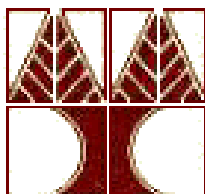


Ατομική Διπλωματική Εργασία

**ΜΕΘΟΔΟΙ ΨΕΥΔΟΔΥΑΔΙΚΗΣ
ΙΚΑΝΟΠΟΙΗΣΙΜΟΤΗΤΑΣ ΣΕ ΠΡΟΒΛΗΜΑΤΑ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΔΡΑΣΗΣ**

Στέφανος Στυλιανού

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Μέθοδοι προτασιακής ικανοποιησιμότητας σε προβλήματα
προγραμματισμού δράσης**

Στέφανος Στυλιανού

Επιβλέπων Καθηγητής

Γιάννης Δημόπουλος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση
των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος
Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2011

Ευχαριστίες

Με το τέλος αυτής της πολύμηνης εργασίας θα ήθελα αρχικά να ευχαριστήσω τον επιβλέποντα καθηγητή Δρ. Γιάννη Δημόπουλο για τη βοήθεια που μου παρείχε σε αυτό το δύσκολο έργο.

Ακόμη θα ήθελα να πω ένα μεγάλο ευχαριστώ στο διδακτορικό φοιτητή του Τμήματος Πληροφορικής Ανδρέα Σιδέρη, ο οποίος ήταν πάντα δίπλα μου, πρόθυμος να βοηθήσει, σε όλες μου τις απορίες. Μέσα από την επικοινωνία μας, με καθοδήγησε σε αρκετές δυσκολίες που αντιμετώπισα.

Τέλος θα ήθελα να ευχαριστήσω θερμά τη Στέλλα και την οικογένειά μου για την αμέριστη στήριξή τους καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Ο προγραμματισμός δράσεων είναι η διαδικασία εύρεσης μιας ακολουθίας από δράσεις, η εκτέλεση των οποίων οδηγεί στην επίτευξη κάποιου στόχου. Πιο συγκεκριμένα σε κάθε πρόβλημα υπάρχει μια αρχική κατάσταση, η οποία είναι η κατάσταση στην οποία βρισκόμαστε και μια τελική κατάσταση, η οποία είναι η κατάσταση στην οποία θέλουμε να φτάσουμε, δηλαδή ο στόχος. Σκοπός είναι ο προσδιορισμός ενός συνόλου από δράσεις, το αποτέλεσμα των οποίων μας οδηγεί από την αρχική κατάσταση στην τελική κατάσταση. Μέσα από αυτή τη διπλωματική εργασία γίνεται προσπάθεια για εύρεση αλγορίθμων και τεχνικών, οι οποίες επιλύουν το πρόβλημα βελτιώνοντας το κόστος επίλυσης σε σχέση με τις κρίσιμες παραμέτρους της απόδοσης και της αποτελεσματικότητας. Με τη χρησιμοποίηση αλγορίθμων, που δεν κάνουν χρήση εξαντλητικής αναζήτησης των δράσεων στο χώρο και τη χρήση ευρετικών συναρτήσεων ανάλογα με την τρέχουσα κατάσταση, γίνεται σε αρκετές περιπτώσεις σημαντική βελτίωση στη διαδικασία επίλυσης ενός προβλήματος προγραμματισμού δράσεων.

Σε κάθε πρόβλημα προγραμματισμού δράσεων, το πρόβλημα διασπάται με τη γλώσσα STRIPS, η οποία αναπαριστά τις δράσεις, τις καταστάσεις και τους στόχους του προβλήματος. Η γλώσσα αυτή δίνει στο πρόβλημα μια λογική δομή ενώ ταυτόχρονα είναι πολύ εκφραστική αλλά και περιοριστική, δίνοντας έτσι τη δυνατότητα για επίλυση των προβλημάτων με αποδοτικούς αλγόριθμους. Διάφοροι αλγόριθμοι χρησιμοποιήθηκαν με βάση τη γλώσσα STRIPS, όπως η προς τα εμπρός αναζήτηση στο χώρο και η προς τα πίσω αναζήτηση στο χώρο. Παράλληλα, σημαντική είναι η παρουσία του γραφήματος προγραμματισμού δράσεων στην επίλυση των προβλημάτων το οποίο κάνει καλύτερους και πιο αξιόπιστους τους υπολογισμούς που γίνονται με ευρετικό τρόπο.

