

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

## Πολυτεχνική Σχολή

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

Σχηματισμός Συμμαχιών στα Πολυπρακτορικά  
Συστήματα

Coalition Formation in Multi-agent Systems

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΟΛΒΑΛΗΣ ΣΤΥΛΙΑΝΟΣ



### Επιβλέποντες Καθηγητές:

Δασκαλοπούλου Ασπασία, Επίκουρη Καθηγήτρια

Τσουκαλάς Ελευθέριος, Καθηγητής

Βόλος, Σεπτέμβριος 2016

## Ευχαριστίες

Η παρούσα αποτελεί Διπλωματική Εργασία με την οποία ολοκληρώνεται ο προπτυχιακός κύκλος σπουδών μου στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας. Θα ήθελα λοιπόν, με την αφορμή αυτή, να ευχαριστήσω όλους εκείνους που στάθηκαν δίπλα μου, σε ολόκληρη τη φοιτητική μου πορεία.

Πριν την παρουσίαση των αποτελεσμάτων στα οποία κατέληξα, αισθάνομαι την υποχρέωση να ευχαριστήσω ορισμένους από τους ανθρώπους που γνώρισα, συνεργάστηκα και έπαιξαν πολύ σημαντικό ρόλο στην πραγματοποίησή της.

Καταρχήν, ιδιαίτερες ευχαριστίες αξίζουν στην οικογένειά μου για τη διαχρονική συμπαράστασή τους, την υλική και ηθική στήριξη των επιλογών μου.

Στη συνέχεια θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια της διπλωματικής μου εργασίας, Επίκουρη Καθηγήτρια Δασκαλοπούλου Ασπασία για την πολύτιμη καθοδήγηση της, την εμπιστοσύνη και εκτίμηση που μου έδειξε. Καθώς επίσης και τον Καθηγητή της επιτροπής της εργασίας μου, Τσουκαλά Ελευθέριο.

Κλείνοντας, θα ήθελα να ευχαριστήσω ολόκληρο το ανθρώπινο δυναμικό της Σχολής, για τις γνώσεις, τις εμπειρίες και τα βιώματα, που με ζήλο, όρεξη και υπομονή, τις περισσότερες φορές, μετέδωσε σε μένα και στους συμφοιτητές μου.

Τέλος, ένα μεγάλο ευχαριστώ θα ήθελα να απευθύνω σε συμφοιτητές και φίλους μου, καθώς, χωρίς τη συνεργασία, την αλληλοϋποστήριξη και την ανταλλαγή ιδεών και εμπειριών, θα ήταν αδύνατη η περάτωση μιας τόσο δύσκολης Σχολής. Ελπίζω να ανταπέδωσα και να συνεχίσω να ανταποδίδω τη βοήθεια που μού προσέφεραν. Θα ήθελα να τους ευχαριστήσω, όμως, όχι μόνο για τη βοήθεια και την αλληλοϋποστήριξη στο πλαίσιο των σπουδών, αλλά κυρίως για το γεγονός ότι έκαναν πολύ όμορφα τα φοιτητικά μου χρόνια.

Σας ευχαριστώ πολύ!

## Σχηματισμός Συμμαχιών στα Πολυπρακτορικά Συστήματα

**Λέξεις-κλειδιά:** πράκτορας, πολυπρακτορικό σύστημα, χρησιμότητα, συμμαχία, δομή συμμαχιών, πρόβλημα παραγωγής δομής συμμαχιών, βέλτιστη δομή συμμαχίας

### ΠΕΡΙΛΗΨΗ

Ο προγραμματισμός και σχεδιασμός ανεξάρτητων πρακτόρων λογισμικού, ικανών να δρουν και να διαπραγματεύονται στα πλαίσια μίας σταθερής συμμαχίας, προσφέρει πολλά οφέλη στους χρήστες. Ο σχηματισμός συμμαχιών είναι ένα από τα θεμελιώδη θέματα για μελέτη στο χώρο της κατανεμημένης τεχνητής νοημοσύνης και των πολυπρακτορικών συστημάτων. Πρόκειται για τη δημιουργία συνεκτικών ομάδων, αποτελούμενων από ξεχωριστούς, αυτόνομους πράκτορες με σκοπό την ικανοποίηση των ατομικών ή συλλογικών τους στόχων. Καθοριστικό ρόλο στην προσπάθεια διαμόρφωσης μιας τέτοιας αποδοτικής ομάδας κατέχει η σωστή επιλογή, μεταξύ των δυνατών, πρακτόρων για την επίτευξη ενός συγκεκριμένου στόχου. Το πρόβλημα αυτό καλείται *παραγωγή δομής συμμαχίας*, το οποίο υπολογίζει τη χρησιμότητας κάθε πιθανής συμμαχίας, δηλαδή μιας τιμής που υποδεικνύει πόσο ωφέλιμη θα ήταν μια συμμαχία αν σχηματιζόταν. Μετά τον υπολογισμό των τιμών αυτών εντοπίζεται ένας συνδυασμός συμμαχιών, με τον περιορισμό πως κάθε πράκτορας ανήκει μόνο σε μία συμμαχία, τέτοιος ώστε το συνολικό κέρδος του συστήματος να μεγιστοποιείται. Η επίλυση αυτού του προβλήματος είναι ιδιαίτερα απαιτητική, καθώς το πλήθος των πιθανών λύσεων προς εξέταση αυξάνεται εκθετικά με τον αριθμό των πρακτόρων του συστήματος.

# Coalition Formation in Multi-agent Systems

**Key-words:** agent, multiagent system, value, coalition, coalition structure, coalition structure generation problem, best coalition structure

## ABSTRACT

Programming and designing independent software agents, that are able to act and negotiate in stable coalitions, would bring many benefits to the users. Coalition formation is one of the fundamental subjects in study of the distributed artificial intelligence and multi-agent systems field. This is about the creating of cohesive groups consisting of separated, autonomous agents in order to satisfy their individual and collective goals. Decisive role in the effort of generation of such an effective group holds the proper selection of available agents to achieve a particular goal. This problem called *coalition structure production*, which estimates the *value* of any possible coalition, that is a value indicating how much beneficial would be a coalition, if it had formed. After the estimation of these values, a combination of coalitions is identified, in which each agent belongs to only one coalition, such one so the total profit of the system is maximized. This problem solution is extremely demanding, as the number of possible solutions to research increases exponentially with the number of system agents.

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΠΕΡΙΛΗΨΗ</b> .....	3
<b>ABSTRACT</b> .....	4
<b>ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ</b> .....	7
1.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΕΔΙΟΥ ΕΦΑΡΜΟΓΗΣ .....	7
1.2 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ .....	9
1.2 ΔΟΜΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ .....	11
2.1 ΕΙΣΑΓΩΓΗ .....	12
2.2 ΤΙ ΕΙΝΑΙ ΕΝΑΣ ΠΡΑΚΤΟΡΑΣ; .....	13
2.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΠΡΑΚΤΟΡΩΝ .....	14
2.4 ΠΕΡΙΒΑΛΛΟΝΤΑ ΕΡΓΑΣΙΩΝ ΤΩΝ ΠΡΑΚΤΟΡΩΝ .....	17
2.5 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΤΩΝ ΕΥΦΥΩΝ ΠΡΑΚΤΟΡΩΝ .....	20
<b>ΚΕΦΑΛΑΙΟ 3: ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ</b> .....	25
3.1 ΕΙΣΑΓΩΓΗ .....	25
3.2 ΟΡΙΣΜΟΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ .....	26
3.3 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΣΤΑ ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ .....	29
3.3.1 ΕΠΙΚΟΙΝΩΝΙΑ ΠΡΑΚΤΟΡΩΝ .....	29
3.3.2 ΣΥΝΤΟΝΙΣΜΟΣ ΠΡΑΚΤΟΡΩΝ .....	31
3.3.3 ΔΙΑΠΡΑΓΜΑΤΕΥΣΗ .....	32
<b>ΚΕΦΑΛΑΙΟ 4: ΣΧΗΜΑΤΙΣΜΟΣ ΣΥΜΜΑΧΙΩΝ</b> .....	34
4.1 ΤΙ ΕΙΝΑΙ ΣΥΜΜΑΧΙΑ ΣΤΗΝ ΚΟΙΝΩΝΙΑ .....	34
4.2 Η ΣΠΟΥΔΑΙΟΤΗΤΑ ΤΩΝ ΣΥΜΜΑΧΙΩΝ .....	35
4.3 ΣΥΜΜΑΧΙΕΣ ΜΕΤΑΞΥ ΠΡΑΚΤΟΡΩΝ .....	36
4.3.1 ΕΙΣΑΓΩΓΗ .....	36
4.3.2 ΣΥΝΕΤΑΙΡΙΣΤΙΚΗ ΘΕΩΡΙΑ ΠΑΙΓΝΙΩΝ .....	38
4.4 ΣΥΜΜΑΧΙΕΣ ΣΤΑ ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ .....	39
4.4.1 ΕΙΣΑΓΩΓΗ .....	39
4.4.2 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ .....	40
4.5 ΣΤΑΔΙΑ ΤΗΣ ΣΥΝΕΤΑΙΡΙΣΤΙΚΗΣ ΕΝΕΡΓΕΙΑΣ .....	43
4.5.1 ΥΠΟΛΟΓΙΣΜΟΣ ΑΞΙΑΣ ΤΗΣ ΣΥΜΜΑΧΙΑΣ (COALITIONAL VALUE CALCULATION) .....	43
4.5.2 ΠΑΡΑΓΩΓΗ ΔΟΜΗΣ ΣΥΜΜΑΧΙΑΣ (COALITION STRUCTURE GENERATION) .....	44
4.5.3 ΚΑΤΑΝΟΜΗ ΚΕΡΔΟΥΣ (PAYOFF DISTRIBUTION) .....	47
<b>ΚΕΦΑΛΑΙΟ 5: ΠΑΡΑΓΩΓΗ ΔΟΜΗΣ ΣΥΜΜΑΧΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΟΥ IP</b> .....	52
5.1 ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ .....	52

5.2 ΑΛΓΟΡΙΘΜΟΣ ΑΚΕΡΑΙΩΝ ΔΙΑΜΕΡΙΣΕΩΝ (IP ALGORITHM) .....	55
5.2.1 ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ ΧΩΡΟΥ ΑΝΑΖΗΤΗΣΗΣ.....	55
5.2.2 ΔΙΧΟΤΟΜΗΣΗ ΤΟΥ ΧΩΡΟΥ ΑΝΑΖΗΤΗΣΗΣ .....	55
5.2.3 ΥΠΟΛΟΓΙΣΜΟΣ ΟΡΙΩΝ ΓΙΑ ΤΟΥΣ ΥΠΟΧΩΡΟΥΣ.....	57
5.3 ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ ΔΟΜΗΣ ΣΥΜΜΑΧΙΑΣ.....	58
5.3.1 ΣΑΡΩΣΗ ΤΗΣ ΕΙΣΟΔΟΥ .....	59
5.3.2 ΕΠΙΛΟΓΗ ΚΑΙ ΑΝΑΖΗΤΗΣΗ ΕΝΟΣ ΥΠΟΧΩΡΟΥ.....	63
5.4 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ .....	71
<b>ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΈΡΕΥΝΑ.....</b>	<b>75</b>
Βιβλιογραφία.....	78

# ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

## 1.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΕΔΙΟΥ ΕΦΑΡΜΟΓΗΣ

Στη σύντομη ιστορία της τεχνολογίας των υπολογιστών, η αντίληψη που έχουμε για τη χρησιμότητα και τις πρακτικές εφαρμογές των ηλεκτρονικών υπολογιστών έχει περάσει από διάφορα στάδια εξέλιξης. Από τις πρώτες υπολογιστικές μηχανές μέχρι και τη δεκαετία του 1960, οι υπολογιστές θεωρήθηκαν ένα εργαλείο που εκτελεί κατ' εξοχήν αριθμητικούς υπολογισμούς. Με την εμφάνιση των βάσεων δεδομένων και τη βελτίωση των συσκευών μαζικής αποθήκευσης, άρχισε να κατακτά κεντρικό ρόλο η έννοια της επεξεργασίας της πληροφορίας. Τα τελευταία χρόνια, χάρη στην καθολική κυριαρχία του Διαδικτύου, εμφανίστηκε η έννοια της αλληλεπίδρασης και της επικοινωνίας των οντοτήτων λογισμικού, χρησιμοποιώντας και επεκτείνοντας τις ικανότητες ταχύτατων υπολογισμών και μαζικής επεξεργασίας πληροφορίας, που κυριάρχησαν στο κοντινό παρελθόν. Η νέα αυτή έννοια εισάγει τον όρο της κοινότητας οντοτήτων λογισμικού, που οδηγεί σε νέους τρόπους αντίληψης, σχεδιασμού και υλοποίησης των υπολογιστικών συστημάτων.

Οι υπολογιστές δεν είναι καλοί στο να γνωρίζουν τι πρέπει να κάνουν κάθε δεδομένη χρονική στιγμή. Κάθε ενέργεια, που υλοποιεί ένας υπολογιστής πρέπει να είναι προσδοκώμενη και ρητώς σχεδιασμένη και γραμμένη σε κώδικα από έναν προγραμματιστή. Αν ένα υπολογιστικό σύστημα αντιμετωπίσει κάποια κατάσταση, την οποία ο σχεδιαστής του δεν είχε προβλέψει, τότε το αποτέλεσμα συνήθως είναι επικίνδυνο για την διατήρηση και επιβίωση του ίδιου του συστήματος (1).

Κατά κύριο λόγο, είναι εύκολο για τους χρήστες των υπολογιστών, να θεωρούν τους δεύτερους ως υπάκουους και πιστούς υπηρέτες. Για τις περισσότερες εφαρμογές, αυτό είναι πλήρως αποδεκτό. Ωστόσο, για ένα συνεχώς αυξανόμενο αριθμό λειτουργιών, απαιτούμε συστήματα τα οποία είναι σε θέση να αποφασίζουν τα ίδια, τι χρειάζεται να εκτελέσουν, με σκοπό να ικανοποιήσουν τους στόχους των σχεδιαστών τους. Τέτοια υπολογιστικά συστήματα, τα αποκαλούμε *πράκτορες* (agents).

Ο πράκτορας αποτελεί το ενοποιητικό θέμα, που κρύβεται πίσω από το μεγάλο επιστημονικό πεδίο της Τεχνητής Νοημοσύνης. Ο κλάδος αυτός αποτελεί τον κατάλληλο χώρο για μελέτη των πρακτόρων, οι οποίοι προσλαμβάνουν αντιλήψεις από το περιβάλλον και πραγματοποιούν ενέργειες. Κάθε ένας τέτοιος πράκτορας υλοποιεί μια συνάρτηση που αντιστοιχίζει ακολουθίες αντιλήψεων σε ενέργειες. Στις παραγράφους που ακολουθούν θα προσπαθήσουμε να εξηγήσουμε

γιατί θεωρείται η τεχνητή νοημοσύνη θέμα άξιο σοβαρής μελέτης, και θα επιχειρήσουμε να προσδιορίσουμε τι ακριβώς είναι.

Ονομάζουμε το είδος μας homo sapiens, δηλαδή άνθρωπος ο σοφός, επειδή οι νοητικές μας ικανότητες είναι πολύ σημαντικές για εμάς. Για χιλιάδες χρόνια προσπαθούμε να κατανοήσουμε το πώς σκεπτόμαστε. Δηλαδή, πώς μια «χούφτα» ύλης μπορεί να αντιλαμβάνεται, να κατανοεί, να προβλέπει και να χειρίζεται έναν κόσμο πολύ μεγαλύτερο και πολύ πιο πολύπλοκο από τον εαυτό της. Το πεδίο της τεχνητής νοημοσύνης (artificial intelligence), για συντομία TN, πηγαίνει ακόμα πιο πέρα. Επιχειρεί όχι μόνο να κατανοήσει αλλά και να κατασκευάσει νοήμονες οντότητες.

Η τεχνητή νοημοσύνη είναι μία από τις νεότερες επιστήμες. Τα πιο αποφασιστικά βήματα στον τομέα αυτό άρχισαν να πραγματοποιούνται λίγο μετά το Δεύτερο Παγκόσμιο Πόλεμο, και ο ίδιος ο όρος εμφανίστηκε το 1956. Σήμερα, η TN συνδυάζει μια τεράστια ποικιλία επιμέρους πεδίων, τα οποία καλύπτουν ένα φάσμα, που ξεκινά από γενικούς τομείς, όπως η μάθηση και η αντίληψη, και φτάνει σε συγκεκριμένες εργασίες, όπως το σκάκι, η απόδειξη μαθηματικών θεωρημάτων, η συγγραφή ποίησης και η διάγνωση ασθενειών. Η TN συστηματοποιεί και αυτοματοποιεί τις διανοητικές εργασίες, γι' αυτό και μπορεί να έχει εφαρμογή σε οποιαδήποτε σφαίρα της ανθρώπινης διανοητικής δραστηριότητας. Με αυτήν την έννοια είναι πραγματικά ένα οικουμενικό γνωστικό πεδίο (2)

Στις προηγούμενες παραγράφους υποστηρίξαμε ότι η TN είναι σημαντική και συναρπαστική. Ας δούμε όμως τώρα πιο αναλυτικά το για τι ακριβώς πρόκειται. Η τεχνητή νοημοσύνη είναι ο κλάδος της πληροφορικής, ο οποίος ασχολείται με τη σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων, που μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς και υπονοούν έστω και στοιχειώδη ευφυΐα. Οι έρευνες που πραγματοποιήθηκαν στο χώρο αυτό είχαν ως αποτέλεσμα τη σημαντική πρόοδο σε βασικά ζητήματα της επιστήμης των υπολογιστών, όπως τα νευρωνικά δίκτυα (neural networks),εξελικτικός προγραμματισμός (evolutionary programming), μηχανική μάθηση (machine learning),επεξεργασία φυσικής γλώσσας (natural language processing) και αντικειμενοστραφής προγραμματισμός (object-oriented programming).

Κατά καιρούς έχουν δοθεί ποικίλοι ορισμοί της επιστήμης της τεχνητής νοημοσύνης. Οι ορισμοί αυτοί διαφέρουν στο γεγονός ότι εστιάζουν το ενδιαφέρον τους σε διαφορετικά σημεία. Κάποια από τα σημεία αυτά ενδιαφέροντος είναι οι διαδικασίες σκέψης και συλλογιστική, η συμπεριφορά, η εγγύτητα προς τις ανθρώπινες επιδόσεις και τέλος η ορθολογικότητα, δηλαδή η έννοια της νοημοσύνης που χαρακτηρίζει ένα σύστημα όταν αυτό κάνει «το σωστό», δεδομένων όσων γνωρίζει. Μερικοί από αυτούς τους ορισμούς δίνονται παρακάτω:

*«Η αυτοματοποίηση των δραστηριοτήτων που συσχετίζουμε με την ανθρώπινη σκέψη, όπως η λήψη αποφάσεων, η επίλυση προβλημάτων, η μάθηση...» (3)*



*«Η μελέτη του πώς μπορούμε να κάνουμε τους υπολογιστές να κάνουν πράγματα, στα οποία προς το παρόν οι άνθρωποι είναι καλύτεροι» (4)*

*«Η μελέτη των υπολογιστικών εργασιών που μας δίνουν τη δυνατότητα να αντιλαμβανόμαστε, να συλλογίζομαστε, και να ενεργούμε» (5)*

*«Η τεχνητή νοημοσύνη ασχολείται με την ευφυή συμπεριφορά των τεχνουργημάτων» (6)*

Υπήρξαν πολλές ενστάσεις απέναντι στην τεχνητή νοημοσύνη και τις δυνατότητες, που παρουσίαζε κατά το πέρασμα των χρόνων. Μέχρι κάποιο σημείο, το γεγονός αυτό είναι κατανοητό, καθώς η ιδέα μιας ευφυούς μηχανής που αποδεικνύεται πιο έξυπνη και που μπορεί να σκεφτεί πιο γρήγορα από τον άνθρωπο στο μέλλον, είναι τρομακτική. Πρόκειται για έναν κλάδο, που έχει τη δύναμη να διαταράξει κάθε πλευρά της καθημερινής μας ζωής και η αβεβαιότητα αυτή μπορεί να μοιάζει απειλητική στους ανθρώπους, που βλέπουν με αμφιβολία τις αλλαγές που μπορούν να επέλθουν στο μέλλον.

Παρά τις ανησυχίες αυτές όμως, είναι κοινά αποδεκτή η συμβολή της στην καθημερινότητα μας με την ανάπτυξη τεχνολογιών, που αποσκοπούν στην βελτίωση της ποιότητας ζωής, όπως η βιοτεχνολογία, η νανοτεχνολογία, η ρομποτική και η γνωσιακή επιστήμη. Η ΤΝ είναι μία συναρπαστική και δυναμική περιοχή έρευνας. Αλλάζει με ταχείς ρυθμούς, και η μελέτη της στην πάροδο του χρόνου γεννά και αναπτύσσει πολλές θαυμάσιες και ενδιαφέρουσες ιδέες. Ωστόσο, πρέπει ακόμα να πετύχουμε τον ανώτερο σκοπό της τεχνητής νοημοσύνης. Έναν σκοπό για τον οποίο αρκετοί άνθρωποι διαφωνούν για το αν θα μπορέσει ποτέ να επιτευχθεί, για πολλούς λόγους. Συνεπώς, οποιοσδήποτε μελετά την ΤΝ θα πρέπει να έχει ανοιχτό μυαλό ως προς την καταλληλότητα των ιδεών που προτείνονται. Θα πρέπει συνεχώς να αναρωτιούνται πόσο καλά αυτές οι ιδέες λειτουργούν και κατά πόσο υπάρχουν καλύτερες ιδέες ή προσεγγίσεις. (7)

## 1.2 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Η ανάγκη για αντιμετώπιση όλο και πιο πολύπλοκων ζητημάτων σε τομείς και πεδία της καθημερινής ζωής του ανθρώπου οδήγησε τους μελετητές της Τεχνητής Νοημοσύνης να επεκτείνουν τις δυνατότητες των πρακτόρων λογισμικού.

Οι αυτόνομοι πράκτορες σχεδιάστηκαν για να πετυχαίνουν στόχους που είναι προκαθορισμένοι από τους δημιουργούς τους. Πολλοί από αυτούς τους στόχους αποδεικνύουν πως η ύπαρξη και μόνο ενός πράκτορα δεν είναι αρκετή για την ολοκλήρωση ενός έργου. Η αδυναμία αυτή κατέστησε αναγκαία την διαμόρφωση κοινοτήτων πρακτόρων, που ονομάζουμε πολυπρακτορικά

συστήματα. Τα συστήματα αυτά αποτελούνται από περισσότερους του ενός ευφυείς πράκτορες, οι οποίοι αλληλοεπιδρούν μέσα σε ένα κοινό περιβάλλον. Χρησιμοποιούνται για την επίλυση προβλημάτων, τα οποία είναι δύσκολο ή ακόμα και αδύνατο να αντιμετωπιστούν αποτελεσματικά από ένα και μόνο πράκτορα ή οποιοδήποτε άλλο μονολιθικό σύστημα. Με τον τρόπο αυτό η ικανοποίηση των στόχων, για τους οποίους σχεδιάστηκαν οι πράκτορες πραγματοποιείται πιο αποτελεσματικά αποφέροντας μεγαλύτερο κέρδος σε κόστος και χρόνο.

Η συνύπαρξη των πρακτόρων μέσα σε ένα κοινό περιβάλλον δεν αποτελεί και τη μοναδική προϋπόθεση για την επίτευξη των στόχων του προγραμματιστή τους. Οι πράκτορες πρέπει να είναι σε θέση να αντιλαμβάνονται πως στο περιβάλλον μέσα στο οποίο ζουν και δρουν «φιλοξενούνται» κι άλλοι όμοιοί τους. Πρέπει να διαθέτουν μηχανισμούς επικοινωνίας και αλληλεπίδρασης για να έρχονται σε επαφή με τους γύρω τους, με σκοπό να ενημερώνονται για τις προθέσεις τους καθώς και τις δυνατότητές τους.

Όπως, αναφέρθηκε ένα πολυπρακτορικό σύστημα χρησιμοποιείται για την κάλυψη αναγκών, για τις οποίες οι ατομικές προσπάθειες ενός πράκτορα δεν είναι αρκετές. Ένας αποδοτικός τρόπος να εκτελούνται εργασίες και να μεγιστοποιείται το κέρδος που μπορεί να αποκομίσει συνολικά το πολυπρακτορικό σύστημα, είναι η ανταλλαγή διαθέσιμων πόρων και η διαμόρφωση συμμαχιών μεταξύ των πρακτόρων που το αποτελούν.

Οι συμμαχίες αυτές αποτελούν ομάδες πρακτόρων, οι οποίοι μέσω μηχανισμών επικοινωνίας, διαπραγμάτευσης και συντονισμού που διαθέτουν συνάπτουν συμφωνίες για συνεργασία και εκτέλεση ενεργειών που αποσκοπούν στην ολοκλήρωση ενός κοινού έργου. Μια τέτοια συμμαχία μπορεί να πραγματοποιηθεί αν και μόνο αν κάθε πράκτορας που συμμετέχει σε αυτήν κερδίζει περισσότερα από αυτά που θα κέρδιζε αν εργαζόταν μόνος του.

Όσο και δελεαστική αν ακούγεται, όμως, η ιδέα της συνεργασίας πρακτόρων για την επίτευξη ενός στόχου αυξάνοντας ταυτόχρονα τις απολαβές, η επιλογή του κατάλληλου συνδυασμού των πρακτόρων για σύναψη συμμαχίας σε ένα πολυπρακτορικό σύστημα παραμένει απαιτητική. Το πρόβλημα αυτό καλείται παραγωγή δομής συμμαχίας (coalition structure generation problem). Οι πράκτορες του συστήματος πρέπει να χωριστούν σε συμμαχίες, με τέτοιο τρόπο ώστε ο συνδυασμός τους να μεγιστοποιεί το συνολικό κέρδος όλης της κοινότητας. Ακόμη, πρέπει να διασφαλίζεται πως κάθε πράκτορας ανήκει σε μία και μόνο συμμαχία κάθε φορά.

Το πρόβλημα εύρεσης ενός τέτοιου αποδοτικού και κερδοφόρου συνδυασμού συμμαχιών είναι ιδιαίτερα απαιτητικό, καθώς το πλήθος των πιθανών λύσεων προς εξέταση αυξάνεται εκθετικά με τον αριθμό των πρακτόρων που συμμετέχουν στη διαδικασία αυτή. Μέχρι σήμερα, αρκετοί αλγόριθμοι έχουν προταθεί για την αντιμετώπιση του προβλήματος, χρησιμοποιώντας διαφορετικές τεχνικές, όπως δυναμικός προγραμματισμός, ακέραιος προγραμματισμός ή

στοχαστική αναζήτηση. Οι δυσκολίες που έχουν όλοι αυτοί οι αλγόριθμοι να υπερπηδήσουν είναι ο χρόνος εκτέλεσης, η ποιότητα του αποτελέσματος καθώς και οι απαιτήσεις σε χώρο μνήμης.

## 1.2 ΔΟΜΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η παρούσα Διπλωματική Εργασία αποτελείται από τα παρακάτω κεφάλαια:

Στο **Κεφάλαιο 1**, *Εισαγωγή*, παρουσιάζεται ο κλάδος της Τεχνητής Νοημοσύνης ως το γενικό πεδίο εφαρμογής και μία σύντομη περιγραφή του αντικειμένου της Διπλωματικής Εργασίας.

Στο **Κεφάλαιο 2**, *Ευφυείς πράκτορες*, γίνεται παρουσιάζονται οι ιδιότητες και τα χαρακτηριστικά που διαθέτουν οι ευφυείς πράκτορες. Ακόμη, γίνεται μια κατηγοριοποίηση των πρακτόρων καθώς και των περιβαλλόντων εργασίας μέσα στα οποία αυτοί δρουν.

Στο **Κεφάλαιο 3**, *Πολυπρακτορικά Συστήματα*, παρουσιάζονται περισσότερες βασικές έννοιες του θεωρητικού υπόβαθρου, που αφορούν στη δημιουργία κοινοτήτων με βασικό συστατικό στοιχείο τον πράκτορα, όπως αυτός παρουσιάστηκε στο **Κεφάλαιο 2**. Αναφέρονται ιδιότητες και χαρακτηριστικά αυτών των συστημάτων και περιγράφονται οι τρόποι αλληλεπίδρασης των πρακτόρων στα πλαίσια ενός πολυπρακτορικού συστήματος.

Στο **Κεφάλαιο 4**, *Σχηματισμός Συμμαχιών*, παρουσιάζεται η δημιουργία συμμαχιών ως αποτέλεσμα της αλληλεπίδρασης μεταξύ των πρακτόρων. Γίνεται μία σύντομη αναφορά στη σπουδαιότητα της ιδέας της συμμαχίας για την κοινωνία. Επίσης, αναφέρεται το πεδίο της Θεωρίας Παιγνίων ως η βασική πηγή εφαρμογών στη διαμόρφωση συμμαχίας στα πλαίσια ενός πολυπρακτορικού συστήματος, καθώς και τα στάδια της συνεταιριστικής ενέργειας για να ολοκληρωθεί η διαδικασία σχηματισμού της συμμαχίας.

Στο **Κεφάλαιο 5**, *Παραγωγή Δομής Συμμαχίας και εφαρμογή αλγορίθμου IP*, αναλύεται το πρόβλημα εύρεσης του αποδοτικότερου συνδυασμού συμμαχιών μέσα από τον αλγόριθμο Integer Partitions (IP) καθώς και την εκτέλεση του για ένα παράδειγμα.

Στο **Κεφάλαιο 6**, *Συμπεράσματα*, παραθέτονται τα συμπεράσματα της εργασίας και προτάσεις για ερευνητική κατεύθυνση μελλοντικών εργασιών

## ΚΕΦΑΛΑΙΟ 2: ΕΥΦΥΕΙΣ ΠΡΑΚΤΟΡΕΣ

### 2.1 ΕΙΣΑΓΩΓΗ

Η τεχνολογία, που αναφέρεται στους ευφυείς πράκτορες αποτελεί ένα αυτόνομο πεδίο με εξαιρετική ερευνητική δραστηριότητα τα τελευταία χρόνια. Συνδυάζει στοιχεία από την Τεχνητή Νοημοσύνη, τον Αντικειμενοστραφή και τον Παράλληλο Προγραμματισμό και βρίσκει εφαρμογές σε πάρα πολλούς τομείς. Η περιοχή των πρακτόρων έλκει το ενδιαφέρον της επιστημονικής κοινότητας λόγω των προβλημάτων, προκλήσεων και προσδοκιών της σύγχρονης τεχνοκρατούμενης εποχής (8), (9), (10). Οι ευφυείς πράκτορες έχουν γίνει τα τελευταία χρόνια ένα δημοφιλές εργαλείο στην ανάπτυξη λογισμικού, καθώς χρησιμοποιούνται σε όλο και περισσότερα είδη εφαρμογών. Ήδη σε τομείς, όπως η αναζήτηση πληροφοριών στο Διαδίκτυο, η υλοποίηση φιλικών διεπαφών (interfaces) και η διαχείριση αλληλογραφίας, η τεχνολογία ευφυών πρακτόρων έχει εφαρμοστεί με επιτυχία.

Η έννοια του πράκτορα δεν είναι νέα στο χώρο της Τεχνητής Νοημοσύνης, και παρ' ότι αρκετοί ερευνητές ασχολούνται με το συγκεκριμένο θέμα, η σύγχυση για το τι είναι ένας πράκτορας είναι αρκετά μεγάλη. Η σύγχυση αυτή φαίνεται και στους τίτλους των συνεδρίων, που διαπραγματεύονται θέματα του επιστημονικού πεδίου των πρακτόρων, αλλά και στη διαφορετική ορολογία, που χρησιμοποιούν αρκετοί επιστήμονες για να περιγράψουν όσο γίνεται καλύτερα αυτή την έννοια, όπως για παράδειγμα πράκτορες, πράκτορες λογισμικού, ευφυείς ή αυτόνομοι πράκτορες, ευφυείς διεπαφές, προσωπικοί βοηθοί, robot λογισμικού κ.α. (11), (12), (13)

Όλη αυτή η σύγχυση θεωρείται πως οφείλεται στη ραγδαία τεχνολογική ανάπτυξη του λογισμικού και του υλικού, η οποία είχε ως αποτέλεσμα τη χρήση πανίσχυρων υπολογιστικών συστημάτων, τα οποία είναι σε θέση να διαχειρίζονται ταυτόχρονα διάφορες εφαρμογές πολυμέσων. Το ρόλο αυτό έρχονται να επιτελέσουν οι πράκτορες.

Η ανάπτυξη εφαρμογών και λειτουργικών συστημάτων, τα οποία θα κάνουν χρήση πιο εύχρηστων και φιλικών διεπαφών, έφερε κοντά, σε κοινά ερευνητικά προγράμματα, ανθρώπους οι οποίοι είχαν διαφορετικό θεωρητικό υπόβαθρο, από πολυμέσα και τρισδιάστατα γραφικά και κίνηση, μέχρι τεχνικές μηχανικής μάθησης, γνωστική ψυχολογία, γλωσσική επεξεργασία και κατανόηση φυσικής γλώσσας. Όλοι αυτοί οι επιστήμονες είναι φυσικό, από τη δική τους πλευρά ο καθένας, να χρησιμοποιούν ή να εφευρίσκουν ορολογίες οι οποίες μπορεί να περιγράψουν καλύτερα το ερευνητικό τους αντικείμενο.

Παρά την ανωτέρω σύγχυση, που αναφέρθηκε, κανένας δεν αμφισβητεί πως τα υπολογιστικά συστήματα, τα οποία βασίζονται στην τεχνολογία των πρακτόρων, προσφέρουν μια νέα άποψη, μια νέα μεθοδολογία, με την οποία μπορούμε να προσεγγίσουμε δύσκολες εφαρμογές για τις οποίες οι υπάρχουσες τεχνικές της

επιστήμης των υπολογιστών δεν προσφέρουν κάποια λύση ή αν το κάνουν είναι αρκετά χρονοβόρα ή/και δαπανηρή. (14)

Πρόκειται για αναπτυσσόμενα εργαλεία, που χρησιμοποιούν οι άνθρωποι με σκοπό να πετύχουν διάφορους στόχους σε ποικίλα προβλήματα. Η κύρια διαφορά μεταξύ των συνηθισμένων εργαλείων και των πρακτόρων είναι πως οι δεύτεροι μπορούν να λειτουργήσουν περισσότερο ή λιγότερο αυτόνομα από τους δημιουργούς τους. Για μεγάλο διάστημα, οι άνθρωποι χρησιμοποιούσαν μόνο άλλους ανθρώπους και μερικές φορές ζώα ως πράκτορες. Η εξέλιξη στην τεχνολογία επεξεργασίας πληροφοριών και δεδομένων, την επιστήμη υπολογιστών και των δικτύων κατέστησε δυνατή την υλοποίηση τεχνητών πρακτόρων. Σήμερα, η πιο δημοφιλής προσέγγιση της Τεχνητής Νοημοσύνης βασίζεται στους πράκτορες.

Οι ευφυείς πράκτορες αποτελούν τη βάση για πολλά είδη προχωρημένων συστημάτων λογισμικού, τα οποία συμπεριλαμβάνουν διαφορετικές μεθοδολογίες, ποικίλες πηγές από διάφορους τομείς γνώσης και μια ποικιλία τύπων δεδομένων. Η προσέγγιση των ευφύων πρακτόρων έχει εφαρμοστεί εκτεταμένα στο χώρο των επιχειρήσεων και πιο πρόσφατα σε συστήματα υποστήριξης ιατρικών αποφάσεων.

## 2.2 ΤΙ ΕΙΝΑΙ ΕΝΑΣ ΠΡΑΚΤΟΡΑΣ;

Δυστυχώς, δεν υπάρχει καθολικά αποδεκτός ορισμός του όρου «πράκτορας», και μάλιστα υπάρχει μεγάλη και συνεχής συζήτηση και διαμάχη, σχετικά με αυτό ακριβώς το ζήτημα (1). Διάφορες επιστημονικές ομάδες προσπάθησαν κατά καιρούς να δώσουν έναν ορισμό για το τι είναι ένας πράκτορας.

