.61	38.64	1.29	0.61	36.54	10.81
	BD	60	56	56	10.6	7.09	57.55	25.64	144.31	61.78	3050	145.83	20.84	12.54	285.28	33.9
Totals	TD	819	810	817	0.36	0.23	8.9	4.67	6.66	4.18	476.61	127.14	0.92	0.63	49.85	15.19
	BD	540	444	444	11.91	7.33	183.92	52.16	88.96	48.67	3050	264.78	19.23	11.86	346.99	68.11
Overall	TD,BD	1359	1254	1261	6.13	3.78	183.92	52.16	47.81	26.42	3050	264.78	10.07	6.24	346.99	68.11

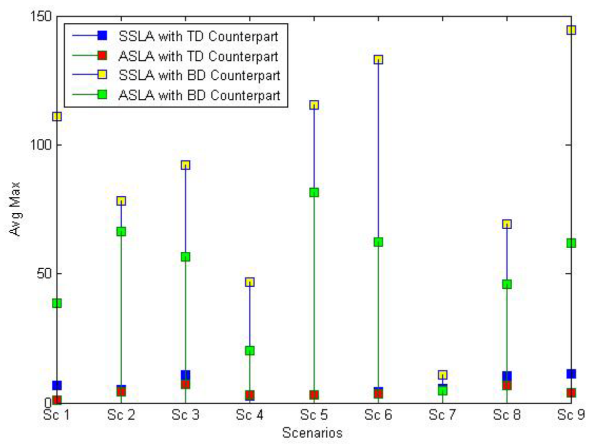
For a more convenient illustration of the results, the comparison of AvgMean, MaxMean, AvgMax, HighestMax, AvgStd and MaxStd incurred to SSLA and ASLA, in cases where counterpart adopts TD and BD strategies, is depicted in Figure 30 (a),(b),(c),(d),(e) and (f) respectively. The values of the SSLA negotiating with a TD and a BD counterpart are illustrated with blue and yellow squares, and the values of the ASLA negotiating with a TD and a BD counterpart are illustrated with red and green squares respectively.



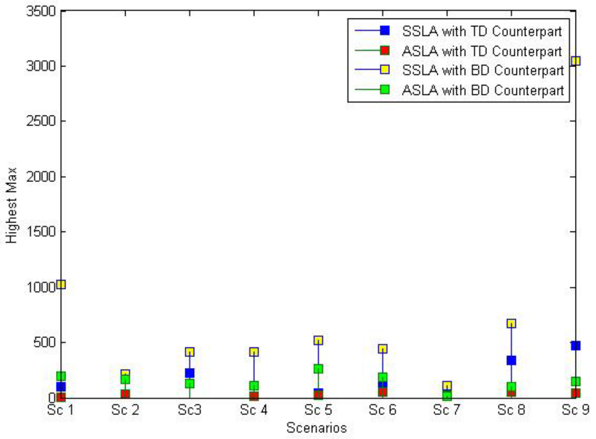
(a)



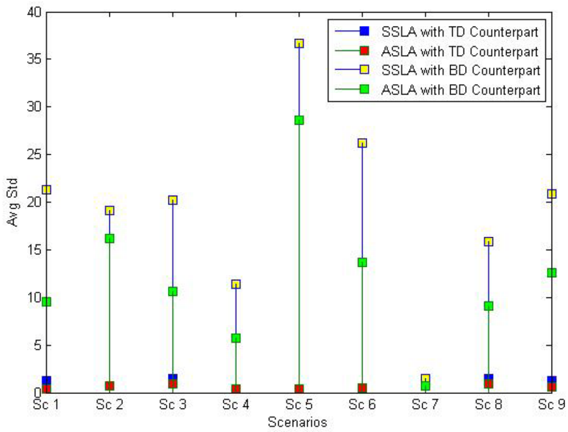
(b)



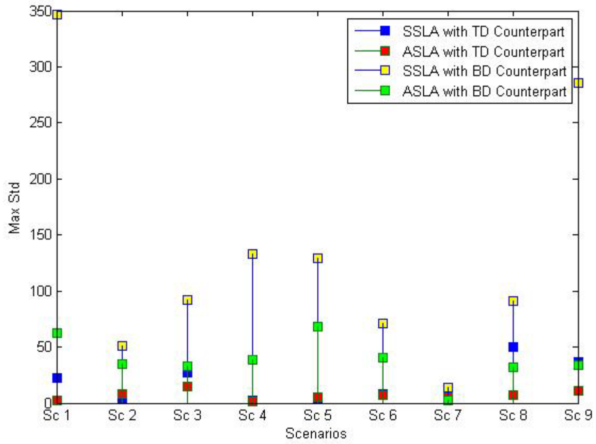
(c)



(d)



(e)



(f)

Figure 30: The values of AvgMean, Max Mean, Avg Max, Highest Max, Avg Std and Max Std in each scenario

The ASLA is shown to be more accurate in the general case since it yields reduction of the mean of absolute errors (AvgMean) by 38.34%, reduction of AvgMax by 44.75% and reduction of AvgStd by 38.03%.

More specifically, when the ASLA deals with counterparts following TD strategies the same measures (AvgMean, AvgMax and AvgStd) are reduced by 36.11%, 37.24%, and 31.52% respectively, while when he deals with counterparts following BD strategies AvgMean, AvgMax and AvgStd are reduced by 38.45%, 45.29%, and 38.32%.

The continuous variables MeanSSLA_TD and MeanASLA_TD, indicating the mean value of the absolute errors in each negotiation instance of SSLA and ASLA when facing an opponent with a time dependent strategy respectively, were tested if they follow the normal distribution using one-sample Kolmogorov-Smirnov non-parametric test. In both cases the normality assumption is violated at a significance level of a=0.05. The means of MeanSSLA_TD and MeanASLA_TD are compared by using Kolmogorov-Smirnov non-parametric tests for two independent samples. In both tests p-value<0.05, therefore there is a statistically significant difference for the error means between SSLA and ASLA agents (MeanASLA_TD is significantly less than MeanSSLA_TD). The same procedure was followed with SSLA and ASLA when facing an opponent with behavior dependent strategy. It is proved that AvgMean of ASLA is significantly less than AvgMean of SSLA.

Similarly, it is proved that AvgMax and AvgStd of an ASLA are significantly less than AvgMax and AvgStd of the SSLA respectively, in all cases (both when counterparts adopt time dependent and behavior dependent strategies).

SSLAs and ASLAs can be safely used in cases where the counterpart's strategy can be expressed by continuous functions. In the scenarios described, these are the cases with TD strategies, yielding to SSLA and ASLA AvgMean of 0.36% and 0.23%, AvgMax of 6.66% and 4.18%, and AvgStd of 0.92% and 0.63%.

On the contrary, when opponents' behavior is sharp (as is the case in BD strategies), neural networks are less accurate and cannot be safely used. In the experiments conducted, cases with BD strategies yield to SSLA and ASLA AvgMean of 11.91% and 7.33%, AvgMax of 88.96% and 48.67%, and AvgStd of 19.23% and 11.86% respectively.

It is expected that in cases where counterparts adopt hybrid strategies, linear combinations of TD and BD, the accuracy of SSLAs and ASLAs will be proportional to the level of time dependency. Since the objective of the forecasting tool is to support agents increase their utility, SSLAs and ASLAs are also compared in terms of the attained utility. It is proved that in TD cases the SSLA's average gain increases by 1.27% and the ASLA's by 2.74% compared to non-learning agents.

10. EXTENDING TO MULTI-ISSUED NEGOTIATIONS

In the previous chapters the value of session-long learning agents was illustrated. SSLAs and ASLAs were proved more accurate than current state of the art Pre-Trained Agents (PTAs), yielding significant gains. Although ASLAs have increased accuracy compared to SSLAs, they have high requirements of computational power and time. For this reason we focus on SSLAs and discuss how their architecture can be extended to support multi-issued negotiations. In section 10.1 we present two ways of applying the MLPs to estimate the counterpart's future offer vectors, and in section 10.2 we illustrate experimental results of the extended SSLAs in the domain of electricity distribution.

10.1 Applying MLPs to estimate future offer vectors

The actions performed by the predictive agents in multi-issued negotiations are the same with the actions performed by the predictive agents in single-issued negotiations. At time step t, the negotiation thread can be analyzed to two time series, one which comprises of the past offers of the predictive agent (Con), and one which comprises of the counterpart's (Pr) responses. The latter time-series is expressed as follows: $\{X^{1}_{(\mathrm{Pr}\to Con)}, X^{3}_{(\mathrm{Pr}\to Con)}, ..., X^{t-(2*J_1+1)}_{(\mathrm{Pr}\to Con)}, ..., X^{t-1}_{(\mathrm{Pr}\to Con)}\}$, where J_1 is the counterpart's previous offers that are taken into account by the predictive agent. The actions the predictive agent undertakes at each round *t* are the following:

Step 1. Receive Opponent's Offer, $X^{t-1}_{(\mathrm{Pr}->Con)}$

Step 2. Update Negotiation Thread by storing the received offer

Step 3. Formulate training set:

Consider a time series of the opponent's past offers: $\{X^{1}_{(\mathrm{Pr}->Con)}, X^{3}_{(\mathrm{Pr}->Con)}, ..., X^{t-1}_{(\mathrm{Pr}->Con)}\}$

Formulate the set of input-output patterns with respect to the number of input nodes of the MLP(s)

Step 4. Use the patterns yielded in step 3 to train the network(s)

Step 5. Formulate current input pattern $\{X^{t-2*J_1+1}_{(\mathrm{Pr}->Con)}, ..., X^{t-1}_{(\mathrm{Pr}->Con)}\}$

Step 6. Apply input to the trained network(s)

Step 7. Obtain forecast of opponent's next offer, $\hat{X}^{t+1}_{\mathrm{Pr}\to Con}$

Step 8. Apply the estimation to generate the next offer according to strategy described in 5.2.1

The difference in the case of multi-issued negotiations lies in the design of the neural networks employed by the predictive agents. Hereby we examine two cases; in the first an MLP is considered for each issue, thus for a negotiation over n negotiable attributes n individual MLPs are constructed. Each MLP comprises of J_1 input nodes representing the counterpart's J_1 previously offered values of the particular issue, J_2 nodes in the hidden layer, and one node in the output layer representing the predicted response. The values of J_1 and J_2 are selected after empirical evaluation. Training using the Levenberg and Marquardt (LM) method is conducted during the negotiation session, as in the case of single-issued negotiations. Each network is initialized with random weights and in every negotiation round the network is re-trained with data extracted from the current thread. Such a network is illustrated in Figure 31.