Πριν από λίγα χρόνια, με σκοπό τη βελτίωση των αλγορίθμων, δημιουργήθηκε το σύστημα SatPlan. Το SatPlan είναι ένα ολοκληρωμένο σύστημα, το οποίο με τη χρήση ενός γραφήματος προγραμματισμού δράσεων, αναζητεί τη λύση του προβλήματος. Το πρόβλημα που δίνεται ως είσοδος είναι σε μορφή STRIPS και μπορεί να δεχθεί ένα σύνολο από διαφορετικές κωδικοποιήσεις του ίδιου προβλήματος. Ακολούθως, με τη

χρήση ενός προτασιακού επιλυτή (π.χ. precosat), προσπαθεί να βρει λύση στο πρόβλημα.

Σκοπός της παρούσας διπλωματικής εργασίας, σε πρώτη φάση είναι η επίλυση προβλημάτων προγραμματισμού δράσης με τη χρήση ψευδοδυαδικών περιορισμών η αποτίμηση των οποίων μπορεί να επεκταθεί σε όλο το σύνολο των ακεραίων αριθμών, μέσω του προτασιακού επιλυτή minisat+. Η δεύτερη φάση, επικεντρώνεται στη λύση των πιο πάνω προβλημάτων αλλά με τη χρήση μιας ενδιάμεσης κατάστασης. Συγκεκριμένα υπολογίζεται αρχικά μια λύση για την οποία η τιμή της αντικειμενικής συνάρτησης υπερβαίνει το κατώφλι, το οποίο ορίζει ο χρήστης. Μετά τον υπολογισμό αυτό επιλύεται ένα νέο πρόβλημα το οποίο έχει σαν αρχική κατάσταση την τελική κατάσταση που υπολογίστηκε στο προηγούμενο βήμα.

Ως μια παραλλαγή της πιο πάνω μεθόδου, εφαρμόζεται στη συνέχεια μια μεθοδολογία η οποία επιχειρεί και πάλι να λύσει το πρόβλημα «τμηματικά» αλλά χρησιμοποιώντας τις μερικές λύσεις με τη μορφή περιορισμών, οι οποίες προστίθενται στο νέο πρόβλημα που επιλύεται σε κάθε φάση. Η μεθοδολογία αυτή, σε αντίθεση με αυτή της ενδιάμεσης κατάστασης, γίνεται σε ένα σύνολο από φάσεις και όχι σε μόνο δύο φάσεις. Η τεχνική αυτή προσπαθεί να ικανοποιεί σε κάθε επόμενη φάση όλο και πιο μεγάλο υποσύνολο στόχων, φτάνοντας έτσι σταδιακά στη τελική λύση.

Ακολούθως, οι μεθοδολογίες αυτές αξιολογούνται πειραματικά σε ένα ευρύ φάσμα προβλημάτων με μετρήσεις που αφορούν κυρίως το μήκος του πλάνου αλλά ασφαλώς και το χρόνο εκτέλεσης μέχρι να βρεθεί η λύση, αν υπάρχει και γίνεται η μεταξύ τους σύγκριση. Παράλληλα γίνεται και σύγκριση της λύσης που δίνουν οι επιλυτές precosat και minisat+ στο ίδιο φάσμα προβλημάτων. Από αυτές τις πειραματικές μετρήσεις διαπιστώνεται ότι η κάθε τεχνική μπορεί να χρησιμοποιηθεί αποδοτικά σε διαφορετικές κατηγορίες προβλημάτων. Επίσης, ο βαθμός πολυπλοκότητας κάθε προβλήματος όπως και οι χρονικοί περιορισμοί (π.χ. προθεσμίες ικανοποίησης περιορισμών – προβλήματα Trucks) που το χαρακτηρίζουν, αποτελούν σημαντική παράμετρο για την αποτελεσματικότητα και την αποδοτικότητα της κάθε τεχνικής.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή	1
	1.1 Ιστορική αναδρομή στον Προγραμματισμό Δράσης	1
	1.2 Στόχος - Περιεχόμενο διπλωματικής εργασίας	2
Κεφάλαιο 2	Η έννοια του προγραμματισμού δράσης και οι κυριότεροι αλγόριθμοι	5
	2.1 Γενικά περί του προγραμματισμού δράσης	5
	2.2 Αναπαράσταση προβλήματος με τη γλώσσα STRIPS	7
	2.2.1 Δομή της γλώσσας STRIPS	7
	2.2.2 Παραδείγματα με τη γλώσσα STRIPS	8
	2.3 Αλγόριθμοι προγραμματισμού δράσης	9
	2.3.1 Εμπρόσθια αναζήτηση στο χώρο (Forward state-space search)	9
	2.3.2 Προς τα πίσω αναζήτηση στο χώρο – (Backward state-space search)	10
	2.3.3 Ευρετική αναζήτηση στο χώρο (Heuristics for state-space search)	11
	2.3.4 Προγραμματισμός με μερική σειρά (Partial-Order Planning)	12
	2.3.5 Ευρετικές μέθοδοι για προγραμματισμό μερικής διάταξης	14
Κεφάλαιο 3	Γράφοι δράσεων και μέθοδος Satplan	15
	3.1 Δομή – Περιγραφή του γράφου δράσεων	15
	3.2 Ευρετικός υπολογισμός με βάση το γράφημα δράσεων	18
	3.3 Ο αλγόριθμος του Γράφου δράσεων	19
	3.4 Ο αλγόριθμος του SatPlan	20
	3.4.1 Η ιδέα του SatPlan	20
	3.4.2 Λειτουργία SatPlan	21