Αρκετοί ορισμοί, καθώς και μια κατηγοριοποίηση των πρακτόρων και των χαρακτηριστικών τους παραθέτονται στο άρθρο των Franklin και Graesser (15). Οι ορισμοί αυτοί ποικίλουν από πολύ απλούς και γενικούς ή πιο πολύπλοκους και ειδικούς. Μερικοί από αυτούς είναι:

*«Πράκτορας είναι οτιδήποτε μπορεί να θεωρηθεί ότι αντιλαμβάνεται το περιβάλλον του μέσω αισθητήρων και επενεργεί σε αυτό το περιβάλλον μέσω μηχανισμών δράσης»,* Rusell και Norvig. (16)

*«Αυτόνομοι πράκτορες είναι υπολογιστικά συστήματα, που δρουν σε ένα πολύπλοκο δυναμικό περιβάλλον, αντιλαμβάνονται και ενεργούν αυτόνομα σε αυτό το περιβάλλον. Με τον τρόπο αυτό επιτυγχάνουν ένα σύνολο από στόχους και εκτελούν καθήκοντα, για τα οποία έχουν σχεδιαστεί»,* Maes. (17)

*«Οι ευφυείς πράκτορες συνεχώς εκτελούν τρεις λειτουργίες: αντίληψη των δυναμικών συνθηκών στο περιβάλλον, ενέργεια για να επιδράσουν στις συνθήκες του περιβάλλοντος και συλλογισμός για την ερμηνεία των αντιλήψεων, την επίλυση*

*προβλημάτων, την εξαγωγή συμπερασμάτων και τον καθορισμό ενεργειών», Hayes-Ruth. (18)*

*«Έξυπνοι πράκτορες είναι οντότητες λογισμικού, οι οποίες φέρουν εις πέρας ορισμένες ενέργειες για λογαριασμό του χρήστη ή ενός άλλου προγράμματος με κάποιο βαθμό ανεξαρτησίας ή αυτονομίας, και το κάνουν έτσι ώστε να εμπεριέχουν γνώση ή αναπαράσταση των στόχων ή των επιθυμιών του χρήστη», IBM.*

Όλοι οι παραπάνω ορισμοί, καταλήγουν στο γεγονός πως οι πράκτορες παρουσιάζουν αυτόνομη συμπεριφορά. Αντιδρούν σε εξωτερικά γεγονότα, που λαμβάνουν χώρα στο περιβάλλον τους και έχουν την ικανότητα να πραγματοποιούν ενέργειες και αλληλεπιδράσεις με άλλες οντότητες, είτε το περιβάλλον είτε άλλοι πράκτορες.

Συνεπώς, είναι λογικό να υποθέσουμε πως ένας πράκτορας είναι οτιδήποτε (ή οποιοσδήποτε) μπορεί να αντιλαμβάνεται το περιβάλλον του μέσω αισθητήρων (sensors) και δρα πάνω σε αυτό μέσω μηχανισμών δράσης (effectors). Για παράδειγμα, ένας ανθρώπινος πράκτορας διαθέτει μάτια, αυτιά και άλλα αισθητήρια όργανα ως αισθητήρες και χέρια, πόδια και άλλα μέλη του σώματος ως μηχανισμούς δράσης. Ένας ρομποτικός πράκτορας χρησιμοποιεί κάμερες, ανιχνευτές υπέρυθρων ακτινών και άλλους μηχανισμούς ως αισθητήρες και διάφορα εξαρτήματα ως μηχανισμούς δράσης. Ένας πράκτορας λογισμικού χρησιμοποιεί κανάλια επικοινωνίας τόσο ως αισθητήρες όσο και ως μηχανισμούς δράσης. (19)

Ο κάθε πράκτορας έχει στη διάθεσή του ένα ρεπερτόριο ενεργειών. Αυτό το σύνολο δυνατών ενεργειών αναπαριστά τη δραστική ικανότητα (effectoric capability) του πράκτορα, δηλαδή την ικανότητά του να τροποποιεί το περιβάλλον του. Πρέπει να σημειωθεί πως δεν μπορούν να εκτελεστούν όλες οι ενέργειες σε όλες τις περιστάσεις. Οι ενέργειες συσχετίζονται με προϋποθέσεις (preconditions), οι οποίες ορίζουν τις πιθανές καταστάσεις στις οποίες μπορούν να εφαρμοστούν.

Το βασικό πρόβλημα που αντιμετωπίζει ένας πράκτορας είναι να αποφασίσει ποιες από τις ενέργειές του θα πρέπει να εκτελέσει προκειμένου να ικανοποιήσει τους σχεδιαστικούς του στόχους με τον καλύτερο τρόπο (1).

### 2.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΠΡΑΚΤΟΡΩΝ

Όπως παρατηρήθηκε προηγουμένως, η απόδοση πολλών διαφορετικών ορισμών στην έννοια «πράκτορας», καθιστά δύσκολη την υιοθέτηση ενός καθολικά αποδεκτού. Παρά το γεγονός αυτό, μεγάλη μερίδα ερευνητών, στην προσπάθεια τους να περιγράψουν από τη δική τους οπτική και περιοχή απασχόλησης, αυτή τη

διφορούμενη έννοια, συγκλίνουν πως οι πράκτορες διαθέτουν κάποια κοινά χαρακτηριστικά.

Συνοπτικά, όπως προκύπτει από τους παραπάνω ορισμούς, οι πράκτορες συγκεντρώνουν κάποια γνωρίσματα που τους κάνουν να ξεχωρίζουν από τα απλά “προγράμματα”, όπως το να λειτουργούν κάτω από αυτόνομο έλεγχο, να αντιλαμβάνονται το περιβάλλον τους, να διατηρούνται για ένα παρατεταμένο χρονικό διάστημα, να προσαρμόζονται στις αλλαγές, και να έχουν την ικανότητα να αναλαμβάνουν τους στόχους κάποιου άλλου (2).

Πιο συγκεκριμένα, έχει προταθεί από τους Wooldridge και Jennings (8) μια λίστα στην οποία απαριθμούνται τα είδη των ικανοτήτων, που περιμένουμε να διαθέτει ένας ευφυής πράκτορας. Σύμφωνα με τη λίστα αυτή οι κύριες ιδιότητες ενός πράκτορα είναι οι εξής:

- *Αυτονομία (autonomy)*: οι πράκτορες λειτουργούν χωρίς την άμεση παρέμβαση των χρηστών ή άλλων πρακτόρων και έχουν κάποιον αυτοέλεγχο στις ενέργειές τους και την εσωτερική τους κατάσταση. Αυτό σημαίνει ότι οι πράκτορες έχουν τη δυνατότητα να επιδιώκουν τους στόχους τους χωρίς να δέχονται συνεχώς εντολές από το χρήστη ή κάποια άλλη εξωτερική πηγή, το οποίο έχει σαν αποτέλεσμα να αφήνει στο χρήστη μόνο τον προσδιορισμό του γενικού στόχου και να τον απελευθερώνει από το βάρος της λήψης των επιμέρους αποφάσεων.
- *Κοινωνική ικανότητα (social ability)*: οι πράκτορες μπορούν να επικοινωνούν και να αλληλοεπιδρούν με άλλους πράκτορες (και πιθανώς με ανθρώπους) προκειμένου να ικανοποιήσουν τους σχεδιαστικούς τους στόχους. Η επικοινωνία αυτή γίνεται μέσω μιας κοινώς κατανοητή γλώσσα, έτσι ώστε να μπορέσουν να συνεργαστούν για την επίτευξη ανεξάρτητων στόχων ή ενός κοινού.
- *Αντιδραστικότητα (reactivity)*: οι πράκτορες μπορούν να αντιλαμβάνονται το περιβάλλον τους και να αντιδρούν μέσα σε συγκεκριμένα χρονικά διαστήματα (έγκαιρα) στις αλλαγές, που συμβαίνουν σε αυτό, προκειμένου να ικανοποιήσουν τους σχεδιαστικούς τους στόχους.
- *Προνοητικότητα (pro-activeness)*: οι ευφυείς πράκτορες δεν αντιδρούν απλά στα ερεθίσματα του περιβάλλοντος, αλλά είναι ικανοί να επιδείξουν και συμπεριφορά που βασίζεται σε στόχους λαμβάνοντας ουσιαστικά πρωτοβουλία ανάλογα με τις συνθήκες, οι οποίες εμφανίζονται στο περιβάλλον τους. Τόσο η προνοητικότητα όσο και η αντιδραστικότητα

απαιτούν σε κάποιο βαθμό τη δυνατότητα συλλογισμού (νοημοσύνης) από τον πράκτορα.

Τα παραπάνω χαρακτηριστικά, προβάλλουν μια ασθενή έννοια του πράκτορα, αφού κάνουν αρκετά δύσκολη τη διάκριση για το τι είναι ένας πράκτορας και που διαφέρει πραγματικά από ένα κοινό πρόγραμμα. Με σκοπό να αποδοθεί μια πιο ισχυρή έννοια του πράκτορα, αρκετοί επιστήμονες, ιδιαίτερα από το χώρο της Τεχνητής Νοημοσύνης, αποπειράθηκαν να περιγράψουν τα στοιχεία που διαθέτει ένας πράκτορας με όρους που συναντάμε σε βιολογικούς οργανισμούς (πεποιθήσεις, επιθυμίες, προθέσεις) και να προσθέσουν και άλλα, πιο δευτερεύοντα, χαρακτηριστικά σε αυτά που ήδη αναφέρθηκαν νωρίτερα. Τα δευτερεύοντα αυτά χαρακτηριστικά είναι (15) (9):

- *Κινητικότητα (mobility)*: Οι πράκτορες δεν είναι πάντα στατικοί, αλλά μπορούν να κινηθούν σε ένα υπολογιστικό περιβάλλον. Μπορούν να μετακινούνται από μηχανή σε μηχανή μέσα σε ετερογενή δίκτυα και να αλληλοεπιδρούν με άλλους στατικούς πράκτορες ή υπηρεσίες του υπολογιστή με σκοπό να φέρουν εις πέρας το στόχο τους.
- *Προσαρμοστικότητα (adaptivity)*: Οι πράκτορες προσαρμόζονται διαρκώς στο περιβάλλον τους ή τις απαιτήσεις του χρήστη. Έχουν δηλαδή, ικανότητα εκμάθησης.
- *Ελικρίνεια (veracity)*: Οι πράκτορες δεν δίνουν επίτηδες ψευδείς πληροφορίες.
- *Καλοσύνη (benevolence)*: Οι πράκτορες δεν έχουν αντικρουόμενους στόχους, και ο καθένας από αυτούς, πάντα προσπαθεί να εκτελέσει αυτό, το οποίο θα του ζητηθεί. Δηλαδή, να επιτύχει τους στόχους, οι οποίοι του έχουν ανατεθεί.
- *Ορθολογισμός (rationality)*: Οι πράκτορες δρουν για να πετύχουν τους στόχους τους, δηλαδή δεν κάνουν αναίτιες ενέργειες και δεν λειτουργούν εναντίον της επίτευξης των στόχων τους.

Φυσικά, δεν έχουν όλοι οι πράκτορες τα παραπάνω χαρακτηριστικά, αλλά κατά καιρούς έχουν αναπτυχθεί διάφορα πρότυπα, τα οποία διαθέτουν ένα μεγάλο αριθμό αυτών σε συνδυασμό.



## 2.4 ΠΕΡΙΒΑΛΛΟΝΤΑ ΕΡΓΑΣΙΩΝ ΤΩΝ ΠΡΑΚΤΟΡΩΝ

Όπως αναφέρθηκε και παραπάνω, ένας πράκτορας είναι μία οντότητα η οποία αντιλαμβάνεται και δρα μέσα σε ένα περιβάλλον. Οι ορισμοί, που δόθηκαν σε προηγούμενη παράγραφο, περιγράφουν τα συστήματα πρακτόρων παραθέτοντας τα χαρακτηριστικά τους, χωρίς να αναφέρονται στο πώς μπορούν να υλοποιηθούν αυτά, δηλαδή πώς να μπορεί να υλοποιηθεί ένας πράκτορας. Το κεντρικό ερώτημα στο οποίο καλείται να απαντήσει κάθε υλοποίηση είναι το πώς ο πράκτορας καθορίζει την επόμενη ενέργεια, που πρέπει να εκτελέσει, δεδομένων της κατάστασης του περιβάλλοντος και του στόχου που καλείται να εκπληρώσει.

Έχουμε δει, πως οι ευφυείς πράκτορες, στη διαδικασία επιλογής μιας ορθολογικής ενέργειας πρέπει να λαμβάνουν υπόψιν τους συγκεκριμένες πληροφορίες. Σε αυτές συμπεριλαμβάνονται πληροφορίες, που συλλέγουν από τους αισθητήρες τους, από τον κόσμο, από προηγούμενες καταστάσεις του κόσμου, από τους στόχους του ίδιου του πράκτορα, καθώς και από τις συναρτήσεις χρησιμότητάς τους που δηλώνουν πόσο «καλή» είναι μία κατάσταση για τον πράκτορα. Χρειάζεται ακόμα να ληφθούν υπόψιν μερικές ιδιαιτερότητες του περιβάλλοντος μέσα στο οποίο λειτουργεί ο πράκτορας.

Μπορούμε να θεωρήσουμε το περιβάλλον ως οτιδήποτε βρίσκεται γύρω από τον πράκτορα, αλλά είναι ευδιάκριτο από αυτόν και τη συμπεριφορά του. Με άλλα λόγια, περιβάλλον καλείται οτιδήποτε περικλείει τον πράκτορα και δεν αποτελεί μέρος του ίδιου του πράκτορα. Είναι το τμήμα του κόσμου, μέσα στο οποίο ο πράκτορας «ζει» ή λειτουργεί, και παρέχει σε αυτόν ερεθίσματα για να αισθάνεται και ένα χώρο για να κινείται μέσα σε αυτό. (7)

Η ποικιλία των περιβαλλόντων εργασίας, που μπορούν να προκύψουν στην Τεχνητή Νοημοσύνη είναι προφανώς τεράστια. Μπορούμε όμως να προσδιορίσουμε ένα σχετικά μικρό αριθμό διαστάσεων με βάση τις οποίες μπορούν τα περιβάλλοντα εργασιών να καταταχθούν σε κατηγορίες. Οι διαστάσεις αυτές προσδιορίζουν σε μεγάλο βαθμό τη σχεδίαση κατάλληλων πρακτόρων και την καταλληλότητα της κάθε κύριας οικογένειας τεχνικών για την υλοποίηση πρακτόρων. Σύμφωνα με τους Russell και Norvig (2), (7), τα περιβάλλοντα εργασιών κατηγοριοποιούνται ως εξής :

- *Πλήρως παρατηρήσιμο ή μερικώς παρατηρήσιμο:* Ένας πράκτορας, μπορεί να θεωρηθεί ως πράκτορας μόνο αν έχει την ικανότητα να παρατηρεί το περιβάλλον του. Αν οι αισθητήρες ενός πράκτορα του παρέχουν πρόσβαση στην πλήρη κατάσταση του περιβάλλοντος σε κάθε χρονική στιγμή, τότε λέμε ότι το περιβάλλον εργασιών είναι πλήρως παρατηρήσιμο (fully observable). Τα πλήρως παρατηρήσιμα περιβάλλοντα είναι βολικά, επειδή ο πράκτορας δε χρειάζεται να διατηρεί κάποια εσωτερική κατάσταση για να

παρακολουθεί τον κόσμο. Ένα περιβάλλον μπορεί να είναι μερικώς παρατηρήσιμο (partially observable) λόγω του θορύβου και της ανακρίβειας των αισθητήρων ή απλώς επειδή κάποια μέρη της κατάστασης λείπουν από τα δεδομένα των αισθητήρων. Συνήθως, τα περιβάλλοντα εργασιών των πρακτόρων είναι μερικώς παρατηρήσιμα, όπως για παράδειγμα το φυσικό περιβάλλον ή το Διαδίκτυο. Όσο πιο παρατηρήσιμο είναι ένα περιβάλλον, τόσο πιο εύκολο είναι να αναπτυχθούν πράκτορες, που θα λειτουργήσουν σε αυτό.

- *Αιτιοκρατικό ή στοχαστικό*: Αν η επόμενη κατάσταση του περιβάλλοντος προσδιορίζεται πλήρως από την τρέχουσα κατάσταση και από την ενέργεια που εκτελείται από τον πράκτορα, τότε λέμε ότι το περιβάλλον είναι αιτιοκρατικό (deterministic). Δηλαδή, κάθε κίνηση δράσης του πράκτορα έχει μία και μοναδική επίδραση στο περιβάλλον, και δεν υπάρχει καμία ασάφεια για την κατάσταση του περιβάλλοντος μετά την εφαρμογή αυτής της δράσης. Σε αντίθετη περίπτωση, το περιβάλλον χαρακτηρίζεται ως στοχαστικό (stochastic) ή μη-αιτιοκρατικό.

Θεωρητικά, σε ένα πλήρως παρατηρήσιμο αιτιοκρατικό περιβάλλον, ο πράκτορας δεν έχει να αντιμετωπίσει καμία αβεβαιότητα. Αν όμως το περιβάλλον είναι μερικώς παρατηρήσιμο, τότε μπορεί να εμφανίζεται ως στοχαστικό από την άποψη του πράκτορα. Αυτό ισχύει ιδιαίτερα αν το περιβάλλον είναι πολύπλοκο, γεγονός που δυσκολεύει την παρακολούθηση όλων των μη παρατηρούμενων απόψεων.

- *Επεισοδιακό ή ακολουθιακό*: Ένα περιβάλλον εργασίας χαρακτηρίζεται ως επεισοδιακό (episodic) αν οι ενέργειες του πράκτορα δεν στηρίζονται στην δράση του παρελθόντος και δεν μπορούν να επηρεάσουν τη μελλοντική του λειτουργία. Η εμπειρία του πράκτορα αποτελείται από ατομικά επεισόδια. Σε κάθε επεισόδιο, ο πράκτορας αντιλαμβάνεται και έπειτα πραγματοποιεί μια μεμονωμένη ενέργεια. Το επόμενο επεισόδιο δεν εξαρτάται καθοριστικά από τις ενέργειες που έγιναν στα προηγούμενα επεισόδια. Στα επεισοδιακά περιβάλλοντα, η επιλογή της ενέργειας σε κάθε επεισόδιο εξαρτάται μόνο από το ίδιο το επεισόδιο. Αντίθετα, ένα περιβάλλον καλείται ακολουθιακό (sequential), όταν η τρέχουσα απόφαση του πράκτορα μπορεί να επηρεάζει όλες τις μελλοντικές αποφάσεις.
- *Στατικό ή δυναμικό*: Ένα περιβάλλον χαρακτηρίζεται ως δυναμικό (dynamic), όταν μπορεί να αλλάζει, ενώ σκέφτεται ο πράκτορας. Σε αντίθετη περίπτωση το περιβάλλον καλείται στατικό (static). Τα στατικά περιβάλλοντα αντιμετωπίζονται ευκολότερα επειδή ο πράκτορας δεν χρειάζεται να συνεχίσει να κοιτάζει τον κόσμο ενώ αποφασίζει για μια ενέργεια, ούτε

χρειάζεται αν τον απασχολεί το πέρασμα του χρόνου. Τα δυναμικά περιβάλλοντα από την άλλη, ζητούν συνεχώς από τον πράκτορα να αποφασίσει τι θα κάνει. Αν το ίδιο περιβάλλον δεν αλλάζει με το πέρασμα του χρόνου, αλλά αλλάζει η βαθμολογία της απόδοσης του πράκτορα, τότε το περιβάλλον χαρακτηρίζεται ως *ημιδυναμικό* (semidynamic).

- *Διακριτό ή συνεχές*: Διακριτό (discrete) περιβάλλον είναι εκείνο, που εγγυημένα μπορεί να βρισκεται μέσα σε ένα πεπερασμένο αριθμό διακριτών καταστάσεων. Ένα συνεχές (continuous) περιβάλλον μπορεί να βρισκεται σε αναρίθμητα πολλές καταστάσεις.

Είναι ευκολότερο να σχεδιάσουμε πράκτορες για διακριτά περιβάλλοντα παρά για συνεχή, για πολλούς λόγους. Ο πλέον προφανής είναι ότι οι ίδιοι οι ψηφιακοί υπολογιστές είναι συστήματα διακριτών καταστάσεων και, αν και μπορούν να προσομοιώσουν συνεχή συστήματα με οποιοδήποτε επιθυμητό βαθμό ακρίβειας, υπάρχει αναπόφευκτα μια αντιστοιχία μεταξύ των δύο τύπων συστημάτων. Κάποιες πληροφορίες χάνονται κατά την αναπαράσταση ενός συνεχούς περιβάλλοντος σε διακριτή μορφή. Έτσι, οι πληροφορίες που χρησιμοποιεί ένας πράκτορας διακριτών καταστάσεων προκειμένου να επιλέξει μια ενέργεια σε ένα συνεχές περιβάλλον θα είναι από τη φύση τους προσεγγιστικές. Τέλος, για τα περιβάλλοντα πεπερασμένων διακριτών καταστάσεων, είναι θεωρητικά δυνατό να απαριθμήσουμε όλες τις ενδεχόμενες καταστάσεις του περιβάλλοντος και τη βέλτιστη ενέργεια προς εκτέλεση για κάθε μία από αυτές. Αυτή η προσέγγιση ως προς το σχεδιασμό πρακτόρων με πίνακα αναζήτησης είναι σπανίως εφικτή στην πράξη, αλλά τουλάχιστον είναι θεωρητικά εφικτή για περιβάλλοντα πεπερασμένων διακριτών καταστάσεων. (1)

- *Μονοπρακτορικό ή πολυπρακτορικό*: Ο πράκτορας μέσα σε ένα μονοπρακτορικό (single-agent) σύστημα μοντελοποιεί τον ίδιο, το περιβάλλον του και τις μεταξύ τους αλληλεπιδράσεις. Είναι μια αυτόνομη οντότητα με τους δικούς του στόχους, ενέργειες και γνώση. Σε ένα μονοπρακτορικό σύστημα καμία άλλη οντότητα δεν αναγνωρίζεται από τον πράκτορα. Ακόμα και αν υπάρχουν άλλοι πράκτορες στον κόσμο, δεν παρουσιάζονται να έχουν στόχους, απλά θωρούνται ως κομμάτι του περιβάλλοντος εργασίας του πράκτορα υπό μελέτη.

Ένα πολυπρακτορικό (multiagent) σύστημα διαφέρει από το μονοπρακτορικό στο γεγονός, πως υπάρχουν περισσότεροι πράκτορες στον κόσμο, καθένας από τους οποίους αναγνωρίζει τους στόχους και τις ενέργειες των υπολοίπων. Μπορεί να υπάρχει άμεση αλληλεπίδραση μεταξύ των πρακτόρων του συστήματος μέσω της επικοινωνίας (communication).

Από την αντίληψη του κάθε πράκτορα ξεχωριστά, τα πολυπρακτορικά συστήματα διαφέρουν από τα μονοπρακτορικά κυρίως στο γεγονός, πως η δυναμική του περιβάλλοντος μπορεί να καθοριστεί από άλλους πράκτορες.

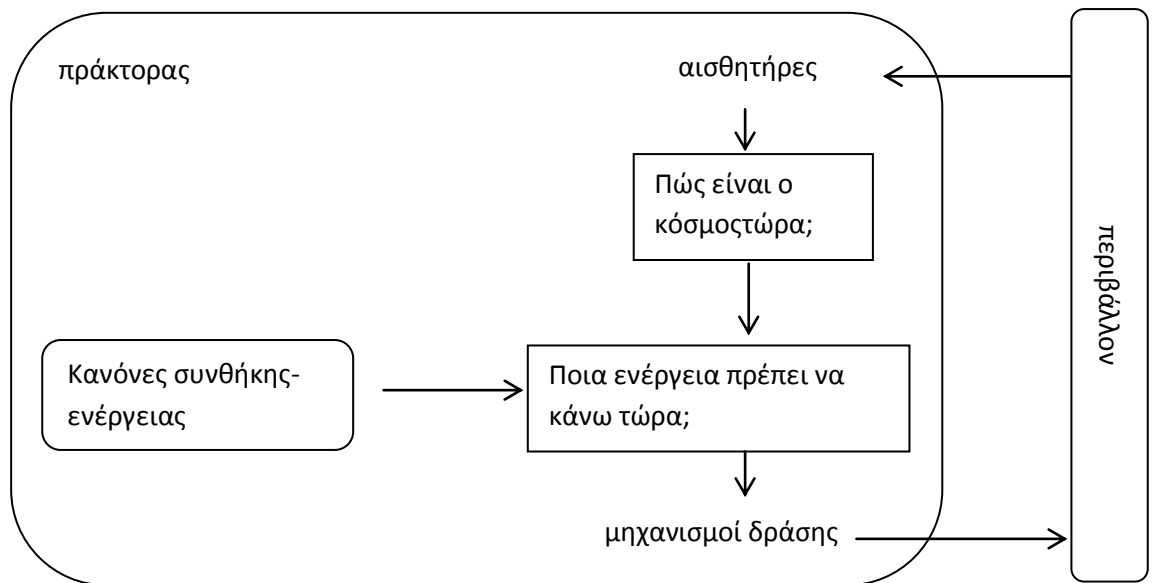
Από την παραπάνω κατηγοριοποίηση των περιβαλλόντων εργασίας, θα περίμενε κανείς, πως η δυσκολότερη περίπτωση είναι τα μερικώς παρατηρήσιμα, στοχαστικά, ακολουθιακά, δυναμικά, συνεχή και πολυπρακτορικά περιβάλλοντα. Τα περιβάλλοντα, που συγκεντρώνουν αυτές τις ιδιότητες συχνά αναφέρονται ως *ανοιχτά περιβάλλοντα*.

Η πολυπλοκότητα του σχεδιασμού πρακτόρων καθορίζεται σε ένα βαθμό από τις ιδιότητες του περιβάλλοντος, αλλά σε καμία περίπτωση δεν καθορίζεται μόνο από αυτές, Σημαντικό ρόλο παίζει και η φύση της αλληλεπίδρασης ανάμεσα στον πράκτορα και το περιβάλλον του.

## 2.5 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΤΩΝ ΕΥΦΥΩΝ ΠΡΑΚΤΟΡΩΝ

Παρόλο, που οι ευφυείς πράκτορες είναι κατά κάποιον τρόπο σχετικά νέες εφαρμογές στο χώρο των υπολογιστικών περιβαλλόντων, εμπορικών ή μη, έχουν αποτελέσει αντικείμενο έρευνας από χρόνια. Κατά τη διάρκεια των χρόνων αυτών έχουν προταθεί αρκετές διαφορετικές κατηγοριοποιήσεις. Μία από αυτές έχει προταθεί από τους Russell και Norvig. Σύμφωνα με την πρόταση αυτή, υπάρχουν τέσσερα βασικά είδη προγραμμάτων πρακτόρων, που ενσωματώνουν τις αρχές στις οποίες βασίζονται σχεδόν όλα τα ευφυή συστήματα:

- *Απλοί αντανakλαστικοί πράκτορες (simple reflex agents)*: Πρόκειται για το απλούστερο είδος πράκτορα. Οι πράκτορες αυτοί επιλέγουν ενέργειες με βάση την τρέχουσα αντίληψη, αγνοώντας το υπόλοιπο ιστορικό αντιλήψεων.



**Εικόνα 1** :Σχηματικό διάγραμμα απλού αντανακλαστικού πράκτορα (2)

απόκριση. Οι άνθρωποι διαθέτουν και αυτοί παρόμοιες αντιδράσεις, μερικές από τις οποίες είναι μαθημένες αποκρίσεις (όπως στην οδήγηση) και μερικές είναι έμφυτα αντανακλαστικά (όπως το ανοιγόκλεισμα των ματιών όταν κάτι πλησιάζει).

Ένας απλός αντανακλαστικός πράκτορας διαθέτει μία βιβλιοθήκη από κανόνες, που ονομάζονται *κανόνες συνθήκης-ενέργειας* (condition-action rules) και είναι της μορφής:

**If συνθήκη then ενέργεια**

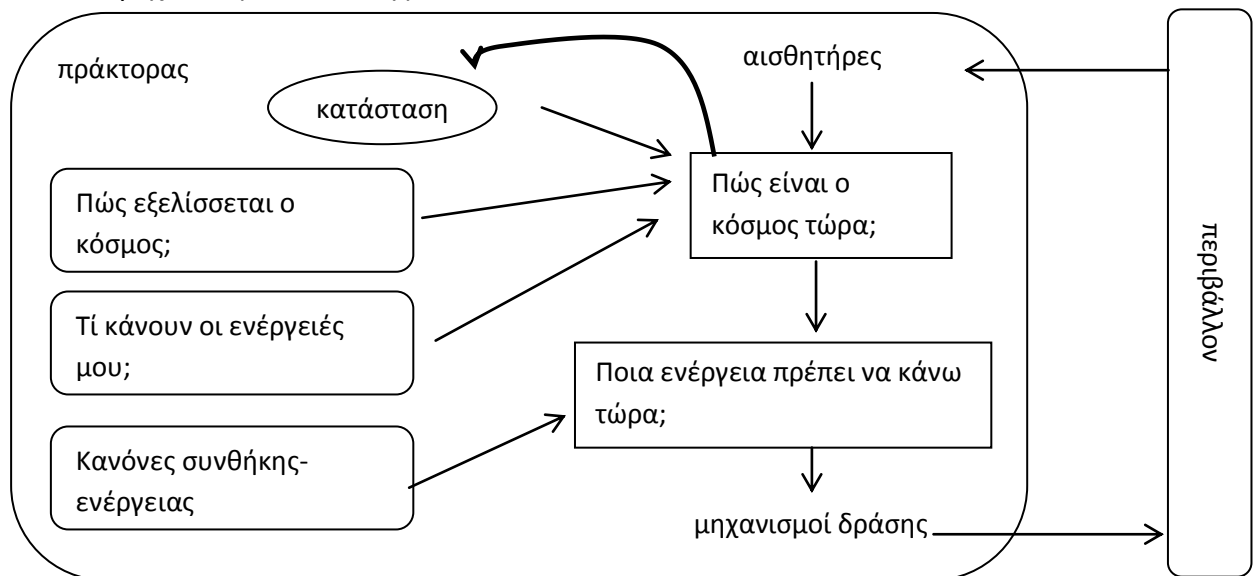
Έτσι, όταν μια κατάσταση του περιβάλλοντος ενεργοποιεί έναν τέτοιο κανόνα, ο πράκτορας γνωρίζει πώς να αντιδράσει.

Οι απλοί αντανακλαστικοί πράκτορες έχουν τη θαυμαστή ιδιότητα να είναι απλοί, αλλά η ευφυΐα τους αποδεικνύεται πολύ περιορισμένη. Λειτουργούν σωστά όταν το περιβάλλον είναι πλήρως παρατηρήσιμο. Ακόμα και ελάχιστη μη παρατηρησιμότητα μπορεί να προκαλέσει μεγάλο πρόβλημα. Στα μη παρατηρήσιμα περιβάλλοντα οι ατέρμονες βρόχοι, δηλαδή η άσκοπη επανάληψη της ίδιας ενέργειας, είναι αναπόφευκτοι. Η έξοδος από μια τέτοια κατάσταση είναι δυνατή αν ο πράκτορας μπορεί να τυχαιοποιεί (randomize) τις ενέργειές του.

- *Αντανακλαστικοί πράκτορες βασισμένοι σε μοντέλο (model-based reflex agents)*: Ένας τέτοιος πράκτορας μπορεί να χειριστεί τη μερική παρατηρησιμότητα, παρακολουθώντας το τμήμα του κόσμου, που δεν μπορεί να δει τώρα. Ο πράκτορας, πρέπει να τηρεί κάποιο είδος εσωτερικής κατάστασης, η οποία εξαρτάται από το ιστορικό των αντιλήψεων και μέσω αυτών αντανακλά τουλάχιστον μερικές από τις μη παρατηρήσιμες απόψεις της τρέχουσας κατάστασης.

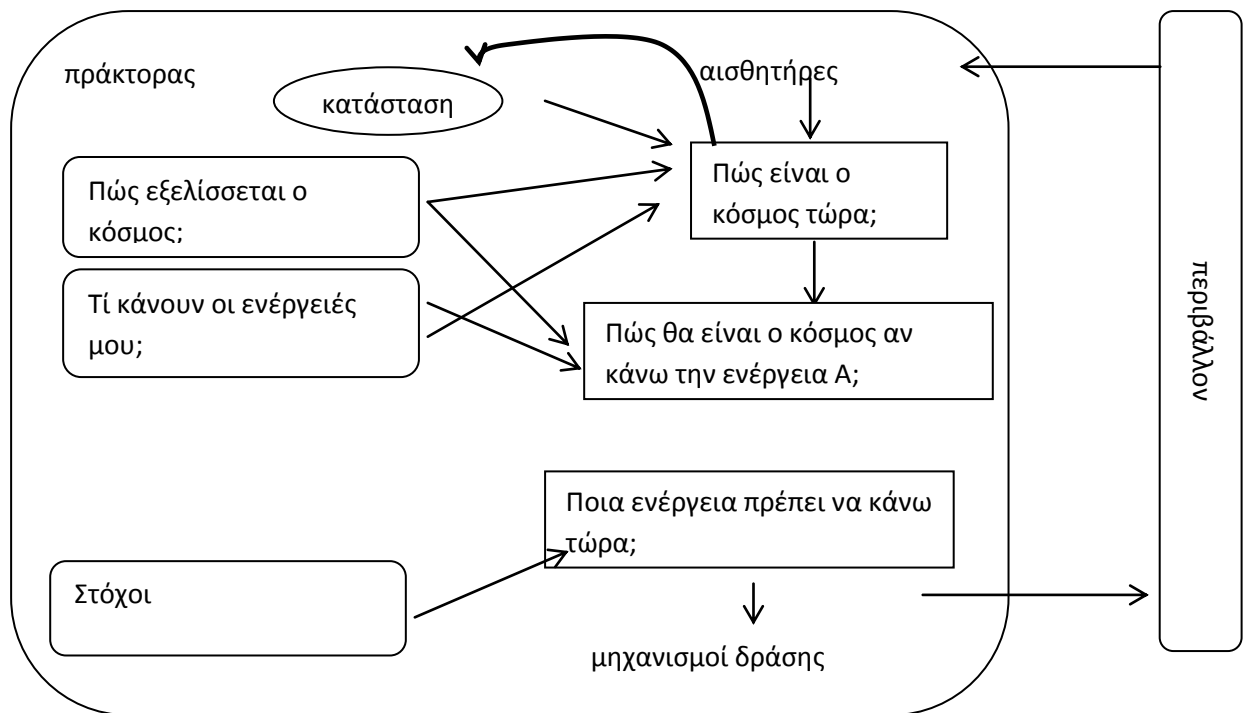
Η ενημέρωση των πληροφοριών αυτής της εσωτερικής κατάστασης στο πέρασμα του χρόνου απαιτεί δύο ειδών κωδικοποιημένη γνώση στο πράκτορα λογισμικού. Πρώτος, χρειαζόμαστε πληροφορίες για το πώς εξελίσσεται ο κόσμος ανεξάρτητα από τον πράκτορα. Δεύτερον, χρειαζόμαστε πληροφορίες για το πώς οι ενέργειες του ίδιου του πράκτορα επηρεάζουν τον κόσμο. Αυτή η γνώση για το “πώς λειτουργεί ο κόσμος”, ονομάζεται μοντέλο του κόσμου.

Στο παρακάτω διάγραμμα παρουσιάζεται η δομή του αντανακλαστικού πράκτορα με εσωτερική κατάσταση και φαίνεται ο τρόπος με τον οποίο η τρέχουσα αντίληψη συνδυάζεται με την προηγούμενη εσωτερική κατάσταση για να παραχθεί η ενημερωμένη περιγραφή της τρέχουσας κατάστασης.



**Εικόνα 2:** Αντανακλαστικός πράκτορας βασισμένος σε μοντέλο (2)

- *Πράκτορες βασισμένοι στο στόχο (goal-based agents):* Οι πράκτορες αυτοί επεκτείνουν τις δυνατότητες των πρακτόρων που βασίζονται σε μοντέλο. Η γνώση της τρέχουσας κατάστασης του περιβάλλοντος δεν αρκεί πάντα για να αποφασιστεί ποια ενέργεια πρέπει να γίνει. Η σωστή απόφαση απαιτεί και πληροφορίες για το στόχο του πράκτορα, οι οποίες περιγράφουν καταστάσεις οι οποίες είναι επιθυμητές. Ο πράκτορας μπορεί να συνδυάζει αυτές τις πληροφορίες με εκείνες, που αφορούν τα αποτελέσματα των δυνατών ενεργειών (τις ίδιες πληροφορίες, που χρησιμοποιήθηκαν για να ενημερωθεί η εσωτερική κατάσταση στον αντανακλαστικό πράκτορα) προκειμένου να επιλέγει ενέργειες που επιτυγχάνουν τους στόχους.