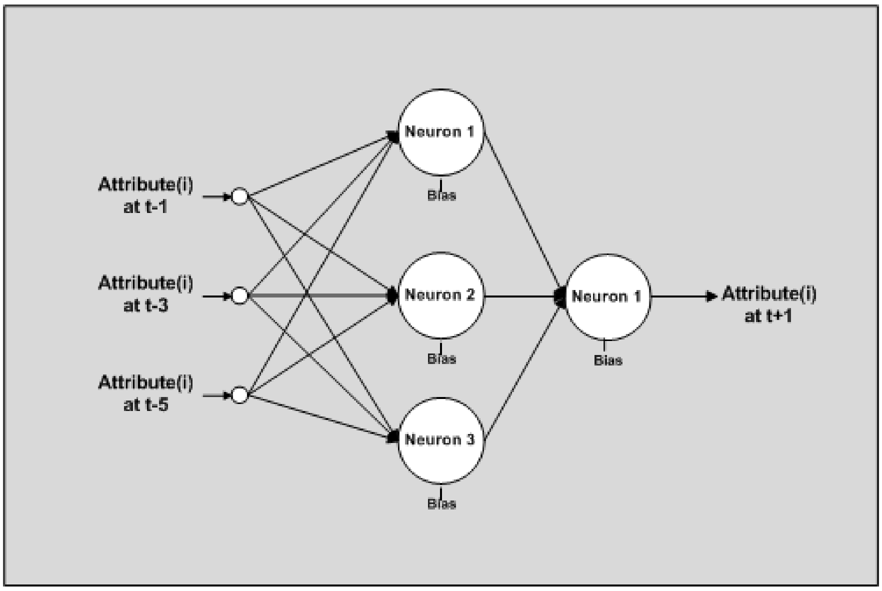


Figure 31: An MLP is used for the prediction of each attribute value

In the second case a single MLP undertakes the task of prediction. If the J_1 previous offers of the opponent are considered for negotiations over n attributes, then an MLP with $n*J_1$ input nodes, J_2 nodes in the hidden layer and n nodes in the output layer is constructed. As in the first case, the network is initialized with random weights and is trained in each round of the predicting agent with data extracted from the current negotiation thread using the LM method. Values of J_1 and J_2 are also empirically evaluated. Such a network is illustrated in Figure 32.

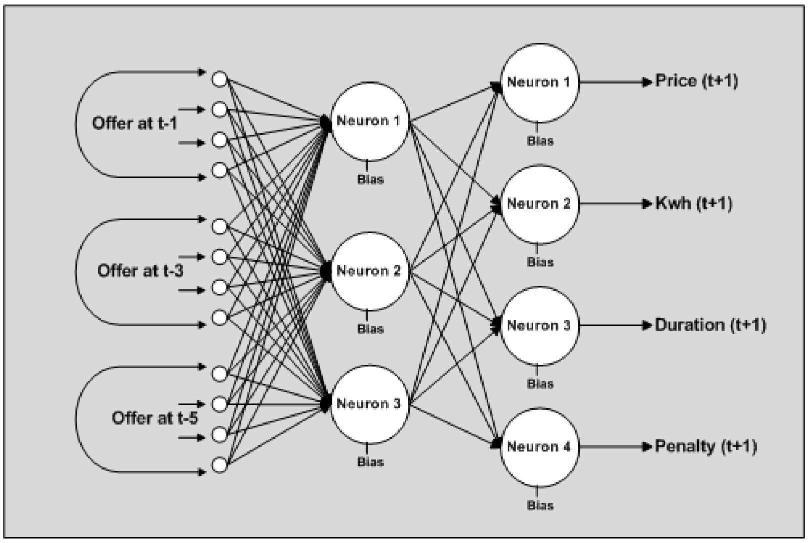


Figure 32: A single MLP is used for the prediction of all attribute values of the offer vector

In both cases, the forecasting tool of the predictive agent makes use of the J_1 previous counterpart's offers to estimate the next offer (at time *t*+1). The network(s) at the beginning of the discourse have random weights. A training set is formulated in each round based on the current negotiation thread (step 3) and it is used to retrain the network (step 4). At time t the consumer formulates a new training set which is constructed from the series of the counterpart's offers as illustrated in Figure 30 and Figure 31. It should be noted that in order to apply the LM method, at least two training patterns are required, therefore the MLP is initially trained and employed at round $t_{init} = 2*J_1+4$. The size of the dataset $|Dataset|$ at time $t \geq t_{init}$ is given by (36):

$$|Dataset| = \frac{t}{2} - J_1 \quad (36)$$

$|Dataset|$ is initially 2 in order to apply the LM method, and increases by 1 at each turn of the predictive agent. At each round, after training the MLP(s), the predictive agent makes use of the network(s) to estimate his counterpart's next offer. The LM method is selected for network training in both cases as it is considered one of the most efficient and popular second order learning methods for networks that are not very large. It also converges much faster than other algorithms, as it is a polynomial time algorithm. In the next section we proceed with the experimental evaluation.

10.2 Experimental results

In Chapter 5 we presented the results of the proposed predictive strategy when agents with "perfect predicting skills" were considered. In this section we present the results when agents employ the neural networks discussed in 10.1. Since focus is set on searching (sub)optimal number of input and hidden nodes, in 10.2.1 we justify the selection of the search space of MLPs' architecture and in 10.2.2 we implement the two cases discussed in 10.1 and outline the results.

10.2.1 Searching (sub)optimal number of input and hidden nodes

Although the optimal network architecture cannot be extracted from theoretical findings, it is rather empirically found that the ratio of learning parameters with respect to the size of the training data should be kept small. As discussed in section 8.2 the bound of the generalization error is given by:

$$E \leq O\left(\frac{1}{J_2}\right) + O\left(\left[\frac{mJ_2 \ln(J_2|Dataset|) - \ln\delta}{|Dataset|}\right]^{1/2}\right) \quad (37)$$

where *m* is the number of input units, J_2 is the number of hidden nodes, δ is a confidence parameter, $\delta \in (0,1)$, and $|Dataset|$ is the size of the dataset. Since in each subsequent step $|Dataset|$ increases, the bound of the generalization error E given is expected to decrease.

There are several rules of thumb that allow as to empirically set (sub)optimal number of input and hidden units. In [87] it is stated that the ratio $\frac{m*J_2}{|Dataset|}$, (where the product $m*J_2$ indicates the parameters that need to be adjusted through the training procedure) must be kept as low as possible if a low bound for the generalization error is desired. Particularly, for a good generalization, we need to have the size of the training set $|Dataset|$ satisfy $|Dataset| = O\left(\frac{m*J_2}{E}\right)$. One rule of thumb discussed by the author is that "with an error of 10%, the number of training examples needed should be about ten times the number of free parameters ($m*J_2$) in the network".

The negotiation settings considered in Chapter 5 comprise of deadlines set to 50, 150, 250 and 350 rounds. As mentioned earlier, the generalization error is expected to decrease in each negotiation round as $|Dataset|$ increases. We assume that the J_1 previous offers and J_2 hidden nodes are selected from the set {2,3,4,5}x{2,3,4,5}. In an attempt to justify the selection of the number of the input and hidden nodes, we have computed the ratio $\frac{m * J_2}{|Dataset|}$ at the expiration of the deadline to see the maximum reduction of the generalization error bound. In the first case, where a single MLP is used for each issue, the number of input nodes m equals J_1. Maximum reduction is achieved when 2 input and 2 hidden nodes are used. The error for the different deadlines (if negotiation reaches the final round) can be reduced to 0.17 for a deadline of 50 rounds, 0.05 for a deadline of 150 rounds, 0.03 for a deadline of 250 rounds, and 0.02 for a deadline of 350 rounds. As expected, increasing the number of input or hidden nodes increases the error bound, which in the case of J_1=5 and J_2=5 is 1.25, 0.35, 0.2, 0.14 for deadlines of 50,150,250 and 350 rounds respectively. The same applies in the case of a single MLP, where there are more parameters that need to be adjusted and the error bound is higher compared to the error in the first case. More specifically, for 4 negotiable attributes m equals $4*J_1$ and the number of input and hidden nodes are selected from the set {4,12,16,20}x{2,3,4,5}. In this case the highest reduction of the error bound is yielded with the selection of 4 input and 2 hidden nodes and is 0.68, 0.2, 0.12 and 0.08 for a deadline of 50, 150, 250 and 350 respectively. The highest error bounds are observed with the selection of 20 input and 5 hidden nodes (5, 1.4, 0.8 and 0.56 for negotiations of 50, 150, 250 and 350 rounds respectively). As the rule of thumb indicates that the ratio is desired to be less than 0.1, we have not considered higher values of J_1 and J_2 with the above settings (the error bound in these cases would surpass the desired value indicated).