3.4.3 Κωδικοποίηση Γράφου δράσεων – Κωδικοποίηση SAT-MAX-PLAN	23
Κεφάλαιο 4 Ψευδοδυαδικοί περιορισμοί και περιορισμοί πληθικότητας	25
4.1 Γενικά	25
4.2 Ψευδοδυαδική συνάρτηση	26
4.3 Ψευδοδυαδικοί περιορισμοί	26
4.3.1 Γραμμικοί Ψευδοδυαδικοί περιορισμοί	27
4.3.2 Μη Γραμμικοί Ψευδοδυαδικοί περιορισμοί	27
4.4 Περιορισμοί Πληθικότητας	28
4.4.1 Κανονικοποίηση Ψευδοδυαδικών περιορισμών	30
4.5 Ικανοποιησιμότητα και Βελτιστοποίηση	31
4.6 Βελτιστοποίηση με ψευδοδυαδικούς περιορισμούς	31
4.7 Λειτουργία αλγορίθμου των ψευδοδυαδικών επιλυτών	32
4.8 Η σημασία των ψευδοδυαδικών περιορισμών	33
Κεφάλαιο 5 Ψευδοδυαδική Ικανοποίηση και Προγραμματισμός Δράσης	34
5.1 Επίλυση προβλημάτων προγραμματισμού δράσης	35
5.2 Γνωστά προβλήματα προγραμματισμού δράσης	36
5.3 Προτασιακοί επιλυτές και κωδικοποιήσεις	39
5.4 Ο προτασιακός επιλυτής Minisat+	42
5.4.1 Γενικά περί του Minisat+	42
5.4.2 Η κωδικοποίηση SMP σε cnf μορφή για επίλυση προβλημάτων με το εργαλείο Minisat+	43
5.5 Ψευδοδυαδική Βελτιστοποίηση	47
5.5.1 Η ψευδοδυαδική συνάρτηση βελτιστοποίησης του minisat+	48
5.5.2 Επίλυση προβλημάτων μέσω ενδιάμεσης κατάστασης βάσει ενός κάτω φράγματος – κατωφλίου	49
5.5.3 Τμηματική επίλυση προβλημάτων	52
5.5.3.1 Τεχνική διατήρησης ικανοποιηθέντων περιορισμών	55
5.5.3.2 Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών	56
5.5.3.3 Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών – συνδυασμός διατήρησης και ενημέρωσης	

περιορισμών	56
Κεφάλαιο 6 Πειραματική Ανάλυση	58
6.1 Επίλυση προβλημάτων και μετρήσεις με το εργαλείο Precosat - χρήση της κωδικοποίησης SMP και BlackBox	58
6.2 Μετρήσεις από την επίλυση προβλημάτων μέσω ενδιάμεσης κατάστασης βάσει ενός κάτω φράγματος – κατωφλίου	60
6.3 Μετρήσεις από την τμηματική επίλυση των προβλημάτων	65
Κεφάλαιο 7 Συμπεράσματα	72
7.1 Σύνοψη και συμπεράσματα από τη Διπλωματική Εργασία	72
7.2 Επεκτάσεις και Μελλοντική Εργασία	77
Βιβλιογραφία	80