Εικόνα 3: Πράκτορας βασισμένος σε μοντέλο και στους στόχους. (2)

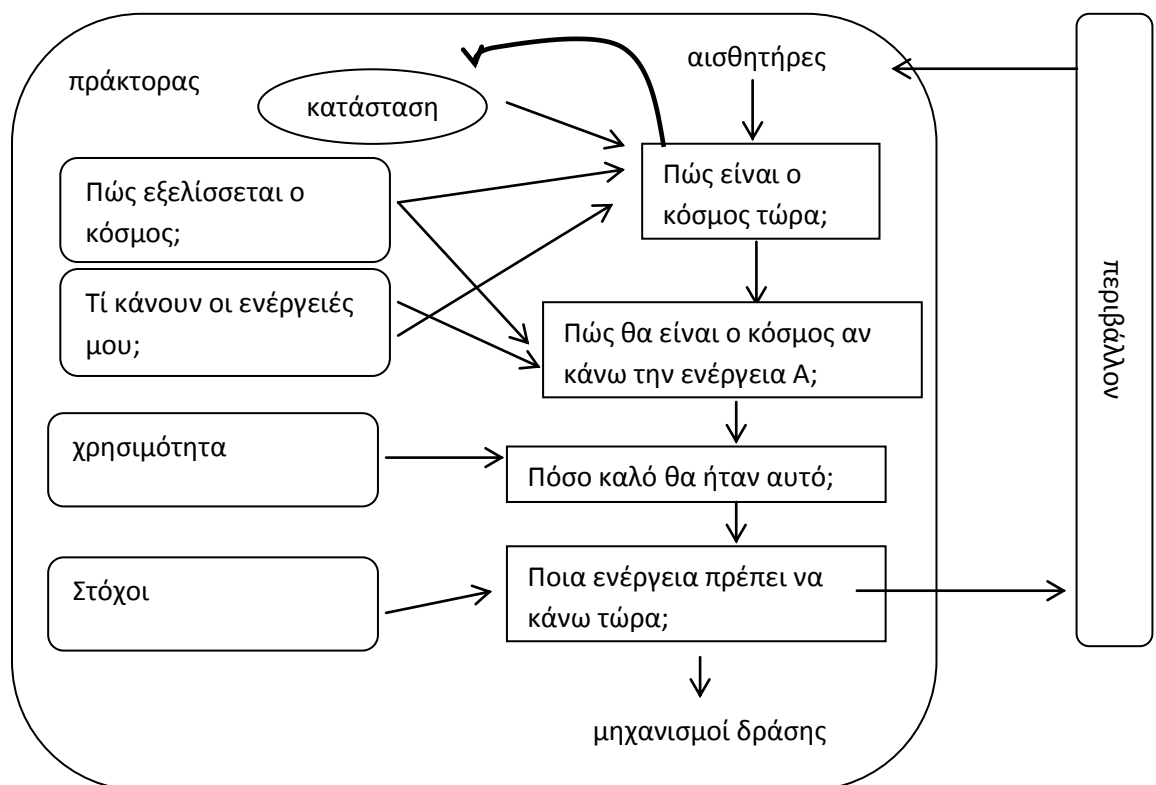
Μερικές φορές η επιλογή ενεργειών με βάση τους στόχους είναι απλή, όταν η επίτευξη του στόχου προκύπτει άμεσα από μία μόνο ενέργεια. Μερικές φορές είναι πιο δύσκολη, όταν ο πράκτορας έχει να εξετάσει μεγάλες ακολουθίες ενεργειών για να βρει έναν τρόπο να επιτύχει το στόχο του.

Αν και ο πράκτορας, που βασίζεται στους στόχους φαίνεται να είναι λιγότερο αποδοτικός, είναι πιο ευέλικτος επειδή η γνώση που υποστηρίζει τις αποφάσεις του αναπαρίσταται ρητά και μπορεί να τροποποιηθεί. Η συμπεριφορά ενός τέτοιου πράκτορα μπορεί να αλλάξει εύκολα για να πάει σε έναν άλλο προορισμό.

- *Πράκτορες βασισμένοι στη χρησιμότητα (utility-based agents):* Οι στόχοι από μόνοι τους δεν αρκούν για να παραχθεί μια συμπεριφορά υψηλής ποιότητας στα περισσότερα περιβάλλοντα. Υπάρχουν πολλές ακολουθίες ενεργειών, που μπορούν να επιτύχουν το στόχο για τον οποίο έχει προγραμματιστεί ο πράκτορας, όμως μερικές είναι γρηγορότερες, ασφαλέστερες, πιο αξιόπιστες ή φθηνότερες από τις άλλες. Οι στόχοι παρέχουν μόνο μία δυαδική διάκριση μεταξύ των καταστάσεων στις οποίες ένας πράκτορας είναι «ευχαριστημένος» ή «δυσανεστημένος». Η ανάγκη για την ανάπτυξη ενός είδους πράκτορα, που θα χρησιμοποιεί ένα πιο γενικό μέτρο της απόδοσης επιτρέποντας τη σύγκριση διαφορετικών καταστάσεων του κόσμου

σύμφωνα με το πόσο ευχαριστημένο θα έκαναν τον πράκτορα αν μπορούσαν να επιτευχθούν, οδήγησε στην ανάπτυξη των πρακτόρων βασισμένων στη *χρησιμότητα* (utility). Αν μία κατάσταση του κόσμου είναι προτιμότερη από μία άλλη, τότε έχει μεγαλύτερη χρησιμότητα για τον πράκτορα.

Μια *συνάρτηση χρησιμότητας* (utility function) αντιστοιχίζει μία κατάσταση (ή μια ακολουθία καταστάσεων) σε έναν πραγματικό αριθμό, ο οποίος περιγράφει τον αντίστοιχο βαθμό ικανοποίησης. Ο πλήρης προσδιορισμός της συνάρτησης χρησιμότητας επιτρέπει να λαμβάνονται ορθολογικές αποφάσεις σε δύο ειδών περιπτώσεις, που οι στόχοι είναι ανεπαρκείς. Πρώτον, όταν υπάρχουν αλληλοσυγκρουόμενοι στόχοι, από τους οποίους μόνο μερικοί μπορούν να επιτευχθούν, η συνάρτηση χρησιμότητας καθορίζει την κατάλληλη εξισορρόπηση. Δεύτερον, όταν υπάρχουν πολλοί στόχοι, τους οποίους μπορεί να επιδιώξει ο πράκτορας, κανένας από τους οποίους δεν μπορεί να επιτευχθεί με βεβαιότητα, η χρησιμότητα παρέχει έναν τρόπο με τον οποίο η πιθανότητα επιτυχίας μπορεί να σταθμιστεί με τη σπουδαιότητα των στόχων.



**Εικόνα 4:** Πράκτορας βασισμένος στη χρησιμότητα (2)



## ΚΕΦΑΛΑΙΟ 3: ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

### 3.1 ΕΙΣΑΓΩΓΗ

Σε προηγούμενη ενότητα αναφέραμε συντόμως τι είναι ένα πολυπρακτορικό περιβάλλον και σε ποια σημεία διαφέρει από ένα περιβάλλον, μέσα στο οποίο δρα μόνο ένας πράκτορας. Αξίζει, στο σημείο αυτό να δοθεί ένας πιο σαφής ορισμός του πολυπρακτορικού συστήματος, αφού άλλωστε αποτελεί το χώρο μελέτης και εφαρμογής μεθόδων στα πλαίσια αυτής της διπλωματικής εργασίας.

Στην επιστήμη της Τεχνητής Νοημοσύνης, τα συστήματα που βασίζονται στην τεχνολογία των πρακτόρων έχουν γίνει αποδεκτά ως ένα νέο τρόπο σύλληψης, σχεδιασμού και εφαρμογής συστημάτων λογισμικού. Οι πράκτορες είναι εκλεπτυσμένα υπολογιστικά προγράμματα, τα οποία ενεργούν αυτόνομα για λογαριασμό των χρηστών τους, μέσα σε ανοιχτά και κατανεμημένα περιβάλλοντα εργασιών, με σκοπό να επιλύσουν έναν αυξανόμενο αριθμό πολύπλοκων προβλημάτων. (20)

Τα πραγματικά προβλήματα συμπεριλαμβάνουν κατανεμημένα, ανοιχτά συστήματα. Ανοιχτό σύστημα χαρακτηρίζεται ένα, του οποίου η δομή έχει τη δυνατότητα να αλλάζει δυναμικά. Τα χαρακτηριστικά ενός τέτοιου συστήματος είναι:

- Τα συστατικά του στοιχεία δεν είναι γνωστά από την αρχή
- Μπορούν να αλλάξουν με την πάροδο του χρόνου
- Μπορούν να αποτελούνται από ετερογενείς πράκτορες, οι οποίοι σχεδιάστηκαν από διαφορετικούς ανθρώπους, διαφορετικές στιγμές, με διαφορετικά εργαλεία λογισμικού και τεχνικές.

Ίσως, το πιο αντιπροσωπευτικό και γνωστό παράδειγμα ενός τέτοιου συστήματος είναι το Διαδίκτυο. (21)

Η επέκταση των ορίων του πραγματικού κόσμου, έτσι ώστε να συμπεριλάβουμε αυτόνομα υπολογιστικά συστήματα, υπήρξε πάντα μία εντυπωσιακή, αν όχι, τρομακτική προοπτική. Ωστόσο, οι απαιτήσεις των σύγχρονων προβλημάτων και εφαρμογών στην Επιστήμη των Υπολογιστών και στους περισσότερους άλλους γνωστικούς κλάδους καθιστούν αναγκαία τη χρήση πολλαπλών πρακτόρων, οι οποίοι μπορούν να εργάζονται από κοινού. Άλλωστε, όπως αναφέρει ο M. Wooldridge (1) «Δεν υπάρχει σύστημα με ένα μόνο πράκτορα».

Η κλάδος της Τεχνητής Νοημοσύνης έχει ωριμάσει και έχει πλέον τη δυνατότητα να αντιμετωπίζει πιο πολύπλοκα, ρεαλιστικά και μακροσκελή προβλήματα. Η ισχύς ενός ευφυούς πράκτορα περιορίζεται από τη γνώση του, τους υπολογιστικούς του πόρους και την αντίληψή του. Αυτός ο περιορισμένος

ορθολογισμός αποτέλεσε έναν από τους καθοριστικότερους λόγους, που οδήγησαν στη δημιουργία ομάδων πρακτόρων για την επίλυση προβλημάτων.

Τα πολυπρακτορικά συστήματα είναι ο κλάδος της Τεχνητής Νοημοσύνης, που αποσκοπεί στην παροχή αρχών τόσο για την κατασκευή πολύπλοκων συστημάτων πολλαπλών πρακτόρων όσο και μηχανισμών συντονισμού της συμπεριφοράς των ανεξάρτητων πρακτόρων, που συμμετέχουν στο σύστημα. Παρά το γεγονός, πως η ικανότητα να αναλογιζόμαστε την συντονισμένη συμπεριφορά των αυτόνομων πρακτόρων είναι καινούρια, ο αντίστοιχος τομέας εξελίσσεται ραγδαία έχοντας ως βάση προϋπάρχουσα εργασία στον τομέα της Κατανεμημένης Τεχνητής Νοημοσύνης (Distributed Artificial Intelligence-DAI).

Η Κατανεμημένη Τεχνητή Νοημοσύνη αποτελεί γνωστικό αντικείμενο της ΤΝ τις τελευταίες δεκαετίες. Διακρίνεται σε δύο τομείς: Την *Κατανεμημένη Επίλυση Προβλημάτων* (Distributed Problem Solving-DPS) και τα Πολυπρακτορικά Συστήματα. Το κυριότερο θέμα, με το οποίο ασχολείται ο πρώτος τομέας, είναι η διαχείριση πληροφοριών, όπως η διαίρεση της εργασίας σε μικρότερες και η σύνθεση της τελικής λύσης, μέσω της επίλυσης των υποεργασιών. Ασχολείται με το πώς ένα συγκεκριμένο πρόβλημα μπορεί να επιλυθεί από έναν αριθμό επεξεργαστικών μονάδων, που συνεργάζονται διαμοιράζοντας γνώση για το πρόβλημα που καλούνται να επιλύσουν, καθώς και τις επιμέρους λύσεις. Τα πολυπρακτορικά συστήματα επιτρέπουν σε αυτά τα υποπροβλήματα, στα οποία έχει διαιρεθεί η αρχική εργασία, να ανατεθούν σε διαφορετικούς πράκτορες επίλυσης προβλημάτων, καθένας από τους οποίους έχει τα δικά του ενδιαφέροντα και τους δικούς του στόχους. (22)

### 3.2 ΟΡΙΣΜΟΣ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Ένα Σύστημα Πολλών Πρακτόρων (Multiagent Systems-MAS) ή αλλιώς ΣΠΠ είναι ένα υπολογιστικό σύστημα, το οποίο αποτελείται από περισσότερους του ενός ευφυείς πράκτορες, μέσα σε ένα περιβάλλον. Πρόκειται για ένα σύστημα το οποίο σχεδιάστηκε και υλοποιήθηκε ως ένα σύνολο πρακτόρων που αλληλοεπιδρούν, δηλαδή συνεργάζονται, συντονίζονται, διαπραγματεύονται μεταξύ άλλων ενεργειών. Πιο συγκεκριμένα πρόκειται για ένα δίκτυο από «χαλαρά» συνδεδεμένους πράκτορες, οι οποίοι δρουν μαζί για να επιλύσουν προβλήματα, για τα οποία είναι δύσκολο ή ακόμα και αδύνατο να δώσει λύση ένα μονοπρακτορικό σύστημα, ίσως γιατί είναι πέραν των δυνατοτήτων και της γνώσης ενός μόνο πράκτορα. Η μεταξύ τους συνεργασία είναι δυναμική με την έννοια ότι οι οντότητες που συμμετέχουν στο σύστημα είναι αυτόνομες και άρα αποφασίζουν οι ίδιες για το πότε και πώς θα συνεργαστούν, ακόμα και σε περιπτώσεις που δεν είχαν αρχικά προβλεφθεί από τον σχεδιαστή τους.

Οι αλληλοεπιδράσεις αυτές μπορεί να είναι συνεργατικές ή «εγωιστικές». Αυτό σημαίνει πως οι πράκτορες επιλέγουν να συνεργάζονται επιλύοντας υποπροβλήματα, ώστε ο συνδυασμός των επιμέρους λύσεων που θα προκύψουν να αποτελέσουν την τελική λύση ή να εργαστούν αυτόνομα ανταλλάσσοντας υπηρεσίες και πληροφορίες προσπαθώντας να επιτύχουν τους δικούς τους ανεξάρτητους στόχους. Έτσι ένα πολυπρακτορικό σύστημα μπορεί να έχει ως στόχους τα παρακάτω:

- Την επίλυση προβλημάτων, που είναι πολύπλοκα για να επιλυθούν αποδοτικά από ένα μόνο πράκτορα.
- Την επίλυση προβλημάτων, τα οποία είναι από τη φύση τους κατανεμημένα, όπως για παράδειγμα προβλήματα στα οποία απαιτείται συλλογή πληροφοριών από κατανεμημένες πηγές, όπως δίκτυα αισθητήρων, κατανεμημένες βάσεις πληροφοριών και έλεγχος εναέριας κυκλοφορίας.
- Την επίλυση προβλημάτων στα οποία η εμπειρογνωμοσύνη είναι κατανεμημένη, όπως για παράδειγμα ροή εργασιών σε κάποιο περιβάλλον εργασίας.
- Τη διασύνδεση και λειτουργία ήδη υπάρχοντων συστημάτων, έτσι ώστε να είναι εύκολη η εκμετάλλευσή τους χωρίς σημαντικές τροποποιήσεις.

Σύμφωνα με την KatiaP. Sycara τα χαρακτηριστικά ενός πολυπρακτορικού συστήματος είναι τα ακόλουθα (21):

- Κάθε πράκτορας, που συμμετέχει σε ένα τέτοιο σύστημα, δεν έχει επαρκείς πληροφορίες ή ικανότητες να επιλύσει ένα πρόβλημα από μόνος του και συνεπώς έχει περιορισμένη οπτική πλευρά
- Δεν υπάρχει παγκόσμιο σύστημα ελέγχου, δηλαδή η συλλογική συμπεριφορά είναι αποτέλεσμα κοινωνικών κανόνων αλληλοεπιδράσεων και δεν επιβλέπεται από κάποια κεντρική αρχή
- Τα δεδομένα είναι αποκεντροποιημένα, δηλαδή οι πόροι που απαιτούνται για την ολοκλήρωση των εργασιών διαμοιράζονται και κατανέμονται στα μέλη του συστήματος.
- Οι υπολογισμοί είναι ασύγχρονοι

Τα κίνητρα για το αυξανόμενο ενδιαφέρον στη μελέτη των πολυπρακτορικών συστημάτων περιλαμβάνουν την ικανότητά τους να κάνουν τα εξής:

Πρώτον, να επιλύουν προβλήματα, τα οποία είναι πολύ μεγάλα για ένα και μόνο πράκτορα, επειδή υπάρχουν περιορισμοί στους υπολογιστικούς πόρους ή λόγω του καθαρού κινδύνου, που διατρέχει ένα κεντροποιημένο σύστημα, όπως μείωση απόδοσης ή κατάρρευσης σε κρίσιμες χρονικές στιγμές.

Δεύτερον, να επιτρέπουν τη διασύνδεση και τη διαλειτουργικότητα πολλών προγενέστερων, ξεπερασμένων συστημάτων. Για να συμβαδίζουν με τις αυξανόμενες απαιτήσεις, τα συστήματα που είχαν δημιουργηθεί στο παρελθόν

πρέπει να ενημερώνονται περιοδικά. Ο σχεδιασμός τέτοιων λογισμικών από την αρχή είναι ιδιαίτερα ακριβός και σε πολλές περιπτώσεις ακόμα και αδύνατος. Για το λόγο αυτό, ο μοναδικός τρόπος για να παραμείνουν τα συστήματα αυτά χρήσιμα είναι η ενσωμάτωσή τους σε μία πιο συνεργατική κοινωνία πρακτόρων, μέσα στην οποία μπορούν να χρησιμοποιηθούν από άλλα τμήματα λογισμικού.

Τρίτον, να παρέχουν λύσεις σε προβλήματα, που μπορούν να αντιμετωπιστούν από μία κοινότητα αυτόνομων αλληλοεπιδρώντων μελών-πρακτόρων.

Τέταρτον, να παρέχουν λύσεις, οι οποίες χρησιμοποιούν αποτελεσματικά πηγές πληροφοριών χωρικά κατανεμημένες. Χαρακτηριστικά παραδείγματα αποτελούν τα δίκτυα αισθητήρων (sensor networks), σεισμική παρακολούθηση (seismic monitoring), και συλλογή πληροφοριών από το Διαδίκτυο.

Πέμπτον, να παρέχουν λύσεις σε καταστάσεις, στις οποίες η τεχνογνωσία είναι κατανεμημένη, όπως η παράλληλη μηχανική (concurrent engineering), η ιατρική φροντίδα (medical care) και η βιομηχανία (manufacturing).

Έκτον, να βελτιώνουν την απόδοση σε όλες τις διαστάσεις του πολυπρακτορικού συστήματος. Πιο συγκεκριμένα, συμβάλλουν στην αύξηση της υπολογιστικής αποδοτικότητας (computational efficiency), καθώς οι υπολογισμοί πραγματοποιούνται παράλληλα από περισσότερους πράκτορες με την προϋπόθεση φυσικά πως το κόστος της επικοινωνίας μεταξύ των μελών του συστήματος παραμένει χαμηλό. Προσφέρουν αξιοπιστία (reliability), ανακάμπτοντας από αποτυχίες σύνθεσης του συστήματος, εντοπίζοντας με δυναμικό τρόπο πράκτορες με ανεπαρκείς δυνατότητες ή όχι αποτελεσματικό συντονισμό μεταξύ αυτών. *Επεκτασιμότητα* (extensibility) , επειδή ο αριθμός και οι δυνατότητες των πρακτόρων, που εργάζονται πάνω σε ένα πρόβλημα μπορούν να διαφοροποιηθούν. Ακόμη, παρέχουν *ευρωστία* (robustness), δηλαδή την ικανότητα του συστήματος να αντιστέκεται στις αλλαγές, χωρίς να προσαρμόζει την αρχική του σταθερή σύνθεση και *διατηρησιμότητα* (maintainability), επειδή ένα σύστημα αποτελούμενο από πολλούς πράκτορες είναι ευκολότερο να συντηρηθεί. Επιπλέον, *ευελιξία* (flexibility), επειδή πράκτορες με διαφορετικές ικανότητες μπορούν να προσαρμοστούν με τέτοιο τρόπο ώστε να οργανωθούν και να επιλύσουν το τρέχον πρόβλημα. Τέλος, προσφέρουν *επαναχρησιμοποίηση* (reuse), καθώς συγκεκριμένοι πράκτορες μπορούν να διατηρηθούν και να αποτελέσουν μέλος διαφορετικών ομάδων πρακτόρων για να επιλύσουν διαφορετικά προβλήματα (21).

Θα ήταν ανόητο να πιστεύουμε πως τα πολυπρακτορικά συστήματα θα πρέπει να χρησιμοποιούνται στο σχεδιασμό κάθε πολύπλοκου συστήματος. Όπως σε κάθε προσέγγιση κάποιου θέματος, υπάρχουν ορισμένες περιπτώσεις, στις οποίες είναι ιδιαίτερα κατάλληλη και σε άλλες όχι τόσο.

Κάποιοι τομείς απαιτούν τη χρήση πολυπρακτορικών συστημάτων. Συγκεκριμένα, αν υπάρχουν διαφορετικοί άνθρωποι ή οργανισμοί με διαφορετικούς, πολλές φορές αντικρουόμενους, στόχους και πληροφορίες, υπάρχει

και η ανάγκη για την ύπαρξη ενός πολυπρακτορικού συστήματος, αρμόδιου για το χειρισμό των αλληλοεπιδράσεών τους.

Οι εφαρμογές των πολυπρακτορικών συστημάτων καλύπτουν μία πληθώρα χώρων μελέτης, όπως είναι η βιομηχανία, έλεγχος εναέριας κυκλοφορίας, διαχείριση πληροφοριών, διαχείριση οικονομικών, τηλεπικοινωνίες, έλεγχος δικτύων, ιατρική περίθαλψη, ηλεκτρονικό εμπόριο, φορητές συσκευές και πολλοί άλλοι.

### 3.3 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΣΤΑ ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

#### 3.3.1 ΕΠΙΚΟΙΝΩΝΙΑ ΠΡΑΚΤΟΡΩΝ

Η αρχιτεκτονική και ο λόγος σχεδιασμού των πολυπρακτορικών συστημάτων καθιστούν αναγκαία την ανάθεση ρόλων στα μέλη πράκτορες, που τα απαρτίζουν. Αλλά για να είναι εφικτή αυτή η απόδοση ρόλων και εργασιών στους πράκτορες στα πλαίσια μιας συλλογικής συμπεριφοράς, όπως αυτή παρατηρείται μέσα σε ένα πολυπρακτορικό σύστημα, χρειάζεται η μεταξύ τους αλληλοεπίδραση. Η επίδραση αυτή του κάθε πράκτορα σε κάθε άλλο, με τον οποίο συνυπάρχει στο περιβάλλον εργασίας του, χρήζει μιας κατάλληλης γλώσσας επικοινωνίας.

Οι πράκτορες αποσκοπούν στο να πραγματοποιήσουν αποτελεσματικά τους στόχους, που τους ανατέθηκαν, να φτάσουν σε κοινά αποδεκτές συμφωνίες και να επιλύσουν ενδεχόμενες συγκρούσεις, οι οποίες προκύπτουν από την επίτευξη των επιμέρους στόχων τους. Όλες αυτές οι λειτουργίες υλοποιούνται χάριν στην ικανότητα των πρακτόρων για επικοινωνία. Η πιο γενική προσέγγιση της επικοινωνίας είναι η αλληλεπίδραση μέσω του περιβάλλοντος, μέσα στο οποίο η ενέργεια ενός πράκτορα, προκαλεί μια επίδραση, η οποία γίνεται αντιληπτή και ερμηνεύεται από τους άλλους πράκτορες. Οι πράκτορες στηρίζονται στην επικοινωνία με τα υπόλοιπα μέλη του συστήματος για να παρουσιάσουν τα αιτήματά τους, να γνωστοποιήσουν τις ικανότητες και τις πληροφορίες τους, να διαπραγματευτούν με άλλους πράκτορες. (23)

Δύο είδη επικοινωνίας κυριαρχούν μεταξύ των πρακτόρων και αυτά είναι η χρησιμοποίηση ενός κοινού σημείου συγκέντρωσης δεδομένων ή αλλιώς *μαυροπίνακας* (blackboard) και η άμεση ανταλλαγή μηνυμάτων (message passing).

Ο *μαυροπίνακας* είναι μια μορφή κοινής μνήμης, στον οποίο όλοι οι πράκτορες μπορούν να γράψουν κάτι ή να τον διαβάσουν. Συνήθως, ο *μαυροπίνακας* χωρίζεται σε περιοχές και οι πράκτορες αποκτούν προνόμια με την ικανότητα να τις διαβάζουν. Έτσι, άλλοι μπορούν να διαβάσουν και να γράψουν σε περισσότερες περιοχές από κάποιους άλλους. Στα συστήματα, που επικοινωνούν με χρήση *μαυροπίνακα*, υπάρχει ένας κοινός χώρος εργασίας για όλους τους πράκτορες του συστήματος, μέσω του οποίου είτε ανταλλάσσουν αποτελέσματα ή

διαμοιράζονται εργασίες. Τονίζεται πως από τη στιγμή, που κάτι αποθηκεύεται στην κοινή αυτή περιοχή, είναι προσπελάσιμο από όλους τους πράκτορες που συμμετέχουν στο σύστημα.

Ο πιο διαδεδομένος τρόπος επικοινωνίας στα σύγχρονα συστήματα πολλαπλών πρακτόρων είναι η ανταλλαγή μηνυμάτων. Πρόκειται για άμεση επικοινωνία μεταξύ των μελών του πολυπρακτορικού συστήματος, με μηνύματα που στέλνονται από τον έναν πράκτορα στον άλλο, και το περιβάλλον χρησιμοποιείται μόνο ως μέσο μετάδοσης. Το περιεχόμενο του μηνύματος διαφοροποιείται σημαντικά από σύστημα σε σύστημα. Αυτό που χρειάζεται στην περίπτωση των μηνυμάτων είναι η κωδικοποίησή τους σε μία γλώσσα επικοινωνίας μεταξύ των πρακτόρων.

Υπάρχουν δύο είδη προβλημάτων στην απόπειρα επικοινωνίας μεταξύ των μελών ενός πολυπρακτορικού συστήματος:

- *Ασυνέπειες* στη χρήση του συντακτικού και του λεξιλογίου. Ένα πρόγραμμα μπορεί να χρησιμοποιεί μία λέξη ή έκφραση για να περιγράψει κάτι, ενώ ένα άλλο πρόγραμμα χρησιμοποιεί την ίδια λέξη ή έκφραση για να περιγράψει κάτι εντελώς διαφορετικό
- *Μη συμβατότητα* μεταξύ διαφορετικών προγραμμάτων, που χρησιμοποιούν διαφορετικές λέξεις ή εκφράσεις για να περιγράψουν το ίδιο πράγμα

Η πρώτη μεγάλη πρόκληση, για την επίτευξη της συνεργασίας μεταξύ διαφορετικών και αυτόνομων πρακτόρων υπήρξε αυτή της αμοιβαίας κατανόησης. Όποιο και να είναι το μοντέλο διασύνδεσης, που υιοθετεί ένα πολυπρακτορικό σύστημα, απαραίτητη προϋπόθεση είναι η ύπαρξη δύο πρωτοκόλλων: εκείνου της επικοινωνίας και εκείνου της αλληλεπίδρασης.

Ένα *πρωτόκολλο επικοινωνίας* καθορίζει τη μορφή των μηνυμάτων (μήκος, επιτρεπτά σύμβολα) και τη σημασία τους, η οποία θα πρέπει να είναι γνωστή και κατανοητή από όλους τους πράκτορες, που συμμετέχουν στο σύστημα. Με άλλα λόγια καθορίζει τον τύπο των μηνυμάτων, που ανταλλάσσονται. Για παράδειγμα, ένα πρωτόκολλο επικοινωνίας είναι δυνατό να περιλαμβάνει ή ακόμη καλύτερα να ορίζει, διαφορετικούς τύπους μηνυμάτων για την ανταλλαγή πληροφορίας, που αφορά προτάσεις, αποδοχές, απορρίψεις, αποσύρσεις, διαφωνίες, αντιπροτάσεις και ενέργειες.

Ένα *πρωτόκολλο αλληλεπίδρασης* δίνει τη δυνατότητα στους πράκτορες να έχουν συζητήσεις, δηλαδή ακολουθίες μηνυμάτων. Για παράδειγμα, ένα πρωτόκολλο αλληλεπίδρασης, που χρησιμοποιείται σε καταστάσεις διαπραγματεύσεων μεταξύ πρακτόρων, θα μπορούσε να ορίζει ότι όταν ένας πράκτορας προτείνει μια σειρά από ενέργειες σε έναν άλλο πράκτορα, ο δεύτερος αφού αξιολογήσει την πρόταση θα πρέπει να απαντήσει με ένα μήνυμα αποδοχής, απόρριψης, διαφωνίας, κτλ (23).

### 3.3.2 ΣΥΝΤΟΝΙΣΜΟΣ ΠΡΑΚΤΟΡΩΝ

Σε ένα πολυπρακτορικό σύστημα, οι πράκτορες έχουν τα δικά τους σχέδια, προθέσεις και γνώση και επιθυμούν να επιλύσουν τους τους προσωπικούς τους στόχους, για τους οποίους σχεδιάστηκαν. Ωστόσο, για την επίλυση κάποιου καθολικού στόχου, απαιτείται ένας κατάλληλος μηχανισμός συντονισμού, έτσι ώστε να αντιμετωπιστούν οι συγκρούσεις που μπορεί να προκύψουν, μεταξύ των πρακτόρων, εξαιτίας των περιορισμένων πόρων ή των αντίθετων προθέσεων που ίσως παρουσιάσουν οι συμμετέχοντες.

Ο συντονισμός είναι μια διαδικασία, στην οποία οι πράκτορες εμπλέκονται για να εξασφαλίσουν το λογικό και ομαλό τρόπο λειτουργίας του πολυπρακτορικού συστήματος. Για το σκοπό αυτό, οι πράκτορες πρέπει να ανταλλάξουν πληροφορίες για τις δραστηριότητές τους. Ένας τρόπος με τον οποίο οι πράκτορες μπορούν να επιτύχουν το συντονισμό είναι μέσω της επικοινωνίας, όπως περιγράφηκε στην προηγούμενη ενότητα. Ένας άλλος τρόπος, χωρίς την ανάγκη για επικοινωνία, είναι κάνοντας την υπόθεση πως οι πράκτορες έχουν ένα «μοντέλο» της συμπεριφοράς των υπολοίπων.

Ο συντονισμός αποφεύγει περιττή δραστηριότητα και επιτρέπει μια πιο ομαλή και δίκαιη κατανομή των διαθέσιμων πόρων στα μέλη πράκτορες του συστήματος. Ακόμη, διαφυλάσσει το σύστημα από αδιέξοδα (deadlocks), δηλαδή σενάρια κατά τα οποία οι ενέργειες ενός ή περισσότερων πρακτόρων είναι αδύνατον να εκτελεστούν, και το διατηρεί σε ασφαλείς συνθήκες, περιορίζοντας τις συγκρούσεις (24).

Σύμφωνα με τους H. Nwana, L. Lee, και N. Jennings (25), κάθε ολοκληρωμένη τεχνική για συντονισμό πρέπει να έχει τέσσερα κύρια συστατικά:

- Ένα σύνολο δομών, που επιτρέπουν την αλληλεπίδραση των πρακτόρων με προβλεπόμενους τρόπους
- Ευελιξία, για να επιτρέπεται στους πράκτορες να λειτουργούν σε δυναμικά περιβάλλοντα και να ανταποκρίνονται επιτυχώς στις δυσκολίες με τη μερική και ανακριβή άποψη, που διαθέτουν, για την κοινωνία
- Ένα σύνολο κοινωνικών δομών, οι οποίες περιγράφουν τον τρόπο με τον οποίο οι πράκτορες θα πρέπει να συμπεριφέρονται στους υπόλοιπους, όταν εμπλέκονται στη διαδικασία του συντονισμού
- Επαρκής γνώση και λογική ικανότητα, για να γίνεται αξιοποίηση τόσο των διαθέσιμων ατομικών και κοινωνικών δομών όσο και της ευελιξίας

Σύμφωνα με τους F.Bergenti και A.Ricci (26) ένα μοντέλο συντονισμού περιγράφεται από τρία στοιχεία: *τους συντονιζόμενους*, δηλαδή τα αντικείμενα του συντονισμού που εδώ είναι οι πράκτορες λογισμικού, *το μέσο συντονισμού*, δηλαδή οτιδήποτε καθιστά ικανή την αλληλεπίδραση μεταξύ των εμπλεκόμενων στη διαδικασία συντονισμού, όπως η γλώσσα επικοινωνίας των πρακτόρων και τέλος *οι κανόνες συντονισμού*, για τον έλεγχο της ομαλής αλληλεπίδρασης μεταξύ

των πρακτόρων και της τήρησης των «νόμων», που ρυθμίζουν τη συμπεριφορά τους, όπως αυτοί περιγράφονται στο πρωτόκολλο αλληλεπίδρασης.

### 3.3.3 ΔΙΑΠΡΑΓΜΑΤΕΥΣΗ

Η *διαπραγμάτευση* (negotiation) είναι ένας σημαντικός τύπος κοινωνικής αλληλεπίδρασης. Στη διοίκηση και σε εργατικά ζητήματα συνάπτεται διαπραγμάτευση για τους όρους των συμβολαίων, στον κόσμο των επιχειρήσεων για την αγορά πρώτων υλών και την πώληση των προϊόντων, στη διπλωματία για τη σύναψη συμφωνιών ειρήνης μεταξύ των αντιπρόσωπων των διαφορετικών εθνών. Η διαπραγμάτευση μεταξύ ανθρώπων μελετάται στους ποικίλους κλάδους των κοινωνικών επιστημών, διεθνών σχέσεων, οικονομίας και κοινωνικής ψυχολογίας. Η αυτοματοποιημένη διαπραγμάτευση είναι μία ενεργή περιοχή μελέτης γενικά στην επιστήμη των υπολογιστών και πιο ειδικά στην τεχνητή νοημοσύνη. Οι απαιτήσεις για συστήματα αποτελούμενα από ευφυείς πράκτορες, που ανήκουν σε διαφορετικούς χρήστες ή οργανισμούς και έχουν την ικανότητα να φτάνουν σε συμφωνίες μέσω της διαπραγμάτευσης, γίνονται συνεχώς μεγαλύτερες. (27)

Οι αυτόνομοι πράκτορες χρησιμοποιούνται όλο και πιο διαδεδομένα σε ένα συνεχώς αυξανόμενο εύρος εφαρμογών. Οι περισσότερες από αυτές τις εφαρμογές συμπεριλαμβάνουν ή απαιτούν τη χρήση πολλαπλών πρακτόρων, οι οποίοι έχουν την ικανότητα να λειτουργούν αποτελεσματικά μέσα σε ένα πολύπλοκο περιβάλλον, σε συνύπαρξη με άλλους. Σε μία τέτοια κατάσταση, κατά την οποία οι πράκτορες πρέπει να μοιράζονται το περιβάλλον και τους διαθέσιμους πόρους που αυτό τους παρέχει με άλλους, οι συγκρούσεις είναι αναπόφευκτες. Ο μηχανισμός, που είναι αρμόδιος για την επίλυση αυτών των συγκρούσεων είναι η διαπραγμάτευση. (28)

Η διαπραγμάτευση αποτελεί μία συχνή μορφή αλληλεπίδρασης μεταξύ των ευφυών πρακτόρων με διαφορετικούς στόχους, σε ένα πολυπρακτορικό σύστημα. Πρόκειται για μια πιο πλούσια τεχνική, η οποία χρησιμοποιείται σε γενικότερες περιπτώσεις, όταν οι πράκτορες καλούνται να συνάψουν συμφωνίες. Πιο συγκεκριμένα, κατά τη διαδικασία της διαπραγμάτευσης δύο ή περισσότεροι πράκτορες, καθένας από τους οποίους προσπαθεί να ολοκληρώσει ένα προσωπικό στόχο ή αποστολή, φτάνουν σε μία απόφαση κοινού συμφέροντος. Οι πράκτορες αρχικά, επικοινωνώντας μεταξύ τους, εκφράζουν τις θέσεις και τις απόψεις τους σχετικά με το υπάρχον πρόβλημα. Οι απόψεις αυτές είναι πολύ πιθανό να είναι αντικρουόμενες. Στο σημείο αυτό, τα μέλη του συστήματος καλούνται να προσπαθήσουν να πλησιάσουν σε μία συμφωνία κάνοντας υποχωρήσεις ή ψάχνοντας για εναλλακτικές λύσεις. (28)



Η τεχνική της διαπραγμάτευσης έχει τέσσερα διαφορετικά συστατικά στοιχεία (1):

- Ένα *σύνολο διαπραγμάτευσης* (negotiation set), το οποίο αναπαριστά το χώρο των ενδεχόμενων προτάσεων που μπορούν να υποβάλουν οι πράκτορες
- Ένα *πρωτόκολλο*, το οποίο ορίζει τις έγκυρες προτάσεις που μπορούν να υποβάλουν οι πράκτορες, ως συνάρτηση του ιστορικού της διαπραγμάτευσης.
- Μια *συλλογή στρατηγικών*, μία για κάθε πράκτορα, η οποία καθορίζει ποιες προτάσεις υποβάλουν οι πράκτορες. Συνήθως, η στρατηγική του κάθε πράκτορα είναι ιδιωτική: το γεγονός ότι ένας πράκτορας χρησιμοποιεί μια συγκεκριμένη στρατηγική δεν είναι γενικά γνωστό στους άλλους πράκτορες που συμμετέχουν στη διαπραγμάτευση (αν και οι περισσότερες περιπτώσεις διαπραγμάτευσης είναι ανοιχτές, με την έννοια ότι οι πραγματικές προτάσεις που υποβάλλονται είναι γνωστές σε όλους όσους συμμετέχουν).
- Έναν *κανόνα*, ο οποίος καθορίζει πότε συνάπτεται συμφωνία και ποια είναι η τελική συμφωνία.

Η διαπραγμάτευση συνήθως πραγματοποιείται σε γύρους, και κάθε πράκτορας υποβάλλει μια πρόταση σε κάθε γύρο. Οι προτάσεις, τις οποίες υποβάλλουν οι πράκτορες ορίζονται από τη στρατηγική τους, προέρχονται από το σύνολο διαπραγμάτευσης, και πρέπει να είναι έγκυρες σύμφωνα με το πρωτόκολλο. Αν επιτευχθεί συμφωνία, με τον τρόπο που ορίζει ο κανόνας συμφωνίας, τότε η διαπραγμάτευση τερματίζεται με τη συμφωνία που έχει συναφθεί.