Moreover, if a few opponents' past offers are considered, the predictive strategy can be applied from an earlier round. As stated in 10.1, the learning mechanism can be applied when at least two input-output patterns are extracted from the negotiation thread. If we consider a window of counterpart's 2,3,4 and 5 previous offers, the first estimation of the counterpart's next offer is derived in the 8th, the 10th, the 12th, and the 14th round respectively. As the window of the counterpart's previous offers increases, application of the learning tool is delayed. Additionally, the number of training patterns is reduced in the first rounds.

Finally, another reason for preferring small MLPs relates to the agent's bounded resources. Storage of the Jacobian matrix ($|Dataset|$xJ_2), as well as computations for matrix inversion that are of order $O(J_2^3)$ are required at each iterative step of the LM method. For this reason the number of hidden units must be kept small.

10.2.2 Experiments

To assess the two cases we generate 192 negotiation environments based on the following settings. Nine different scenarios with respect to deadline and overlap of agreement zones of the two negotiators are considered. ({ $T_{\max}^{Con} = T_{\max}^{\Pr}$, $T_{\max}^{Con} < T_{\max}^{\Pr}$, $T_{\max}^{Con} > T_{\max}^{\Pr}$} × {Φ=0, Φ=0.33, Φ=0.66}), where $T_{\max}^{a}$ $\in$ [50:100:350], α={Con,Pr}, and Φ is the parameter indicating overlap of the agreement zones. In each scenario the concession curves, defined by parameter β = {0.8, 3}, are considered in order to build the default strategies of the opposing agents.

The above settings are illustrated in Table 9.

Table 9: Negotiation settings

Overlap :	Φ=0		Φ=0.33		Φ=0.66	
Parameters	**Consumer**	**Provider**	**Consumer**	**Provider**	**Consumer**	**Provider**
Kwh(min)	20	20	20	79.4	20	138.8
Kwh(max)	200	200	200	259.4	200	318.8
Price(min)	10	10	10	39.7	10	69.4
Price(max)	100	100	100	129.7	100	159.4
Penalty(min)	5	5	5	29.75	5	54.5
Penalty(max)	80	80	80	104.75	80	129.5
Duration(min)	10	10	10	16.6	10	23.2
Duration(max)	30	30	30	36.6	30	43.2
T_{max}^{σ}	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]
S^{σ}	TD β=[0.8, 3]	TD β=[0.8, 3]	TD β=[0.8,3]	TD β=[0.8, 3]	TD β=[0.8, 3]	TD β=[0.8, 3]

In both cases, preprocessing, in terms of normalization, is applied to the input data set. The original input and output patterns (InputX and OutputY respectively) are normalized and the matrices NormalInputX and NormalOutputY that are returned fall in the interval [–1,1]. The minimum and maximum values of the original inputs (InputXmin, InputXmax) and outputs (OutputYmin and OutputYmax) are stored. After the network has been trained InputXmin and InputXmax are used to transform the new input that is applied to the network. OutputYmin and OutputYmax are used to convert the networks' output to the original scale (that of the output patterns).We have used Matlab's mapminmax function for the normalization process, which transforms $r \in [r_{min}, r_{max}]$ to $t \in [t_{min}, t_{max}]$, based on the following formula:

$$t = \frac{(t_{max} - t_{min}) * (r - r_{min})}{r_{max} - r_{min}} + t_{min}$$

Furthermore, error calculation is performed similarly in both cases. In each decision making step t the consumer makes an estimation of his counterpart's next offer $\hat{X}_{b\to a}^{t+1} = \left(\hat{x}_{1(b\to a)}^{t+1}, \hat{x}_{2(b\to a)}^{t+1}, ..., \hat{x}_{n(b\to a)}^{t+1}\right)^T$. This estimation is compared to the true offer vector of the counterpart at time t+1 $X_{b\to a}^{t+1} = \left(x_{1(b\to a)}^{t+1}, x_{2(b\to a)}^{t+1}, ..., x_{n(b\to a)}^{t+1}\right)^T$ and the absolute error is computed in terms of Euclidean distance. The absolute error signal yielded by the estimation of each attribute i at time t is defined as the distance of the computed output $\hat{x}_{i(b\to a)}^{t+1}$ with the desired output $x_{i(b\to a)}^{t+1}$. This relation is given by:

$$e_i^t = x_{i(b \to a)}^{t+1} - \hat{x}_{i(b \to a)}^{t+1} \quad (38)$$

The error of the prediction at round t is then computed by the following equation:

$$E^t = \sqrt{\sum_{j=1}^{n} (e_i^t)^2} \quad (39)$$

It should be noted that the outputs are transformed in the original scale before calculating the error. The mean of the absolute errors in each discourse is used as an indicative measure to compare the networks applied.

Taking the first case, the predicting agent constructs an MLP for each negotiable issue. Negotiations with RP set to 100% are conducted and the average error of the predictive mechanism is computed. The subset of input features J_1 expressing the past offers of the opponent for a particular issue, as well as the number of hidden nodes are searched in the space {2,3,4,5}x{2,3,4,5}. The search space comprises of 16 neural networks and is selected to be small since only a few patterns extracted from the current thread will be available for training. At the end of each negotiation, the mean of the absolute errors is computed for each network. The same procedure is also repeated in the second case, where a single neural network is used to predict the counterpart's next offer vector. For an offer which consists of n=4 attributes and for the case where the $J_1 \in$ {2,3,4,5} previously sent offers of the opponent are considered, the (sub)optimal number of input and hidden nodes is searched in the space {8,12,16,20}x{2,3,4,5}.

For each case 192 negotiation environments are generated and 16 ANNs are tested, leading to a total of 3072 experiments. The overall mean of the absolute errors is used to assess the predictive models.

Results show that the neural network yielding the smallest error and smallest standard deviation comprises of 5 input and 4 hidden nodes, when an MLP is constructed separately for each issue (first case). For this ANN the average increase in utility attained by the predictive agent is 10.78%. On the other hand, in the second case where a single MLP is employed for the prediction of the counterpart's response, the smallest average error is yielded when 8 input nodes (stemming from the counterpart's 2 previous offer vectors) and 5 hidden nodes are used. This model returns an average increase in utility of 10.5%. The smallest average standard deviation is yielded when 20 input and 5 hidden nodes are used. The last ANN yields 10.34% average increase in utility. The low value of the average standard deviation signifies smoother predictive curves, as estimations do not deviate much from the mean. Table 10 summarizes the results with respect to the combination of input-hidden nodes.

Table 10: Mean errors and mean standard deviations for each combination of (input,hidden) nodes in the MLP, are illustrated for each case. Minimum values are depicted in bold style.

	Mean Error				Mean Std Deviation				Avg Increase in Utility (%)
Hidden / **Input**	**2**	**3**	**4**	**5**	**2**	**3**	**4**	**5**	
Case 1									
2	1.62	**1.36**	1.6	1.81	3.49	**1.67**	1.97	2.07	**6.85**
3	1.60	1.20	**1.17**	1.26	3.03	1.72	**1.38**	1.47	**6.86**
4	1.69	1.11	**1.03**	1.08	3.44	1.37	**1.19**	1.29	**7.5**
5	1.57	1.07	**0.98**	1.07	2.91	1.32	**1.16**	1.26	**10.78**
Case 2									
8	1.19	0.63	0.53	**0.49**	3.63	1.38	1.03	**0.97**	**10.5**
12	1.14	0.64	**0.50**	0.51	3.21	1.18	**0.88**	0.92	**10.34**
16	1.00	0.61	0.55	**0.51**	2.27	1.14	0.98	**0.89**	**10.09**
20	1.10	0.69	0.53	**0.52**	3.02	1.35	0.97	**0.83**	**9.54**

The error measured in the two cases is not directly comparable to other related work, as the negotiation domains are not the same. It should be noted that aforementioned work involving single-lag predictions considers only single-issued negotiations between automated agents. In [6] although negotiations with four issues are conducted, the domain is static and the negotiable issues take predefined discrete values. The simple MLP employed by the SSLAs is herein extended to support multi-issued negotiations, leading to the two different designs illustrated in Figures 31 and 32. From the experiments conducted it is shown that extending the proposed MLP to support multi-issued negotiations can also capture the negotiation dynamics, as in both cases the proposed networks yield low mean of the absolute errors and mean standard deviation, and incur a significant increase to the predictive agent's utility.

11. EPILOGUE

Predictive decision making is characteristic to current state of the art socio-technical systems that guide negotiation processes under electronic settings. From semi-automated negotiation support systems, to fully automated platforms where all processes are undertaken by software agents, the back end participants are particularly benefitted by the use of models of computational intelligence. Such models provide estimations about the behavior of the negotiator's counterparts and allow users or agents acting on their behalf, to adapt their strategy and evaluate risks and dynamics of current negotiation. The first four chapters of this thesis provide the foundations related to the negotiation domain, terminologies and classifications, research methodologies and software platforms, as well as description of negotiation protocols and strategies that constitute state of the art negotiating agents.

In the fifth chapter, a predictive strategy employed by an autonomous agent who engages in multi-issued negotiations is presented and assessed. The strategy allows the predictive agent to adapt his consequent offers with respect to the estimations of his counterpart's responses. As different attitudes towards risk may emerge, a risk-related parameter is also embodied to the strategy. We have considered a number of negotiation environments and we have measured the average increase in utility that incurs to the predictive agent compared to the non-learning one. An agent with a highly risk-seeking attitude achieves on average 12.05% increase in utility, while a predictive agent with a more conservative behavior (risk-averse) achieves 0.94% increase in utility. However, the trade-off of the highly increased utility is the decrease of the number of agreements, which is due to prolongation of the negotiation time. In the case of the highly risk-seeking agent the number of agreements is decreased by 20.78% compared to the non-learning case. To address this issue the risk-related parameter must be appropriately set in each negotiation discourse. Our proposed approach to appropriately set the parameter requires estimation of the counterpart's deadline. To illustrate the proposed decision-making rule we have assumed knowledge of the counterpart's deadline and we have reproduced the same experimental settings. From the experiments conducted, the average increase in utility is 12.017% and approaches the average increase of the risk-seeking agent. At the same time the average decrease of agreements is reduced to 0.61%.