Κεφάλαιο 1

Εισαγωγή

1.1 Ιστορική αναδρομή στον Προγραμματισμό Δράσης	1
1.2 Στόχος - Περιεχόμενο διπλωματικής εργασίας	2

1.1 Ιστορική αναδρομή στον Προγραμματισμό Δράσης

Το πρόβλημα του προγραμματισμού δράσης, δηλαδή της εύρεσης μιας ακολουθίας από δράσεις η εκτέλεση των οποίων οδηγούν στο επιθυμητό αποτέλεσμα, ξεκίνησε από πολύ παλιά. Χρησιμοποιήθηκαν αρχικά για την επίλυση τέτοιου είδους προβλημάτων, αλγόριθμοι κυρίως σειριακοί που εκτελούσαν δηλαδή μια δράση κάθε φορά, την μια μετά την άλλη. Τέτοιοι αλγόριθμοι όμως απέτυχαν εξαιτίας του ότι απαιτούσαν μεγάλο χρόνο εκτέλεσης, αφού έλεγχαν όλες τις πιθανές λύσεις και είχαν γενική μορφή και όχι εξειδικευμένα.

Η ανάπτυξη της τεχνολογίας αλλά κυρίως η δημιουργία νέων αλγορίθμων έδωσε νέα ώθηση στο πρόβλημα του προγραμματισμού δράσης. Η δημιουργία παράλληλων αλγορίθμων είχε ως αποτέλεσμα την παράλληλη επεξεργασία των δεδομένων του προβλήματος και συνάμα βοήθησε να ξεφύγουμε από τους φυσικούς περιορισμούς των υπολογιστών, όπως για παράδειγμα η μνήμη, η ταχύτητα επεξεργαστή κ.τ.λ.

Γύρω στο 1992, οι Kautz & Selman εισήγαγαν την ιδέα τους για μετασχηματισμό του προβλήματος του planning σε πρόβλημα ικανοποιησιμότητας προτασιακής λογικής. Αργότερα, το 1996 οι Kautz & Selman με το Satplan, που είναι ένα ολοκληρωμένο σύστημα που στηρίζεται σε προτασιακή λογική, απέδειξαν ότι είναι εξίσου αποδοτικό με τις υπόλοιπες τεχνολογίες προβλημάτων δράσης και μπορεί να είναι ανταγωνιστικό. Μέχρι το 2004, το SatPlan κέρδισε το πρώτο βραβείο στο συνέδριο ICAPS 2004 που αφορούσε στο βέλτιστο προγραμματισμό δράσεων [5].

Το 2006, δημιουργήθηκε μια νέα έκδοση του SatPlan, το SatPlan-2006 το οποίο βελτίωσε την αποδοτικότητα της λύσης με πιο γρήγορους υπολογισμούς, λαμβάνοντας υπόψη τους αμοιβαία αποκλειόμενους περιορισμούς που αφορούν στις συνθήκες και στις συνέπειες των δράσεων.

Το πρόβλημα του προγραμματισμού δράσεων είναι ένα πρόβλημα «ανοικτό» μέχρι σήμερα. Αρκετοί ερευνητές, μελετούν το πρόβλημα και αναπτύσσουν διάφορα εργαλεία και τεχνικές προτασιακής ικανοποιησιμότητας, ευελπιστώντας ότι θα επιτύχουν ταχύτερες και πιο αποτελεσματικές λύσεις. Δημιουργούνται συνεχώς νέοι προτασιακοί επιλυτές, οι οποίοι μέσω κοινής αναπαράστασης επικοινωνούν με το SatPlan και μπορούν να δώσουν κάποιο αποτέλεσμα στο πρόβλημα, δηλαδή αν υπάρχει λύση και ποια είναι αυτή. Αυτό επιτρέπει τη σύγκριση της αποτελεσματικότητας και της αποδοτικότητας μεταξύ διαφορετικών αλγορίθμων – προτασιακών επιλυτών.