Οι ερευνητές του χώρου της Τεχνητής Νοημοσύνης έχουν αφιερώσει πολλή προσοχή και ενδιαφέρον στην αυτοματοποιημένη διαπραγμάτευση τα τελευταία χρόνια. Πολλοί από αυτούς ακολούθησαν μία πιο πρακτική προσέγγιση και κυρίως ασχολήθηκαν με την ανάπτυξη υπολογιστικών συστημάτων, τα οποία έχουν την ικανότητα να διαπραγματεύονται. Άλλοι ερευνητές ακολούθησαν μια πιο θεωρητική προσέγγιση και ασχολήθηκαν κυρίως με την ανάπτυξη θεωριών της διαπραγμάτευσης, όπως οικονομικών μεθόδων και μεθόδων της θεωρίας παιγνίων. Οι περισσότεροι από τους τελευταίους εστίασαν στη διεξαγωγή δημοπρασιών (auctions), σύναψη συμφωνιών, και σχηματισμός συμμαχιών (coalitions). Η δημιουργία συμμαχιών αποτελεί και το αντικείμενο ενδιαφέροντος μας για το υπόλοιπο αυτής της διπλωματικής εργασίας.

## ΚΕΦΑΛΑΙΟ 4: ΣΧΗΜΑΤΙΣΜΟΣ ΣΥΜΜΑΧΙΩΝ

### 4.1 ΤΙ ΕΙΝΑΙ ΣΥΜΜΑΧΙΑ ΣΤΗΝ ΚΟΙΝΩΝΙΑ

Συχνά, πολλά κοινωνικά προβλήματα ή ζητήματα είναι αρκετά μεγάλα και πολύπλοκα, για να αντιμετωπιστούν από ένα άτομο ή μία ομάδα ατόμων. Σε αυτές τις περιπτώσεις, ο σχηματισμός μιας συμμαχίας, που αποτελείται από μεμονωμένες οντότητες και ομάδες τέτοιων, μπορεί να αποδειχτεί μια αποτελεσματική στρατηγική για θέματα εκπαίδευσης, επιχειρήσεων, κυβερνητικά και άλλων συναφών τομέων για την επίλυση συγκεκριμένων προβλημάτων και επίτευξη καθορισμένων στόχων.

Με απλά λόγια, μία *συμμαχία* (coalition) είναι μία ομάδα ατόμων ή/και ομάδων αυτών με κοινά ενδιαφέρον την σύναψη συμφωνίας για συνεργασία προσανατολισμένοι σε έναν κοινό στόχο. Ο στόχος αυτός, θα μπορούσε να είναι τόσο περιορισμένου ενδιαφέροντος, όπως η απόκτηση χρηματοδότησης για κάποιο σκοπό, όσο και πιο ευρέος ενδιαφέροντος, όπως η βελτίωση της συνολικής ποιότητας διαβίωσης μέσα σε μια κοινωνία ανθρώπων.

Υπάρχουν πολλοί λόγοι για τους οποίους η δημιουργία μιας συμμαχίας μπορεί να αποδειχτεί καλή ιδέα. Μερικοί από αυτούς είναι η αντιμετώπιση ενός επείγοντος ζητήματος, η αποδοχή ή παροχή υπηρεσιών, η συγκέντρωση πόρων, η αύξηση της επικοινωνίας μεταξύ των ομάδων και ο προγραμματισμός πρωτοβουλιών σε μια μεγάλη ποικιλία ζητημάτων. Σε γενικές γραμμές, μπορεί να διατηρήσει το ενδιαφέρον της κοινωνίας εστιασμένο σε ένα ορισμένο πρόβλημα, να δημιουργήσει συνασπισμό μεταξύ εκείνων που σε φυσιολογικές συνθήκες δεν θα μπορούσαν να συνεργαστούν και να κρατήσει συνεπή τη προσέγγιση των ζητημάτων από την πλευρά της κοινωνίας. Η συνέπεια μπορεί να αποτελέσει σημαντικό παράγοντα στην αντιμετώπιση ενός κοινωνικού θέματος, ειδικά αν υπάρχουν ήδη άτομα ή ομάδες, που εργάζονται πάνω σε αυτό. Αν οι προσεγγίσεις τους διαφέρουν σημαντικά, και δεν συνεργάζονται, μπορεί να οδηγηθούν σε μια χαοτική κατάσταση κατά την οποία το κέρδος που θα επιτευχθεί θα είναι πολύ μικρό. Αντιθέτως, αν ακολουθήσουν συνεργατική συμπεριφορά και συμφωνήσουν σε έναν κοινό τρόπο αντιμετώπισης του θέματος για να πετύχουν κοινούς στόχους, είναι πολύ πιο πιθανό να σημειώσουν πρόοδο.

Οι συμμαχίες, μπορεί να είναι χαλαρές ενώσεις τα μέλη των οποίων εργάζονται για ένα σύντομο διάστημα με σκοπό την ολοκλήρωση ενός συγκεκριμένου έργου, και μετά να διασπαστούν. Αντίστοιχα, θα μπορούσαν να χαρακτηρίζονται από μονιμότητα και να μετατραπούν από μόνες τους σε οργανισμούς με ειδικές ευθύνες προς τη κοινωνία. Μπορεί να έχουν σχηματιστεί έτσι ώστε να εργάζονται για λογαριασμό μιας κοινωνίας, μιας περιοχής, μιας πόλης ή ακόμα και ενός ολόκληρου έθνους. Ανεξάρτητα από το μέγεθος ή τη δομή της, ο λόγος ύπαρξης των συμμαχιών εξακολουθεί να παραμένει ένας: η προσπάθεια και η υποστήριξη αυτής της προσπάθειας για την εκπλήρωση ενός συνόλου στόχων. (29)

#### 4.2 Η ΣΠΟΥΔΑΙΟΤΗΤΑ ΤΩΝ ΣΥΜΜΑΧΙΩΝ

Η ανάγκη για αντιμετώπιση προβλημάτων, των οποίων η επίλυση είναι αδύνατο να επιτευχθεί από μία και μόνο οντότητα οδηγεί στη δημιουργία συμμαχιών μεταξύ μεμονωμένων μελών και ομάδων. Οι συμμαχίες αυτές χαρακτηρίζονται από κοινές αξίες, στόχους και ενδιαφέροντα. Στα πλαίσια αυτού του συνασπισμού, τα μέλη που συμμετέχουν έχουν τη δυνατότητα να συνδυάζουν τους πόρους, που έχει στη διάθεσή του το καθένα, και να γίνονται πιο ισχυρά δουλεύοντας σε συνύπαρξη με άλλους όμοιούς τους, παρά να ενεργούσαν ατομικά. (30)

Η ικανότητα να κατασκευάζονται συμμαχίες είναι μία βασική δεξιότητα αυτών που επιδιώκουν να αποκτήσουν και να διατηρήσουν δύναμη και επιρροή (31). Μέσα στα πλαίσια των συμμαχιών τα πιο αδύναμα μέλη, μπορούν να αυξήσουν τη δύναμή τους και να υποστηρίξουν πιο αποτελεσματικά τις αξίες και τα ενδιαφέροντα τους, αφού τα μοιράζονται με άλλους.

Η δημιουργία συμμαχίας μπορεί να μετατοπίσει την ισορροπία της υπεροχής σε μία κατάσταση σύγκρουσης καθώς και να μεταβάλει τη μελλοντική έκβαση αυτής. Αυτοί, που συμμετέχουν, συγκεντρώνουν τους πόρους και τις πληροφορίες τους, γεγονός που τους επιτρέπει την αποδοτικότερη προώθηση των ιδεών και των απόψεών τους. Επιπλέον, η σύμπραξη αυτή συμβάλλει στην αποτελεσματικότερη αντίσταση εναντίον συγκεκριμένων απειλών, που προέρχονται από το εξωτερικό περιβάλλον της συμμαχίας, καθώς και την αντιμετώπιση και «αντεπίθεση» κατά αυτών των απειλών. Σε γενικές γραμμές, οι ομάδες με μικρή ισχύ παρουσιάζουν μεγαλύτερη επιτυχία στην υπεράσπιση των ενδιαφερόντων τους, αν εργάζονται στα πλαίσια μιας συμμαχίας, εναντίον μιας επικρατέστερης ομάδας. Η συμμετοχή ενός ατόμου σε μια συμμαχία, μπορεί οδηγήσει και στην εσωτερική του ενδυνάμωση, καθιστώντας το ικανό να ανταπεξέλθει και σε καταστάσεις μετά τη διάσπαση της συμμαχίας.

Μία συμμαχία από άτομα, ομάδες ή συνδυασμό αυτών μπορεί να αποφέρει μεγαλύτερο κέρδος και σε παραπάνω από έναν τύπο προβλημάτων και να αυξήσει το ποσοστό επιτυχούς επίλυσης αυτών. Ακόμη, επιτυγχάνεται η απολαβή περισσότερων πόρων και ειδικών γνώσεων για την αντιμετώπιση πιο πολύπλοκων

ζητημάτων, για τα οποία οι πληροφορίες και το δυναμικό της κάθε ομάδας ξεχωριστά θα κρινόταν ανεπαρκή. Οι απολαβές αυτές δε απευθύνονται αποκλειστικά σε φυσικούς και οικονομικούς πόρους, αλλά και πρόσβαση σε επαφές, επικοινωνίες και σχέσεις που διαθέτουν τα υπόλοιπα άτομα της συμμαχίας.

Η συνεργασία αυτή, μπορεί να οδηγήσει στην ανάδειξη νέων ηγετών. Πεπειραμένοι αρχηγοί των συστατικών ομάδων της συμμαχίας αναλαμβάνουν την ηγεσία ολόκληρου του συνόλου και έτσι προωθούνται οι επόμενοι για άνοδο στην ιεραρχία αναλαμβάνοντας την πρώτη θέση στις ανεξάρτητες ομάδες. Τέλος, στα παραπάνω πλεονεκτήματα μπορεί να προστεθεί και η αύξηση της επιρροής των ενεργειών της συμμαχίας. Η συμμετοχή σε μία συμμαχία σημαίνει πως υπάρχουν περισσότεροι, που κατανοούν τις απόψεις σου και που είναι πρόθυμοι να στηρίξουν την προσπάθειά σου.

Το κλειδί για μια επιτυχή δημιουργία συμμαχίας είναι, τα άτομα που συμμετέχουν να μην έχουν συνεργαστεί ξανά στο παρελθόν. Η προέλευση τους από διαφορετικά υπόβαθρα και όχι ίδιες οπτικές πλευρές, οδηγεί τα μέλη στην προσπάθεια να βρουν τον τρόπο πώς να σέβονται τη διαφορετικότητα των άλλων και να πετύχουν έναν απώτερο σκοπό. Πρέπει να συμφιλιωθούν με την ιδέα ότι συνυπάρχουν και συνεργάζονται με άλλους, έτσι ώστε κάθε ομάδα και οι αντιπρόσωποί τους ξεχωριστά να δώσουν σημασία στη διαφορετική αλλά πολύτιμη συνεισφορά του κάθε συστατικού μέλους της συμμαχίας στην συνολική στρατηγική, που ακολουθείται.

#### 4.3 ΣΥΜΜΑΧΙΕΣ ΜΕΤΑΞΥ ΠΡΑΚΤΟΡΩΝ

##### 4.3.1 ΕΙΣΑΓΩΓΗ

Ακολουθώντας τη φιλοσοφία του σχηματισμού συμμαχιών μεταξύ μελών και ομάδων στα πλαίσια μιας ανθρώπινης κοινωνίας για την επίτευξη στόχων υιοθετώντας μία κοινή στρατηγική και μία κοινή προσέγγιση του προβλήματος προς λύση, η έννοια αυτή επεκτείνεται και στο χώρο των πολυπρακτορικών συστημάτων. Η σύναψη συμφωνιών μεταξύ πρακτόρων λογισμικού για την ίδρυση συμμαχιών μέσα στην οποία θα αλληλοεπιδρούν, θα διαχειρίζονται κοινούς πόρους και δεδομένα, και θα προσπαθούν για τη μεγιστοποίηση του συνολικού κέρδους εμφανίζει παρόμοια πλεονεκτήματα με την ίδια σπουδαιότητα και για ένα πολυπρακτορικό σύστημα, όπως ακριβώς συζητήθηκε παραπάνω.

Σε προηγούμενες ενότητες συζητήθηκε πως οι εφαρμογές, που βασίζονται σε κατανομημένους υπολογισμούς, βρίσκουν μεγάλη ανταπόκριση στη σύγχρονη καθημερινή μας ζωή. Στις περισσότερες περιπτώσεις οι εφαρμογές αυτές αποτελούνται από περισσότερες της μιας οντότητας ή για να μιλήσουμε με όρους

με τους οποίους στο σημείο αυτό θα πρέπει να είμαστε οικείοι περισσότερους του ενός λογικού πράκτορα. Καθένας από αυτούς τους πράκτορες χαρακτηρίζεται από τους δικούς του στόχους και λόγους ύπαρξης. Σε τέτοια πολύπλοκα συστήματα, η συνύπαρξη πολλών διαφορετικών πρακτόρων προκαλεί την εμφάνιση εξαρτήσεων μεταξύ αυτών, που δεν μπορούν να προβλεφθούν από τον αρχικό τους σχεδιαστή.

Οι πράκτορες που μελετώνται διακρίνονται ως αντιδραστικοί (ανταποκρίνονται στις αλλαγές του περιβάλλοντος μέσα στο οποίο δρουν), δραστήριοι (εκδηλώνουν συμπεριφορά κατευθυνόμενη από τους στόχους τους και αναλαμβάνουν πρωτοβουλίες όταν το κρίνουν απαραίτητο) και κοινωνικοί (αλληλοεπιδρούν με άλλους πράκτορες ή ακόμα και ανθρώπους). Συγκεκριμένα για το τελευταίο χαρακτηριστικό, αυτό της κοινωνικότητας, αναφέρθηκε παραπάνω πως οι πράκτορες προβαίνουν σε μηχανισμούς επικοινωνίας, συντονισμού και διαπραγμάτευσης για την αρμονική και αποδοτική λειτουργία του συστήματος. Άλλωστε, η ικανότητα αυτή των πρακτόρων να αλληλοεπιδρούν μεταξύ τους και να εξισορροπούν τις ανεπάρκειες των άλλων, είναι ο σημαντικότερος λόγος που καθιστά τα πολυπρακτορικά συστήματα ένα τόσο χρήσιμο εργαλείο σε μια πληθώρα εφαρμογών όπως βιομηχανικών, οικονομικών, ιατρικών και διασκέδασης. (32)

Σε αυτές τις εφαρμογές οι πράκτορες μπορούν να ανήκουν σε έναν και μόνο σχεδιαστή και σε αυτήν την περίπτωση θεωρούνται ως συνεργατικοί (cooperative), δηλαδή κάθε πράκτορας ενδιαφέρεται να μεγιστοποιήσει τις συνολικές απολαβές ολόκληρου το συστήματος, ακόμα κι αν αυτό σημαίνει πως δεν βελτιώνει την προσωπική του χρησιμότητα. Άλλες εφαρμογές συμπεριλαμβάνουν έναν αριθμό πρακτόρων που αντιπροσωπεύουν διαφορετικούς χρήστες με τους δικούς του ανεξάρτητους στόχους προτιμήσεις. Σε αυτήν την περίπτωση, οι πράκτορες θεωρούνται ως “εγωιστές” (self-interested), δηλαδή ενεργούν με τέτοιο τρόπο ώστε να μεγιστοποιούν την δική τους χρησιμότητα, ανεξάρτητα από τις συνέπειες που μπορεί να επιφέρει η επιλογή αυτή στη χρησιμότητα των υπόλοιπων πρακτόρων μέσα στο σύστημα.

Κλείνοντας, αυτήν την εισαγωγική ενότητα αξίζει να σημειωθεί πως και στην περίπτωση των συνεργατικών αλλά και στην περίπτωση των εγωιστών πρακτόρων, ο σχεδιαστής του συστήματος χρειάζεται να εξασφαλίσει πως οι οντότητες, που συμμετέχουν σε αυτό, είναι αρμονικά οργανωμένες. Αρμονικά οργανωμένες με τέτοιο τρόπο, ώστε οι ρόλοι, οι σχέσεις και η διάρθρωση της εξουσίας, που ρυθμίζουν τη συμπεριφορά των πρακτόρων, έχουν οριστεί σαφώς. Πολλά παραδείγματα οργανωτικής προσπάθειας από τη μεριά των σχεδιαστών περιλαμβάνουν ιεραρχίες, ομάδες ή ομοσπονδίες. Κάθε ένα από αυτά τα παραδείγματα έχει τα δικά του πλεονεκτήματα και αδυναμίες, καθιστώντας το κατάλληλο για συγκεκριμένα προβλήματα και όχι τόσο για άλλα. Μεταξύ αυτών των παραδειγμάτων οργάνωσης των πρακτόρων, που κερδίζουν συνεχώς έδαφος στο χώρο των πολυπρακτορικών συστημάτων, βρίσκεται και ο σχηματισμός συμμαχιών.

Αυτό οφείλεται στο γεγονός, πως από τη φύση τους οι πρακτορικές συμμαχίες μπορούν να ανταποκριθούν αποτελεσματικά σε σενάρια, κατά τα οποία δεν υπάρχει μια κεντροποιημένη αρχή για την επίλυση πιθανών συγκρούσεων μεταξύ των μελών του συστήματος.

#### 4.3.2 ΣΥΝΕΤΑΙΡΙΣΤΙΚΗ ΘΕΩΡΙΑ ΠΑΙΓΝΙΩΝ

Σε πολλά σενάρια των πολυπρακτορικών συστημάτων η δημιουργία συνεργατικών σχέσεων μεταξύ των μελών τους δεν είναι δυνατή. Αυτό μπορεί να οφείλεται στο γεγονός πως οι πράκτορες δεν μπορούν να εμπιστευτούν τις υποσχέσεις του άλλου. Καλούνται να λάβουν κάποια απόφαση βασιζόμενοι πιστά στις πληροφορίες που διαθέτουν τη δεδομένη στιγμή, προσπαθούν να μεγιστοποιήσουν τη χρησιμότητά τους και υποθέτουν πως και οι άλλοι ακολουθούν την ίδια τακτική. Επίσης, οι πράκτορες μπορεί να μην ενδιαφέρονται για το συλλογικό κέρδος. Το μόνο που επιθυμούν να καταφέρουν είναι η επίτευξη μεγαλύτερης προσωπικής χρησιμότητας και υποθέτουν πως και οι άλλοι πράκτορες συμπεριφέρονται ομοίως.

Σε πολλές καταστάσεις του πραγματικού κόσμου όμως θεωρούμε πως τέτοιες υποθέσεις δεν υφίστανται. Οι πράκτορες ενός συστήματος καταφεύγουν σε δημιουργία συμμαχιών για την επίτευξη μεγαλύτερων συλλογικών και συνεπώς ατομικών απολαβών. Η άρνηση των παραπάνω υποθέσεων μας οδηγεί στη μελέτη της *συνεταιριστική θεωρίας παιγνίων* (cooperative game theory). (14)

Σε συντομία θα αναφέρουμε πως η *θεωρία παιγνίων* (game theory) πρόκειται για τη μελέτη των συγκρούσεων και της συνεργασίας. Οι ιδέες της εφαρμόζονται όποτε οι ενέργειες των πρακτόρων είναι αλληλεξαρτώμενες. Αυτοί οι πράκτορες μπορεί να είναι ξεχωριστές οντότητες, ομάδες ή οποιοσδήποτε συνδυασμός αυτών. Οι έννοιες της θεωρίας παιγνίων παρέχουν μια γλώσσα για τη διατύπωση, τη δομή, την ανάλυση και κατανόηση των στρατηγικών σεναρίων. Η εσωτερική συνοχή και οι μαθηματικές θεμελιώσεις της θεωρίας παιγνίων, την καθιστούν ένα πρωταρχικό εργαλείο για τη μοντελοποίηση και σχεδιασμό συστημάτων αυτόματης λήψης αποφάσεων σε διαδραστικά περιβάλλοντα. (33)

Το αντικείμενο μελέτης της θεωρίας παιγνίων είναι το *παίγνιο* (game). Αποτελεί μία μαθηματική μέθοδο ανάλυσης προβλημάτων, που αφορά τον τρόπο λήψης αποφάσεων σε καταστάσεις σύγκρουσης και συνεργασίας. Σε αυτό μπορούν να περιέχονται περισσότεροι του ενός παίκτες. Ένας παίκτης μπορεί να είναι ένα πρόσωπο, μία οργάνωση, ένα κράτος ή ένας συνασπισμός. Διάφορα προβλήματα της πολιτικής επιστήμης, της ψυχολογίας, των κοινωνικών και οικονομικών επιστημών μπορούν να μοντελοποιηθούν ως παίγνια.

Βασική υπόθεση της θεωρίας παιγνίων είναι αυτή της ευφυούς και λογικής συμπεριφοράς των παικτών. Ένας παίκτης χαρακτηρίζεται ως ευφυής, όταν έχει

τέλεια γνώση του πώς να παίξει το παίγνιο, και λογικός, όταν παίζει με αντικειμενικό στόχο τη μεγιστοποίηση του προσωπικού του οφέλους. Το όφελος του κάθε παίκτη δεν εξαρτάται μόνο από την επιλογή του, αλλά και από τις επιλογές των υπόλοιπων παικτών, οι οποίοι μπορεί να μην αντιμετωπίζονται απαραίτητα ως αντίπαλοί του.

Προχωρώντας ένα βήμα παραπέρα, ένα *συνεταιριστικό παίγνιο* (cooperative game) είναι αυτό στο οποίο ομάδες παικτών εκδηλώνουν συνεργατική συμπεριφορά, καθώς πρόκειται για ανταγωνισμό μεταξύ συμμαχιών και όχι μεμονωμένων παικτών. Το ενδιαφέρον τώρα εστιάζεται στις απολαβές, που μπορεί να αποκτήσει μια πιθανή συμμαχία μέσω της συνεργασίας των μελών της. Αποτελείται από δύο στοιχεία: ένα σύνολο από παίκτες και μία χαρακτηριστική συνάρτηση, που δηλώνει το υποσύνολο του αρχικού, με τους παίκτες που συμμετέχουν στην εκάστοτε συμμαχία.

Η θεωρία παιγνίων διαιρείται σε δύο κλάδους: τη *μη συνεταιριστική* (non-cooperative) και τη *συνεταιριστική* (cooperative). Οι δύο αυτοί κλάδοι διαφέρουν στον τρόπο με τον οποίο αντιμετωπίζουν τις αλληλεξαρτήσεις μεταξύ των παικτών.

Στην περίπτωση της μη συνεταιριστικής θεωρίας παιγνίων το ενδιαφέρον εστιάζεται στην διαδικασία λήψης αποφάσεων, με γνώμονα μόνο το δικό τους συμφέρον. Ασχολείται δηλαδή με την ανάλυση των ανεξάρτητων στρατηγικών επιλογών από τους παίκτες. Αντίθετα, η συνεταιριστική θεωρία παιγνίων ασχολείται και περιγράφει τα αποτελέσματα που προκύπτουν όταν οι παίκτες συνεργάζονται, σε διάφορους συνδυασμούς, μεταξύ τους.

## 4.4 ΣΥΜΜΑΧΙΕΣ ΣΤΑ ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

### 4.4.1 ΕΙΣΑΓΩΓΗ

Η μελέτη των πολυπρακτορικών συστημάτων έχει αποδείξει ξανά και ξανά πως οι ομάδες πρακτόρων, που συνεργάζονται, έχουν την τάση να πετυχαίνουν καλύτερα αποτελέσματα από αυτά που επιφέρουν οι ατομικές ενέργειες των μεμονωμένων πρακτόρων. Αυτές οι επιτυχίες γίνονται εμφανείς με πολλές μορφές: με την ολοκλήρωση στόχων γρηγορότερα ή με χαμηλότερο κόστος, την παραγωγή καλύτερης ποιότητας αποτελέσματος, ή την επίτευξη λογικής απόδοσης ακόμα και σε περιπτώσεις με μεγαλύτερο αριθμό «τσιγγούνηδων» πρακτόρων.

Η περισσότερη έρευνα, που έχει πραγματοποιηθεί, πάνω στη δημιουργία συμμαχιών στα πολυπρακτορικά συστήματα, περιλαμβάνει το ξεκίνημα με μία ομάδα πρακτόρων και παρέχοντάς της μια μεθοδολογία, που επιτρέπει τη συνεργασία μεταξύ των μελών της, ή ένα μηχανισμό για τη βελτίωση της συλλογικής απόδοσης. Ο πράκτορας συμμετέχει από προεπιλογή σε μία ομάδα και

αυτοί που καλούνται να αποφασίσουν, εκ των προτέρων, αν αξίζει να συνεργαστεί με άλλους πράκτορες σε μία ομάδα είναι οι μελετητές.

Μπορεί αυτή η προσέγγιση να είναι βολική σε καταστάσεις που μελετάμε τις επιδράσεις της αύξησης του μεγέθους της ομάδας σε ένα μηχανισμό συντονισμού ή που πειραματιζόμαστε με τη φύση του τρόπου με τον οποίο οι πράκτορες που ανήκουν ήδη σε μία ομάδα αλληλοεπιδρούν. Ωστόσο, οι περισσότερες ομάδες πρακτόρων του πραγματικού κόσμου δεν εμφανίζονται ξαφνικά και πλήρως «εξοπλισμένοι», αλλά προκύπτουν από αλληλοεπιδράσεις και κοινές απόψεις και ιδέες μεταξύ των μελών τους. Έχοντας κατά νου αυτό, βγαίνουν στο προσκήνιο πολλά ζητήματα που αγνοούνται στις περισσότερες έρευνες στα πολυπρακτορικά συστήματα. Σε πολλές ομάδες του πραγματικού κόσμου, για παράδειγμα, η πιθανότητα κάθε πιθανό μέλος να προσφέρει την ίδια χρησιμότητα στο σύνολο, δηλαδή να διαθέτει το ίδιο επίπεδο ικανοτήτων και να προσφέρει την ίδια ποιότητα εργασίας με τα υπόλοιπα μέλη, δεν είναι μεγάλη. Για αυτό το λόγο, γενικά, αξιολογούμε τη συμμετοχή κάποιων μελών σε σχέση με άλλων και μαθαίνουμε να μην συνεργαζόμαστε με άτομα λιγότερων ικανοτήτων. Σε κάποιες περιπτώσεις, τα μέλη μιας ομάδας μάλιστα πρέπει να αποφασίσουν αν αλληλοεπιδράσουν με όχι μόνο τόσο επιδέξιους αλλά πολλές φορές και κακόβουλους πράκτορες.

Υποθέτοντας πως υπάρχει μια χρήσιμη διαδικασία σχηματισμού ομάδων πρακτόρων, η μελέτη των πολυπρακτορικών συστημάτων προσπαθεί να απαντήσει σε διάφορα σημαντικά ερωτήματα. Με ποιον θα πρέπει να δημιουργήσει μια ομάδα ένας συγκεκριμένος πράκτορας; Πότε πρέπει να σχηματιστεί μια τέτοια ομάδα; Γιατί να θέλει ένας πράκτορας να σχηματίσει μια ομάδα; Πώς κατασκευάζεται αυτή η ομάδα; Τι περιορισμοί επιβάλλονται στους πράκτορες που συμμετέχουν σε μια τέτοια ομάδα;

Με σκοπό να βρεθούν απαντήσεις για τα παραπάνω και άλλα συναφή ερωτήματα, οι ερευνητές έχουν στρέψει την προσοχή τους στη μελέτη του σχηματισμού συμμαχιών στα πλαίσια ενός πολυπρακτορικού συστήματος. Ο στόχος της μελέτης αυτής είναι η διαμόρφωση ευφύων, συνεκτικών ομάδων μέσα στις οποίες ωφελούνται όλοι, αποκλείονται ενεργά αυτοί που παρακωλύουν την απόδοση της ομάδας, και αποφασίζεται αν πρέπει να πραγματοποιηθεί σύναψη άμεσων συνεργατικών σχέσεων με άλλους. (34)

#### 4.4.2 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

Ο σχηματισμός συμμαχιών στα πλαίσια ενός πολυπρακτορικού συστήματος είναι μία βασική μορφή αλληλεπίδρασης, η οποία επιτρέπει την δημιουργία συνεκτικών ομάδων από ξεχωριστούς, αυτόνομους πράκτορες με σκοπό την αποδοτική επίτευξη των ατομικών ή συλλογικών στόχων τους. Η συγκρότηση



αποτελεσματικών συμμαχιών αποτελεί μία μεγάλη ερευνητική πρόκληση στο χώρο των πολυπρακτορικών συστημάτων. (32)

Μια συμμαχία αποτελεί το αποτέλεσμα μιας ειδικής οργάνωσης μεταξύ των πρακτόρων. Σύμφωνα με τους Horling και Lesser, υπάρχουν κάποια κύρια χαρακτηριστικά, που διαχωρίζουν τις συμμαχίες από τις υπόλοιπες μορφές οργάνωσης.

*«Οι συμμαχίες στη γενική περίπτωση είναι προσανατολισμένες σε στόχους και δεν είναι μόνιμες. Σχηματίζονται στηριζόμενες σε έναν σκοπό και διασπώνται όταν αυτός ο σκοπός δεν υφίσταται πλέον, ή όταν πάψουν να είναι κατάλληλες για τον σκοπό αυτό ή όταν η αποδοτικότητα χάνεται με την αποχώρηση πρακτόρων»*, (35)

Ένα άλλο καθοριστικό χαρακτηριστικό των συμμαχιών είναι πως μέσα σε κάθε μία από αυτές, οι συμμετέχοντες πράκτορες συντονίζουν τις ενέργειές τους έτσι ώστε να πετύχουν τους στόχους της συμμαχίας. Όμως, καμία διαπραγμάτευση δεν λαμβάνει χώρα μεταξύ πρακτόρων που ανήκουν σε διαφορετικές συμμαχίες, εκτός αν οι στόχοι τους αλληλοεπιδρούν. Επιπλέον, η δομή μιας συμμαχίας συνήθως είναι ισοδύναμη, δηλαδή όλοι οι πράκτορες έχουν την ίδια αξία. Ωστόσο, μπορεί να υπάρξει ένας ηγέτης της συμμαχίας, ο οποίος ενεργεί ως αντιπρόσωπος ολόκληρου του συνόλου.

Κάθε πράκτορας, οφείλει να σκεφτεί την ιδέα της συμμετοχής του σε μία συμμαχία αν μπορεί να εκτελέσει μια εργασία που του έχει ανατεθεί, αλλά η συνεργασία του με άλλους πράκτορες θα αυξήσει τις απολαβές ή θα μειώσει το κόστος για την ολοκλήρωση αυτής της εργασίας. Επίσης, ένας πράκτορας καταφεύγει στη δημιουργία συνεργατικών σχέσεων με άλλους στο ίδιο σύστημα όταν δεν έχει τη δυνατότητα να εκτελέσει από μόνος του τα καθήκοντα με τα οποία τον έχει επιβαρύνει ο σχεδιαστής του ή μπορεί να τα εκτελέσει αλλά υπάρχουν άλλοι πράκτορες οι οποίοι είναι πιο αποδοτικοί στην εκτέλεση των ίδιων καθηκόντων, είτε γιατί απαιτούν λιγότερους πόρους είτε γιατί λειτουργούν γρηγορότερα.

Η πρώτη και η τρίτη συνθήκη, δεν είναι απαραίτητα εύκολα αντιληπτές από έναν πράκτορα, ειδικά σε περιπτώσεις ανεπαρκών πληροφοριών σχετικά με τις δυνατότητες των άλλων πρακτόρων του συστήματος και το αναμενόμενο κέρδος της εκτέλεσης μιας εργασίας από αυτούς. Παρ' όλα αυτά, ο συλλογισμός για τη συνολική χρησιμότητα της συνεργασίας και την εφαρμογή συλλογικών συνεταιριστικών στρατηγικών στηρίζεται σημαντικά σε αυτό το αναμενόμενο κέρδος. Υπολογίσιμα προσδοκώμενα κέρδη, μπορούν να συγκριθούν έτσι ώστε να αποφασιστεί πιο είναι το προτιμότερο και να οριστούν οι κατάλληλες ενέργειες της συμμαχίας για την επίτευξή τους. Οι απολαβές της συμμαχίας από την εκτέλεση μιας σειράς ενεργειών αξιολογούνται και εκφράζονται από τις συναρτήσεις χρησιμότητας. Το ενδιαφέρον σε μια μελέτη συμμαχιών εστιάζεται κυρίως στα

οφέλη ολόκληρου του συστήματος πρακτόρων, που συμμετέχουν σε αυτές και όχι τόσο στο κέρδος των ανεξάρτητων πρακτόρων. (36)

Έχοντας κατά νου αυτό το θεωρητικό υπόβαθρο, μπορούμε να είμαστε αντικειμενικοί υποστηρίζοντας πως ο σχηματισμός συμμαχιών έχει προσελκύσει το ενδιαφέρον για συστηματική μελέτη και έχει αποδείξει πως πρόκειται για ένα χρήσιμο εργαλείο σε ένα μεγάλο αριθμό σεναρίων του πραγματικού κόσμου και των πολυπρακτορικών συστημάτων. Για παράδειγμα, στο ηλεκτρονικό εμπόριο, οι αγοραστές μπορούν να σχηματίσουν μια συμμαχία για τη μαζική αγορά ενός προϊόντος για να επωφεληθούν από τις εκπτώτικές τιμές. Στα καταναμημένα δίκτυα αισθητήρων, οι συμμαχίες αισθητήρων μπορούν να συνεργαστούν για τον εντοπισμό στόχων ενδιαφέροντος. Τέλος, η δημιουργία συμμαχιών μπορεί να χρησιμοποιηθεί και στη συλλογή πληροφοριών με πολλούς εξυπηρετητές (servers) να συνεργάζονται με άλλους για να ανταποκρίνονται σε ερωτήματα (queries).

Στο σημείο αυτό ας δώσουμε μια πιο τεχνική μορφή στην έννοια της συμμαχίας χρησιμοποιώντας μερική ορολογία, την οποία θα ακολουθήσουμε και στις επόμενες ενότητες στα πλαίσια της διπλωματικής εργασίας.

Θα ασχοληθούμε με καταστάσεις, στις οποίες συμμετέχουν  $n$  πράκτορες και κατά βάση  $n > 2$  και θεωρούμε  $Ag = \{1, \dots, n\}$ , το σύνολο αυτών των πρακτόρων. Θα αναφερόμαστε σε ένα υποσύνολο του  $Ag$  με τον όρο συμμαχία και θα χρησιμοποιούμε τους συμβολισμούς  $C, C', C_I$  για να δηλώνουμε μια συμμαχία. Η χρήση του όρου συμμαχία στην περίπτωση των πρακτόρων μπορεί να δημιουργήσει μια μικρή σύγχυση, καθώς η έννοια αυτή στην καθομιλουμένη δηλώνει μία ομάδα ανθρώπων, που παρουσιάζουν σε κάποιο βαθμό δέσμευση σε μία συνεργατική ενέργεια. Στο χώρο των πολυπρακτορικών συστημάτων ο όρος σημαίνει απλά ένα σύνολο από πράκτορες, που μπορεί ή όχι να εργάζονται από κοινού. Αναφερόμαστε με τον όρο *γενική συμμαχία* (grand coalition), στη συμμαχία που αποτελείται από όλους τους πράκτορες, δηλαδή  $C = Ag$ . Με τον όρο *μοναδιαία συμμαχία* (singleton coalition) αναφερόμαστε σε αυτήν η οποία περιλαμβάνει μόνο έναν πράκτορα εκ του συνόλου  $Ag$ . Γενικά, θεωρούμε πως μια συμμαχία έχει την ικανότητα να αποκτά μια συγκεκριμένη χρησιμότητα (utility), η οποία μπορεί να μοιραστεί μεταξύ των μελών της.

Αναφέραμε νωρίτερα, πως το παίγνιο αποτελεί το βασικό αντικείμενο μελέτης των συνεργατικών σχέσεων και αλληλεπιδράσεων, που αναπτύσσονται μεταξύ των μελών μιας συμμαχίας. Ένα συνεταιριστικό παίγνιο συμβολίζεται με  $\mathcal{G}$  και μοντελοποιείται με τη δυάδα  $\mathcal{G} = (Ag, v)$ , όπου  $Ag$  όπως είδαμε είναι ένα σύνολο από πράκτορες και  $v: 2^{Ag} \rightarrow \mathcal{R}$ , ονομάζεται η *χαρακτηριστική εξίσωση* (characteristic function) του παιγνίου.

Η ιδέα πίσω από την χαρακτηριστική εξίσωση είναι πως αναθέτει σε κάθε δυνατή συμμαχία μία πραγματική τιμή, που ενστικτωδώς αναπαριστά το κέρδος που πρέπει να διαμοιραστεί μεταξύ των μελών αυτής της συμμαχίας. Αυτό σημαίνει πως αν  $v(C) = k$ , τότε η συμμαχία  $C$  μπορεί να εργαστεί με τέτοιο τρόπο ώστε να

αποκτήσει χρησιμότητα  $k$ , η οποία μπορεί κατανεμηθεί σε όλους τους πράκτορες που συμμετέχουν. Ο τρόπος με τον οποίο θα γίνει αυτή η κατανομή είναι τελείως ανεξάρτητος από το ίδιο το παίγνιο. Οι πράκτορες είναι αυτοί, που καλούνται να συμφωνήσουν μεταξύ τους για τη διαίρεση της χρησιμότητας.