In the remainder of this thesis the skill of forecasting the counterpart's future offers is further investigated and selection of Multi-layer Perceptrons (MLPs) is preferred to other learning models. The sixth Chapter provides a brief overview and comparison of the forecasting tools employed by negotiators, and the seventh Chapter provides a justification of the selection of MLPs, based on bibliographical research. Current systems which base their learning models on data acquired from previous interactions or from synthetic data provide satisfying results in static negotiation environments (where data distributions do not change). Such systems are once trained in an offline mode and are thereafter expected to operate in a real environment. However, when data distributions change, the systems no longer provide accurate estimations. A new perspective to the issue is introduced, by highlighting the need of learning during the negotiation session, as discussed in Chapter 8. Such an approach is viable in open, dynamic negotiation environments. A number of experiments are conducted to support this argument and disclose the inability of initially pre-trained networks to capture the dynamics of changing distributions. "Session-long learning" agents, trained with the data of the current negotiation thread, prove capable of capturing the negotiation dynamics. In Chapter 8 we introduce Static Session-long Learning Agents (SSLAs), which employ a simple static neural network model. To illustrate the superiority of SSLAs compared to agents that employ pre-trained networks (Pre-Trained Agents, PTAs) in cases where data distributions change, we have conducted a number of experiments considering single-issued negotiations and we have computed the absolute error yielded in each decision making step. SSLAs are proved more accurate, as the mean of the errors is reduced by 92.67% compared to the PTAs. However, they do not yield satisfying results in negotiations with short deadline. This is due to the small size of the training data set compared to the number of parameters of the neural network that need to be learned. The incorporation of an optimization technique for the selection of the networks' architecture to address this issue is discussed in Chapter 9, with the introduction of an Adaptive Session-long Learning Agent (ASLA). ASLA evolves its structure and input features with the use of a genetic algorithm in each negotiation round. We empirically prove the superiority of ASLAs compared to SSLAs as far as accuracy is concerned. More specifically, we have conducted a number of experiments and we have computed the mean, the standard deviation and the maximum values of the absolute errors at the end of each negotiation. ASLAs are proved more accurate as the average mean of the absolute errors is reduced by 38.34%, the average of the maximum values is reduced by 44.75% and the average of the standard deviation is reduced by 38.03%. ASLA is a smoother predictive model as it proves more accurate with decreased standard deviation and maximum error values. However it is not as fast as SSLA and has higher storage requirements, which makes it difficult to apply in real situations. In the appendix we also examine the employment of a simple evolving connectionist structure (eMLP), which adapts its structure with each new training pattern, and is much faster than ASLA as it conducts one-pass learning. However agents enhanced with eMLP structures are less accurate than agents enhanced with MLP structures.

The idea of static session-long learning agents is extended to support multi-issued negotiations. Forecasting is again conducted with the use of Multilayer Perceptrons (MLPs) and the training set is extracted during the negotiation session. Two cases are examined: one where separate MLPs are used to estimate each negotiable attribute and one where a single MLP is used to estimate the counterpart's response. It is shown that simple MLPs with one hidden layer are adequate for forecasting the counterpart's offer vectors, and are tested in order to find the appropriate number of nodes on input and hidden layer. The network that yields the lowest error, incurs to the predictive agent 10.78% average absolute increase of his individual utility (gain), which is close to the increase in utility incurred by the highly risk-seeking agent enhanced with a perfect forecasting tool.

This thesis contributes to the field of negotiation with the proposal of a predictive strategy that incorporates different attitudes towards risk, as well as to the field of application of neural networks in negotiations with the introduction of session-long learning agents. It is concluded with the discussion of future research issues.

12. Future research issues

In this section we outline a number of issues that can be considered for future research. The first relates to the domain of application of the predictive strategy. The decision rule of the risk-seeking agent, as well as the decision rule discussed in [8], cannot safely be used in the case where counterparts adopt pure Behavior Dependent strategies, as the latter would imitate the 'smart' behavior of the predictive agents and would push back from the agreement as well. For this reason the predictive agents presented in this thesis were only tested with non-learning counterparts who followed Time Dependent strategies. However, in hybrid strategies, where linear combinations of time and behavior dependent tactics are considered, success of the proposed strategy is expected to depend on the weight of the time dependent tactic. In order to broaden the applicability of the proposed strategy, estimation of behavior dependency could also be enhanced to the decision rule. An interesting approach concerning estimation of time and behavior dependent weights is based on the difference method and is found in [5]. Another issue left for future research is the investigation of predictive strategies in co-operative environments, where the objective is to maximize the joint rather than the individual utility. Moving to the realm of the employed forecasting tool, ASLAs have proved more efficient than SSLAs. However the trade-off is the increased computational resources and time of convergence. An issue left for future research is therefore to test other adaptive and more efficient structures with the predictive agents. Examples of such structures are the Evolving Fuzzy Neural Networks (EFuNNs) and DENFIS, which are Evolving Connectionist Systems (ECoS) that continuously, evolve their structure and functionality to capture the dynamics of turbulent settings.

TERMINOLOGY

English Term	Greek Term
Risk averse	Αποστροφή κινδύνου
Electronic markets	Ηλεκτρονικές αγορές
Socio-technical systems	Κοινωνικο-τεχνικά συστήματα
Negotiation domain	Περιοχή διαπραγμάτευσης
Negotiation software agent	Πράκτορας λογισμικού διαπραγμάτευσης
Adaptive session-long learning agent	Πράκτορας με δυναμικό μοντέλο μάθησης που εκπαιδεύεται με δεδομένα της τρέχουσας διαπραγμάτευσης
Session-long learning agent	Πράκτορας με μοντέλο μάθησης που εκπαιδεύεται με δεδομένα της τρέχουσας διαπραγμάτευσης
Pre-Trained Agent	Πράκτορας με μοντέλο που εκπαιδεύεται μία φορά στη φάση σχεδιασμού
Static session-long learning agent	Πράκτορας με στατικό μοντέλο μάθησης που εκπαιδεύεται με δεδομένα της τρέχουσας διαπραγμάτευσης
Risk seeking	Ροπή προς τον κίνδυνο

ACRONYMS

ART	Adaptive Reasonance Theory
ASLA	Adaptive Session-long Learning Agent
ADEPT	Advanced Decision Environment for Process Tasks
AI	Artificial Intelligence
ANN	Artificial Neural Network
AR	Autoregressive
ARCH	Autoregressive Conditional Heteroscedasticity
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BD	Behavior Dependent
BATNA	Best Alternative to Negotiating Agreement
CBR	Case Based Reasoning
CG	Conjugate Gradient
E-Market	Electronic Marketplace
ENS	Electronic Negotiation System
ENT	Electronic Negotiation Table
eNAs	e-Negotiation Agents
ECoS	Evolving Connectionist Systems
EFuNN	Evolving Fuzzy Neural Network
F2F	Face to Face
FFNN	Feedforward Neural Network
FeNAs	Fuzzy e-Negotiation Agents
GARCH	Generalized Autoregressive Conditional Heteroscedasticity
GA	Genetic Algorithm
HV	High Voltage
ITA	Intelligence Trading Agency
INSS	InterNeg Support System
LMS	Least Mean Squares
LM	Levenberg and Marquardt
LV	Low Voltage
MSE	Mean Squarred Error
MV	Medium Voltage
MP	Meeting Point
MA	Moving Average
MAGNET	Multi Agent Negotiation Testbed
MLP	Multi-Layer Perceptron
NAA	Negotiation Agent Assistant
NSA	Negotiation Software Agent
NSS	Negotiation Support System
OSS	One Step Secant
PTA	Pre-Trained Agent
RBF	Radial Basis Function
RBFN	Radial Basis Function Network

RNN	Recurrent Neural Network
RAE	Regulatory Authority for Energy
RD	Resource Dependent
RP	Risk Portion
SLA	Service Level Agreement
SSLA	Static Session-long Learning Agent
SVM	Support Vector Machine
TD	Time Dependent

APPENDIX

Application of a simple ECoS (eMLP) to estimate the next offer

Considering evolving structures when data are extracted from the current negotiation thread (case of session-long learning agents) has been highlighted in Chapter 9, where the employed ANNs evolve their structure with the use of a genetic algorithm. Since time is crucial in negotiations and GAs require a lot of interactions to converge, research has been guided to the use of Evolving Connectionist Systems (ECoS) which employ fast learning algorithms. In this section we illustrate an agent engaging in on-line one-pass learning to predict his counterpart's response with the use of a simple ECoS, eMLP [156]. In the next section we describe the eMLP and its advantages over classical neural networks. Results of the ECoS-based negotiator are also presented.