1.2 Στόχος – Περιεχόμενο διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία, έχει ως σκοπό τη μελέτη του ολοκληρωμένου συστήματος SatPlan-2006. Μελετώνται πειραματικά οι χρόνοι εκτέλεσης για την επίλυση συγκεκριμένων προβλημάτων προτασιακής ικανοποιησιμότητας από συγκεκριμένους προτασιακούς επιλυτές όπως minisat+, precosat, siege. Δίνεται ιδιαίτερη έμφαση στην κωδικοποίηση προτασιακής ικανοποιησιμότητας SAT-MAX-PLAN. Ακολουθώντας, γίνεται ενσωμάτωση της SAT-MAX-PLAN αλλά οι περιορισμοί έχουν ψευδοδυαδική μορφή. Ο προτασιακός επιλυτής που χρησιμοποιείται γι' αυτήν την μορφή και ευρύτερα για τη ψευδοδυαδική βελτιστοποίηση είναι ο minisat+. Γίνονται πειραματικές μετρήσεις αρχικά για να διαφανεί ότι η λύση που προκύπτει από αυτήν την κωδικοποίηση βρίσκεται στον ίδιο αριθμό χρονικών βημάτων - επιπέδων με αυτήν που δίνει ο προτασιακός επιλυτής precosat με κωδικοποίηση και πάλι τη SAT-MAX-PLAN.

Εφόσον εξασφαλιστεί αυτή η ισοδυναμία, το επόμενο βήμα είναι η επίλυση προβλημάτων με τον minisat+ αλλά αυτή τη φορά με τη χρήση αντικειμενικής συνάρτησης. Το εκάστοτε πρόβλημα επιλύεται τμηματικά. Πιο συγκεκριμένα σε αυτή τη φάση προσπαθούμε να βρούμε μια λύση για την οποία η τιμή της αντικειμενικής συνάρτησης υπερβαίνει το κατώφλι που ορίζεται από τον χρήστη. Μετά τον

υπολογισμό αυτό επιλύεται ένα νέο πρόβλημα το οποίο έχει σαν αρχική κατάσταση την τελική κατάσταση (ενδιάμεση λύση) που υπολογίστηκε στο προηγούμενο βήμα. Ως μια διαφορετική τεχνική του πιο πάνω είναι η επίλυση των προβλημάτων τμηματικά, σε φάσεις. Η διαφορά εδώ είναι ότι ξεκινούμε συνήθως από μικρό κατώφλι και προσπαθούμε να επιλύσουμε το πρόβλημα αυξάνοντας το κατώφλι σταδιακά μέχρι να ικανοποιηθούν όλοι οι στόχοι του προβλήματος. Οι μερικές λύσεις που προκύπτουν, προστίθενται στο νέο πρόβλημα που επιλύεται σε κάθε επόμενη φάση υπό μορφή περιορισμών. Τέλος, οι μέθοδοι αξιολογούνται πειραματικά και γίνεται μία μεγάλης κλίμακας συγκριτική αξιολόγηση σε ένα μεγάλο εύρος προβλημάτων Προγραμματισμού Δράσης.

Η ολοκλήρωση της διπλωματικής εργασίας έγινε ουσιαστικά σε τρεις φάσεις. Στην πρώτη φάση σκοπός ήταν κυρίως η απόκτηση του θεωρητικού υποβάθρου, που αφορά το πρόβλημα του προγραμματισμού δράσης και των μεθόδων Ψευδο-δυαδικής Βελτιστοποίησης. Έγινε μελέτη επιστημονικών άρθρων αλλά και εμπέδωση κεφαλαίων από θεωρητικά βιβλία τα οποία είναι σχετικά με το θέμα. Επίσης έχει εξεταστεί σε βάθος ο κώδικας του συστήματος SATPLAN και έχουν εντοπιστεί τα τμήματα τα οποία πρέπει να αντικατασταθούν ώστε να ενσωματωθούν οι τεχνικές Ψευδο-δυαδικής Βελτιστοποίησης. Παράλληλα, ολοκληρώθηκε μία προκαταρκτική υλοποίηση της ενσωμάτωσης αυτής όπου ο προτασιακός επιλυτής του SatPlan έχει αντικατασταθεί από το ψευδο-δυαδικό σύστημα minisat+. Η φάση αυτή διήρκεσε από το Σεπτέμβριο μέχρι το Δεκέμβριο.

Η δεύτερη φάση που ήταν και η πιο σημαντική αφορούσε την επίλυση προβλημάτων με το ψευδο-δυαδικό σύστημα minisat+ που αναφέρθηκε πιο πάνω μέσω ενδιάμεσης κατάστασης αλλά και τμηματικά. Πιο συγκεκριμένα κάθε πρόβλημα προγραμματισμού δράσης επιλύεται τώρα σε φάσεις, όπου σε κάθε φάση ικανοποιείται ένα υποσύνολο του συνόλου του αρχικού στόχου, και οι στόχοι που ικανοποιούνται σε κάθε φάση ενσωματώνονται στην επόμενη φάση ως περιορισμοί στο νέο πρόβλημα. Αυτό επαναλαμβάνεται μέχρι να ικανοποιηθεί ο αρχικός στόχος. Ουσιαστικά η φάση αυτή είχε σκοπό να βελτιώσει το κόστος χρόνου εύρεσης της λύσης. Η φάση αυτή διήρκεσε από τις αρχές Ιανουαρίου μέχρι τις αρχές Απριλίου.