Δεδομένου ενός τέτοιου παιγνίου, η συνεταιριστική θεωρία παιγνίων προσπαθεί να απαντήσει σε ερωτήματα του τύπου ποιες συμμαχίες μπορούν να σχηματιστούν από πράκτορες και πώς οι απολαβές της συμμαχίας μπορούν να διαιρεθούν με λογικό τρόπο στα μέλη της.

#### 4.5 ΣΤΑΔΙΑ ΤΗΣ ΣΥΝΕΤΑΙΡΙΣΤΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

Ο τρόπος με τον οποίο οι πράκτορες είναι οργανωμένοι σε ένα σύστημα επηρεάζει ή ακόμα καθορίζει και τις μεταξύ τους αλληλοεπιδράσεις. Για παράδειγμα, αν οι πράκτορες οργανωθούν με ιεραρχικό τρόπο, τότε ο κάθε ένας από αυτούς θα έχει τη δυνατότητα να συντονίζεται μόνο με τους προγόνους και/ή τους απογόνους του στο ιεραρχικό δένδρο. Προφανώς, υπάρχουν πολλές εναλλακτικές μέθοδοι οργάνωσης ενός συστήματος εκτός της ιεραρχικής με τις δικές τους αδυναμίες και αντίστοιχα δυνατά σημεία. Όπως έχουμε αναφέρει, μια από αυτές τις μεθόδους είναι η δημιουργία συμμαχιών. (37)

Έχουμε απαριθμήσει μια σειρά προβλημάτων στα οποία η συγκρότηση των πρακτόρων σε συμμαχίες βρίσκει εύφορο έδαφος για εφαρμογή. Άλλωστε αυτό που προσπαθούμε να πετύχουμε είναι η ανάδειξη των συμμαχιών σε ένα από τα καταλληλότερα εργαλεία για την αποτελεσματική αντιμετώπιση της πλειοψηφίας των προβλημάτων, με τα οποία ασχολείται η τεχνητή νοημοσύνη. Πρέπει να τονιστεί πως το πρόβλημα σχηματισμού δομών συμμαχιών εξυπηρετεί ως ένα πρώτο βήμα στην κατανόηση και πρόταση λύσεων ζητημάτων του πραγματικού κόσμου. Σε κάθε περίπτωση, που χρησιμοποιούνται συμμαχίες πρακτόρων, η διαδικασία διαμόρφωσής τους περιλαμβάνει τρεις κύριες ενέργειες.

##### 4.5.1 ΥΠΟΛΟΓΙΣΜΟΣ ΑΞΙΑΣ ΤΗΣ ΣΥΜΜΑΧΙΑΣ (COALITIONAL VALUE CALCULATION)

Το πρώτο στάδιο της διαμόρφωσης μιας συμμαχίας απαιτεί από τους πράκτορες τον υπολογισμό της χρησιμότητας κάθε συμμαχίας, που μπορεί να σχηματιστεί δεδομένου ενός συνόλου  $n$  πρακτόρων. Η χρησιμότητα αυτή μπορεί να αποδειχτεί υπολογιστικά πολύπλοκη καθώς το σύνολο των δυνατών συμμαχιών, που μπορούν να διαμορφωθούν, αυξάνεται με τον αριθμό των πρακτόρων και συγκεκριμένα πρόκειται για ένα πλήθος της τάξης  $2^n - 1$ . Το πλήθος αυτό δεν είναι  $2^n$ , γιατί θεωρούμε πως δεν υπάρχει νόημα μελέτης μιας κενής συμμαχίας, δηλαδή να μην περιέχει κανέναν πράκτορα.

Ένας αριθμός αλγορίθμων σχηματισμού συμμαχιών έχουν αναπτυχθεί για τον προσδιορισμό των πιθανών συμμαχιών που μπορούν πραγματικά να δημιουργηθούν. Για να επιτευχθεί αυτό, οι αλγόριθμοι υπολογίζουν την *αξία της συμμαχίας* (coalition value), η οποία παρέχει μια ένδειξη των προσδοκώμενων απολαβών που μπορούν να προέλθουν από το σχηματισμό της πρώτης. Έχοντας υπολογίσει τις αξίες όλων των συμμαχιών, η απόφαση για το ποια ή ποιες είναι οι ιδανικές από αυτές είναι πολύ εύκολη.

Ο τρόπος με τον οποίο η τιμή αυτή υπολογίζεται εξαρτάται από το πρόβλημα που βρίσκεται υπό μελέτη και η πολυπλοκότητα του υπολογισμού ποικίλει από γραμμική σε εκθετική (38). Σε έναν ηλεκτρονικό χώρο αγορών για παράδειγμα, η τιμή της αξίας μια συμμαχίας καταναλωτών μπορεί να υπολογιστεί ως η διαφορά μεταξύ του αθροίσματος των εξόδων από τις κρατήσεις όλων των μελών της συμμαχίας και του ελάχιστου κόστους που απαιτείται για την ικανοποίηση των αιτημάτων όλων των πελατών. Στη συλλογή πληροφοριών, η αξία αυτή μπορεί να αναπαριστά ένα μέτρο για το πόσο στενά συσχετισμένοι είναι οι τομείς πληροφοριών των πρακτόρων, που συμμετέχουν. Σε περιπτώσεις στις οποίες ο ορθολογισμός των πρακτόρων είναι δεσμευμένος εξαιτίας της υπολογιστικής πολυπλοκότητας, η αξία της συμμαχίας μπορεί να αναπαριστά το καλύτερο δυνατό αποτέλεσμα που μπορεί να πετύχει η συμμαχία, έχοντας στη διάθεσή της περιορισμένους πόρους, για την επίλυση ενός προβλήματος.

Μία από τις κύριες προκλήσεις όμως σε αυτό το πρώτο στάδιο της συλλογικής δράσης εντοπίζεται στο πλήθος των αξιών προς υπολογισμό, που είναι εκθετικό ως προς τον αριθμό των πρακτόρων του συστήματος. Ένας τρόπος να αντιμετωπιστεί αυτός ο καταίγισμός από πράξεις είναι να γίνει κατανομή των υπολογισμών μεταξύ των πρακτόρων, παρά να συγκεντρώνονται σε έναν και μόνο πράκτορα. Έτσι, η διαδικασία των υπολογισμών των αξιών είναι γρηγορότερη και η δυσκολία αυτής δεν επιβαρύνει μόνο έναν. Για να γίνει αυτό, ωστόσο, απαιτείται η ανάπτυξη αλγορίθμων που υποδεικνύουν σαφώς πώς πρέπει να διεξαχθούν αυτοί οι υπολογισμοί με αποδοτικό κατανεμημένο τρόπο.

#### 4.5.2 ΠΑΡΑΓΩΓΗ ΔΟΜΗΣ ΣΥΜΜΑΧΙΑΣ (COALITION STRUCTURE GENERATION)

Το επόμενο βήμα προς τη συλλογική δράση καθώς και το πρόβλημα που θα μας απασχολήσει στη συνέχεια της παρούσας μελέτης περιλαμβάνει την επιλογή, από τον κάθε πράκτορα, της συμμαχίας στην οποία θα συμμετέχει. Πώς ένας πράκτορας επιλέγει την καταλληλότερη συμμαχία για να συμμετέχει; Ποιες πληροφορίες έχει στη διάθεσή του, που θα τον οδηγήσουν στην λογικότερη επιλογή; Η βασική υπόθεση, που κάνουμε σε αυτό το σημείο, είναι πως η μοναδική πληροφορία που είναι διαθέσιμη στους πράκτορες του συστήματος είναι η χαρακτηριστική

συνάρτηση του παιγνίου. Αυτό σημαίνει πως τους είναι γνωστό το κέρδος που μπορεί να εξασφαλίσει κάθε συμμαχία.

Ο σκοπός του κάθε πράκτορα είναι να μεγιστοποιήσει τη χρησιμότητά του. Για το λόγο αυτό ενστικτωδώς, θα επιχειρήσει να συνάψει συμμαχία με μία ομάδα, της οποίας οι απολαβές είναι υψηλές. Όλοι οι πράκτορες όμως θα προσανατολίσουν τις ενέργειές τους με αυτόν τον ορθολογισμό και αυτό είναι και το σημείο στο οποίο οι σκέψεις στρατηγικής εισχωρούν στο χώρο της θεωρίας παιγνίων.

Τι εννοούμε όμως με την έκφραση «σκέψεις στρατηγικής»; Ας υποθέσουμε πώς υπάρχουν δύο πράκτορες υπό μελέτη: ένας σκληρά εργαζόμενος και ένας όχι τόσο. Είναι πολύ πιθανό, ο δεύτερος να επιθυμεί να συμμετέχει σε μια συμμαχία με τον πρώτο, καθώς γνωρίζει πως εκτελεί μεγάλο ποσοστό έργου και το κέρδος που θα αποκομίσει θα είναι σαφώς μεγαλύτερο από αυτό που θα ήταν αποτέλεσμα των ατομικών του ενεργειών. Παρόμοιες σκέψεις γίνονται όμως και από την πλευρά του εργατικού. Μπορεί να παρουσιάσει διστακτική διάθεση να συνεργαστεί με κάποιον απρόθυμο για εργασία, και να προσπαθήσει να συμμετέχει σε μία συμμαχία που αποτελείται από πιο παραγωγικούς πράκτορες. Μία τέτοια συμμαχία, που θα αποτελούταν από αυτούς τους δύο πράκτορες χαρακτηρίζεται ως *ασταθής* (unstable), καθώς ο πρώτος έχει το κίνητρο να αποχωρήσει. Θα μιλήσουμε για τη σταθερότητα μιας συμμαχίας παρακάτω. (14)

Ας δώσουμε στο σημείο αυτό μια πιο συμβολική περιγραφή του προβλήματος της παραγωγής δομής συμμαχιών γνωστό και στην ξένη ορολογία ως *CSG*. Δεδομένων των προσδοκώμενων κερδών της κάθε δυνατής προς κατασκευή συμμαχίας, καλούμαστε να διαιρέσουμε (partition) το αρχικό σύνολο πρακτόρων  $Ag$  σε εξαντλητικές και ανεξάρτητες συμμαχίες. Ένας τέτοιος διαχωρισμός ονομάζεται *δομή συμμαχίας* (coalition structure).

Μέχρι τώρα, θεωρούσαμε σιωπηρά πως ένας πράκτορας θα ενεργήσει στρατηγικά, όπως στα μη συνεταιριστικά παίγνια, προσπαθώντας να μεγιστοποιήσει την ατομική του χρησιμότητα. Αν ολόκληρο το σύστημα ελέγχεται και «ανήκει» σε έναν και μόνο σχεδιαστή, τότε η απόδοση των ανεξάρτητων πρακτόρων που το απαρτίζουν ίσως και να μην είναι ο κυριότερος προβληματισμός μας. Αντίθετα, ίσως είναι υψίστης σημασίας το ενδιαφέρον μας να εστιάζεται στην αύξηση των συλλογικών απολαβών για ολόκληρο το σύστημα, μέσω της αύξησης του αθροίσματος των κερδών των επιμέρους συμμαχιών.

Ας θεωρήσουμε το παρακάτω παράδειγμα για να γίνει πιο κατανοητό το πρόβλημα που παρουσιάζεται. Έστω πως έχουμε το σύνολο πρακτόρων  $Ag = \{a_1, a_2, a_3\}$ . Τότε, υπάρχουν επτά πιθανές συμμαχίες:

$$\{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_2, a_3\}, \{a_3, a_1\}, \{a_1, a_2, a_3\}.$$

Επιπλέον, υπάρχουν πέντε διαφορετικές πιθανές δομές συμμαχιών, που δηλώνουν τους διαφορετικούς συνδυασμούς συμμαχιών στους οποίους μπορεί να διαχωριστεί το σύστημα. Αυτές οι δομές είναι οι εξής:

$$\{\{a_1\}, \{a_2\}, \{a_3\}\}, \{\{a_1\}, \{a_2, a_3\}\}, \{\{a_2\}, \{a_3, a_1\}\}, \{\{a_3\}, \{a_1, a_2\}\}, \{\{a_1, a_2, a_3\}\}.$$

Δεδομένου ενός συνεταιριστικού παιγνίου  $\mathcal{G} = (Ag, v)$ , λέμε πως η ιδανική δομή συμμαχιών  $CS^*$ , για το  $\mathcal{G}$  δίνεται ως εξής:

$$CS^* = \arg \max_{CS \in \text{διαχωρισμό του } Ag} V(CS)$$

$$\text{όπου,} \quad V(CS) = \sum_{C \in CS} v(C).$$

Δυστυχώς, ο χώρος των πιθανών δομών συμμαχιών μεγαλώνει ραγδαία με την αύξηση του αριθμού των οντοτήτων που συμμετέχουν, εδώ του αριθμού των πρακτόρων του συστήματος. Το γεγονός αυτό καθιστά σημαντικά δύσκολο τον εντοπισμό της ιδανικής διχοτόμησης του αρχικού συνόλου πρακτόρων σε συμμαχίες. Η αναζήτηση της ιδανικής περίπτωσης μέσα σε ολόκληρο το φάσμα των πιθανών λύσεων είναι συνήθως ανεπιθύμητη και στη χειρότερη περίπτωση ανέφικτη. Συγκεκριμένα, ο Sandholm μελέτησε το πρόβλημα αυτό και απόδειξε πως η εύρεση του καλύτερου διαχωρισμού σε συμμαχίες είναι NP πολυπλοκότητας. Για την αντιμετώπιση αυτής της πολυπλοκότητας, αναπτύχθηκαν αρκετοί αλγόριθμοι τα τελευταία χρόνια, χρησιμοποιώντας διάφορες τεχνικές, όπως δυναμικός προγραμματισμός, ακέραιος προγραμματισμός και στοχαστική ανάλυση.

Τέλος, πρέπει να σημειώσουμε πως η ιδανική λύση στο πρόβλημα CSG είναι μία που μεγιστοποιεί το συνολικό κέρδος του συστήματος. Αντίθετα με τα συνεργατικά περιβάλλοντα, στα οποία οι πράκτορες ενδιαφέρονται κυρίως για την επίτευξη του μεγαλύτερου συλλογικού κέρδους, στα μη συνεργατικά οι πράκτορες ενεργούν περισσότερο εγωιστικά και δρουν με μόνο σκοπό την αύξηση των ατομικών τους απολαβών. Αυτό, όμως, δεν σημαίνει πως ένας αλγόριθμος που σχεδιάστηκε για το πρόβλημα CSG δεν μπορεί να εφαρμοστεί στα εγωιστικά πολυπρακτορικά συστήματα. Η ελαστικότητα αυτών των αλγορίθμων οφείλεται στο γεγονός πως οι σχεδιαστές τέτοιων συστημάτων συνήθως ενδιαφέρονται για την βελτίωση της συνολικής απόδοσης του συστήματος και σε πολλές περιπτώσεις αυτός ο στόχος επιτυγχάνεται μέσω της μεγιστοποίησης των συνολικών απολαβών.

### **Σταθερότητα συμμαχιών (coalition stability)**

Ο ορθολογισμός, που χαρακτηρίζει τους πράκτορες που μελετάμε, προβλέπει πως ένα μέλος του πολυπρακτορικού συστήματος θα ενεργήσει με τέτοιο τρόπο για συμμετοχή σε μια συμμαχία, που θα του αποφέρει μεγαλύτερο κέρδος από αυτό

που θα προέκυπτε από την ατομική του δράση. Οφείλουμε να υποθέσουμε πως όλοι οι πράκτορες του συστήματος θα σκεφτούν με παρόμοιο τρόπο , και θα επιχειρήσουν να εισαχθούν σε μία όσο το δυνατόν περισσότερο παραγωγική συμμαχία. (14)

Αυτό σημαίνει πως ένας πράκτορας δεν μπορεί να γίνει μέλος όποιας συνεργατικής οργάνωσης διαλέξει. Μία συμμαχία μπορεί να διαμορφωθεί αν όλοι οι πράκτορες σε αυτήν επιθυμούν να ανήκουν εκεί όπου ανήκουν. Συγκεκριμένα, μια συμμαχία θα σχηματιστεί μόνο αν κανένας από τους συμμετέχοντες πράκτορες δεν μπορεί να πετύχει καλύτερα αποτελέσματα αν αποσκιρτήσει και δημιουργήσει μια συμμαχία από μόνος του. Στην περίπτωση αυτή, λέμε πως η συγκεκριμένη συμμαχία χαρακτηρίζεται από *σταθερότητα* (stability). Πρόκειται για μία άκρως επιθυμητή ιδιότητα των συμμαχιών σε πολυπρακτορικά συστήματα, καθώς εγγυάται πως οι πράκτορες θα αφιερώσουν τους διαθέσιμους πόρους τους στην συμμαχία που έχουν επιλέξει να συμμετέχουν και δεν θα τους σπαταλούν στις διαπραγματεύσεις και μετακινήσεις τους σε άλλες συμμαχίες. Με τον τρόπο αυτό, εξασφαλίζεται μεγαλύτερη περίοδος ζωής για τη συμμαχία, αρκετή για να ικανοποιηθούν οι στόχοι της. (32)

Στην παραπάνω συζήτηση για την παραγωγή δομών συμμαχίας αναφέραμε πως ένας πράκτορας, πιθανώς να εμφανίσει διάθεση για αποχώρηση από μία συμμαχία στην οποία ανήκει. Αυτό μπορεί να οφείλεται στη συνεργασία του με πράκτορες, οι οποίοι δεν συνεισφέρουν εξίσου ισοδύναμα με τον ίδιο στην συλλογική δράση. Η ύπαρξη πρακτόρων με την διάθεση για διάλυση της συμμαχίας και αναζήτηση άλλων πιο κερδοφόρων καθιστά μία συμμαχία ασταθή (unstable).

Συνεπώς, για να είναι σε θέση ο πράκτορας να επιλέξει την κατάλληλη συμμαχία για να γίνει μέλος της, ουσιαστικά περιορίζει το ερώτημα «σε ποια συμμαχία να εισέλθω;» στο «ποιες από τις συμμαχίες στις οποίες μπορώ να εισέλθω είναι σταθερές;». Πρέπει να τονίσουμε πως η ιδιότητα της σταθερότητας είναι χρήσιμη αλλά όχι επαρκής συνθήκη για τη σύνθεση μιας συμμαχίας. Αυτό για το οποίο μπορούμε να είμαστε σίγουροι είναι πως μια ασταθής συμμαχία δεν θα σχηματιστεί ποτέ. Ωστόσο, το γεγονός ότι μία συμμαχία είναι σταθερή δεν σημαίνει απαραίτητα ότι θα σχηματιστεί.

#### 4.5.3 ΚΑΤΑΝΟΜΗ ΚΕΡΔΟΥΣ (PAYOFF DISTRIBUTION)

Το τελευταίο αυτό στάδιο της συλλογικής δράσης περιλαμβάνει την απόφαση για τι απολαβές θα επιφέρει αυτή η συνεργασία, που επιτεύχθηκε στα προηγούμενα στάδια, σε κάθε πράκτορα που συμμετείχε. Έχοντας αποφασίσει ποιες συμμαχίες θα πρέπει να σχηματιστούν μέσα στο πολυπρακτορικό σύστημα, είναι σημαντικό να οριστούν και οι ανταμοιβές προς κάθε μέλος, έτσι ώστε να επιθυμεί να παραμείνει στην ίδια συμμαχία καθιστώντας την παράλληλα σταθερή.

Αν τα οφέλη της συλλογικής δράσης συσσωρεύονται για λογαριασμό ολόκληρης της συμμαχίας, τα μέλη αυτής καλούνται να συμφωνήσουν τον τρόπο διαίρεσης των πρώτων ανάμεσά τους. Γενικά, ο στόχος είναι αυτή η διαίρεση να γίνει με τέτοιο τρόπο ώστε να ικανοποιούνται συγκεκριμένα κριτήρια, όπως αυτό της *δικαιοσύνης* (fairness). Δηλαδή, η αμοιβή του κάθε πράκτορα οφείλει να ανταποκρίνεται στη συνεισφορά του στο παίγνιο. Όλες οι προσεγγίσεις αυτού του θέματος, που έχουν πραγματοποιηθεί, προσπαθούν να απαντήσουν στο ίδιο ερώτημα: πόση αμοιβή πρέπει να λάβει ο κάθε πράκτορας με βάση τη συνεισφορά του στη συμμαχία.

Η ανάλυση των τρόπων αποδοτικής και δίκαιης διαμοίρασης των κερδών ενός παιγνίου στα μέλη που αποτελούν τη συμμαχία έχει μελετηθεί στα πλαίσια της θεωρίας των παιγνίων. Πολλές λύσεις έχουν προταθεί για την επίτευξη αυτού του σκοπού, οι οποίες βασίζονται σε διαφορετικές απόψεις της σταθερότητας. Δύο από τις πιο γνωστές λύσεις αυτή του *πυρήνα* και της *τιμής Sharpley*.

### **Κατανομή κέρδους με πυρήνα (core)**

Η ιδέα του πυρήνα αποτελεί ένα τρόπο που αναπτύχθηκε για να δώσει απαντήσεις στη κύρια θεμελιώδη ερώτηση της συνεταιριστικής θεωρίας παιγνίων, δηλαδή πώς να διαμοιραστεί το κέρδος ως αποτέλεσμα της συλλογικής δράσης μιας συμμαχίας σε όλα τα μέλη της. Με άλλα λόγια, ποια δεσμευτική συμφωνία πρέπει να συναφθεί μεταξύ των πρακτόρων που συμμετέχουν στη συμμαχία. Μέσα από την ανάλυση που θα ακολουθήσει θα γίνει και πιο ξεκάθαρη και η ιδέα της σταθερότητας μιας συμμαχίας, που αναφέρθηκε παραπάνω.

Η βασική ιδέα του πυρήνα (core) είναι πως μια συμφωνία μεταξύ των πρακτόρων, που ανήκουν στο σύνολο  $Ag$ , μπορεί να είναι δεσμευτική αν κάθε συμμαχία  $C \subset Ag$  λαμβάνει, συνολικά, τουλάχιστον τη χρησιμότητα που μπορεί να παράγει ή να διεκδικήσει όπως ορίζεται στη χαρακτηριστική εξίσωση του παιγνίου, δηλαδή την τιμή που συμβολίζεται με  $v(C)$ . Για τα παρακάτω θα βασιστούμε στη σταθερότητα της γενικής συμμαχίας (grand coalition), δηλαδή αυτής που αποτελείται από όλους τους πράκτορες του συνόλου  $Ag$ . (39)

Ο πυρήνας ενός συνεταιριστικού παιγνίου αποτελεί το σύνολο των εφικτών κατανομών των απολαβών στα μέλη μιας συμμαχίας  $C$ , κατά τις οποίες καμία συμμαχία  $C' (C' \subset C)$ , δεν μπορεί να εκφράσει διαφωνία. Αλλά ας μιλήσουμε με μεγαλύτερη ακρίβεια και σαφήνεια. Ένα αποτέλεσμα  $x$  για μια συμμαχία  $C$  σε ένα παίγνιο  $\mathcal{G} = (Ag, v)$  είναι μία κατανομή της χρησιμότητας της  $C$  στα μέλη της, δηλαδή,  $x = \langle x_1, \dots, x_k \rangle$ . Η κατανομή αυτή είναι εφικτή για τη  $C$ , με την έννοια πως η  $C$  μπορεί όντως να λάβει το κέρδος που αναπαριστά το  $x$  και παράλληλα αποδοτική, με την έννοια πως διατίθεται ολόκληρη η χρησιμότητα της  $C$  στα μέλη της. Τυπικά, ένα αποτέλεσμα της μορφής  $x = \langle x_1, \dots, x_k \rangle$  για μια συμμαχία  $C = \{1, \dots, k\}$  πρέπει να ικανοποιεί την εξής ιδιότητα:



$$\sum_{i \in C} x_i = v(C).$$

Για παράδειγμα ας θεωρήσουμε πως έχουμε ένα παίγνιο με  $Ag = \{1,2\}$  έτσι ώστε  $v(\{1\}) = 5$ ,  $v(\{2\}) = 5$  και  $v(\{1,2\}) = 20$ . Τότε, τα πιθανά αποτελέσματα είναι  $\langle 20,0 \rangle$ ,  $\langle 19,1 \rangle$ ,  $\langle 18,2 \rangle, \dots, \langle 0,20 \rangle$ . Το πρώτο αποτέλεσμα σημαίνει πως ο πράκτορας 1 θα λάβει όλη τη χρησιμότητα της συμμαχίας ενώ ο πράκτορας 2 δεν θα κερδίσει τίποτα. Στο δεύτερο, ο πράκτορας 1 θα λάβει 19 από τη συνολικό κέρδος και ο 2 θα λάβει 1 κ.ο.κ.

Μια συμμαχία  $C$  αντιτίθεται σε μια προτεινόμενη κατανομή κέρδους της γενικής συμμαχίας, αν υπάρχει κάποια άλλη εναλλακτική που οδηγεί όλα τα μέλη της  $C$  σε πιο βελτιωμένη θέση από την πλευρά των απολαβών. Πιο αναλυτικά, μία συμμαχία  $C \subset Ag$  διαφωνεί με το προτεινόμενο αποτέλεσμα  $\langle x_1, \dots, x_k \rangle$  της γενικής συμμαχίας αν υπάρχει αποτέλεσμα  $\langle x'_1, \dots, x'_k \rangle$ , τέτοιο ώστε  $x'_i > x_i$  για όλα τα  $i \in C$ .

Προφανώς, μια προτεινόμενη κατανομή κέρδους δεν θα πραγματοποιηθεί αν διαφωνεί μια συμμαχία και θα βρισκόταν σε πιο ευνοϊκή θέση αν αποσκιρτούσε και εργαζόταν ανεξάρτητα από τους υπόλοιπους πράκτορες του παιγνίου. Αναφέραμε πως ο πυρήνας είναι το σύνολο των αποτελεσμάτων για την γενική συμμαχία, για τα οποία δεν εκφράζει διαφωνία καμία μικρότερη συμμαχία. Αν ο πυρήνας δεν είναι κενός, τότε υπάρχει κάποιος τρόπος για την γενική συμμαχία να συνεργαστεί και να κατανέμει την παραχθείσα χρησιμότητα μεταξύ των μελών με τέτοιο τρόπο ώστε καμία μικρότερη συμμαχία να μπορεί να πετύχει καλύτερα αποτελέσματα από μόνη της.

Αν επιστρέψουμε στο παράδειγμά μας, ο προφανής τρόπος να εξετάσουμε αν ο πυρήνας είναι κενός ή όχι, είναι να ελέγξουμε όλα τα αποτελέσματα με τη σειρά έχοντας στο μυαλό μας την ερώτηση να μπορεί να διαφωνήσει κάποια συμμαχία. Ξεκινώντας με το  $\langle 20,0 \rangle$ , παρατηρούμε πως είναι λογικό για τον πράκτορα 2 να διαφωνήσει για συμφωνία καθώς μπορεί να πετύχει περισσότερα από μόνος του,  $v(\{2\}) = 5$ . Ομοίως μια κατανομή όπως η  $\langle 4,16 \rangle$ , θα είχε ως αποτέλεσμα ένσταση από τη μεριά του πράκτορα 1 καθώς, αν εργαστεί ανεξάρτητα θα λάμβανε μεγαλύτερη χρησιμότητα,  $v(\{1\}) = 5$ .

Η ουσία της ιδέας του πυρήνα είναι η μελέτη της σταθερότητας και όχι της δικαιοσύνης. Πρόκειται για μια ευκολονόητη προσέγγιση της σταθερότητας και πιθανώς να είναι η ιδέα που έχει μελετηθεί περισσότερο στα συνεργατικά παίγνια. Φαίνεται να είναι ένας χρήσιμος μηχανισμός για να εκτιμήσουμε τη σταθερότητα μιας συμμαχίας, ωστόσο η μέθοδος αυτή δεν μπορεί με κανέναν τρόπο να καθορίσει ποιο αποτέλεσμα κατανομής κέρδους θα εφαρμοστεί τελικά. Για το λόγο αυτό προτάθηκε η τιμή *Shapley*. (14)

### Κατανομή κέρδους με τιμή *Shapley* (*Shapley value*)

Όπως είδαμε ο πυρήνας είναι ένας τρόπος κατανομής των απολαβών, ως αποτέλεσμα μιας συνεργατικής δράσης στα μέλη μιας συμμαχίας, δίνοντας

ιδιαίτερη σημασία στη σταθερότητα αυτής. Η τιμή *Shapley* χρησιμοποιείται ακριβώς για τον ίδιο σκοπό με τη διαφορά πως εστιάζει περισσότερο σε μια πιο δίκαιη διαμοίραση του κέρδους της γενικής συμμαχίας στα μέλη της.

Το πρώτο πράγμα που θα σκεφτούμε, ενστικτωδώς, όταν ακούσουμε την έννοια «δίκαιη διαμοίραση» είναι, όλοι οι πράκτορες να λάβουν την ίδια αμοιβή. Μια τέτοια προσέγγιση όμως, θα καθιστούσε την κατανομή πλήρως ανεξάρτητη από την απόδοση και τη συνεισφορά των πρακτόρων και θα τους στερούσε το κίνητρο για να συνεχίσουν να εργάζονται το ίδιο αποτελεσματικά. Η τιμή *Shapley* προσπαθεί να ξεπεράσει αυτό το πρόβλημα, προσφέροντας υψηλότερη ανταμοιβή σε κάποιον πράκτορα που πρόσφερε περισσότερο στη συμμαχία από αυτό που πρόσφεραν οι ομότιμοί του. (14)

Η τιμή (value) είναι ένας τελεστής  $\mu$  που αναθέτει σε κάθε συμμαχία ένα διάνυσμα απολαβών,  $\mu(C) = (\mu_1, \mu_2, \dots, \mu_n)$ . Θεωρούμε πως  $\mu_i(C)$  είναι το ποσό που συνεισφέρει ο πράκτορας  $i$  στη συνολική χρησιμότητα της συμμαχίας  $C$  ή με άλλα λόγια η δύναμη του πράκτορα  $i$  στη συμμαχία. Ο τελεστής αυτός έχει την τιμή (40):

$$\mu_i(C) = v(C \cup \{i\}) - v(C).$$

Αξίζει να σημειώσουμε πως αν  $\mu_i(C) = v(\{i\})$ , τότε δεν υπάρχει κανένα κέρδος από τη συμμετοχή του πράκτορα  $i$  στη συμμαχία, καθώς η χρησιμότητα που κερδίζει η συμμαχία από τη συνεισφορά του είναι ακριβώς η ίδια με την ποσότητα κέρδους αν εργαστεί ανεξάρτητα. Επίσης, με  $sh_i$  συμβολίζουμε την αμοιβή την οποία κερδίζει ένας πράκτορας  $i \in Ag$  σε ένα παίγνιο  $\mathcal{G} = (Ag, v)$ .

Ο *Shapley* ισχυρίστηκε πως οποιαδήποτε προσπάθεια για δίκαιη κατανομή κέρδους μεταξύ των πρακτόρων μιας συμμαχίας πρέπει να ικανοποιεί τρία αξιώματα, τα οποία όρισε ο ίδιος:

- *Συμμετρία (symmetry)*: Το πρώτο αξίωμα απαιτεί οι πράκτορες που συνεισφέρουν το ίδιο σε κάθε συμμαχία, που μπορούν να συμμετέχουν να ανταμείβονται το ίδιο. Οι πράκτορες αυτοί καλούνται ανταλλάξιμοι (interchangeable). Συνεπώς δύο πράκτορες  $i$  και  $j$  είναι ανταλλάξιμοι αν  $\mu_i(C) = \mu_j(C)$  για κάθε  $C \subseteq Ag$ . Σύμφωνα με το αξίωμα της συμμετρίας  $sh_i = sh_j$ . (14)
- *Αξίωμα του «ανόητου» παίκτη (dummy player)*: Ανόητος παίκτης ονομάζεται αυτός ο οποίος συνεισφέρει σε οποιαδήποτε συμμαχία αυτό που μπορεί να κερδίσει και μόνος του έξω από αυτήν. Δηλαδή ένας πράκτορας  $i \in Ag$  είναι ανόητος αν  $\mu_i(C) = v(\{i\})$  για κάθε συμμαχία  $C \subseteq Ag$ . Σύμφωνα με το αξίωμα αυτό οι ανόητοι πράκτορες θα πρέπει να λαμβάνουν αμοιβή ακριβώς ίση με το ποσό που θα κέρδιζαν μόνοι τους, δηλαδή  $sh_i = v(\{i\})$ . (14)
- *Προσθετικότητα (additivity)*: Για το αξίωμα αυτό κάνουμε την υπόθεση πως έχουμε στη διάθεσή μας δύο διαφορετικά συνεταιριστικά παίγνια, που ορίζονται από δύο διαφορετικές χαρακτηριστικές εξισώσεις  $v_1$  και

$v_2$  στα οποία συμμετέχει το ίδιο σύνολο πρακτόρων. Σύμφωνα με το αξίωμα της προσθετικότητας, αν θεωρήσουμε τα δύο αυτά διαφορετικά παίγνια ένα, κατά το οποίο κάθε συμμαχία  $C$  πετυχαίνει κέρδος  $v_1(C) + v_2(C)$ , οι πληρωμές των πρακτόρων σε κάθε συμμαχία θα είναι το άθροισμα των πληρωμών που θα κατάφεραν να λάβουν σε κάθε παίγνιο, από τα δύο, ξεχωριστά. Πιο συμβολικά, έστω πως  $\mathcal{G}_1 = (Ag, v_1)$  και  $\mathcal{G}_2 = (Ag, v_2)$  τα δύο συνεργατικά παίγνια με το ίδιο σύνολο πρακτόρων  $Ag$ ,  $i \in Ag$  ένας πράκτορας,  $sh_i^1$  και  $sh_i^2$  η χρησιμότητα του πράκτορα  $i$  στα παίγνια  $\mathcal{G}_1$  και  $\mathcal{G}_2$  αντίστοιχα και τέλος  $\mathcal{G}_{1+2} = (Ag, v_{1+2})$  το παίγνιο τέτοιο ώστε  $v_{1+2}(C) = v_1(C) + v_2(C)$ . Τότε, η τιμή Sharpley για τον πράκτορα  $i$  στο παίγνιο  $\mathcal{G}_{1+2}$  είναι  $sh_i^1 + sh_i^2$ . (41)

Περιγράψαμε τις ιδιότητες που πρέπει να ικανοποιεί μια κατανομή κέρδους για να χαρακτηρίζεται δίκαιη. Η τιμή Sharpley υπολογίζεται από τη σχέση:

$$\mu_i(C) = \frac{1}{|Ag|!} \sum_{C \subseteq Ag \setminus \{i\}} |C|! (|Ag| - |C| - 1)! [v(C \cup \{i\}) - v(C)].$$

Η παραπάνω έκφραση μπορούμε να θεωρήσουμε πως αποτυπώνει την κατά μέσο όρο ελάχιστη συνεισφορά (average marginal contribution) του πράκτορα  $i$ , με την οποία υπολογίζουμε το μέσο όρο όλων των διαφορετικών αλληλουχιών εισόδων των πρακτόρων για να σχηματιστεί η γενική συμμαχία από την αρχή. Πιο συγκεκριμένα, φανταζόμαστε πως μια συμμαχία διαμορφώνεται ξεκινώντας χωρίς κανένα πράκτορα να συμμετέχει σε αυτήν, και προσθέτοντας ένα πράκτορα κάθε φορά με την επιλογή του πράκτορα προς εισαγωγή να είναι τελείως τυχαία. Μέσα σε κάθε τέτοια αλληλουχία, εξετάζουμε την ελάχιστη συνεισφορά του πράκτορα  $i$  τη στιγμή, που εισέρχεται στη συμμαχία. Αν γίνει μέλος της συμμαχίας  $C$ , η συνεισφορά του σε αυτήν είναι  $v(C \cup \{i\}) - v(C)$ . Πολλαπλασιάζουμε αυτή την ποσότητα με τον όρο  $|C|!$ , δηλαδή τον αριθμό των διαφορετικών τρόπων που μπορεί να σχηματιστεί το σύνολο  $C$ , πριν την εισαγωγή του  $i$  και με τον όρο  $(|Ag| - |C| - 1)!$ , δηλαδή τον αριθμό των διαφορετικών αλληλουχιών με τις οποίες μπορούν να εισαχθούν οι εναπομείναντες πράκτορες του  $Ag$  στο σύνολο  $C$ , μετά το  $i$ . Τέλος, προσθέτουμε όλα τα πιθανά σύνολα  $C$ , που μπορούν να σχηματιστούν, και λαμβάνουμε το μέσο όρο διαιρώντας με τον όρο  $|Ag|!$ , δηλαδή τον αριθμό των πιθανών διατάξεων όλων των πρακτόρων.

Η τιμή Sharpley υπάρχει πάντα και είναι μοναδική, ενώ ο πυρήνας δεν εγγυάται αυτές τις συνθήκες της ύπαρξης και της μοναδικότητας. Και οι δύο μέθοδοι εξασφαλίζουν τη διαμοίραση ολόκληρης της χρησιμότητας της συμμαχίας στα μέλη που την αποτελούν. Επίσης, εξασφαλίζουν πως οι ανεξάρτητοι πράκτορες καθώς και η γενική συμμαχία έχουν κίνητρο να επιθυμούν την παραμονή τους στην ίδια δομή συμμαχίας και όχι να αποσκιρτούν. Ωστόσο, αντίθετα με τον πυρήνα, η τιμή Sharpley δεν εγγυάται πως όλα τα υποσύνολα των πρακτόρων βρίσκονται σε

ευνοϊκότερη θέση αν εξακολουθούν να παραμένουν σε μία συμμαχία, από το να σχηματίσουν μία ξεχωριστά από μόνοι τους.