Integrating ECoS with automated negotiators

ECoS are flexible structures that are capable of accommodating new data without forgetting previously learned ones (local learning), keeping the training time low. More specifically, "an ECOS is an adaptive, incremental learning and knowledge representation system that evolves its structure and functionality, where in the core of the system is a connectionist architecture that consists of neurons and connections between them" [156]. ECoS have the following attractive features: they may evolve in open space, engage in incremental lifelong learning in an online mode, learn both as individual systems and as evolutionary populations of such systems, partition the problem space locally, allowing for fast adaptation, have evolving structures and trace the evolving processes over time. Hereby we present the integration of a simple ECoS, eMLP, with a negotiating agent who adopts the strategy described in section 5.2.1. The characteristic feature of the eMLP is the creation of rule nodes that provide appropriate mappings from input to output subspaces. As new patterns are presented, the eMLP changes its structure either by creating a new rule node to represent the new association or by adjusting the centers of an already associated rule node. In more detail eMLPs have three layers of neurons: an input layer which represents the input features, an evolving layer which comprises of the rule nodes that represent prototypes of input-output data associations and an output layer which represents the output features. Each rule node R_j in the evolving layer is associated with the center of a hypersphere from the input space, represented by a weight vector $W_1(R_j)$ and with the center of a hypersphere from the output space, represented by a weight vector $W_2(R_j)$. $W_1(R_j)$ and $W_2(R_j)$ constitute the interconnection weights from input to evolving layer and from evolving to output layer respectively. Rule nodes "move" to accommodate new input-output examples. A new example (x,y) is considered in association with a rule node R_j if x falls in the input receptive field and y falls in the output reactive field of the rule. The first condition is satisfied if the distance of the input x with the center $W_1(Rj)$ of the rule node is less than a threshold (specified by the Radius of R_j). Similarly, the second condition is satisfied if the distance of output y with the calculated output is less than an error threshold. Distances are measured as normalized Hamming distances. As long as input x falls in the input receptive and output reactive field of the most highly activated rule node, the one for which the distance of its input center $W_1(R_j)$ and input x is minimum, the weight vector $W_1(R_j)$ is adjusted through unsupervised learning depending on the distance of x and $W_1(R_j)$, while $W_2(R_j)$ is adjusted through supervised learning based on the Widrow-Hoff Least Mean Squares (LMS) delta algorithm. More details about the weight adjustment formulas can be found in [156]. If the new example (x,y) cannot be associated with any of the existing nodes, a new rule node is created by setting its initial weights $W_1(Rj)$ to x and $W_2(R_j)$ to y. Initially the network does not contain any rule nodes in the evolving layer and it is gradually built. When eMLP is employed by predictive negotiating agents, it is trained in each decision-making step by propagating the data patterns extracted from the current negotiation thread. At time step t, the negotiation thread can be analyzed to two time series, one which comprises of the past offers of the predictive agent (agent b), and one which comprises of the counterpart's (agent α) responses. If the predictive agent is the one who initiates the interaction at time step 0 (eg. consumer agent), the latter time-series is expressed as follows: $\{X^{1}_{(a\to b)}, X^{3}_{(a\to b)}, ..., X^{t-(2*J_1+1)}_{(a\to b)}, ..., X^{t-1}_{(a\to b)}\}$. The previously J_1 offers can be extracted from the latter series by considering the subsequent offers from time t_1=t-(2*J_1+1) until time t_2= t-1. These offers are propagated to the input layer of the eMLP to infer the prediction of the counterpart's offer at time t+1. At the same time the training set is augmented with the addition of a new data pattern extracted from the J_1 +1 counterpart's responses. More specifically the offers sent from t_1'= t-(2*J_1+3) until t_2' = t-3 constitute the new input pattern while the offer sent at t-1 constitutes the related output pattern. The learning algorithm employed uses one-pass-learning, thus only one data pattern is propagated to the eMLP at each decision making step. In the following paragraph we illustrate results of the integration of eMLP with negotiating agents.

Illustrative results

A number of negotiations are conducted between a provider and a consumer agent α={Con,Pr}, over service terms of electricity trade, characterized by four negotiable attributes: number of Kwh, Price per Kwh, Penalty terms, and Duration of service provision. The latter agent uses the predictive strategy discussed in section 5.2.1 setting RP to 100%. The experimental workbench issues various scenarios with respect to deadline $T^{a}_{\max}$, and overlap of agreement zones of the two negotiators. The overlap of agreement zones is defined by a parameter Φ $\in$ [0,1]. When Φ is set to 0 the two agents have equal reservation values, while when Φ is set to 1, there does not exist a solution in accord with the preferences set by the two agents. Various concession curves of TD group of strategies, defined by a parameter β are considered in order to build the default strategies of the agents. The parameters of the 192 generated negotiation environments are depicted in Table 11.

Table 11: Negotiation Settings

Overlap :	Φ=0		Φ=0.33		Φ=0.66	
Parameters	**Consumer**	**Provider**	**Consumer**	**Provider**	**Consumer**	**Provider**
Kwh(min)	20	20	20	79.4	20	138.8
Kwh(max)	200	200	200	259.4	200	318.8
Price(min)	10	10	10	39.7	10	69.4
Price(max)	100	100	100	129.7	100	159.4
Penalty(min)	5	5	5	29.75	5	54.5
Penalty(max)	80	80	80	104.75	80	129.5
Duration(min)	10	10	10	16.6	10	23.2

Duration(max)	30	30	30	36.6	30	43.2
$T^{\text{ȣ}}_{max}$	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]	[50:100:350]
$S^{\smile}_{(TD)}$	β=[0.8, 3]	β=[0.8, 3]	β=[0.8,3]	β=[0.8, 3]	β=[0.8, 3]	β=[0.8, 3]

The predictive agent constructs an eMLP for each negotiable issue. Error threshold and initial value of the evolving node's radius are set to 0.001 and maximum value of the radius of each node's input hypersphere is set to 0.1, to test the performance of eMLPs. The number of the counterpart's previously offered values J_1 are obtained from the set {2,3,4,5}, thus 4*192= 768 experiments are conducted and the average error of the predictive mechanism is computed. In each decision making step the agent makes an estimation of the counterpart's next offer. This estimation is compared to the true offer vector of the counterpart and the absolute error is computed in terms of Euclidean distance. At the end of each negotiation, the mean, the average standard deviation and the maximum value of the absolute errors is computed. Overall assessment of the 768 experiments is provided through the computation of mean and maximum values of the above measures. Additionally the increase in terms of utility which incurs to the predictive agent compared to the non learning one is also computed. Results are illustrated in Table 12.

Table 12: Results of predictive strategy when agents are enhanced with e-MLP

Measures:	**Mean of Abs Errors**		**Avg of Std of Abs Errors**		**Max of Abs Errors**		**Avg Utility Increase(%)**
#Previous Offers	**Mean**	**Max**	**Mean**	**Max**	**Mean**	**Max**	
2	5.1	5.841	1.09	2.92	7.46	14.73	5.316
3	5.32	6.18	1.27	3.51	8.19	17.18	5.315
4	4.98	5.99	1.08	2.96	7.36	15.12	5.318
5	**4.8**	**5.84**	**0.9**	**2.58**	**6.79**	**14.11**	**5.327**

The values in Table 12 are not normalized; they rather express maximum and average values of error vectors, computed as Euclidean distances. The values related to the Mean of Absolute errors are desired to be low for a model to be accurate. The low values of average standard deviations signify that there are not high oscillations around the mean of the absolute errors, and that the predictive curves are quite smooth. Finally the low values of Maximum absolute errors are also desirable since high values could misguide the predictive agents. The eMLP which yields the minimum mean error and incurs the maximum increase in utility to the predictive negotiator is attained when the 5 previously sent offers of the counterpart constitute the input features. It is worth noticing that all models have very low average standard deviations, which justifies the decision of integrating ECoS with automated negotiators.

REFERENCES

[1] G. E. Kersten, E. Chen, D. Neumann, R. Vahidov and C. Weinhardt, “On Comparison of Mechanisms of Economic and Social Exchanges: The Times Model”, *Negotiation, Auctions and Market Engineering,* H. Gimpel et al., eds., Berlin Heidelberg: Springer-Verlag, 2008a, pp. 16-43.

[2] C. Hou, “Predicting Agents Tactics in Automated Negotiation”, *Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, Washington DC: IEEE Computer Society, 2004, pp. 127-133.

[3] J. Brzostowski, and R. Kowalczyk, “Adaptive Negotiation with On-Line Prediction of Opponent Behaviour in Agent-Based Negotiations”, *Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology,* Washington DC: IEEE Computer Society, 2006a, pp. 263-269.

[4] J. Brzostowski, and R. Kowalczyk, “Modelling Partner’s Behaviour in Agent Negotiation”, *AI 2005,* LNCS 3809, Berlin Heidelberg: Springer-Verlag, 2005, pp. 653-663.

[5] J. Brzostowski, and R. Kowalczyk, “Predicting partner's behaviour in agent negotiation”, *Proc. 5th Int. Joint Conf. on Autonomous Agents and Multiagent Systems,* New York: ACM, 2006b, pp. 355-361.

[6] R. Carbonneau, G. E. Kersten, and R. Vahidov, Predicting opponent's moves in electronic negotiations using neural networks, *Expert Systems with Applications: An International Journal*, vol. 34, no. 2, 2008, pp. 1266-1273.

[7] C. C. Lee, and C. Ou-Yang, A neural networks approach for forecasting the supplier's bid prices in supplier selection negotiation process, *Expert Systems with Applications*, vol. 36, no. 2, 2009, pp. 2961-2970.

[8] M. Oprea, “The Use of Adaptive Negotiation by a Shopping Agent in Agent-Mediated Electronic Commerce”, *Proc. 3rd Int. Central and Eastern European Conf. on Multi-Agent Systems (CEEMAS 03)*, Heidelberg Berlin: Springer Verlag, 2003, pp. 594-605.

[9] I. V. Papaioannou, I. G. Roussaki, and M. E. Anagnostou, “Comparing the Performance of MLP and RBF Neural Networks Employed by Negotiating Intelligent Agents”, *Proc. of the IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, Washington DC: IEEE Computer Society, 2006a, pp. 602-612.

[10] I. V. Papaioannou, I. G. Roussaki, and M. E. Anagnostou, “Towards Successful Automated Negotiations based on Neural Networks”, *Proc. of the 5th IEEE/ACIS Int. Conf. on Computer and Information Science*, Washington DC: IEEE Computer Society, 2006b, pp. 464-472.

[11] I. Papaioannou, I. Roussaki, and M. Anagnostou, “Detecting Unsuccessful Automated Negotiation Threads When Opponents Employ Hybrid Strategies”, *Proc. of the 4th Int. Conf. on Intelligent Computing: Advanced Intelligent Computing theories and Applications - with Aspects of Artificial intelligence*, D. Huang, D. C. Wunsch, D. S. Levine, and K. Jo, eds., Heidelberg Berlin: Springer, 2008, pp. 27-39.