Τέλος, στην τρίτη φάση, εφόσον έγιναν οι πιο πάνω υλοποιήσεις σκοπός ήταν η πειραματική αξιολόγηση των μεθόδων αυτών σε ένα μεγάλο σύνολο από προβλήματα. Από αυτήν την αξιολόγηση προκύπτουν συμπεράσματα που αφορούν στο κόστος εκτέλεσης (χρονικό κόστος) και στην αποτελεσματικότητα (τους στόχους που ικανοποιήθηκαν). Επίσης, έγινε η συγγραφή της διπλωματικής εργασίας. Η φάση αυτή διήρκησε τους μήνες Απρίλιο και Μάιο.

Κεφάλαιο 2

Η έννοια του προγραμματισμού δράσης και οι κυριότεροι αλγόριθμοι

2.1 Γενικά περί του προγραμματισμού δράσης	5
2.2 Αναπαράσταση προβλήματος με τη γλώσσα STRIPS	7
2.2.1 Δομή της γλώσσας STRIPS	7
2.2.2 Παραδείγματα με τη γλώσσα STRIPS	8
2.3 Αλγόριθμοι προγραμματισμού δράσης	9
2.3.1 Εμπρόσθια αναζήτηση στο χώρο (Forward state-space search)	9
2.3.2 Προς τα πίσω αναζήτηση στο χώρο (Backward state-space search)	10
2.3.3 Ευρετική αναζήτηση στο χώρο (Heuristics for state-space search)	11
2.3.4 Προγραμματισμός με μερική σειρά (Partial-Order Planning)	12
2.3.5 Ευρετικές μέθοδοι για προγραμματισμό μερικής διάταξης	14

2.1 Γενικά περί του προγραμματισμού δράσης

Ο προγραμματισμός δράσεων όπως αναφέρθηκε και στην εισαγωγή αφορά προβλήματα στα οποία δεδομένου ότι βρισκόμαστε σε μια αρχική κατάσταση, σκοπός είναι να βρούμε μια ακολουθία από δράσεις οι συνέπειες των οποίων οδηγούν σε κάποιο στόχο. Ωστόσο η επίλυση προβλημάτων προγραμματισμού δράσης δεν είναι απλή, ούτε μπορεί να γενικευθεί μέσω κανόνων. Έτσι, αυτό είχε ως αποτέλεσμα να χρησιμοποιηθούν εμπρόσθιοι (από την αρχική στην τελική κατάσταση) και οπίσθιοι (από το στόχο προς την αρχική κατάσταση) αλγόριθμοι με τη βοήθεια ευρετικών συναρτήσεων οι οποίες λαμβάνουν υπόψη τη δομή της αναπαράστασης του προβλήματος και μπορούν να λύσουν ικανοποιητικά ένα τέτοιο πρόβλημα [1].

Σε ένα πρόβλημα προγραμματισμού δράσης, υπάρχουν δυο στάδια κατά τη διαδικασία επίλυσης. Το πρώτο στάδιο είναι να διαπιστωθεί κατά πόσο το πρόβλημα μπορεί να ικανοποιηθεί ή όχι με βάση πάντα τους περιορισμούς του. Σχετίζεται δηλαδή με τη

διαπίστωση ότι υπάρχει ή όχι κάποιο σχέδιο δράσης. Το δεύτερο στάδιο, δεδομένου ότι υπάρχει κάποιο σχέδιο δράσης, είναι η εύρεση του ίδιου του σχεδίου δράσης. Αφορά δηλαδή τη δημιουργία σχεδίου δράσης, το οποίο περιλαμβάνει μια σειρά από πράξεις η εκτέλεση των οποίων οδηγεί στον επιθυμητό στόχο.