## **ΚΕΦΑΛΑΙΟ 5: ΠΑΡΑΓΩΓΗ ΔΟΜΗΣ ΣΥΜΜΑΧΙΑΣ ΚΑΙ ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΟΥ IP**

### **5.1 ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ**

Τα πολυπρακτορικά συστήματα γίνονται καθημερινά όλο και πιο απαραίτητα. Ένας λόγος είναι πως η τεχνολογία προωθεί συνεχώς αναπτυσσόμενες υποδομές επικοινωνίας, πάνω από τις οποίες ξεχωριστά σχεδιασμένοι πράκτορες, που ανήκουν σε διαφορετικούς οργανισμούς, μπορούν να αλληλοεπιδρούν σε ένα ανοιχτό περιβάλλον σε πραγματικό χρόνο και να φέρουν εις πέρας συναλλαγές με ασφάλεια. Ο δεύτερος λόγος είναι ο καθοριστικός ρόλος που έχουν στη συμμετοχή τους στη διαδικασία λήψης αποφάσεων. Οπότε, έχοντας κάποιος κατά νου τα πλεονεκτήματα ενός πράκτορα είναι πολύ εύκολο να φανταστεί το όφελος μετά τη συνεργασία αυτών των οντοτήτων και σύναψη συμμαχιών μεταξύ τους. (42)

Η τεχνολογία των πολυπρακτορικών συστημάτων διευκολύνει την αυτοματοποιημένη δημιουργία τέτοιων δυναμικών συμμαχιών. Αυτή η αυτοματοποίηση οδηγεί σε εξοικονόμηση εργασίας από την πλευρά των ανθρώπων για διαπραγματεύσεις. Επιπλέον, οι ευφυείς πράκτορες αποδεικνύονται πιο

αποτελεσματικοί στην εύρεση κερδοφόρων βραχυπρόθεσμων συμμαχιών σε σχέση με τους ανθρώπους. (43)

Η ικανότητα για σχηματισμό αποτελεσματικών συμμαχιών είναι ένας από τους στόχους κλειδιά της μελέτης των πολυπρακτορικών συστημάτων. Τα συνεταιριστικά παίγνια είναι μοντέλα που εκμεταλλεύονται τις ευκαιρίες για σύναψη συνεργασίας αποτυπώνοντας λεπτομερώς την ικανότητα των πρακτόρων να πραγματοποιούν συντονισμένες ενέργειες.

Στο μεγαλύτερο μέρος της βιβλιογραφίας, που αναφέρεται στη διαμόρφωση συμμαχιών μεταξύ των πρακτόρων στα πλαίσια ενός πολυπρακτορικού συστήματος, το ενδιαφέρον εστιάζεται στα παίγνια με χαρακτηριστική εξίσωση, στα οποία η αποτελεσματικότητα μιας συμμαχίας δεν επηρεάζεται από τον τρόπο με τον οποίο οι πράκτορες είναι ταξινομημένοι μέσα στο σύστημα. Σε αυτό το είδος παιγνίων η χρησιμότητα κάθε συμμαχίας  $C$  δίνεται από μία χαρακτηριστική εξίσωση  $v(C)$ , για την οποία κάνουμε την υπόθεση πως ισχύει  $v(C) \geq 0$ .

Όπως συζητήθηκε στο προηγούμενο κεφάλαιο, ένα από τα στάδια της συνεταιριστικής ενέργειας μεταξύ των οντοτήτων ενός συστήματος, αποτελεί η παραγωγή δομής συμμαχίας (coalition structure generation-CSG). Η διαδικασία αυτή αποτελεί ένα απαιτητικό πρόβλημα και περιλαμβάνει την εύρεση μιας εξαντλητικής και διεξοδικής διαίρεσης του συνόλου των πρακτόρων του συστήματος σε συμμαχίες, με τέτοιο τρόπο ώστε η απόδοση του του συστήματος να βελτιστοποιείται.

Το πρόβλημα της παραγωγής δομής συμμαχίας είναι ισοδύναμο με το πρόβλημα της πλήρους διαίρεσης ενός συνόλου, ενός από τα βασικότερα προβλήματα στη συνδυαστική βελτιστοποίηση (combinatorial optimization). Το θέμα αυτό έχει εφαρμογές σε πολλούς επιστημονικούς χώρους, όπως στις πολιτικές και οικονομικές επιστήμες και στην επιστήμη των υπολογιστών και ασχολείται με την εύρεση μίας βέλτιστης λύσης ανάμεσα σε ένα σύνολο πιθανών λύσεων. Σε ζητήματα για την αντιμετώπιση των οποίων ακολουθείται μια τέτοια μέθοδος η εξαντλητική αναζήτηση όλων των πιθανών αποτελεσμάτων δεν είναι πάντα εφικτή. Η προσέγγιση αυτή λειτουργεί σε προβλήματα στα οποία το σύνολο των εφικτών λύσεων είναι ευδιάκριτο ή μπορεί να περιοριστεί για να γίνει ευδιάκριτο και επομένως η διαδικασία της αναζήτησης του καλύτερου αποτελέσματος να είναι διευκολύνεται.

Σε ένα πρόβλημα παραγωγής δομής συμμαχίας, διαθέτουμε ένα σύνολο  $Ag$  με  $n$  στοιχεία και μια συνάρτηση αποτίμησης  $v$ . Ακόμη,  $\wp(Ag)$  είναι το δυναμοσύνολο του  $Ag$  και το ζητούμενο είναι να διαιρέσουμε το δεδομένο σύνολο σε ξεχωριστά υποσύνολα, ή στην περίπτωση μας συμμαχίες,  $C_1, \dots, C_m$ , με τέτοιο τρόπο έτσι ώστε το συνολικό άθροισμα των χρησιμότητων,  $\sum_{i=1}^m v(C_i)$ , να μεγιστοποιείται. Επομένως, το τελικό επιθυμητό αποτέλεσμα είναι ο εντοπισμός εκείνης τον πιο «πολύτιμο» συνδυασμό συμμαχιών επί του συνόλου  $Ag$ .

Τα προβλήματα που απαιτούν διαίρεση του συνόλου των οντοτήτων που συμμετέχουν σε αυτά, εντοπίζονται σε ένα μεγάλο εύρος περιοχών πρακτικών εφαρμογών, όπως στη διαχείριση παραδόσεων και παραλαβών (delivery management), στον προγραμματισμό (scheduling), στη δρομολόγηση (routing) ή σε προβλήματα στα οποία επιθυμούμε να διασφαλίσουμε πως κάθε πελάτης εξυπηρετείται αποκλειστικά από μία τοποθεσία ή άτομο. Κοινώς γνωστά προβλήματα που ανήκουν σε αυτές τις κατηγορίες είναι ο προγραμματισμός των πτήσεων μιας αεροπορικής, κατά τον οποίο κάθε πτήση μιας αεροπορικής εταιρείας πρέπει να ανατεθεί σε ένα και μόνο πλήρωμα. Επίσης, δημογραφικά προβλήματα, κατά τα οποία κάθε περιοχή είναι χωρισμένη σε πολλά εκλογικά κέντρα και κάθε πολίτης είναι εγγεγραμμένος σε ένα και μόνο τέτοιο κέντρο. Πρόσφατα, το CSG έγινε κύριο θέμα μελέτης και στο χώρο της Τεχνητής Νοημοσύνης και των πολυπρακτορικών συστημάτων, ως εργαλείο των αυτόνομων πρακτόρων για τη διαμόρφωση αποτελεσματικών ομάδων. Για παράδειγμα, στο ηλεκτρονικό εμπόριο πράκτορες-αγοραστές συγκεντρώνουν τις απαιτήσεις τους στο ίδιο μέρος με σκοπό να εξασφαλίσουν μαζικές εκπτώσεις. Στις ηλεκτρονικές επιχειρήσεις, μπορούν να σχηματιστούν συμμαχίες με σκοπό την ικανοποίηση συγκεκριμένων τομέων της αγοράς, καθώς ανταποκρίνονται σε ποικίλες παραγγελίες. Ακόμη, στην κατανεμημένη δρομολόγηση οχημάτων, οι συμμαχίες μεταξύ των επιχειρήσεων παραλαβής και παράδοσης, μπορούν να περιορίσουν το κόστος μεταφοράς μοιράζοντας τα δρομολόγια. Άλλες σημαντικές εφαρμογές του CSG περιλαμβάνουν συλλογή πληροφοριών κατά την οποία πολλοί εξυπηρετητές πληροφορίας συνεργάζονται για να ανταποκρίνονται σε αιτήματα και δίκτυα πολλών αισθητήρων στα οποία οι αισθητήρες σχηματίζουν δυναμικές συμμαχίες σε σενάρια παρακολούθησης ευρείας περιοχής.

Κατά καιρούς, έχουν αναπτυχθεί διάφοροι αλγόριθμοι για την επίλυση του προβλήματος της παραγωγής δομής συμμαχίας. Ο Sandholm et al. (1999) πρότεινε έναν αλγόριθμο οποιουδήποτε χρόνου με εγγυήσεις χειρότερης περίπτωσης. Ωστόσο, η εύρεση της βέλτιστης δομής συμμαχίας, ολοκληρώνεται μετά τον έλεγχο όλων των πιθανών συνδυασμό συμμαχιών. Με αυτόν τον τρόπο, η πολυπλοκότητα ενός τέτοιου αλγορίθμου είναι  $O(n^n)$ , όπου με  $n$  δηλώνεται ο αριθμός των πρακτόρων. Από την άλλη, οι αλγόριθμοι που βασίζονται στο δυναμικό προγραμματισμό (Dynamic Programming-DP) εξασφαλίζουν την εύρεση της καλύτερης λύσης με πολυπλοκότητα  $O(3^n)$ . Επιπλέον, οι Shehory και Kraus (1998) πρότειναν έναν άπληστο αλγόριθμο, ο οποίος θέτει περιορισμούς στο πιθανό μέγεθος των συμμαχιών.

Αδιαμφισβήτητα, ο αποδοτικότερος αλγόριθμος που έχει προταθεί μέχρι σήμερα είναι ο αλγόριθμος των ακέραιων διαμερίσεων (Integer Partition-IP Algorithm) από τον Talal Rahwan. Πρόκειται και αυτός για έναν αλγόριθμο οποιουδήποτε χρόνου, ο οποίος διχοτομεί τον χώρο έρευνας σε τμήματα, βασισμένα σε ακέραιες διαμερίσεις. Στη θεωρία των αριθμών, μία διαμέριση ενός

θετικού ακέραιου αριθμού  $n$ , ή αλλιώς ακέραια διαμέριση, είναι ένας τρόπος να αποτυπώσουμε τον  $n$  ως άθροισμα θετικών ακεραίων. Για παράδειγμα, ο αριθμός 4 μπορεί να διαμεριστεί με πέντε διαφορετικούς τρόπους: 4, 3+1, 2+2, 2+1+1, 1+1+1+1. Παρατηρείται πως αλλαγή στη σειρά των αριθμών σε μία διαμέριση οδηγεί στην ίδια διαμέριση, 3+1=1+3. Στην ενότητα που ακολουθεί γίνεται αναλυτική παρουσίαση και επεξήγηση αυτού του αλγορίθμου.

## 5.2 ΑΛΓΟΡΙΘΜΟΣ ΑΚΕΡΑΙΩΝ ΔΙΑΜΕΡΙΣΕΩΝ (IP ALGORITHM)

### 5.2.1 ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ ΧΩΡΟΥ ΑΝΑΖΗΤΗΣΗΣ

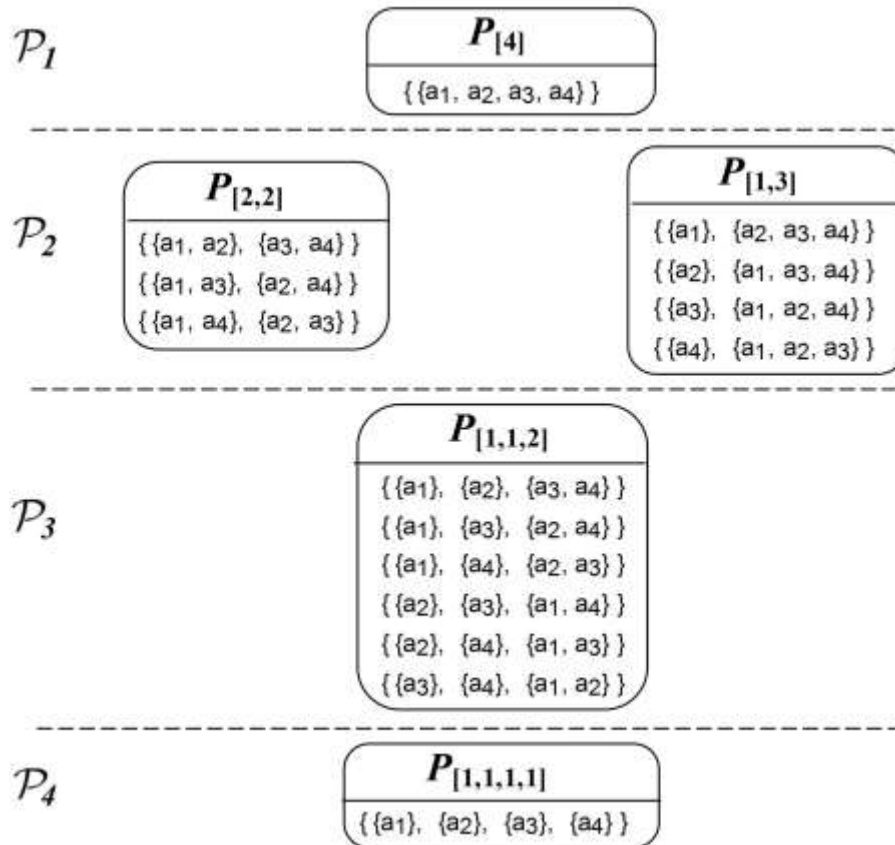
Στην ενότητα αυτή, περιγράφεται η αναπαράσταση του χώρου αναζήτησης (search space), δηλαδή του χώρου των πιθανών δομών συμμαχίας. Οι περισσότεροι υπάρχοντες αλγόριθμοι αυτή τη στιγμή αναπαριστούν τον χώρο αυτό με ένα η κατευθυνόμενο γράφο, οι κόμβοι του οποίου αποτυπώνουν δομές συμμαχίας. Η αναπαράσταση αυτή ωστόσο, οδηγεί στην στον έλεγχο όλων των πιθανών λύσεων έτσι ώστε να εξασφαλίσει τον εντοπισμό της βέλτιστης λύσης. Ο αλγόριθμος, που αναλύεται, παρουσιάζει ως ιδανική αναπαράσταση του χώρου αναζήτησης, μία που θα επιτρέπει τον υπολογισμό λύσεων οποιαδήποτε στιγμή, ενώ ταυτόχρονα ορίζει όρια στην ποιότητά τους και επιτρέπει τον περιορισμό του χώρου για την επιτάχυνση της αναζήτησης. Συγκεκριμένα, αυτός ο τρόπος αναπαράστασης υποστηρίζει μια αποδοτική αναζήτηση για πολλούς λόγους. Πρώτον, διχοτομεί τον χώρο αναζήτησης σε μικρότερους, ανεξάρτητους υποχώρους, για καθένα από τους οποίους μπορούμε να υπολογίσουμε άνω και κάτω όρια, και με αυτόν τον τρόπο να βρούμε ένα όριο για τις λύσεις που εντοπίζονται κατά τη διάρκεια της αναζήτησης. Δεύτερον, μπορούμε να περικόψουμε τους περισσότερους από αυτούς τους υποχώρους, καθώς είμαστε σε θέση να αναγνωρίσουμε αυτούς, που δεν υπάρχει πιθανότητα να περιέχουν μία λύση καλύτερη από αυτήν που έχει βρεθεί μέχρι εκείνο το σημείο. Τρίτον, αφού αυτή η αναπαράσταση προκαθορίζει το μέγεθος των συμμαχιών κάθε υποχώρου, οι πράκτορες μπορούν να ισορροπήσουν την προτίμηση τους προς συμμαχίες συγκεκριμένου μεγέθους με το κόστος του υπολογισμού της λύσης για αυτούς τους υποχώρους. Ακολουθούν ο τρόπος διχοτόμησης του χώρου αναζήτησης, η περιγραφή των αλγεβρικών ιδιοτήτων και ο τρόπος υπολογισμού ορίων χειρότερης περίπτωσης για την ποιότητα της λύσης που η αναπαράσταση αυτή μας επιτρέπει να παράγουμε.

### 5.2.2 ΔΙΧΟΤΟΜΗΣΗ ΤΟΥ ΧΩΡΟΥ ΑΝΑΖΗΤΗΣΗΣ

Διχοτομούμε το χώρο αναζήτησης  $\mathcal{P}$ , ορίζοντας υποχώρους που περιέχουν δομές συμμαχίας παρόμοιες σύμφωνα με κάποιο κριτήριο. Το συγκεκριμένο αυτό κριτήριο βασίζεται στις ακέραιες διαμερίσεις του πλήθους των πρακτόρων που συμμετέχουν στη διαδικασία. Υπενθυμίζουμε πως μία ακέραια διαμέριση ενός αριθμού  $n$ , είναι ένα σύνολο θετικών ακεραίων το άθροισμα των οποίων ισούται με τον αριθμό  $n$ . Για παράδειγμα, δεδομένου  $n = 4$ , οι πέντε ξεχωριστές διαμερίσεις είναι:  $[4]$ ,  $[3, 1]$ ,  $[2, 2]$ ,  $[2, 1, 1]$  και  $[1, 1, 1, 1]$ . Μπορεί πολύ εύκολα να φανεί πως οι διαφορετικοί τρόποι διχοτόμησης ενός συνόλου  $n$  στοιχείων να απεικονιστούν άμεσα με τις ακέραιες διαμερίσεις του  $n$ , όπου τα μέρη τα ακέραιας διαμέρισης να αντιστοιχούν στα μεγέθη των συμμαχιών μέσα σε μία δομή συμμαχίας. Για παράδειγμα, οι δομές συμμαχίας  $\{\{a_1, a_2\}, \{a_3\}, \{a_4\}\}$  και  $\{\{a_4, a_1\}, \{a_2\}, \{a_3\}\}$  μπορούν να απεικονιστούν στην ακέραια διαμέριση  $[2, 1, 1]$ , καθώς και οι δύο περιέχουν μία συμμαχία μεγέθους 2 και δύο συμμαχίες μεγέθους 1. Ορίζουμε την παραπάνω απεικόνιση με τη συνάρτηση  $F: \mathcal{P} \rightarrow \mathcal{G}$ , όπου  $\mathcal{G}$  είναι το σύνολο των ακέραιων διαμερίσεων του  $n$ . Με τον τρόπο αυτό, η  $F$  ορίζει μία σχέση ισοδυναμίας  $\sim$  στο  $\mathcal{P}$  έτσι ώστε  $CS \sim CS''$ , όπου  $CS$  και  $CS''$  δύο διαφορετικές δομές συμμαχίας, αν και μόνο  $F(CS) = F(CS'')$ , δηλαδή τα μεγέθη των συμμαχιών στο  $CS$  είναι τα ίδια με αυτών στο  $CS''$ . Σύμφωνα με τον τρόπο αυτό, η αντίστροφη εικόνα μιας ακέραιας διαμέρισης  $G$ , που συμβολίζεται με  $\mathcal{P}_G = F^{-1}[\{G\}]$ , περιέχει όλες τις δομές συμμαχίας, που αντιστοιχούν στο στην ίδια ακέραια διαμέριση  $G$ . Κάθε τέτοια αντίστροφη εικόνα αναπαριστά έναν υπόχωρο του αρχικού χώρου αναζήτησης. Αυτό υποδηλώνει πως το πλήθος των υποχώρων με αυτήν την αναπαράσταση είναι ίσο με το πλήθος των πιθανών ακέραιων διαμερίσεων, το οποίο αυξάνεται εκθετικά με τον αριθμό των πρακτόρων  $n$ .

Κατηγοριοποιούμε τους υποχώρους σε επίπεδα, με βάση τον αριθμό των μελών μέσα στις ακέραιες διαμερίσεις. Συγκεκριμένα, το επίπεδο  $\mathcal{P}_i = \{ \mathcal{P}_G : |G| = i \}$  περιέχει όλους τους υποχώρους που αντιστοιχούν σε μία ακέραια διαμέριση με  $i$  μέλη. Στο σχήμα που ακολουθεί βλέπουμε ένα παράδειγμα για 4 πράκτορες.





Εικόνα 5: Παράδειγμα της αναπαράστασης του χώρου αναζήτησης για 4 πράκτορες (44)

### 5.2.3 ΥΠΟΛΟΓΙΣΜΟΣ ΟΡΙΩΝ ΓΙΑ ΤΟΥΣ ΥΠΟΧΩΡΟΥΣ

Για κάθε υπόχωρο  $\mathcal{P}_G$ , είναι δυνατό να υπολογίσουμε ένα άνω και κάτω φράγμα στην αξία, ή όπως αλλιώς έχουμε εξηγήσει από την αρχή της μελέτης μας, στη χρησιμότητα της καλύτερης δομής συμμαχίας που μπορεί να βρεθεί σε αυτόν. Επιπλέον, θεωρούμε ως  $L_s$  μία λίστα με τις συμμαχίες μεγέθους  $s$ , και ορίζουμε ως  $max_s$ ,  $min_s$  και  $avg_s$  τη μέγιστη, την ελάχιστη και τη μέση χρησιμότητα των συμμαχιών στη  $L_s$  αντίστοιχα. Επίσης, δεδομένης μιας ακέραιας διαμέρισης  $G$ , θεωρούμε ως  $T_G$  ως το καρτεσιανό γινόμενο των λιστών  $L_s$ :  $s \in G$ . Αυτό σημαίνει πως  $T_G = \prod_{s \in G} (L_s)^{G(s)}$ , όπου το  $G(s)$  είναι η πολλαπλότητα του  $s$  στο  $G$ . Για παράδειγμα, δεδομένου  $G = [6,3,3,3,2,2]$ , έχουμε  $T_G = (L_6)^1 \times (L_3)^3 \times (L_2)^2$ . Εδώ πρέπει να σημειωθεί πως το  $T_G$  περιέχει πολλούς συνδυασμούς συμμαχιών που θεωρούνται άκυρες δομές συμμαχίας. Αυτό οφείλεται στο γεγονός πως μερικές από τις συμμαχίες μέσα σε αυτούς τους συνδυασμούς μπορεί να επικάλύπτονται. Για παράδειγμα, το  $T_{[2,1,1]}$  περιέχει τον συνδυασμό  $\{\{a_1, a_2\}, \{a_1\}, \{a_3\}\}$ , ο οποίος δεν αποτελεί έγκυρη δομή συμμαχίας καθώς ο πράκτορας  $a_1$  εμφανίζεται σε δύο

συμμαχίες. Τώρα, αν θεωρήσουμε της χρησιμότητας το κάθε στοιχείου, δηλαδή της κάθε δομής συμμαχίας, στο  $T_G$  να είναι το άθροισμα των χρησιμοτήτων όλων των συμμαχιών σε αυτή τη δομή, τότε η μέγιστη τιμή που μπορεί να λάβει ένα στοιχείο στο  $T_G$  ορίζεται ως  $MAX_G$  και υπολογίζεται ως εξής:  $MAX_G = \sum_{s \in G} max_s \times G(s)$ . Βασιζόμενοι σε αυτό, είναι πολύ εύκολο να αποδείξουμε πως το  $MAX_G$  είναι ένα άνω φράγμα στη χρησιμότητα της καλύτερης δομής συμμαχίας στο  $\mathcal{P}_G$ .

Ομοίως, η ελάχιστη τιμή που μπορεί να λάβει ένα στοιχείο στο  $T_G$  ορίζεται ως  $MIN_G$  και υπολογίζεται ως εξής:  $MIN_G = \sum_{s \in G} min_s \times G(s)$ . Αν και αυτή η τιμή ενστικτωδώς θα μπορούσε να αποτελέσει ένα κάτω φράγμα στη χρησιμότητα της καλύτερης δομής συμμαχίας στο  $\mathcal{P}_G$ , δείχνουμε πως στην πραγματικότητα είναι πιθανό να υπολογίσουμε ένα υψηλότερο κάτω φράγμα από το  $MIN_G$ .

Πιο λεπτομερώς, ας θεωρήσουμε ως  $AVG_G$  τη μέση χρησιμότητα όλων των δομών συμμαχίας στο  $\mathcal{P}_G$ . Τότε, η τιμή  $AVG_G$  θα αποτελούσε ένα κάτω φράγμα στην αξία της καλύτερης δομής συμμαχίας στο  $\mathcal{P}_G$ , καθώς μία μέση τιμή είναι πάντα μεγαλύτερη ή ίση από την ελάχιστη τιμή. Το σημαντικό σε αυτό το σημείο είναι πως μπορούμε να υπολογίσουμε την τιμή  $AVG_G$ , χωρίς να ελέγξουμε κάθε δομή συμμαχίας στο  $\mathcal{P}_G$ . Αντίθετα, μπορούμε να την υπολογίσουμε πολύ απλά, προσθέτοντας τις μέσες τιμές όλων των λιστών συμμαχιών  $L_s$ , ο υπολογισμός των οποίων μπορεί να γίνει πολύ εύκολα με την είσοδο των δεδομένων του αλγορίθμου, όπου γνωρίζουμε τη χρησιμότητα κάθε δυνατής συμμαχίας. Οπότε,  $AVG_G = \sum_{i=1}^{|G|} avg_i$ .

### 5.3 ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ ΔΟΜΗΣ ΣΥΜΜΑΧΙΑΣ

Υποθέτοντας πως η χρησιμότητα κάθε συμμαχίας  $C$  δίνεται από μία χαρακτηριστική εξίσωση  $v(C) \in \mathbb{R}$  και πως η τιμή της κάθε δομής συμμαχίας δίνεται από τη συνάρτηση  $V(CS) = \sum_{C \in CS} v(C)$ , ο στόχος είναι να πραγματοποιηθεί αναζήτηση στο σύνολο των πιθανών δομών συμμαχίας, που συμβολίζεται με  $\mathcal{P}$ , με σκοπό να εντοπιστεί ένας βέλτιστος συνδυασμός, που υπολογίζεται ως εξής:

$$CS^* = \arg \max_{CS \in \mathcal{P}} V(CS)$$

Ο αλγόριθμος IP αποτελείται από τα εξής δύο βασικά στάδια:

1. Σάρωση της εισόδου με σκοπό τον υπολογισμό των φραγμάτων για κάθε υπόχωρο. Στη διάρκεια αυτού του σταδίου, μπορούμε με πολύ μικρό κόστος να βρούμε την καλύτερη δομή συμμαχίας σε συγκεκριμένους υποχώρους, να «κλαδέψουμε» άλλους υποχώρους με βάση τα άνω όριά τους και να θέσουμε ένα όριο χειρότερης περίπτωσης για την ποιότητα της καλύτερης λύσης που έχουμε βρει μέχρι αυτό το σημείο.

2. Αναζήτηση στους εναπομείναντες υποχώρους, με τεχνικές που μας επιτρέπουν να αποφύγουμε αχρείαστες συγκρίσεις μεταξύ συμμαχιών, σε περίπτωση που επικαλύπτονται, για την παραγωγή έγκυρων συνδυασμών, να αποφύγουμε τον υπολογισμό του ίδιου συνδυασμού περισσότερες από μία φορές και να εφαρμόσουμε ελέγχους για να περιορίσουμε ακόμα περισσότερο τον χώρο αναζήτησης.

Στις επόμενες ενότητες αναλύονται τα παραπάνω στάδια με περισσότερες λεπτομέρειες. Στη συνέχεια, θα χρησιμοποιείται ο συμβολισμός  $CS'$ , για να δηλώνουμε την καλύτερη δομή συμμαχίας που βρέθηκε μέχρι τώρα, και  $G' \subseteq G$ , για τις ακέραιες διαμερίσεις που αναπαριστούν τους υποχώρους, που δεν έχουν εξεταστεί.

### 5.3.1 ΣΑΡΩΣΗ ΤΗΣ ΕΙΣΟΔΟΥ

Η είσοδος του προβλήματος CSG είναι οι χρησιμότητες που συσχετίζονται με κάθε συμμαχία, δηλαδή η χαρακτηριστική εξίσωση  $v(C)$  για κάθε  $C \in 2^{Ag} \setminus \{\emptyset\}$ . Ένας τρόπος αναπαράστασης αυτής της εισόδου είναι ένας πίνακας, που περιέχει κάθε συμμαχία μαζί με τη χρησιμότητά της. Ένας άλλος τρόπος είναι να συμφωνηθεί μία ταξινόμηση των συμμαχιών και να χρησιμοποιηθεί μόνο μία λίστα που θα περιέχει μόνο τις χρησιμότητες των ταξινομημένων συμμαχιών, δηλαδή η πρώτη χρησιμότητα αντιστοιχεί στην πρώτη συμμαχία και συνεχίζουμε ομοίως. Για τις ανάγκες του παρόντος αλγορίθμου, χρησιμοποιείται η τελευταία αναπαράσταση, καθώς δεν απαιτείται διατήρηση των συμμαχιών στη μνήμη. Πιο συγκεκριμένα, υποθέτουμε πως η είσοδος του αλγορίθμου δίνεται ως εξής:  $v(L_s) \forall s \in \{1, 2, \dots, n\}$ , όπου  $v(L_s)$  είναι μία λίστα με τις χρησιμότητες όλων των συμμαχιών μεγέθους  $s$ . Επιπλέον, υποθέτουμε πως οι συμμαχίες στη λίστα  $L_s$  είναι ταξινομημένες λεξικογραφικά. Για παράδειγμα, η συμμαχία  $\{a_1, a_2, a_4\}$ , έχει τα στοιχεία της με τη σειρά που ορίζουν οι δείκτες της και βρίσκεται πάνω από τη συμμαχία  $\{a_1, a_2, a_3\}$  και κάτω από τη συμμαχία  $\{a_1, a_3, a_4\}$  στη λίστα  $L_3$ . Η ταξινόμηση αυτή μπορεί πολύ εύκολα να παραχθεί εφαρμόζοντας τις τεχνικές που χρησιμοποίησαν οι Rawhan και Jennings (2007). Στη συνέχεια, περιγράφουμε τα βήματα του αλγορίθμου μας για την διαδικασία της σάρωσης της εισόδου, όπως φαίνεται στον Αλγόριθμο 1.

Αρχικά, εξετάζουμε τη χρησιμότητα της μοναδικής συμμαχίας μεγέθους  $n$ , που είναι γνωστή από την είσοδο. Ουσιαστικά πρόκειται για τη συμμαχία που περιλαμβάνει όλους τους πράκτορες του συστήματος και τη μοναδική συμμαχία που βρίσκεται στον υπόχωρο  $P_{[n]}$ , δηλαδή το μοναδικό υπόχωρο στο επίπεδο  $\mathcal{P}_1$ . Έπειτα, σαρώνουμε τις χρησιμότητες όλων των συμμαχιών μεγέθους 1, τις μοναδιαίες συμμαχίες, και προσθέτοντας αυτές τις τιμές, παράγουμε τη χρησιμότητα της μοναδικής δομής συμμαχίας στον υπόχωρο  $P_{[1, 1, \dots, 1]}$ , που είναι και ο μοναδικός υπόχωρος στο επίπεδο  $\mathcal{P}_n$ . Στο σημείο αυτό (βήμα 1), μπορούμε να

υπολογίσουμε την καλύτερη δομή συμμαχίας ( $CS'$ ) μέχρι στιγμής μεταξύ της γενικής συμμαχίας και τα δομής που περιέχει  $n$  μοναδιαίες συμμαχίες.

Έχοντας ολοκληρώσει την αναζήτηση στα επίπεδα  $P_1$  και  $P_n$ , τώρα ο αλγόριθμος συνεχίζει με την αναζήτηση στο επίπεδο  $P_2$ , με πολύ μικρό κόστος κατά την σάρωση της εισόδου. Για το σκοπό αυτό, θεωρούμε  $G^2 = \{G \in \mathcal{G} : |G| = 2\}$  ως το σύνολο των ακέραιων διαμερίσεων, που περιέχουν δύο συμμαχίες η κάθε μία. Ως αποτέλεσμα του τρόπου ταξινόμησης της εισόδου, ο οποίος υιοθετήθηκε για λόγους σύμβασης, δύο συμπληρωματικές συμμαχίες  $C$  και  $C'$  μέσα σε μία δομή συμμαχίας  $CS = \{C, C'\}$ , είναι πάντα διαμετρικά τοποθετημένες στις λίστες  $L_C$  και  $L_{C'}$  αντίστοιχα και αυτό συμβαίνει ακόμα και αν ισχύει  $|C| = |C'|$ . Για παράδειγμα, μπορούμε να παρατηρήσουμε εύκολα πως δεδομένων 4 πρακτόρων, οι συμμαχίες  $\{a_3\}$  και  $\{a_1, a_2, a_4\}$  είναι διαμετρικά ταξινομημένες στις λίστες  $L_1$  και  $L_3$  αντίστοιχα, όπως επίσης και οι συμμαχίες  $\{a_2, a_4\}$  και  $\{a_1, a_3\}$  είναι διαμετρικά ταξινομημένες στη λίστα  $L_2$  (όπως φαίνεται στην εικόνα 6 το παράδειγμα για 4 πράκτορες).

$L_1$	$L_2$	$L_3$	$L_4$
$a_4$	$a_3, a_4$	$a_2, a_3, a_4$	$a_1, a_2, a_3, a_4$
$a_3$	$a_2, a_4$	$a_1, a_3, a_4$	
$a_2$	$a_2, a_3$	$a_1, a_2, a_4$	
$a_1$	$a_1, a_4$	$a_1, a_2, a_3$	
	$a_1, a_3$		
	$a_1, a_2$		

*Εικόνα 6:* Ένα παράδειγμα του τρόπου ταξινόμησης της εισόδου των λιστών με τις συμμαχίες για 4 πράκτορες

Βασιζόμενοι σε αυτή τη διαμετρική τοποθέτηση των συμμαχιών στις λίστες της εισόδου, για κάθε ακέραια διαμέριση  $G = [g_1, g_2] \in G^2$ , υπολογίζουμε τις χρησιμότητες όλων των δομών συμμαχίας στο  $P_G$  απλά προσθέτοντας τις χρησιμότητες των συμμαχιών καθώς σαρώνουμε τις λίστες  $v(L_{g_1})$  και  $v(L_{g_2})$ , ξεκινώντας από διαφορετικό άκρο για κάθε λίστα. Αφού έχει ολοκληρωθεί η σάρωση αυτών των λιστών (βήματα 10-24), έχουμε τη δυνατότητα να λάβουμε τις δύο χρησιμότητες για τις οποίες το άθροισμα είναι το μεγαλύτερο.