[12] I. Roussaki, I. Papaioannou, and M. Anagnostou, “Building Automated Negotiation Strategies Enhanced by MLP and GR Neural Networks for Opponent Agent Behaviour Prognosis”, *IWANN 2007*, F. Sandoval, A. Prieto, J. Cabestany, and M. Grana, eds., Heidelberg Berlin: Springer, 2007, pp. 152-161.

[13] P. H. Gulliver, *Disputes and Negotiations: A Cross-Cultural Perspective,* New York, Academic Press, 1979.

[14] W. N. Robinson, and V. Volkov, Supporting the negotiation life cycle, *Communications of the ACM,* vol. 41, no. 5, 1998, pp. 95-102.

[15] L. Putnam, and M. Roloff, (eds.), *Communication Perspectives on Negotiation*, Communication and Negotiation, Newbury Park CA: Sage, 1992, pp. 1-17.

[16] H. Raiffa, *The Art and Science of Negotiation,* Cambridge Massachusetts and London: The Belknap Press of Harvard University Press, 1982.

[17] G. E. Kersten, “E-negotiations: Towards Engineering of Technology-based Social Processes*”, Proc. of the 11th European Conf. on Information Systems (ECIS 2003),* 2003.

[18] M. Bichler, G. Kersten and S. Strecker, Towards a Structured Design of Electronic Negotiations, *Group Decision and Negotiation,* vol. 12, no. 4, 2003, pp. 311–335.

[19] R. R. Blake, and J.S. Mouton, *The managerial Grid,* Houston: Gulf Publishing Co, 1964.

[20] P. Braun, J. Brzostowski, G. Kersten, J. Kim, R. Kowalczyk, S. Stecker, R. Vahidov, “e-Negotiation Systems and Software Agents: Methods, Models, and Applications”, *Intelligent decision-making support systems: foundations, applications and challenges,*

Part II, chapter 15, J. N. D. Gupta, G. A. Forgionne and M. T. Manuel, (eds.), 2006, pp. 271-300.

[21] J. S. Rosenscein, and G. Zlotkin, *Rules of Encounter*, Cambridge Massachusetts: The MIT Press, 1994.

[22] A. Rubinstein, Perfect Equilibrium in a Bargaining Model, *Econometrica*, vol. 50, no. 1, 1982, pp. 97-109.

[23] R. Avenhaus, and T. Krieger, "Formal Methods for Forecasting Outcomes of Negotiations on Interstate Conflicts", *Diplomacy Games*, W. Zartman, ed., Berlin Heidelberg: Springer, 2007, pp. 123-148.

[24] I. A. Vetsikas, and N. R. Jennings, "Generating Bayes-Nash Equilibria to Design Autonomous Trading Agents", *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence*, 2007, pp. 1543-1550.

[25] P. Faratin, C. Sierra, and N. R. Jennings, Negotiation Decision Functions for Autonomous Agents, *International Journal of Robotics and Autonomous Systems*, vol. 24, no. 3-4, 1998, pp. 159-182.

[26] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg, Argumentation-based Negotiation, *The Knowledge Engineering Review,* vol. 18, no. 4, 2004, pp. 343-375.

[27] H. A. Simon, *The New Science of Management Decisions,* New York: Harper and Row, 1960.

[28] G. E. Kersten, "Decision Making and Decision Support", *Decision Support Systems for Sustainable Development in Developing Countries*, G. E. Kersten, Z. Mikolajuk and A. Yeh, eds., Kluwer Academic, 1999.

[29] J. Von Neumann, and O. Morgenstern, *Theory of Games and Economic Behaviour*, Princeton NJ: Princeton Univ. Press, 1944.

[30] C.-J. Tsay, and M. H. Bazerman, A Decision Making Perspective to Negotiation: A Review of the Past and a Look into the Future, *Negotiation Journal*, vol. 25, no. 4, 2009, pp. 467-480.

[31] G. F. Loewenstein, L. Thompson, and M. H. Bazerman, Social utility and decision making in interpersonal contexts, *Journal of Personality and Social Psychology*, vol. 57, no. 3, 1989, pp. 426-441

[32] L. Thompson, and G. Loewenstein, Egocentric interpretations of fairness and interpersonal conflict, *Organizational Behavior and Human Decision Processes*, vol. 51, no. 2, 1992, pp. 176-197.

[33] R. J. Lewicki, and N. Stark, What is ethically appropriate in negotiations: An empirical examination of bargaining tactics. *Social Justice Research* vol. 9, no. 1, 1996, pp. 69-95.

[34] J. P. Forgas, On feeling good and getting your way: Mood effects on negotiator cognition and bargaining strategies, *Journal of Personality and Social Psychology,* vol. 74, no. 3, 1998, pp. 565-577.

[35] C. K. W. De Dreu, P.J. D. Carnevale, B. J. M. Emans, E. van de Vliert, Outcome Frames in Bilateral Negotiation: Resistance to Concession Making and Frame Adoption. *European Review of Social Psychology*, vol. 6, 1995, pp. 97-125.

[36] R. E. Petty, and J. T. Cacioppo, *Communication and persuasion: Central and peripheral routes to attitude change*, New York: Springer-Verlag, 1986.

[37] J. K. Sebenious, “Negotiation Analysis: Between Decisions and Games”, Advances in Decision Analysis from Foundations to Applications, 2007, pp. 469-488.

[38] G. E. Kersten, and H. Lai, Satisfiability and Completeness of Protocols for Electronic Commerce, *European Journal of Operational Research*, vol. 180, 2007, pp. 922-937.

[39] G. E. Kersten,and H. Lai, “Negotiation Support and e-Negotiation Systems”, *Handbook on Decision Support Systems 1: International Handbooks on Information Systems,* Springer Berlin Heidelberg, 2008, pp. 469-508.

[40] Y. Yuan, M. Head, and M. Du, The Effects of Multimedia Communication on Web-based Negotiation, *Group Decision and Negotiation*, vol. 12, no. 2, 2003, pp. 89–109.

[41] G. E. Kersten, “Negotiation Support Systems and Negotiating Agents”, *Modèles et Systèmes Multi-Agents pour la Gestion de l’Environement et des Territoire (Colloque SMAGET)*, Clermont-Ferrand France, 1999b, pp. 307–316

[42] S. Strecker, G. Kersten, J. Kim, K. P. Law, “Electronic Negotiation Systems: The Invite Prototype”, *Proceedings of the Collaborative business MKWI’06*, Passau Germany, 2006.

[43] R. Kowalczyk, and V. Bui, “On Constraint-based Reasoning in e-Negotiation Agents”, *Agent Mediated Electronic Commerce III, LNAI 2000*, F. Dignum and U. Cortés eds., Springer-Verlag, pp. 31-46, 2000.

[44] R. Kowalczyk, Fuzzy e-Negotiation Agents, *Journal of SoftComputing, Special Issue on Fuzzy Logic and the Internet*, vol. 6, no. 5, 2002, pp. 337-347.

[45] I. Rahwan, R. Kowalczyk, and H. H. Pham, “Intelligent Agents for Automated one-to-many e-commerce Negotiation”, *Proc. of the 25th Australian Conference on Computer Science, Melbourne, Victoria, Australia*, 2002, pp. 197-204.

[46] A. Chavez, and P. Maes, “Kasbah: An Agent Marketplace for Buying and Selling Goods”, *Proc. of the 1st Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK*, 1996.

[47] “Cybersettle”; http://www.cybersettle.com/pub/home/about.aspx. [Last Accessed October 2012]

[48] R. Guttman, and P. Maes, “Agent-mediated integrative negotiation for retail electronic commerce”, *Proc. of the Workshop on Agent-Mediated Electronic Trading (AMET 98), Minneapolis*, 1998.

[49] “Magnet: Multi Agent Negotiation Testbed”; http://magnet.cs.umn.edu/. [Last accessed October 2012]

[50] E. Chen, G. E. Kersten, et al., Agent-Supported Negotiations on E-marketplace, *International Journal of Electronic Business*, vol. 3, no. 1, 2004, pp. 28-49.

[51] M. Liu, and F. F. Wu, Portfolio Optimization in Electricity Markets, *Electric Power Systems Research*, vol. 77, no. 8, 2007, pp. 1000-1009.

[52] H. Li, J. J. Jeng, J. Y. Chung, “Commitment-Based Approach to Categorizing, Organizing and Executing Negotiation Processes”, *Proc. of IEEE Int. Conf. on E-Commerce (CEC 03)*, 2003.

[53] G. E. Kersten, S. Strecker, K. P. “Law, Protocols for Electronic Negotiation Systems: Theoretical Foundations and Design Issues”, *E-Commerce and Web Technologies*, K. Bauknecht, M. Bichler and B. Pröll, eds., Berlin Heidelberg: Springer-Verlag, 2004, pp. 265-268.

[54] E.H. Gerding, D.D. Bragt, J.A. “La Poutre, Scientific Approaches and Techniques for Negotiation”, *A Game Theoretic and Artificial Intelligence Perspective,* Technical

Report, UMI Order Number SEN R-0005, Centre for Mathematics and Computer Science (CWI), Amsterdam, Netherlands, 2000.

[55] C. Tsatsoulis, “A Case-Based Reflective Negotiation Model”, Air Force Research Laboratory, Information Directorate, 2003.

[56] H. L. Cardoso, and E. Oliveira, “Using and Evaluating Adaptive Agents for Electronic Commerce Negotiation”, *Proc.of the Int. Joint Conf., 7th Ibero-American Conference on AI: Advances in Artificial intelligence*, M. C. Monard and J. S. Sichman, eds., London: Springer-Verlag, 2000, pp. 96-105.

[57] E. Oliveira, and A. P. Rocha, “Agents Advanced Features for Negotiation in Electronic Commerce and Virtual Organizations Formation Process”, *Agent Mediated Electronic Commerce, the European AgentLink Perspective*, F. Dignum and C. Sierra, eds., Heidelberg: Springer Berlin, 2001, pp.77-97.

[58] C. Claus, and C. Boutilier, “The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems”, *Proc. of the 15th national/10th conference on Artificial intelligence/Innovative applications of artificial intelligence*, Menlo Park, CA: American Association for Artificial Intelligence, 1998, pp. 746-752.

[59] J. R. Oliver, A machine-learning approach to automated negotiation and prospects for electronic commerce, *J. Manage. Inf. Syst.*, vol. 13, no. 3, 1996, pp. 83-112.

[60] N. Matos, C. Sierra, and N. Jennings, “Determining Successful Negotiation Strategies: An Evolutionary Approach”, *Proc. of the 3rd Int. Conf. on Multi Agent Systems*, Washington DC: IEEE Computer Society, 1998, pp. 182-189.

[61] E. H. Gerding, D. D. van Bragt, and J. A. La Poutre, “Multi-Issue Negotiation processes by Evolutionary Simulation: Validation and Social Extensions”, *Technical Report. UMI Order Number: SEN-R0024., CWI (Centre for Mathematics and Computer Science)*, 2000b.

[62] R. Y. Lau, “Adaptive negotiation agents for e-business”, *Proc. of the 7th Int. Conf. on Electronic Commerce*, New York: ACM, 2005, pp. 271-278.

[63] R.R Nelson, and S. G. Winter, *An Evolutionary theory of economic change*, Cambridge, MA and London: The Belknap Press of Harvard University Press, 1982.

[64] K. Sycara, “Resolving Goal Conflicts via Negotiation”, *Proc. of the 7th National Conference on Artificial Intelligence (AAAI-88)*, 1988.

[65] Y. W. Wong, D. M. Zhang, and M. Kara-Ali, "Negotiating with Experience. Knowledge-Based Electronic Markets", *Papers from the AAAI Workshop, Technical Report,* AAAI Press, 2000.

[66] J. Brzostowski, and R. Kowalczyk, "On possibilistic case-based reasoning for selecting partners for multi-attribute agent negotiation", *Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, New York: ACM, 2005b, pp. 273-279.

[67] D. Zeng, and K. Sycara, Bayesian learning in negotiation, *Int. J. Hum.-Comput. Stud.*, vol. 48, no. 1, 1998, pp. 125-141.

[68] S. Buffett, and B. Spencer, A Bayesian classifier for learning opponents' preferences in multi-object automated negotiation, *Electronic Commerce Research and Applications*, vol. 6, no. 3, 2007, pp. 274-284.

[69] H. H. Bui, S. Venkatesh, and D. Kieronska, Learning other agents' preferences in multi-agent negotiation using the Bayesian classifier, *International Journal of Cooperative Information Systems*, vol. 8, no. 4, 1999, pp. 275-294.

[70] J. S. Dyer, and R. K. Sarin, Relative Risk Aversion, *Management Science*, vol. 28, no. 8, 1982, pp. 875-886

[71] S. F. Richard, Multivariate Risk Aversion, Utility Independence and Separable Utility Functions, *Management Science*, vol. 22, no. 1, 1975, pp. 12-21.

[72] J. Norstad, "An Introduction to Utility Theory" 1999; http://homepage.mac.com/j.norstad/finance/util.pdf. [Last Accessed November 2011]

[73] S. Schneider, Framing and Conflict: Aspiration Level Contingency, the status quo, and current theories of risky choice, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 18, 1992, pp. 1040-1057.

[74] M. Masvoula, C. Halatsis, D. Martakos, "Predictive Automated Negotiators Employing Risk-Seeking and Risk-Averse Strategies", *Proc. of 12th EANN/7th AIAI Joint Conf. (September 15-18, 2011, Corfu, Greece)* (*EANN/AIAI*), 2011b, pp. 325-334.

[75] M. Masvoula, C. Halatsis, D. Martakos, "Predictive Automated Negotiators with different attitudes towards risk", *Engineering Intelligent Systems Journal*, vol. 20, no. 1, 2012b , pp. 89-105.

[76] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control 3rd edition,* New Jersey: Prentice-Hall PTR, 1994.

[77] P. Abelson, and R. Joyeux, *Economic Forecasting,* Australia: Allen & Unwin, 2000.

[78] R. F. Engle, Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Ination, *Econometrica*, vol. 50, 1982, pp. 987-1008.

[79] T. Bollerslev, Generalized Autoregressive Conditional Heteroscedasticity, *Journal of Econometrics*, vol. 31, 1986, pp. 307-327.

[80] C.L. Jain, and J. Malehorn, *Benchmarking Forecasting Practices: A Guide to Improving Forecasting Performance*, New York: Graceway Publishing Company, 2006.

[81] R. H. Kilmann, and K. W. Thomas, Developing a Forced-Choice Measure of Conflict-Handling Behavior: The "Mode" Instrument, *Educational and Psychological Measurement*, vol. 37, no. 2, 1977, pp. 309-325.

[82] S. A. Ludwig, G. E. Kersten, X. Huang, "Towards a Behavioural Agent-based Assistant for e-Negotiations", *Proc. of Montreal Conf. on e-Technologies (MCETECH)*, 2006.

[83] T.S Breusch, and A.R. Pagan, Simple test for heteroscedasticity and random coefficient variation, *Econometrica (The Econometric Society)*, vol. 47, no. 5, 1979, pp. 1287-1294.

[84] D. A. Freedman, *Statistical Models: Theory and Practice. New York: Cambridge University Press*, 2005.

[85] K. M. Hornik, M. Stinchombe, and H. White, Multilayer feed-forward networks are universal approximators, *Neural Networks*, vol. 2, no. 5, 1989, pp. 359-366

[86] I.V. Papaioannou, "Optimizing Automated Negotiations with Various Learning Techniques", Doctoral Dissertation, National Technical University of Athens, Athens, 2009.

[87] S. Haykin, *Neural Networks: A Comprehensive Foundation, Second Edition*, Upper Saddle River, New Jersey: Prentice Hall Inc, 1999.

[88] B. Krose, and P. van der Smagt, *An Introduction to Neural Networks, Eight Edition,* The University of Amsterdam, 1996.

[89] R. Rojas, *Neural Networks: A Systematic Introduction*, Berlin: Springer-Verlag, 1996.

[90] D.O. Hebb, *The Organization of Behavior: A Neuropsychological Theory,* New York: Willey, 1949.

[91] N. Rochester, J. H. Holland, L.H Haibt, W.L Duda, Tests on a Cell Assembly Theory of the Action of the Brain Using a Large Digital Computer. *IRE Transactions on Information Theory*, vol. 2, 1956, pp. 80-93.

[92] F. Rosenblatt, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, *Psychological Review*, vol. 65, no. 6, 1958, pp. 386-408.

[93] B. Widrow, and M.E. Hoff, Adaptive Switching Circuits, *IRE WESCON Convension Record*, vol. 4, 1960, pp. 96-104.

[94] M.L. Minsky, and S.A. Papert, *Perceptrons*, Cambridge, MA: MIT Press, 1969.

[95] D.E. Rumelhart, G.E. Hilton, R.J. Williams, Learning Representations of Back-Propagation Errors, *Nature*, vol. 323, 1986, pp. 533-536.

[96] D. B. Parker, Learning-logic: Casting the Cortex of the Human Brain in Silicon, *Technical Report TR-47, Center for Computational Research in Economics and Management Science,* Cambridge, MA: MIT Press, 1985.

[97] C. Von der Malsburg, Self-organization of Orientation Sensitive Cells in the Striate Cortex, *Kybernetik*, vol. 14, 1973, pp. 85-100.

[98] D.J. Willshaw and von der Malsburg, “How Patterned Neural Connections can be set up by Self-organization”, *Proceedings of the Royal Society of London Series B*, vol. 194, 1976, pp. 431-445.

[99] S. Grossberg, Neural Expectation: Cerebellar and Retinal Analogs of Cells fired by Learnable or Unlearned Pattern Classes, *Kybernetik*, vol. 10, 1972, pp. 49-57.

[100] S. Grossberg, Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Detectors, *Biological Cybernetics*, vol. 23, 1976a, pp. 121-134.

[101] S. Grossberg, Adaptive Pattern Classification and Universal Recording: II. Feedback, Expectation, Olfaction, Illusions, *Biological Cybernetics*, vol. 23, 1976b, pp. 187-202.

[102] J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. of the National Academy of Sciences*, USA, vol. 79, 1982, pp. 2554-2558.

[103] T. Kohonen, Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics*, vol. 43, 1982, pp. 59-69.

[104] T. Poggio, and F. Girosi, "Networks for Approximation and Learning", *Proc. of the IEEE*, vol. 78, 1990, pp. 1481-1497.

[105] B. Boser, I. Guyon, and V.N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", *5th Annual Workshop on Computational Learning Theory*, San Mateo, CA: Morgan Kaufmann, 1992, pp. 144-152.

[106] C. Cortes, and V. Vapnik, Support Vector Networks, *Machine Learning*, vol. 20, 1995, pp. 273-297.

[107] V.N. Vapnik, *The Nature of Statistical Learning Theory,* New York: Springer-Verlag, 1995.

[108] V. N. Vapnik, *Statistical Learning Theory*, New York: Wiley, 1998.

[109] K.-L. Du, and M.N.S. Swamy, *Neural Networks in a Soft Computing Framework,* London: Springer-Verlag, 2006.

[110] J. J. More, "The Levenberg-Marquardt algorithm: Implementation and theory", *Numerical Analysis, LNM 630*, G. Watson, ed., Berlin, Heidelberg: Springer-Verlag, 1978, pp. 105–116.