---

**Αλγόριθμος 1:** scanInput( ) – σάρωση της εισόδου, υπολογισμός αρχικής λύσης και φραγμάτων

---

---

```

Είσοδος:  $n, \{v(L_s)\}_{s \in \{1, 2, \dots, n\}}$ 
1:  $CS' \leftarrow \arg \max_{CS \in \{\{a_1, \dots, a_n\}, \{\{a_1\}, \dots, \{a_n\}\}} V(CS)$ 
2: for  $s=1$  to  $\lfloor \frac{n}{2} \rfloor$  do
3:    $\hat{s} \leftarrow n-s$ 
4:   if  $\hat{s} = s$  then /*αν εξετάζουμε την ίδια λίστα από δύο κατευθύνσεις*/
5:      $end = \lfloor \frac{|v(L_s)|}{2} \rfloor$ 
6:   else
7:      $end = |v(L_s)|$ 
8:   end if
9:   Θέτουμε τα  $max_s, max_{\hat{s}}, v_{max}$  ίσα με  $-\infty$  και τα  $sum_s, sum_{\hat{s}}$  ίσα με  $\emptyset$ 
10:  for  $x=1$  to  $end$  do /*εξετάζουμε τις λίστες  $v(L_s)$  και  $v(L_{\hat{s}})$ */
11:     $\hat{x} \leftarrow |v(L_s)| - x + 1$ 
12:     $v \leftarrow v(L_s)^x, \hat{v} \leftarrow v(L_{\hat{s}})^{\hat{x}}$  /*συλλέγουμε τα στοιχεία στις θέσεις  $x$ 
    και  $\hat{x}$  από τις λίστες  $v(L_s)$  και  $v(L_{\hat{s}})$ */
13:    if  $v_{max} < v + \hat{v}$  then
14:       $v_{max} \leftarrow v + \hat{v}$ 
15:       $x_{max} = x$  /*καταχωρούμε τη θέση στη  $v(L_s)$  που είναι το  $v$ 
16:    end if
17:    if  $max_s < v$  then
18:       $max_s \leftarrow v$  /*καταχωρούμε τη μέγιστη τιμή στη  $v(L_s)$ */
19:    end if
20:    if  $max_{\hat{s}} < \hat{v}$  then
21:       $max_{\hat{s}} \leftarrow \hat{v}$ 
22:    end if
23:     $sum_s \leftarrow sum_s + v, sum_{\hat{s}} \leftarrow sum_{\hat{s}} + \hat{v}$ 
24:  end for
25:   $\hat{x}_{max} \leftarrow |v(L_s)| - x_{max} + 1$ 
26:  if  $V(CS') < v(\{L_s^{x_{max}}, L_{\hat{s}}^{\hat{x}_{max}}\})$  then
27:     $CS' \leftarrow \{L_s^{x_{max}}, L_{\hat{s}}^{\hat{x}_{max}}\}$  /*ενημερώνουμε την καλύτερη δομή ως
    τώρα*/
28:  end if
29:   $avg_s \leftarrow sum_s / |v(L_s)|, avg_{\hat{s}} \leftarrow sum_{\hat{s}} / |v(L_{\hat{s}})|$ 
30: end for
31:  $G' \leftarrow G \setminus G^2$  /*αφαιρώ τον υπόχωρο  $G^2$  γιατί ελέγχθηκε/*
32: for  $G \in G'$  do
33:    $MAX_G \leftarrow \sum_{s \in G} max_s \times G(s)$ . /*Υπολογισμός των άνω και των κάτω
    ορίων
34:    $AVG_G \leftarrow \sum_{s \in G} avg_s \times G(s)$ . /*για κάθε υπόχωρο στο σύνολο  $G'$ 
35: end for
36:  $UB^* \leftarrow \max[V(CS'), \max_{G \in G'} [MAX_G]]$ 
37:  $LB^* \leftarrow \max[V(CS'), \max_{G \in G'} [AVG_G]]$ 
38:  $G' \leftarrow \text{prune}(G', \{MAX_G\}_{G \in G'}, LB^*)$  /*αφαιρούμε τους υποχώρους που έχουν
    άνω όριο μικρότερο από το  $LB^*$ */
39:  $\beta \leftarrow \min[\frac{n}{2}, \frac{UB^*}{V(CS^*)}]$ 
40: return  $CS', \beta, \{max_s\}_{s \in \{1, \dots, n\}}, G', \{MAX_G\}_{G \in G'}, \{AVG_G\}_{G \in G'}$ 

```

---

Επιπλέον, μπορούμε να καταχωρήσουμε τις θέσεις στις λίστες, στις οποίες εμφανίζονται αυτές οι δύο χρησιμότητες (βλέπουμε πως υπολογίζονται οι θέσεις  $x_{max}$  και  $\hat{x}_{max}$  στα βήματα 15 και 25 αντίστοιχα). Έχοντας καταχωρήσει αυτές τις τοποθεσίες, πλέον γνωρίζουμε ακριβώς τις θέσεις στις λίστες  $L_{g1}$  και  $L_{g2}$ , που βρίσκονται οι δύο συμμαχίες που ανήκουν στην καλύτερη δομή συμμαχίας στο επίπεδο  $\mathcal{P}_{[g1,g2]}$  (αυτό απορρέει από το γεγονός πως η θέση κάθε χρησιμότητας στη λίστα  $v(L_s)$ :  $s \in \{1, \dots, n\}$  είναι ακριβώς η θέση της αντίστοιχης συμμαχίας στη λίστα  $L_s$ . Επομένως, χρησιμοποιούμε αυτές τις δύο θέσεις για να συνθέσουμε την καλύτερη δομή συμμαχίας που εντοπίστηκε στον υπόχωρο προς αναζήτηση, ο οποίος περιλαμβάνει τους συνδυασμούς 2 συμμαχιών, και αφού συγκρίνουμε τη χρησιμότητά της με αυτή της δομής που θεωρήθηκε καλύτερη μέχρι τώρα (βήμα 2) ανανεώνουμε κατάλληλα τη μεταβλητή  $CS'$  (βήματα 26 και 27).

Κατά τη διάρκεια της σάρωσης των λιστών  $v(L_{g1})$  και  $v(L_{g2})$ , υπολογίζουμε ταυτόχρονα τις τιμές  $max_{g1}$  και  $max_{g2}$  (βήματα 17-22), καθώς επίσης και τις τιμές  $avg_{g1}$  και  $avg_{g2}$  (βήμα 29). Αφού υπολογίσαμε τις παραπάνω τιμές για κάθε μέγεθος συμμαχίας  $s$ , μπορούμε τώρα να υπολογίσουμε τα άνω και κάτω φράγματα για κάθε υπόχωρο (βήματα 32-34). Χρησιμοποιώντας αυτά τα όρια, έχουμε τη δυνατότητα να ορίσουμε ένα άνω φράγμα  $UB^*$  και ένα κάτω φράγμα  $LB^*$  στη χρησιμότητα της καλύτερης δομής συμμαχίας (βήματα 36 και 37). Γνωρίζοντας αυτές τις τιμές, κάθε υπόχωρος  $\mathcal{P}_G$  που έχει άνω όριο  $MAX_G < LB^*$  μπορεί να αφαιρεθεί από το χώρο αναζήτησης, καθώς δεν υπάρχει περίπτωση να περιέχει δομή συμμαχίας καλύτερη από αυτήν που έχει βρεθεί μέχρι στιγμής. Η συνάρτηση της αφαίρεσης υποχώρων (βήμα 38), υλοποιείται στον Αλγόριθμο 2.

---

**Αλγόριθμος 2:**  $prune(G', \{MAX_G\}_{G \in \mathcal{G}}, LB^*)$  – αφαίρεση υποχώρων

---

```

1: for  $G \in \mathcal{G}'$  do
2:   if  $MAX_G \leq LB^*$  /*αν το άνω όριο του  $\mathcal{P}_G$  είναι μικρότερο από το  $LB^*$ */
3:      $G' \leftarrow G' \setminus G$  /*αφαίρεση του υποχώρου  $G^*$ */
4:   end if
5: end for
6: return  $G'$ 

```

---

Ένα άλλο πλεονέκτημα της διαδικασίας σάρωσης ης εισόδου είναι πως μας επιτρέπει να υπολογίσουμε ένα φράγμα χειρότερης περίπτωσης  $\beta$  στη χρησιμότητα της  $CS'$  ως εξής:  $\beta = \min \left[ \frac{n}{2}, \frac{UB^*}{V(CS')} \right]$  (βήμα 39). Αυτό προκύπτει από την απόδειξη του Sandholm et al. (1999), πως η χρησιμότητα της καλύτερης δομής συμμαχίας στα επίπεδα  $\mathcal{P}_1, \mathcal{P}_2$  και  $\mathcal{P}_n$  βρίσκεται εντός ενός ορίου  $\frac{n}{2}$  από τη βέλτιστη.

Μέχρι αυτό το σημείο, σαρώνοντας απλώς την είσοδο, έχουμε υπολογίσει τις τιμές  $max_s$  και  $avg_s$  για όλα τα  $s \in \{1, \dots, n\}$ , έχουμε ολοκληρώσει την αναζήτηση για τη  $CS'$  στα επίπεδα  $\mathcal{P}_1, \mathcal{P}_2$  και  $\mathcal{P}_n$ , έχουμε υπολογίσει τις τιμές  $MAX_G$  και  $AVG_G$  για

κάθε υπόχωρο στα εναπομείναντα επίπεδα ( $P_3, \dots, P_{n-1}$ ), έχουμε αφαιρέσει κάποιους από αυτούς τους χώρους, και ορίσαμε ένα φράγμα χειρότερης περίπτωσης  $\beta$  για την ποιότητα του καλύτερου συνδυασμού  $CS'$ , που βρέθηκε μέχρι αυτή τη στιγμή. Ακόμη, υπάρχει η δυνατότητα να καθορίσουμε ένα όριο  $\beta^* \geq 1$  εντός του οποίου κάθε λύση είναι αποδεκτή. Συγκεκριμένα, αν η καλύτερη λύση που έχει βρεθεί μέχρι αυτή τη στιγμή βρίσκεται εντός αυτού του καθορισμένου ορίου, δηλαδή  $\beta \leq \beta^*$ , τότε η αναζήτηση μπορεί να διακοπεί. Σε αντίθετη περίπτωση, πρέπει να γίνει έλεγχος και στους υποχώρους που δεν έχουν ακόμα αφαιρεθεί από το χώρο αναζήτησης.

### 5.3.2 ΕΠΙΛΟΓΗ ΚΑΙ ΑΝΑΖΗΤΗΣΗ ΕΝΟΣ ΥΠΟΧΩΡΟΥ

Έχοντας στη διάθεση μας το σύνολο των υπόλοιπων χώρων προς αναζήτηση μετά την περάτωση της διαδικασίας της σάρωσης της εισόδου, επιλέγουμε έναν υπόχωρο και βρίσκουμε τον καλύτερο συνδυασμό συμμαχιών σε αυτόν. Ύστερα από αυτό αφαιρούμε όλους τους εναπομείναντες υποχώρους που έχουν άνω όριο μικρότερο από την καλύτερη τιμή που βρήκαμε μέχρι στιγμής. Η διαδικασία της επιλογής, αναζήτησης και αφαίρεσης επαναλαμβάνεται μέχρι να ικανοποιηθεί μία από τις παρακάτω συνθήκες τερματισμού:

- Η χρησιμότητα του καλύτερου συνδυασμού συμμαχιών που έχει βρεθεί μέχρι τώρα βρίσκεται εντός του καθορισμένου ορίου  $\beta^*$
- Όλοι οι υπόλοιποι υπόχωροι που έχουν παραμείνει στο χώρο αναζήτησης έχουν εξεταστεί ή αφαιρεθεί

Αυτό μπορεί να φανεί στον Αλγόριθμο 3, σύμφωνα με τον οποίο επιλέγεται μία ακέραια διαμέριση προς αναζήτηση, δηλαδή ο αντίστοιχος υπόχωρος που συμβολίζεται ως  $P_G$  (βήμα 2). Αφού ολοκληρωθεί ο έλεγχος στον υπόχωρο αυτόν (βήμα 3), αφαιρείται από το σύνολο του χώρου αναζήτησης (βήμα 4). Μετά από αυτό, ελέγχουμε αν η βέλτιστη λύση  $CS'$  έχει τροποποιηθεί κατά τη διάρκεια της τελευταίας αναζήτησης (βήμα 5), και αν ισχύει αυτό, τότε κάθε υπόχωρος με άνω όριο μικρότερο από την τιμή  $V(CS')$ , αφαιρείται (βήμα 6).

Μπορούμε να ελέγξουμε αν το  $CS'$  ανήκει στο  $P_G$ , πολύ εύκολα εξετάζοντας αν τα μεγέθη των συμμαχιών στο  $CS'$ , ανταποκρίνονται στα μεγέθη των μελών του  $G$ . Στη συνέχεια οι τιμές  $UB^*$  και  $\beta$  ενημερώνονται (βήματα 8 και 9 αντίστοιχα) και αν η τρέχουσα βέλτιστη λύση βρίσκεται εντός των ορίων που θέτει η τιμή  $\beta^*$  τότε αυτή επιστρέφεται και η αναζήτηση τερματίζεται (βήματα 10 και 11). Σε αντίθετη περίπτωση, ολόκληρη η διαδικασία που μόλις εξηγήθηκε επαναλαμβάνεται με την επιλογή ως  $G$  του επόμενου υποχώρου (αν υπάρχει άλλος).

---

**Αλγόριθμος 3:** searchRemainingSpace() – έλεγχος και αφαίρεση υπόλοιπων υποχώρων

---

**Απαιτήσεις:**  $G'$ ,  $\{MAX_G\}_{G \in G'}$ ,  $A$ ,  $\beta^*$

```
1: while  $G' \neq \emptyset$  do
2:   Select  $G''$  /*επιλέγουμε την ακέραια διαμέριση που αναπαριστά τον
           τον επόμενο υπόχωρο για αναζήτηση*/
3:    $CS' \leftarrow$  searchList ( $G''$ , 1, 1,  $A$ ,  $CS'$ ,  $\overline{CS}$ ) /*αναζήτηση στο
           /*χώρο  $P_{G''}$  και ενημέρωση της  $CS'$ */
4:    $G' \leftarrow G' \setminus G''$  /*αφαίρεση του  $P_{G''}$  από το χώρο αναζήτησης που έμεινε*/
5:   if  $CS' \in P_{G''}$ , then /*αν η  $CS'$  άλλαξε κατά την αναζήτηση στο  $P_{G''}$ */
6:      $G' \leftarrow$  prune ( $G'$ ,  $\{MAX_G\}_{G \in G'}$ ,  $V(CS')$ )
           /*αφαίρεση των υποχώρων με άνω όρια μικρότερα από το  $V(CS')$ */
7:   end if
8:    $UB^* \leftarrow$  max [ $V(CS')$ , max $_{G \in G'} [MAX_G]$ ] /*ενημέρωση του άνω φράγματος
           στη χρησιμότητα της βέλτιστης δομής συμμαχίας*/
9:    $\beta \leftarrow$  min [ $\frac{UB^*}{V(CS')}$ ,  $\beta$ ] /*ενημέρωση του ορίου χειρότερης περίπτωσης*/
10:  if  $\beta \leq \beta^*$  then /*αν το  $CS'$  είναι εντός του ορίου*/
11:    return  $CS'$ 
12:  end if
13: end while
14: return  $CS'$ 
```

---

### 5.3.2.1 ΕΠΙΛΟΓΗ ΕΝΟΣ ΥΠΟΧΩΡΟΥ

Είναι εύκολα κατανοητό, πως αν δεν εξεταστούν οι υπόχωροι, ο οποίοι έχουν άνω φράγμα μεγαλύτερο από την τιμή  $V(CS')$ , δεν μπορούμε να εξασφαλίσουμε πως η  $CS'$  είναι όντως μία βέλτιστη λύση. Αυτό υπονοεί πως η τιμή  $\beta$  πρέπει να διατηρείται μεγαλύτερη από 1 μέχρι να ολοκληρωθεί η αναζήτηση και στους υπόλοιπους υποχώρους:  $\{ P_G : MAX_G \geq V(CS') \}$ . Αυτό μπορεί να επιτευχθεί επιλέγοντας τον επόμενο υπόχωρο προς αναζήτηση χρησιμοποιώντας τον ακόλουθο κανόνα:

$$\text{Select } G = \arg \max_{G \in G'} (MAX_G)$$

Ως αποτέλεσμα αυτής της τεχνικής, όλοι οι υπόχωροι με άνω φράγμα μικρότερο από  $V(CS')$  δεν θα εξεταστούν, και το κέρδος που αποκομίζουμε είναι ένας πιθανός περιορισμός του χώρου αναζήτησης. Ένα άλλο αποτέλεσμα είναι πως πάντα θα είναι ωφέλιμο η αναζήτηση ενός υποχώρου, ακόμα και αν αυτός δεν περιέχει καλύτερη λύση από αυτήν που έχει ονομαστεί μέχρι τη δεδομένη στιγμή ως βέλτιστη. Αυτό οφείλεται στο γεγονός πως η παραπάνω τακτική επιλογής εξασφαλίζει πως η τιμή  $UB^*$  μειώνεται κάθε φορά που ένας υπόχωρος εξετάζεται με αποτέλεσμα να



βελτιώνεται η τιμή  $\beta$  για την χειρότερη περίπτωση στην ποιότητα της τρέχουσας καλύτερης λύσης.

Ένα άλλο πλεονέκτημα της δυνατότητας να ελέγχουμε ποιοι υποχώροι θα περάσουν από τη διαδικασία της αναζήτησης είναι πως οι πράκτορες μπορούν να επιλέγουν τον τύπο των δομών συμμαχιών που διαμορφώνουν σύμφωνα με τους υπολογιστικούς τους πόρους ή τις προσωπικές τους προτιμήσεις. Για παράδειγμα, έχει συζητηθεί στο παρελθόν πως ο χρόνος υπολογισμού μπορεί να μειωθεί περιορίζοντας το μέγεθος των συμμαχιών που μπορούν να σχηματιστούν. Ωστόσο, αυτός είναι ένας άκρως δαπανηρός και αυθαίρετος περιορισμός, καθώς πιθανώς να αγνοείται ένας σημαντικός αριθμός αποδοτικών λύσεων. Αντίθετα, με τη χρήση του αλγορίθμου IP, υπάρχει η δυνατότητα, πριν την πραγματοποίηση της αναζήτησης, να καθοριστεί για ποιους υποχώρους υπάρχουν μεγαλύτερες προοπτικές εύρεσης βέλτιστης λύσης σύμφωνα με τα άνω και τα κάτω φράγματά τους. Με τον τρόπο αυτό ο χρόνος για τους υπολογισμούς εστιάζεται σε αυτά τα τμήματα του συνολικού χώρου αναζήτησης και τα οφέλη εξισορροπούν τον χρόνο που απαιτείται.

### 5.3.2.2 ΑΝΑΖΗΤΗΣΗ ΣΕ ΕΝΑΝ ΥΠΟΧΩΡΟ

Δεδομένης μιας ακέραιας διαμέρισης  $G = [g_1, g_2, \dots, g_{|G|}] \in \mathcal{G}$ , χρειάζεται να διατρέξουμε όλες τις δομές συμμαχιών που ανήκουν στο  $\mathcal{P}_G$  με σκοπό να εντοπιστεί η βέλτιστη μεταξύ αυτών. Χωρίς απώλεια της γενίκευσης θεωρούμε πως  $g_1 \leq g_2 \leq g_{|G|}$ . Χρησιμοποιείται μία μεταβλητή  $\vec{CS} = \langle C_1, C_2, \dots, C_{|G|} \rangle$  για να διατρέχουμε όλους τους συνδυασμούς συμμαχιών στο  $\mathcal{P}_G$  όπως περιγράφεται στη συνέχεια. Αρχικά, το  $C_1$  αναθέεται σε μία συμμαχία της λίστας  $L_{g_1}$ . Έπειτα, το  $C_2$  χρησιμοποιείται για να διατρέξει τη λίστα  $L_{g_2}$  μέχρι να βρεθεί μία συμμαχία που δεν επικαλύπτεται με τη  $C_1$ . Το  $C_3$  χρησιμοποιείται για να διατρέξει τη λίστα  $L_{g_3}$  μέχρι να βρεθεί μία συμμαχία που δεν επικαλύπτεται με τη  $\{C_1, C_2\}$ . Η διαδικασία αυτή επαναλαμβάνεται μέχρι κάθε  $C_k \in \vec{CS}$  να ανατεθεί σε μία συμμαχία της λίστας  $L_{g_k}$ . Με αυτόν τον τρόπο το  $\vec{CS}$  θα είναι μία έγκυρη δομή συμμαχιών, που ανήκει στο  $\mathcal{P}_G$ . Η χρησιμότητα αυτής της δομής υπολογίζεται και συγκρίνεται με την μέγιστη τιμή που έχει βρεθεί μέχρι τη δεδομένη χρονική στιγμή. Στη συνέχεια, οι συμμαχίες στο  $\vec{CS}$  ενημερώνονται έτσι ώστε να θέσουμε το  $\vec{CS}$  σε άλλο συνδυασμό συμμαχιών μέσα στο  $\mathcal{P}_G$ . Στο σημείο αυτό πρέπει να τονιστεί πως μία συμμαχία  $C_k$  ανανεώνεται μόνο όταν έχουν εξεταστεί όλες οι πιθανές περιπτώσεις για τις συμμαχίες  $C_{k+1}, \dots, C_{|G|}$ , που δεν επικαλύπτονται με τις συμμαχίες  $\{C_1, \dots, C_k\}$ . Αυτό εξασφαλίζει πως κάθε δυνατός συνδυασμός συμμαχιών στο  $\mathcal{P}_G$  έχει εξεταστεί.

Στη συνέχεια παρουσιάζουμε πως η παραπάνω διαδικασία μπορεί να πραγματοποιηθεί χωρίς την αποθήκευση των λιστών  $L_{g_1}, L_{g_2}, \dots, L_{g_{|G|}}$ . Θεωρούμε

ως  $LC_{g_k}^n : 1 \leq g_k \leq n$  τη λίστα των συνδυασμών μεγέθους  $g_k$ , που λαμβάνουμε από το σύνολο  $\{1, 2, \dots, n\}$  και στην οποία οι συνδυασμοί είναι ταξινομημένοι λεξικογραφικά. Με βάση αυτό, και οι δύο λίστες που διαθέτουμε  $LC_{g_k}^n$  και  $L_{g_k}$  περιέχουν τα υποσύνολα του μεγέθους  $g_k$ , όπως προκύπτουν από ένα σύνολο μεγέθους  $n$ . Η μόνη διαφορά μεταξύ των δύο λιστών βρίσκεται στο γεγονός πως η πρώτη πρόκειται για μία λίστα συνδυασμών αριθμών, ενώ η δεύτερη για μία λίστα συμμαχιών μεταξύ πρακτόρων. Για την ανάγκη αυτή οι Rawhan και Jennings (2007), παρουσίασαν τον τρόπο με τον οποίο διατρέχουμε τους συνδυασμούς στη λίστα  $LC_{g_k}^n$ , χωρίς να χρειάζεται να αποθηκεύουμε ολόκληρη τη λίστα στη μνήμη. Αντίθετα, μόνο ένας συνδυασμός αποθηκεύεται κάθε φορά. Αυτό βασίζεται στην υπόθεση πως η λίστα είναι ταξινομημένη με τέτοιο τρόπο ώστε ο τελευταίος συνδυασμός της λίστας  $LC_{g_k}^n$  είναι πάντα ο  $\{1, 2, \dots, g_k\}$ . Αυτή η ταξινόμηση σημαίνει πως δεδομένου ενός οποιουδήποτε συνδυασμού στη θέση  $x$  της λίστας, όπου  $1 < x \leq LC_{g_k}^n$ , είναι δυνατό να υπολογίσουμε το συνδυασμό που βρίσκεται στη θέση  $x - 1$ . Για να προσπελάσουμε όλες τις συμμαχίες στη λίστα  $L_{g_k}$ , χρησιμοποιούμε μία μεταβλητή  $M_k$  για να διατρέχουμε όλους τους συνδυασμούς στη λίστα  $LC_{g_k}^n$  και για κάθε περίπτωση του  $M_k$ , εξάγουμε την αντίστοιχη συμμαχία  $C_k \in L_{g_k}$  ακολουθώντας την παρακάτω σχέση:

$$C_k = \{a_i \in Ag \mid i \in M_k\}$$

Για παράδειγμα, δεδομένης της τιμής  $M_k = \{2, 4, 5\}$ , η αντίστοιχη συμμαχία θα ήταν  $\{a_2, a_4, a_5\}$ . Καθώς υπάρχει μία άμεση αντιστοίχιση για κάθε συνδυασμό στη λίστα  $LC_{g_k}^n$  σε μία συμμαχία στη λίστα  $L_{g_k}$ , έχοντας τη μεταβλητή  $M_k$  να διατρέχει όλους τους συνδυασμούς στην πρώτη λίστα, τότε μπορούμε να διατρέξουμε όλες τις συμμαχίες στη δεύτερη λίστα.

Η ιδέα αυτής της διαπέρασης των λιστών με μια πρώτη ματιά φαίνεται αποτελεσματική και εγγυάται την εξαγωγή της βέλτιστης δυνατής δομής συμμαχιών. Ωστόσο, παρουσιάζει κάποιους μικρούς περιορισμούς, που ενώ δεν απομακρύνουν το αποτέλεσμα από τη βέλτιστη λύση, επιδρούν αρνητικά στο χρόνο εκτέλεσης. Συγκεκριμένα, πραγματοποιεί συγκρίσεις σε μεγαλύτερο χώρο αναζήτησης από αυτόν που χρειάζεται στην πραγματικότητα παράγει σύνολα συμμαχιών που περιέχουν τις ίδιες συμμαχίες αλλά με διαφορετική σειρά, με άλλα λόγια τις ίδιες δομές συμμαχιών.

Αυτό που θα ήταν επιθυμητό είναι να βρεθεί ένας τρόπος να διατρέξουμε τις λίστες  $L_{g_1}, L_{g_2}, \dots, L_{g_{|G|}}$  με τέτοιο τρόπο ώστε να παράγονται μόνο έγκυροι συνδυασμοί. Με άλλα λόγια, θα ήταν επιθυμητό αν η μεταβλητή  $C_k$  εξέταζε μόνο τις έγκυρες συμμαχίες της λίστας  $L_{g_k}$ , παρά να διατρέχει κάθε συμμαχία της ίδια λίστας για να βεβαιωθεί πως δεν επικαλύπτεται με τις συμμαχίες  $\{C_1, \dots, C_{k-1}\}$ . Επιπλέον, με σκοπό να αποφύγουμε περιττές λειτουργίες, θα ήταν επιθυμητό αν η διαδικασία της προσπέλασης εγγυόταν την μη ύπαρξη διπλότυπων δομών

συμμαχιών. Στον αλγόριθμο 4 παρουσιάζουμε μία τέτοια διαδικασία, που ανταποκρίνεται σε αυτές τις απαιτήσεις.

Η βασική ιδέα είναι η αρχική κλήση της συνάρτησης *searchList* για την προσπέλαση της λίστας  $L_{g_1}$ . Για κάθε μία από τις συμμαχίες στη λίστα αυτή, η συνάρτηση καλείται αναδρομικά για να προσπελάσουμε και τις συμμαχίες στη λίστα  $L_{g_2}$ , που δεν επικαλύπτονται με την πρώτη συμμαχία, δηλαδή αυτή που λάβαμε από τη λίστα  $L_{g_1}$ . Ομοίως, καθώς διατρέχουμε τη λίστα  $L_{g_2}$  καλούμε και πάλι αναδρομικά τη συνάρτηση *searchList* για την προσπέλαση των συμμαχιών της επόμενης λίστας  $L_{g_3}$ , που δεν επικαλύπτονται με τις δύο πρώτες συμμαχίες. Η διαδικασία αυτή επαναλαμβάνεται μέχρι η συνάρτηση κληθεί για την προσπέλαση των συμμαχιών της λίστας  $L_{g_{|G|}}$ , οπότε και έχουμε μία έγκυρη δομή συμμαχίας που συμβολίζεται με  $\overrightarrow{CS}$  στον Αλγόριθμο 4 και ανήκει στο  $\mathcal{P}_G$ . Στη συνέχεια, αν η δομή  $\overrightarrow{CS}$  έχει χρησιμότητα μεγαλύτερη από την τιμή  $V(CS')$ , τότε η  $CS'$  ανανεώνεται ανάλογα. Στη συνέχεια της ενότητας αυτής εξηγείται πώς ο Αλγόριθμος 4 αποφεύγει την παραγωγή μη έγκυρων και περιττών δομών συμμαχιών και πώς επιταχύνει την αναζήτηση παραλείποντας να κάνει την εκτέλεση ανώφελων συγκρίσεων.

---

**Αλγόριθμος 4:** *searchList*( $G, k, a, A_k, \overrightarrow{CS'}, \overrightarrow{CS}$ ) – έλεγχος ενός υποχώρου

---

**Απαιτήσεις:**  $\{max_s\}_{s \in \{1, \dots, n\}}$ ,  $\{v(L_s)\}_{s \in \{1, 2, \dots, n\}}$ ,  $UB^*$ ,  $\beta^*$

```

1: if  $k > 1$  &&  $g_k \neq g_{k-1}$  then /*αν τα μεγέθη στο Δ δεν επαναλαμβάνονται*/
2:    $a \leftarrow 1$  /*επααναφορά του a*/
3: end if
4: for  $M_k \in LC_{g_k}^{|A_k|}$  τ.ω.  $a \leq M_{k,1} \leq n + 1 - \sum_{i=1}^k (g_i \times G(g_i))$  do
5:    $C_k \leftarrow \{A_{k,i} \mid i \in M_k\}$  /*εξαγωγή της  $C_k$  δεδομένων των  $M_k$  και  $A_k$ */
6:   if  $k = |G|$  &&  $V(CS') < V(\overrightarrow{CS})$  then
7:      $CS' \leftarrow \overrightarrow{CS}$  /*ενημέρωση της τρέχουσας καλύτερης λύσης*/
8:   else if  $V(CS') < \sum_{s \in \{g_1, \dots, g_k\}} v(C_s) + \sum_{s \in \{g_{k+1}, \dots, g_n\}} max_s$ 
/*είσοδος στη συνθήκη μόνο αν υπάρχει περίπτωση εύρεσης καλύτερης
/*δομής συμμαχιών από τη  $CS'$ */
9:      $CS' \leftarrow searchList(G, k+1, M_{k,1}, A \setminus C, CS', \overrightarrow{CS})$ 
/*συνέχεια στην επόμενη συμμαχία*/
10:  end if
11:  if  $\frac{UB^*}{V(CS')} \leq \beta^* \mid \mid V(CS') = MAX_G$  then /*τερματισμός της διαδικασίας
/* αν η ζητούμενη λύση έχει βρεθεί ή η τρέχουσα καλύτερη λύση
/* είναι ίση με το άνω φράγμα αυτού του υποχώρου*/
12:    return  $CS'$ 
13:  end if
14: end for
15: return  $CS'$ 

```

---

### Αποφυγή μη έγκυρων δομών συμμαχιών

Δεδομένης μιας ακέραιας διαμέρισης  $G = [g_1, g_2, \dots, g_{|G|}]$ , ορίζουμε την ακόλουθη ταξινόμηση συνόλων των πρακτόρων:  $A_1, A_2, \dots, A_{|G|}$ , όπου το  $A_1$  περιέχει  $n$  πράκτορες και το  $A_k : 2 \leq k \leq |G|$  περιέχει  $n - \sum_{i=1}^{k-1} g_i$  πράκτορες. Επιπλέον, υποθέτουμε πως οι πράκτορες στο  $A_k : 1 \leq k \leq |G|$  είναι ταξινομημένοι με αύξουσα σειρά με βάση τη θέση τους στο  $A$ , δηλαδή αν το  $A_k$  περιέχει τους πράκτορες  $a_5, a_7$ , και  $a_2$ , τότε η σειρά θα είναι  $a_5 = \langle a_2, a_5, a_7 \rangle$ . Με άλλα λόγια, υποθέτουμε πως:  $A_{k,1} < A_{k,2} < \dots < A_{k,|A_k|}$ , όπου  $A_{k,i}$  είναι ο  $i$ -στος πράκτορας στο  $A_k$ . Δεδομένου ενός αριθμού συμμαχιών  $C_1, C_2, \dots, C_{L_{g_k}-1}$ , που έχουν εξαχθεί από τις λίστες  $L_{g_1}, L_{g_2}, \dots, L_{g_{k-1}}$ , αντίστοιχα, δείχνουμε πως διατρέχουμε της συμμαχίες στη λίστα  $L_{g_k}$  που δεν επικαλύπτονται με τις παραπάνω, και αυτό πραγματοποιείται χωρίς την αποθήκευση της λίστας  $L_{g_k}$  στη μνήμη. Πιο λεπτομερώς αυτό γίνεται ως εξής:

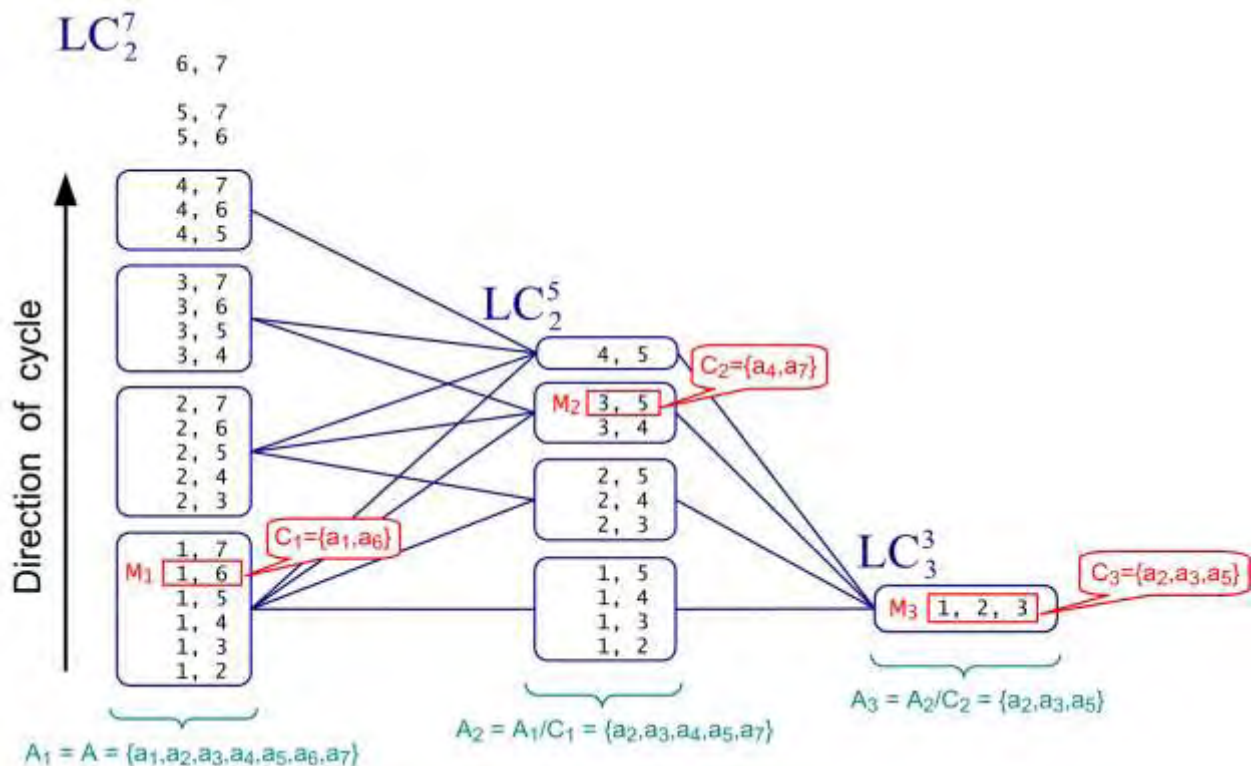
- Αντί να χρησιμοποιούμε τη μεταβλητή  $M_k$  για να διατρέχουμε τους συνδυασμούς στη λίστα  $LC_{g_k}^n$ , όπως εξηγήσαμε στη βασική ιδέα νωρίτερα, τη χρησιμοποιούμε για να διατρέχουμε τους συνδυασμούς της λίστας  $LC_{g_k}^{|A_k|}$ .
- Για κάθε διαφορετική τιμή της μεταβλητής  $M_k$ , εξάγουμε την αντίστοιχη συμμαχία  $C_k \in L_{g_k}$  ακολουθώντας την εξής τακτική:  $C_k = \{A_{k,i} \mid i \in M_k\}$  (βήμα 5 στον Αλγόριθμο 4). Για παράδειγμα, δεδομένου  $M_k = \{1, 3, 5\}$ , η αντίστοιχη συμμαχία δεν είναι αυτή που περιέχει τους πράκτορες  $a_1, a_3$ , και  $a_5$  αλλά περιέχει το πρώτο, τρίτο και πέμπτο στοιχείο του συνόλου  $A_k$ .

Οι παραπάνω δύο προσθήκες εξασφαλίζουν πως η τιμή  $M_k$  διατρέχει όλες τις πιθανές συμμαχίες μεγέθους  $g_k$ , που λαμβάνουμε από το σύνολο  $A_k$ . Με βάση αυτού, αν ορίσουμε  $A_k = A \setminus \{C_1, \dots, C_{k-1}\}$ , τότε διασφαλίζουμε πως κάθε περίπτωση για το  $C_k$  δεν επικαλύπτεται με καμία από τις συμμαχίες  $C_1, \dots, C_{k-1}$ .

Η εικόνα 7 παρουσιάζει ένα παράδειγμα δεδομένων συνόλου  $A=A_1=\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$  και ακέραιας διαμέρισης  $G = [2, 2, 3]$ . Όπως φαίνεται, εξετάζοντας για  $M_1=\{1, 6\}$  σημαίνει πως η συμμαχία  $C_1$  περιλαμβάνει τους πράκτορες  $a_1$  και  $a_6$ .

Γνωρίζοντας τους πράκτορες στη  $C_1$ , μπορούμε να αναθέσουμε το  $A_2$  σε αυτούς που δεν ανήκουν στο  $C_1$ , οπότε και  $A_2=\{a_2, a_3, a_4, a_5, a_7\}$ . Όπως αναφέρθηκε νωρίτερα, η μεταβλητή  $M_2$  θα διατρέξει όλες τις πιθανές συμμαχίες μεγέθους 2 από το σύνολο πρακτόρων  $A_2$  και καμία από αυτές τις συμμαχίες δεν πρέπει να επικαλύπτεται με τη  $C_1$ . Ομοίως, έχοντας (τυχαία για λόγω του παραδείγματος, καθώς εξετάζονται όλοι οι συνδυασμοί)  $M_2=\{3, 5\}$ , σημαίνει πως η συμμαχία  $C_2$  περιλαμβάνει το τρίτο και πέμπτο στοιχείο του συνόλου  $A_2$ , δηλαδή  $C_2 = \{a_4, a_7\}$ . Γνωρίζοντας τώρα, τους πράκτορες που ανήκουν στη  $C_2$ , αναθέτουμε το  $A_3$  στους πράκτορες που δεν περιέχονται στο  $C_1$  αλλά ούτε και στο  $C_2$ , οπότε  $A_3=\{a_2, a_3, a_5\}$ , και συνεχίζουμε με τον ίδιο τρόπο τη διαδικασία. Με τον τρόπο, που μόλις

περιγράψαμε παράγονται όλες οι δομές συμμαχιών στο  $\mathcal{P}_G$  και το πετυχαίνουμε χωρίς συγκρίσεις μεταξύ των συμμαχιών και για το λόγο αυτό μπορούν να παραχθούν διπλότυποι συνδυασμοί. Η τελευταία τροποποίηση που θα κάνουμε αμέσως τώρα στη βασική ιδέα που παρουσιάσαμε στην αρχή αυτής της ενότητας επιλύει αυτόν τον τελευταίο περιορισμό.



Εικόνα 7: Παράδειγμα της διαδικασίας προσπέλασης, δεδομένων συνόλου  $A=A_1=\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$  και ακέραιας διαμέρισης  $G = [2, 2, 3]$  (44)

### Αποφυγή διπλότυπων δομών συμμαχιών

Παρατηρούμε πως χρησιμοποιώντας τον παραπάνω τρόπο προσπέλασης των λιστών  $LC_{g_k}^n$ , υπάρχει η πιθανότητα της παραγωγής της ίδια δομής συμμαχιών δύο φορές αν υπάρχουν επαναλαμβανόμενα μέλη της ακέραιας διαμέρισης  $G$ , όπως στο παράδειγμα, που χρησιμοποιήσαμε στην εικόνα 7,  $G = [2, 2, 3]$ . Αυτό οφείλεται στο γεγονός πως παράγονται ταξινομημένα σύνολα  $\langle C_1, \dots, C_{|G|} \rangle$ , που περιέχουν ανεξάρτητες μεταξύ τους συμμαχίες των οποίων τα μεγέθη αντιστοιχούν στα μέλη του  $G$ . Με βάση αυτό, αν παραχθεί ένα ταξινομημένο σύνολο  $\overline{CS}$ , τότε οποιοδήποτε άλλο ταξινομημένο σύνολο  $\overline{CS}'$ , που περιέχει τις ίδιες συμμαχίες αλλά με διαφορετική σειρά, θα παραχθεί και αυτό, εφόσον τα μεγέθη των συμμαχιών του ανταποκρίνονται στα μέλη της ακέραιας διαμέρισης  $G$ . Αυτό, φυσικά, μπορεί να συμβεί μόνο αν ισχύει  $g_k = g_j : k \neq j$ . Γνωρίζοντας αυτά, η ιδέα μας χρειάζεται την

παρακάτω μικρή τροποποίηση για τις περιπτώσεις που επαναλαμβάνονται τα μέλη στο  $G$ .

- Καθώς διατρέχουμε τους συνδυασμούς στη λίστα  $LC_{g_k}^n$ , εξασφαλίζουμε πως το πρώτο στοιχείο της μεταβλητής  $M_k$ , που συμβολίζεται με  $M_{k,1}$  ικανοποιεί την εξής συνθήκη:  $a \leq M_{k,1} \leq n + 1 - \sum_{i=1}^k (g_i \times G(g_i))$ , όπου  $a=M_{k-1,1}$  αν  $g_k = g_{k-1}$  και  $a=1$  σε αντίθετη περίπτωση (βήμα 4 του Αλγορίθμου 4).

Αυτό φαίνεται και στο παράδειγμά μας στην εικόνα 6, με τη χρήση των συνδεδεμένων κουτιών. Πιο συγκεκριμένα, το  $M_1$  διατρέχει όλους τους συνδυασμούς της λίστας  $LC_2^7$ , που περιέχονται στα κουτιά (δεν διατρέχει τους συνδυασμούς  $\{5,6\}$ ,  $\{5,7\}$  και  $\{6,7\}$ , καθώς δεν ικανοποιούν την παραπάνω συνθήκη). Επιπλέον, το  $M_2$  διατρέχει μόνο τους συνδυασμούς της λίστας  $LC_2^5$ , που περιέχονται σε κουτιά και συνδέονται μέσω ακμών με αυτόν που διατρέχει την τρέχουσα χρονική στιγμή το  $M_1$ . Αυτή η τροποποίηση, διασφαλίζει πως  $M_{k+1,1} \geq M_{k,1}$  όταν  $g_{k+1}=g_k$ . Για παράδειγμα, όσο το  $M_1$  διατρέχει το κουτί στη λίστα  $LC_2^7$  που περιέχει τους συνδυασμούς, των οποίων το πρώτο στοιχείο είναι 3, έχουμε  $M_{1,1}$ . Στην περίπτωση αυτή, το  $M_2$ , διατρέχει μόνο τα κουτιά της λίστας  $LC_2^5$  που περιέχουν τους συνδυασμούς των οποίων το πρώτο στοιχείο είναι 3 ή 4, και έτσι διασφαλίζεται πως  $M_{2,1} \geq M_{1,1}$ . Η διαδικασία προσπέλασης που μόλις περιγράψαμε εξασφαλίζει πως όλοι οι συνδυασμοί συμμαχιών θαπραχθούν ακριβώς μία φορά. Τέλος, δεδομένου του εκθετικού μεγέθους του χώρου  $\mathcal{P}_G$ , θα ήταν πιο επιθυμητό να μπορούσαμε να αποφύγουμε την παραγωγή δομών συμμαχιών, οι οποίες δεν υπάρχει πιθανότητα να έχουν χρησιμότητα μεγαλύτερη από τη μέγιστη που έχει βρεθεί μέχρι αυτή τη στιγμή. Στη συνέχεια παρουσιάζεται η τεχνική, η οποία υλοποιεί την παραπάνω ιδέα, η οποία αποτελεί και το τελευταίο στάδιο του αλγορίθμου ακέραιων διαμερίσεων, IP.

### Αποφυγή εξαντλητικής αναζήτησης

Όπως αναφέρθηκε νωρίτερα, κατά τη διάρκεια σάρωσης των δομών συμμαχιών στο  $\mathcal{P}_G$ , ανανεώνουμε τη  $C_k$ , αφού έχουμε εξετάσει όλες τις πιθανές περιπτώσεις για τις συμμαχίες  $C_{k+1}, \dots, C_{|G|}$ , που δεν επικαλύπτονται με τις συμμαχίες  $C_1, \dots, C_k$ . Με άλλα λόγια, ενημερώνουμε τη  $C_k$ , αφού έχουμε ελέγξει όλους τους συνδυασμούς συμμαχιών που ξεκινούν με τη  $C_1, \dots, C_k$ . Ωστόσο, αν γνωρίζαμε εκ των προτέρων πως καμία από αυτές τις δομές συμμαχιών δεν θα μπορούσε να πετύχει χρησιμότητα μεγαλύτερη από τη βέλτιστη που έχει βρεθεί μέχρι εκείνη τη στιγμή, θα μπορούσαμε να την ανανεώσουμε τη  $C_k$  αμέσως (χωρίς να εξετάσουμε όλες τις περιπτώσεις για τη  $C_{k+1}, \dots, C_{|G|}$ ). Για να το πετύχουμε αυτό, υπολογίζουμε ένα άνω φράγμα στις χρησιμότητες των συμμαχιών που μπορούν να προστεθούν στις  $C_1, \dots, C_k$ . Συγκεκριμένα, έχοντας υπολογίσει την τιμή  $max_s$  για κάθε πιθανή

συμμαχία μεγέθους  $s \in \{1, 2, \dots, n\}$ , μπορούμε να υπολογίσουμε ένα άνω φράγμα, που συμβολίζεται με  $\text{MAX}_{[g_{k+1}, \dots, g_{|G|}]}$ , ως εξής:

$$\text{MAX}_{[g_{k+1}, \dots, g_{|G|}]} = \sum_{i=k+1}^{|G|} \max_{g_i}$$

Αν ορίσουμε ως  $V(\{C_1, \dots, C_k\})$ , το άθροισμα των χρησιμοτήτων των συμμαχιών  $C_1, \dots, C_k$ , το οποίο είναι  $V(C_1, \dots, C_k) = \sum_{i=1}^k v(C_i)$ , τότε το άθροισμα  $V(\{C_1, \dots, C_k\}) + \text{MAX}_{[g_{k+1}, \dots, g_{|G|}]}$  αναπαριστά ένα άνω φράγμα στη χρησιμότητα της δομής συμμαχιών που μπορεί να ανακτηθεί και ξεκινάει με τη  $C_1, \dots, C_k$  και τελειώνει με τα μεγέθη συμμαχιών  $[g_{k+1}, \dots, g_{|G|}]$ .

Συνεπώς, ισχύοντας  $V(CS') \geq V(\{C_1, \dots, C_k\}) + \text{MAX}_{[g_{k+1}, \dots, g_{|G|}]}$ , σημαίνει πως κανένας από τους συνδυασμούς συμμαχιών που αρχίζει με  $C_1, \dots, C_k$  και τελειώνει με τα μεγέθη συμμαχιών  $[g_{k+1}, \dots, g_{|G|}]$  δεν έχει τιμή μεγαλύτερη από  $V(CS')$  (ο έλεγχος αυτός πραγματοποιείται στο βήμα 8 του Αλγορίθμου 4). Από την άλλη, αν ισχύει η αντίθετη συνθήκη, δηλαδή  $V(CS') < V(\{C_1, \dots, C_k\}) + \text{MAX}_{[g_{k+1}, \dots, g_{|G|}]}$ , σημαίνει πως θα μπορούσε να εντοπιστεί ένας συνδυασμός συμμαχιών που αρχίζει με  $C_1, \dots, C_k$  και είναι καλύτερος από το βέλτιστο που έχει βρεθεί μέχρι τη δεδομένη χρονική στιγμή. Ωστόσο, αυτό και πάλι δεν σημαίνει πως όλες οι δομές θα πρέπει να εξεταστούν. Αυτό οφείλεται στο γεγονός, πως όταν ο αλγόριθμος μετακινείται στην επόμενη λίστα, ίσως βρει ότι υπάρχουν συγκεκριμένες δομές συμμαχιών οι οποίες δεν είναι καλύτερες από την μέχρι τότε βέλτιστη λύση. Τυπικά, για κάθε συμμαχία  $C_j : k < j < |G|$ , μπορούμε να έχουμε:

$$V(CS') > V(\{C_1, \dots, C_j\}) + \text{MAX}_{[g_{j+1}, \dots, g_{|G|}]}$$

## 5.4 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Για τις ανάγκες της απόδειξης του αλγορίθμου IP ως του πιο κατάλληλου για το πρόβλημα της παραγωγής δομών συμμαχιών, πραγματοποιήθηκε σύγκριση των χρόνων περάτωσης της διαδικασίας για εξαγωγή της βέλτιστου συνδυασμού συμμαχιών μεταξύ των αλγορίθμων IP και IDP. Ο κώδικας για τον αλγόριθμο IP υλοποιήθηκε σε γλώσσα προγραμματισμού C, σε εικονικό περιβάλλον Linux έκδοσης Ubuntu, μέσα από λειτουργικό σύστημα Windows 10. Τα πειράματα πραγματοποιήθηκαν σε υπολογιστή με επεξεργαστή AMD 3.50 GHz F™-6300 Six-Core και μνήμη RAM 768MB. Χρησιμοποιήθηκε κώδικας για τον αλγόριθμο IDP υπό τις ίδιες δυνατότητες συστήματος.

Πραγματοποιήθηκαν πειράματα για τυχαία πολυπρακτορικά συστήματα των 4, 6 και 10 πρακτόρων, δίνοντας ως είσοδο τις χρησιμότητες (values) των πιθανών συμμαχιών όλων των διαφορετικών μεγεθών, με σειριακό τρόπο. Και στις 6

περιπτώσεις, δηλαδή το αποτέλεσμα κάθε αλγορίθμου μεταξύ των δύο για κάθε είσοδο πρακτόρων, ήταν το επιθυμητό. Έχοντας στη διάθεσή μας σχετικά μικρό αριθμό δομών συμμαχιών, ήταν αρκετά εύκολο να γίνει αντιληπτή εκ των προτέρων η πιθανή βέλτιστη λύση. Πιο συγκεκριμένα, δεδομένων 4 πρακτόρων ο αλγόριθμος IDP εκτελέστηκε σε χρόνο 0,000567ms, ενώ με τον αλγόριθμο IP σε 0,0005660075ms. Για 10 πράκτορες, οι χρόνοι εκτέλεσης ήταν αντίστοιχα 0,000582 ms και 0,0005809815 αντίστοιχα. Τέλος, για 10 πράκτορες οι χρόνοι τερματισμού του προγράμματος ήταν 0,000673ms και 0,00667182225ms.

Παρατηρούμε πως σε κάθε περίπτωση υπάρχει βελτίωση ως προς τον χρόνο εκτέλεσης των πειραμάτων με τον αλγόριθμο IP. Οι χρόνοι εκτέλεσης ενδεχομένως να ήταν καλύτεροι αν τα πειράματα πραγματοποιούνταν σε σύστημα υψηλότερων δυνατοτήτων και στο βασικό λειτουργικό σύστημα του υπολογιστή. Επιπλέον, παρατηρούμε πως η αύξηση του αριθμού των πρακτόρων στην είσοδο οδηγεί και σε υψηλότερο χρόνο εκτέλεσης. Δεν πραγματοποιήθηκαν έλεγχοι για πολύ μεγάλο αριθμό πρακτόρων εξαιτίας της δυσκολίας σε σειριακή είσοδο όλων των χρησιμότητων. Ωστόσο, δεν υπάρχει καμία αμφιβολία πως ο αλγόριθμος εκτελείται αποτελεσματικά και για μεγαλύτερα σύνολα πρακτόρων, με εκθετική αύξηση στο χρόνο εκτέλεσης επί του μεγέθους του συνόλου αυτού.

Στη συνέχεια εμφανίζονται τα στιγμιότυπα του αλγορίθμου IP για τα αποτελέσματα του κώδικα στην περίπτωση των 4 πρακτόρων εισόδου. Υπενθυμίζουμε πως ο πίνακας των λιστών εισόδου για το κάθε δυνατό μήκος συμμαχίας είναι ο παρακάτω:

$L_1$	$L_2$	$L_3$	$L_4$
$a_4$	$a_3, a_4$	$a_2, a_3, a_4$	$a_1, a_2, a_3, a_4$
$a_3$	$a_2, a_4$	$a_1, a_3, a_4$	
$a_2$	$a_2, a_3$	$a_1, a_2, a_4$	
$a_1$	$a_1, a_4$	$a_1, a_2, a_3$	
	$a_1, a_3$		
	$a_1, a_2$		

Ο αντίστοιχες χρησιμότητες  $v(L_s)$  για την κάθε δυνατή συμμαχία του παραπάνω πίνακα, όπως δόθηκαν σειριακά από την είσοδο φαίνονται στον πίνακα που ακολουθεί.

$v(L_1)$	$v(L_2)$	$v(L_3)$	$v(L_4)$
45	80	110	140
25	70	100	
40	55	120	
30	80	90	
	60		



Στο πρώτο βήμα του αλγορίθμου αρχικοποιούμε την καλύτερη λύση με τη δομή συμμαχιών με τη μεγαλύτερη χρησιμότητα μεταξύ αυτής που ο κάθε πράκτορας εδν ανήκει σε καμία συμμαχία και ενεργεί ατομικά και αυτής στην οποία όλοι οι πράκτορες ανήκουν στην ίδια συμμαχία. Σύμφωνα με τις τιμές χρησιμοτήτων που επιλέχθηκαν στην είσοδο, οι δύο αυτές δομές διαθέτουν την ίδια χρησιμότητα, ίση με 140. Στην περίπτωση αυτή έχουμε ορίσει να επιλέγεται η δεύτερη δομή με όλους τους πράκτορες σε μία σύμμαχια, ως η καλύτερη λύση, όπως φαίνεται και από το στιγμιότυπο εκτέλεσης του κώδικά μας.

```
The initial best coalition structure is:
({a1,a2,a3,a4} with utility=140
```

*Εικόνα 8 Αρχικοποίηση καλύτερης λύσης με τη grand coalition*

Συνεχίζεται η διαδικασία αναζήτησης βέλτιστης δομής συμμαχιών στον υπόλοιπο χώρο αναζήτησης και συγκεκριμένα στις ακέραιες διαμερίσεις μεγέθους 2, δηλαδή στις (3,1) και (2,2). Αυτές αποτελούνται από τον συνδυασμό των συμμαχιών των λιστών  $L_3, L_4$  και  $L_2$  αντίστοιχα. Αναζητούμε μία δομή συμμαχίας με χρησιμότητα μεγαλύτερη από το 140.

```
Search for a better coalition structure in integer partition (3,1)
({a4},{a1,a2,a3}) with utility=135)
({a3},{a1,a2,a4}) with utility=145)
A better coalition found with utility=145, the ({a3},{a1,a2,a4})
({a2},{a1,a3,a4}) with utility=140)
({a1},{a2,a3,a4}) with utility=140)
```

*Εικόνα 9 Αναζήτηση καλύτερης λύσης στην ακέραια διαμέριση (3,1)*

Συνεπώς για να εντοπίσουμε ία δομή συμμαχίας με μεγαλύτερη χρησιμότητα από αυτή που έχουμε ήδη βρει, πρέπει να εντοπιστεί στο χώρο (2,2) μία με χρησιμότητα μεγαλύτερη από το 145.

```
Search for a better coalition structure in integer partition (2,2)
({a3,a4},{a1,a2}) with utility=130)
({a2,a4},{a1,a3}) with utility=130)
({a2,a3},{a1,a4}) with utility=135)
```

*Εικόνα 10 Αναζήτηση καλύτερης λύσης στην ακέραια διαμέριση (2,2)*

Στη συνέχεια αφαιρούμε από το χώρο αναζήτησης τις ακέραιες διαμερίσεις μεγέθους 2 και παρατηρούμε ότι η μοναδική που έχει παραμείνει είναι η διαμέριση (2,1,1). Για το χώρο αυτό υπολογίζουμε το άνω και κάτω φράγμα  $MAX_G$  και  $AVG_G$  και αν το άνω φράγμα είναι μικρότερο από το κάτω όριο για την καλύτερη λύση  $LB^*$  που υπολογίσαμε, τότε δεν υπάρχει λόγος να αναζητήσουμε βέλτιστη δομή συμμαχιών στη διαμέριση αυτή.

```
The remaining integer partitions in search space are: (2,1,1)
The MAX_G of (2,1,1) is: 170
The LB* is= 145
We don't prune the (2,1,1) because MAX_G > LB*
```

*Εικόνα 11 Υπολογισμός ορίων και διαγραφή υποχώρων*

Στο τελευταίο βήμα αναζητούμε καλύτερη δομή συμμαχιών στο χώρο (2,1,1) από αυτήν που έχουμε ανακαλύψει μέχρι αυτή τη στιγμή, δηλαδή την  $(\{a_1, a_2, a_4\}, \{a_3\})$ . Το βήμα αυτό φαίνεται στο παρακάτω στιγμιότυπο.

```
Search for a better coalition structure in integer partition (2,1,1)
({a3,a4},{a2},{a1}) with utility=150
A better coalition structure found with utility=150, the ({a3,a4},{a2},{a1})({a2,a4},{a3},{a1}) with utility=125
({a2,a3},{a4},{a1}) with utility=130
({a1,a4},{a3},{a2}) with utility=145
({a1,a3},{a4},{a2}) with utility=145
({a1,a2},{a3},{a4}) with utility=120
```

*Εικόνα 12 Αναζήτηση λύσης στην τελευταία ακέραια διαμέριση (2,1,1) και εντοπισμός καλύτερης λύσης*

Ο αλγόριθμός μας, μετά το βήμα αυτό, τερματίζει καθώς ο χώρος αναζήτησης έχει εξεταστεί ολόκληρος και εμφανίζεται και ο χρόνος εκτέλεσης της όλης διαδικασίας.

```
The best coalition structure found by IP algorithm is:
({a3,a4},{a2},{a4}) with utility 150
The time for the execution of the whole procedure was: 0.0005660075 ms
```

*Εικόνα 13 Εμφάνιση τελικής βέλτιστης δομής συμμαχιών και χρόνου εκτέλεσης*

## ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΈΡΕΥΝΑ

Ο αλγόριθμος, που βασίζεται στις αέριες διαμερίσεις του χώρου αναζήτησης και αναλύθηκε λεπτομερώς στο Κεφάλαιο 5, σχεδιάστηκε για να ικανοποιεί ορισμένες ιδιότητες:

*Βελτιστοποίηση:* Η πλήρης εκτέλεση του αλγορίθμου, εγγυάται την εξαγωγή ενός αποτελέσματος, το οποίο μεγιστοποιεί τη συλλογική χρησιμότητα του πολυπρακτορικού συστήματος.

*Ικανότητα για περιορισμό του χώρου αναζήτησης:* Ο αλγόριθμος πρέπει να αναγνωρίζει τους υποχώρους, οι οποίοι δεν υπάρχει πιθανότητα να περιέχουν βέλτιστη λύση, και να τους αφαιρεί από το συνολικό χώρο αναζήτησης. Η ιδιότητα αυτή είναι κρίσιμης σημασίας καθώς η φύση του προβλήματος που καλούμαστε να επιλύσουμε είναι εκθετική.

*Διάκριση:* Ο αλγόριθμος πρέπει να έχει τη δυνατότητα να αντιλαμβάνεται πως έχει εντοπιστεί η βέλτιστη λύση, κατά τη διάρκεια της αναζήτησης και να μην συνεχίζει με την ελπίδα πως μπορεί να βρεθεί μία καλύτερη λύση.

*Οποιοδήποτε χρόνου:* Ο αλγόριθμος πρέπει να έχει τη δυνατότητα να επιστρέφει άμεσα μία αρχική λύση και έπειτα να βελτιώνει την ποιότητά της, καθώς σαρώνει μεγαλύτερο τμήμα του χώρου αναζήτησης μέχρι να βρει μία βέλτιστη. Η ιδιότητα αυτή είναι ιδιαίτερα σημαντική, καθώς οι πράκτορες ίσως να μη διαθέτουν αρκετό χρόνο μέχρι τον τερματισμό του αλγορίθμου, ειδικά αν σκεφτεί κανείς το εκθετικό μέγεθος του χώρου αναζήτησης. Επιπλέον, ο αλγόριθμος γίνεται πιο ανθεκτικός σε αποτυχίες του συστήματος. Σε περιπτώσεις που η εκτέλεση του αλγορίθμου τερματιστεί απροσδόκητα, τότε και πάλι θα παραχθεί ένα αποτέλεσμα μεταξύ των πρακτόρων, το οποίο θα είναι καλύτερο από το αρχικό.

*Εγγυήσεις χειρότερης περίπτωσης:* Ο αλγόριθμος θα πρέπει να παρέχει εγγυήσεις χειρότερης περίπτωσης για την ποιότητα της λύσης. Σε αντίθετη περίπτωση, το εξαγόμενο αποτέλεσμα θα μπορούσε να είναι αυθαίρετα χειρότερο από το βέλτιστο. Τέτοιες εγγυήσεις είναι σημαντικές όταν επιθυμούμε μία χρυσή τομή μεταξύ της ποιότητας της λύσης και του χρόνου αναζήτησης. Για παράδειγμα, αν ποιότητα της παρούσας λύσης δεν είναι χειρότερη από το 95% τα βέλτιστης και υπάρχει ακόμα σημαντικό μέρος του χώρου αναζήτησης που δεν εξεταστεί ακόμα,

τότε οι πράκτορες ίσως να αποφασίσουν πως δεν αξίζει περαιτέρω έλεγχος. Προφανώς, όσο καλύτερες είναι οι εγγυήσεις, τόσο πιθανότερο οι πράκτορες να καταλήξουν στην απόφαση να διακόψουν τη διαδικασία αναζήτησης καλύτερης λύσης, από αυτήν που έχουν ήδη βρει.

Ο αλγόριθμος, που εφαρμόστηκε για την επίλυση του προβλήματος *Παραγωγής Δομών Συμμαχιών*, συνεισφέρει σημαντικά στη μέχρι τώρα δυνατότητα και γνώση για αντιμετώπιση του εν λόγω ζητήματος. Πολλοί αλγόριθμοι έχουν χρησιμοποιηθεί για τον ίδιο σκοπό, με την ικανότητα να παράγουν έναν συνδυασμό συμμαχιών ως βέλτιστη λύση του προβλήματος, παρουσιάζοντας ωστόσο κάποιους περιορισμούς και αδυναμίες. Ο αλγόριθμος IP καλύπτει αυτές τις αστοχίες και δεν θεωρείται άδικα ως ο πλέον αποδοτικός στην οργάνωση του πολυπρακτορικού συστήματος σε συμμαχίες. Η μεγάλη συνεισφορά του αλγορίθμου αυτού στηρίζεται στα εξής:

- Παρουσιάζει έναν νέο τρόπο αναπαράστασης του χώρου των πιθανών δομών συμμαχιών. Η αναπαράσταση αυτή, διαιρεί το χώρο αναζήτησης σε μικρότερα ανεξάρτητα τμήματα που μπορούν να εξεταστούν ξεχωριστά για την εύρεση μιας βέλτιστης λύσης. Σε αντίθεση με τον πιο διαδομένο τρόπο αναπαράστασης, σύμφωνα με τον οποίο οι δομές συμμαχιών κατηγοριοποιούνται με βάση του αριθμού των συμμαχιών που περιέχουν, ο αλγόριθμος IP κατηγοριοποιεί τις δομές με βάση το μέγεθος των συμμαχιών σε αυτές. Ένα σημαντικό πλεονέκτημα στην προσέγγιση που ακολουθεί ο αλγόριθμος IP είναι πως αμέσως μετά τη σάρωση της εισόδου (τις χρησιμότητες των συμμαχιών), είναι δυνατός ο υπολογισμός της μέσης χρησιμότητας των δομών συμμαχιών μέσα σε έναν υπόχωρο. Επιπλέον, κατά τη διάρκεια της σάρωσης της εισόδου, είναι δυνατός ο υπολογισμός ενός άνω και κάτω φράγματος στην αξία της βέλτιστης δομής, που μπορεί να βρεθεί μέσα σε αυτούς τους υποχώρους. Στη συνέχεια, συγκρίνοντας αυτά τα φράγματα μπορούν να εντοπιστούν τα τμήματα του χώρου αναζήτησης, που είναι βέβαιο πως δεν περιέχουν καλύτερη λύση και δεν συμμετέχουν στη διαδικασία. Ένα δεύτερο πλεονέκτημα εντοπίζεται στο γεγονός πως οι πράκτορες αναλύουν τη χρυσή τομή μεταξύ του μεγέθους ενός υποχώρου και της βελτίωσης που μπορεί να επέλθει στην τρέχουσα λύση εξαιτίας των ορίων του. Συνεπώς, αντί να περιορίζεται η λύση σε προκαθορισμένα μεγέθη, όπως πρότειναν οι Shehory και Kraus (1998), οι πράκτορες στον IP αποφασίζουν πιο εμπειριστατωμένα για το μέγεθος των συμμαχιών που θα σχηματίσουν.
- Πρόκειται για έναν οποιοδήποτε χρόνου αλγόριθμος, βασισμένος σε ακέραιες διαμερίσεις του χώρου και παρέχει υψηλές εγγυήσεις για την ποιότητα των λύσεων της.

- Έχει τη δυνατότητα να εξετάζει διαδοχικά συνδυασμούς συμμαχιών αποδοτικά, αποφεύγοντας περιττές και άτοπες λύσεις. Η τεχνική που εφαρμόζεται στη διαδικασία αυτή της εξέτασης επιτρέπει την αποφυγή εξαντλητικής αναζήτησης και τον περιορισμό του χώρου που χρειάζεται να περάσει από αυτήν τη διαδικασία.
- Ενώ πολλοί αλγόριθμοι για το πρόβλημα CSG υλοποιήθηκαν χρησιμοποιώντας την κατανομή της εισόδου, όπως προτάθηκε από τους Larson και Sandholm (2000), αποδεικνύεται πως αυτή η κατανομή εμφανίζεται να είναι μεροληπτική ως προς το πρόβλημα CSG. Για τον αλγόριθμο IP προτείνεται μία νέα κατανομή, αυτή που χρησιμοποιείται και στην ανάλυση του Κεφαλαίου 5, πιο κατάλληλη για την επίλυση του προβλήματος CSG.
- Σχετικά με την αξιολόγηση του χρόνου που απαιτείται για την παραγωγή μιας βέλτιστης λύσης, συγκρίνεται ο αλγόριθμος IP με τον γρηγορότερο αλγόριθμο που εγγυάται εξαγωγή βέλτιστου αποτελέσματος, τον IDP (Improved Dynamic Programming) αλγόριθμο, των Rahwan και Jennings (2008). Η σύγκριση αυτή δείχνει πως ο αλγόριθμος IP είναι γρηγορότερος.
- Τέλος, συγκρίνοντας τον αλγόριθμο IP με αντίστοιχους, που υλοποιήθηκαν για την επίλυση του ίδιου προβλήματος, καταλήγουμε στο συμπέρασμα πως παρέχει σημαντικά καλύτερες εγγυήσεις για την ποιότητα των λύσεων που παράγει της τάξης περί του 90% της βέλτιστης λύσης.

Μελλοντική έρευνα θα μπορούσε να πραγματοποιηθεί για την εισαγωγή και χρήση στοιχείων και από άλλα είδη προγραμματισμού, όπως ακέραιος και γραμμικός στις τεχνικές υπολογισμού των φραγμάτων. Περαιτέρω προσπάθειες για όσο το δυνατόν μεγαλύτερη μείωση στο χώρο προς αναζήτηση και του χρόνου σάρωσης των λιστών, θα πρέπει να γίνονται συνεχώς. Τέλος, ωφέλιμο θα ήταν πραγματοποίηση μελέτης για πιθανή εφαρμογή του αλγορίθμου και σε άλλα παρόμοια προβλήματα βελτιστοποίησης, που απαιτείται διαίρεση συνόλων και διαμερίσεων του χώρου αναζήτησης.

## Βιβλιογραφία

1. **M., Wooldridge.** *Εισαγωγή στα Πολυπρακτορικά Συστήματα.* Αθήνα : Εκδόσεις Κλειδάριθμος, 2008.
2. **Rusell Stuart, Norvig Peter.** *Τεχνητή Νοημοσύνη:Μια σύγχρονη προσέγγιση.* Αθήνα : Εκδόσεις Κλειδάριθμος, 2011.
3. **Bellman, R.E.** *An Introduction to Artificial Intelligence:Can Computers Think?* San Fransisco : Boyd & Fraser Publishing Company, 1978.
4. **Rich, E. and Knight, K.** *Artificial Intelligence (second ed.).* New York : McGraw-Hill, 1991.
5. **Winston, P. H.** *Artificial Intelligence (Third (ED.)).* . Massachuseetts : Addison-Wesley, Reading, 1992.
6. **Nilsson, N. J.** *Artificial Intelligence: A New Synthesis.* California : Morgan Kaufmann, 1998.
7. **Teahan, William John.** *Artificial Intelligence - Agents and Environments.* s.l. : William John Teahan & Ventus Publishing ApS, 2010.
8. **Βλαχάβας Ιωάννης, Κεφαλάς Πέτρος, Βασιλειάδης Νικόλαος, Κόκκορας Φώτης, Σακελλαρίου Ηλίας.** *Τεχνητή Νοημοσύνη, Β' Έκδοση.* Θεσσαλονίκη : Εκδόσεις Γαρταγάνη, 2005.
9. **Ferber, Jackues.** *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence.* Great Britain : Addison-Wesley, 1999.
10. **(Ed.), Jeffrey M.Bradshaw.** *Software Agents.* Menlo Park, CA, USA : AAI Press/ The MIT Press, 1997.
11. **J.C., Brustoloni.** *Autonomous Agents: Characterization and Requirements.* Pittsburgh: Carnegie Mellon : Carnegie Mellon Technical Report, 1991.
12. *Intelligence Without Reason.* **R.A., Brooks.** Sydney, Australia : Morgan Kaufmann, 1991. Proceedings of 12th Int. Joint.
13. **P., Maes.** How to Do the Right Thing. *Connection Science Journal, Vol. 1,No. 3, Taylor & Francis Group.* 1989, σσ. 291-323.
14. *Issues in agent-based software engineering.* **M., Wooldridge.** Heidelberg, : s.n., 1997. Proceedings of the First International Workshop on Cooperative Information Agents volume 1202 of LNAI. σσ. 1-18.

15. *Is it an Agent, or just a Program?.* **Graesser, Stan Franklin and Art.** 1996. Proceedings of the Third International Workshop on Agent Theories,.
16. **Russell, Stuart J. and Peter Norvig.** *Artificial Intelligence: A Modern Approach.* Englewood Cliffs, NJ : Prentice Hall, 1995.
17. **Maes, Pattie.** Artificial Life Meets Entertainment: Life like Autonomous Agents. *Communications of the ACM.* 1995, σσ. 108-114.
18. **Hayes-Roth, B.** An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence: Special Issue on Agents and Interactivity.* 1995, σσ. 329-365.
19. **Dodig-Crnkovic, Mark Burgin and Gordana.** *A Systematic Approach to Artificial Agents.* Department of Mathematics, University of California, Los Angeles, California, USA, School of Innovation, Design and Engineering, Mälardalen University, Sweden : s.n., 2009.
20. **Carneie Mellon Uniersity:School of Compter Science.** [Ηλεκτρονικό] <http://www.cs.cmu.edu/~softagents/multi.html>.
21. **Sycara, Katia P.** Multiagent. *AI Magazine Volume 19 Number 2.* (1998).
22. **Veloso, Peter Stone and Manuela.** Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robotics volume 8, number 3.* July 2000.
23. (Ed.), **Matthiaw Klusch.** *Intelligent Information Agents: Agent-Based Information Discovery an Management on the Internet.* Chemnitz, Germany : s.n., 1999.
24. **Oprea, Mihaela.** *Applications of Multi-agent Systems.* Ploiesti, Romania : University of Ploiesti, Departement of Informatics.
25. **H.Nwana, L.Lee and N.Jennings.** Coordination in Software Agent Systems. *BT Technology Jouranl.* 1996.
26. *Three Approaches to the Coordination of Multiagent Systems.* **F.Bergenti and A, Ricci,** Madrid, Spain : s.n., 2002. Proceedings of he international ACM confrece SAC 2002.
27. **Fernando Lopes, MichaelWooldridge and A. Q. Novais.** Negotiation among autonomous computational agents:principles, analysis and challenges. 2009.
28. **Weiss, Gerhard.** *Multiagent Systems-A Modern Approach to Distributed Modern Approach to Artificial Intelligence.* s.l. : The MIT Press Cambridge, Massachusetts,London, England, 1999.
29. **COMMUNITY TOOLBOX.** [Ηλεκτρονικό] <http://ctb.ku.edu/en/table-of-contents/assessment/promotion-strategies/start-a-coalition/main>.
30. "Coalition Building" (Boulder, CO: Conflict Research Consortium, 1998, accessed on January 30, 2003), <http://www.colorado.edu/conflict/peace/problem/coalition.htm>, available from και Internet. [Ηλεκτρονικό]

31. Rosegrant, Michael Watkins and Susan. *Breakthrough International Negotiation: How Great Negotiators Transformed the World's Toughest Post-Cold War Conflicts*, Review. San Francisco: : s.n., 2001.
32. Rahwan, Talal. *Algorithms for Coalition Formation in Multi-Agent Systems*. Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, UNIVERSITY OF SOUTHAMPTON : s.n., 2007.
33. Theodore L. Turocy, Bernhard von Stengel. *Game Theory*. s.l. : CDAM Research Report LSE-CDAM-2001-09, 2001.
34. Anderson, Michael van de Vijzel and John. *Coalition Formation in Multi-Agent Systems under Real-World Conditions*. s.l. : Autonomous Agents Laboratory, Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, R3T 2N2, Canada.
35. Lesser, Bryan Horling and Victor. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, Volume 19, Issue 4. 2004.
36. Onn M. Shehory, Katia Sycara and Somesh Jha. *Multi-agent Coordination through Coalition Formation*. s.l. : The Robotics Institute.
37. Talal Rahwan, Tomasz P. Michalak , Michael Wooldridge, Nicholas R. Jennings. *Coalition structure generation: A survey*. 2015.
38. *Issues in automated negotiation and electronic commerce: extending the contract net protocol*. V., Sandholm T. and Lesser. San Fransisco : s.n., 1995. Proceedings of the First International Multi-Agent Systems.
39. Gilles, R.P. *The Cooperative Game Theory of Networks and Hierarchies. Theory and Decision*. s.l. : Springer-Verlag Berlin Heidelberg, 2010.
40. Winter, Eyal. Chapter 53 The shapley value. *Handbook of Game Theory with Economic Applicatnios*. s.l. : Published by Elsevier B.V, 2002, σσ. 2025-2054.
41. Yoav Shoham, Kevin Leyton-Brown. *MULTIAGENT SYSTEMS: Algorithmic, Game-Theoretic, and Logical Foundations*. 2009.
42. Tuomas Sandholm, Kate Larson , Martin Andersson, Onn Shehory , Fernando Tohmé. *Coalition structure generation with worst case guarantees*. s.l. : Washington University, Department of Computer Science, One Brookings Drive, Campus Box 1045, St. Louis, 1999.
43. Talal Rahwan, Tomasz Michalak , Michael Wooldridge , Nicholas R. Jennings. *Anytime coalition structure generation in multi-agent systems with positive or negative externalities*. 2012.
44. Talal Rahwan, Sarvapali D. Ramchurn, Nicholas R. Jennings, Andrea Giovannucci. An Anytime Algorithm for Optimal Coalition Structure Generation. *Journal of Artificial Intelligence Research* 34. 2009.



45. Spangler, Brad. **Beyond Intractability** . [Ηλεκτρονικό] 2003.  
<http://www.beyondintractability.org/essay/coalition-building>.