[111] K. Levenberg, A Method for the Solution of Certain Problems in Least Squares, *Quart. Appl. Math.*, 1944, vol. 2, pp. 164-168.

[112] D. Marquardt, An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *SIAM J. Appl. Math.*, vol. 11, 1963, pp. 431-441.

[113] M.T. Hagan, and M. Menhaj, Training Feedforward Neural Networks with the Marquardt Algorithm, *IEEE Transactions on Neural Networks*, vol. 5, no. 6, 1994, pp. 989-993.

[114] T.M. Cover, Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition, *IEEE Transactions on Electronic Computers,* vol. 14, 1965, pp. 326-334.

[115] MJD. Powell, “Radial basis functions for multivariable interpolation: A review”, *Algorithms for Approximation,* JC Mason, MG Cox, eds., Clarendon Press, Oxford, 1987, pp. 143–167.

[116] D.S. Broomhead, and D. Lowe, Multivariate Functional Interpolation and Adaptive Networks, *Complex Systems,* vol. 2, 1988, pp. 321-355.

[117] C.A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constr Approx*, vol. 2, 1986, pp. 11–22.

[118] A.N. Tikhonov, “On solving incorrectly posed problems and method of regularization”, *Doklady Akademii Nauk USSR*, vol. 151, 1963, pp. 501–504.

[119] K. Hindriks, C. M. Jonker, D. Tykhonov, “Negotiation Dynamics: Analysis, Concession Tactics, and Outcomes”, *Proc. of the 2007 IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology*, 2007, pp. 427-433.

[120] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control,Signals,Syst.*, vol. 2, 1989, pp. 303-314.

[121] K. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks*, vol. 2, no. 3, 1989, pp. 183–192.

[122] C. Xiang, S.Q. Ding, T.H. Lee. Geometrical interpretation and architecture selection of MLP, *IEEE Transactions on Neural Networks*, vol. 16, no. 1, 2005, pp. 84–96.

[123] J. Park, I.W. Sanberg, Universal approximation using radial-basis function networks, *Neural Computation*, vol. 3, no. 2, 1991, pp. 246–257

[124] Y. Liao, S.-C. Fang, H.L.W. Nuttle, Relaxed conditions for radial-basis function networks to be universal approximators, *Neural Networks*, vol. 16, no. 7, 2003, pp. 1019–1028.

[125] A.R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Transactions on Information Theory*, vol. 39, no. 3, 1993, pp. 930–945.

[126] C. Li, J. Giampapa, and K. Sycara, Bilateral Negotiation Decisions with Uncertain Dynamic Outside Options, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Application and Reviews,* vol. 36, no. 1, 2006, pp. 31-44.

[127] D.D. Van Bragt, and J.A. La Poutré, Why Agents for Automated Negotiations Should Be Adaptive, *Netnomics*,vol. 5, no. 2, 2003, pp. 101-118.

[128] B.M. Wilamowski, and H. Yu, Improved Computation for Levenberg-Marquardt Training, *Transactions on Neural Networks*, vol. 21, no. 6, 2010, pp. 930-937.

[129] G. Lera, and M. Pinzolas, “A quasi-local Levenberg-Marquardt algorithm for Neural Network Training”, Proc. of the 1998 Int. Joint Conf. on Neural Networks, 1998, pp. 2242-2246.

[130] P. Niyogi, and F. Girosi, Generalization bounds for function approximation from scattered noisy data, *Advances in Computat Math*, vol. 10, 1999, pp. 51–80

[131] A.R. Barron, Approximation and Estimation Bounds of Artificial Neural Networks, *Machine Learning*, vol. 14, no. 1, 1994, pp. 115-133.

[132] O. Goldreich, *Computational complexity: A Conceptual Perspective,* New York: Cambridge University Press, 2006.

[133] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning 1st edition,* Boston, MA: Addison−Wesley Longman Publishing Co,Inc, 1989.

[134] J. Yang, and V. Honavar, Feature Subset Selection using a Genetic Algorithm, *IEEE Intelligent Systems and their Applications*, vol. 13, no. 2, 1998, pp. 44-49.

[135] D. Shanthi, G. Sahoo, and N. Saravanan, Evolving Connection Weights of Artificial Neural Networks Using Genetic Algorithm with Application to the Prediction of Stroke Disease, *International Journal of Soft Computing*, vol. 4, no. 2, 2009, pp. 95-102.

[136] S.H. Ling, F.H.F. Leung, H.K. Lam, Y. Lee, and P.K.S. Tam. A Novel Genetic-Algorithm-Based Neural Network for Short-Term Load Forecasting, *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, 2003, pp. 793-799.

[137] A.U. Khan, T.K. Bandopadhyaya, and S. Sharma, Genetic Algorithm Based Backpropagation Neural Network performs better than Backpropagation Neural Network in Stock Rates Prediction, *International Journal of Computer Science and Network Security*, vol. 8, no. 7, 2008, pp. 162-166.

[138] V. P. Plagianakos, D.G. Sotiropoulos, and M.N. Vrahatis, “Evolutionary Algorithms for Integer Weight Neural Network Training”, *Technical Report No TR98-04*, Patras, Greece: Department of Mathematics, University of Patras, 1998.

[139] Z. Liu, A. Liu, C. Wang, Z. Niu, Evolving Neural Network using Real Coded Genetic Algorithm (GA) for Multispectral Image Classification, *Future Gener. Comput. Syst.*, vol. 20, no. 7, 2004, pp. 1119-1129.

[140] S. Chakravorty, and A. Chakrabarti, “Hidden Layer Optimization of Neural Network using Computational Technique”, *Proc. of the International Conference on Advances in Computing, Communication and Control, (ICAC3 09),* New York: ACM, 2009, pp. 559-561.

[141] R. S. Sexton, Identifying irrelevant input variables in chaotic time series problems: Using the genetic algorithm for training neural networks, *Journal of Computational Intelligence in Finance*, 1998, pp. 34-41.

[142] A. Fiszelew, P. Britos, A. Ochoa, H. Merlino, E. Fernández, and R. García-Martínez, Finding Optimal Neural Network Architecture Using Genetic Algorithms, *Research in Computing Science*, vol. 27, 2007, pp. 15-24.

[143] J. Dávila, “Genetic Optimization of NN Topologies for the Task of Natural Language Processing”, *Proc. of the Int. Joint Conf. on Neural Networks*, 1999, pp. 821-826.

[144] T. Rajkumar, and D.E. Thompson, D. E. “Optimization of a Model to Predict Salinity Intrusion in San Francisco Bay Estuary Using a Genetic Algorithm”, *Proc. of the Int. Environmental Modeling &Software Society (IEMSS ’02),* 2002, pp. 142-147.

[145] G.-H. Kim, J.-E. Yoon, S.-H. An, H.-H. Cho, and K.-I. Kang, Neural network model incorporating a genetic algorithm in estimating construction costs, *Building and Environment*, vol. 39, no. 11, 2004, pp. 1333-1340.

[146] X. Zhang, X.H. Tong, and M.L. Liu, “Genetic Algorithm Optimized Neural Network Classification and Overlapping Pixel Change Detection Based on Remote Sensing for Urban Sprawl: A Case Study in Jiading District of Shanghai, China”, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37 (Part B7), 2008, pp. 743-748.

[147] X. Yao, “Evolving Artificial Neural Networks”, *Proc. of the IEEE*, 1999, pp. 1423-1447.

[148] D.M. Rodvold, “Method for Simultaneously Optimizing Artificial Neural Network Inputs and Architectures using Genetic Algorithms”, United States Patent Application 20020059154, 2002; http://freepatentsonline.com/y2002/0059154.html. [Last accessed October 2012]

[149] P. Makvanti, J. Jassbi, and S. Khanmohammadi, “Application of Genetic Algorithm and Neural Network in Forecasting with Good Data”, *Proc. of the 6th WSEAS Int. Conf. on Neural Networks,* 2005, pp. 56-61.

[150] M. Sheikhan, and B. Movaghar, Exchange Rate Prediction Using an Evolutionary Connectionist Model, *World Applied Sciences Journal*, vol. 7(Special Issue of Computer & IT), 2009, pp. 8-16.

[151] F. Ahmad, N.A. Mat-Isa, and Z. Hussain, „Simultaneous Input Selection and Hidden Nodes Optimization for Sunspot Time Series Prediction“, *Proc. of the 4th Int. Conf. on Mathematical/Analytical Modelling and Computer Simulation (AMS ’10)* Washington, DC: IEEE Computer Society, 2010, pp. 56-59.

[152] J. Peralta, G. Gutierrez, and A. Sanchis, “Design of Artificial Neural Networks based on Genetic Algorithms to Forecast Time Series”, *Innovations in Hybrid Intelligent Systems, Advances in Soft Computing*, E. Corchado et. al, eds., Berlin, Heidelberg: Springer-Verlag, 2007, pp. 231-238.

[153] Y. Chen, “Extending the Scalability of Linkage Learning Genetic Algorithms”, *Studies in Fuzziness and Soft Computing,* Berlin, Heidelberg : Springer-Verlag, 2006.

[154] X. Yao, and Y. Lin, A new evolutionary system for evolving artificial neural networks, *IEEE Transactions on Neural Networks*, vol. 8, no. 3, 1997, pp. 694-713.

[155] J. Brzostowski, “Predictive Decision-making Mechanisms based on Off-line and On-line Reasoning”, doctoral dissertation, Swinburne University of Technology, Melbourne, Australia, 2007.

[156] N. Kasabov, *Evolving Connectionist Systems: The Knowledge Engineering Approach*, London: Springer-Verlag, 2007.

[157] W.S. Mc Culloch, and W. Pitts, A Logical Calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, vol. 5, 1943, pp. 115-133.