Οι πιο πάνω αλγόριθμοι γενικής μορφής αντιμετώπιζαν δυσκολίες στο να λύσουν τέτοιου είδους προβλήματα. Αλγόριθμοι όπως οι A^* και η αναζήτηση σε βάθος στην προσπάθεια τους να λύσουν προβλήματα μεγάλης πολυπλοκότητας απέτυχαν. Αυτό διότι ακολουθούσαν τυποποιημένους κανόνες για όλα τα προβλήματα, και εξερευνούσαν την κάθε πιθανή λύση. Για κάθε δράση εξεταζόταν κατά πόσο οι συνέπειες της ικανοποιούν το στόχο. Εγκλωβίζονταν δηλαδή σε μια εξαντλητική ανίχνευση των περιπτώσεων του προβλήματος.

Κλασσικό παράδειγμα από την καθημερινή μας ζωή που τονίζει τις αδυναμίες τέτοιων αλγορίθμων είναι το εξής: Έστω ότι θέλουμε να βρούμε το άτομο του οποίου ο αριθμός ταυτότητας είναι 1220951 και η εύρεση του σχετίζεται με 10 εκατομμύρια δράσεις. Σε αυτή την περίπτωση ο αλγόριθμος θα εξετάσει όλες τις δράσεις μέχρι να βρει αυτή που ικανοποιεί το στόχο. Αντίθετα ένας άλλος σαφώς πιο αποδοτικός αλγόριθμος θα αναζητούσε την ταυτότητα ξεκινώντας από το στόχο, εφόσον η ταυτότητα είναι γνωστή θα εκτελούσε μόνο την αντίστοιχη πράξη που έχει ως συνέπεια αυτή την ταυτότητα. Θα ξεκινούσε δηλαδή από την τελική κατάσταση και θα έφτανε στην αρχική, χωρίς να χρειαζόταν να εξετάσει όλες τις δράσεις.

Σημαντικές παράμετροι στα προβλήματα προγραμματισμού δράσης είναι η ευρετική συνάρτηση και η διάσπαση του προβλήματος [1]. Η ευρετική συνάρτηση σίγουρα μπορεί να αποκλείσει κάποιες περιπτώσεις του προβλήματος, ωστόσο δημιουργείται το πρόβλημα της αυτονομίας του επιλυτή. Επίσης είναι σημαντικό να γίνεται διάσπαση του προβλήματος σε ισομεγέθη υποπροβλήματα ανάλογα με το στόχο, καθώς αυτό μειώνει σε μεγάλο βαθμό το σύνολο των περιπτώσεων που θα εξεταστούν. Ωστόσο, η επίλυση των υποπροβλημάτων και ο συνδυασμός της λύσης τους θα πρέπει να γίνει με τέτοιο τρόπο ώστε οι λύσεις να μην αναιρούν άλλους υποστόχους.

2.2 Αναπαράσταση προβλήματος με τη γλώσσα STRIPS

2.2.1 Δομή της γλώσσας STRIPS

Έχει συζητηθεί ότι η επίλυση των προβλημάτων προγραμματισμού δράσης, εμπεριέχει δυσκολίες οι οποίες προκύπτουν κυρίως από την πολυπλοκότητα των προβλημάτων. Η δυσκολία ήταν στο πώς να αναπαρασταθεί ένα τέτοιο πρόβλημα, ώστε να μειώσει το βαθμό δυσκολίας της επίλυσης του. Γι' αυτό το λόγο δημιουργήθηκε η γλώσσα STRIPS η οποία αναπαριστά ξεκάθαρα ποιοι είναι οι στόχοι, οι δράσεις και οι καταστάσεις. Περιγράφει έτσι την αρχική κατάσταση, τις δράσεις που μπορούν να εφαρμοστούν και ποιο το αποτέλεσμα-συνέπειες τους, καθώς και την τελική κατάσταση που είναι ο στόχος. Είναι μια γλώσσα η οποία δίνει στο πρόβλημα λογική δομή, είναι αρκετά εκφραστική ώστε να μπορεί να περιγράψει πληθώρα προβλημάτων αλλά και περιοριστική, γεγονός που επιτρέπει σε αποδοτικούς αλγόριθμους να τη χρησιμοποιήσουν.

Κάθε κατάσταση (state) αντικατοπτρίζει λογικές συνθήκες του κόσμου και παρουσιάζεται ως μια σύζευξη από κυριολεκτήματα (literals) π.χ. Άνδρας $\wedge$ Ηλικιωμένος ή $At(vase, table) \wedge \neg At(vase, floor)$.

Κάθε στόχος, είναι ουσιαστικά μια κατάσταση η οποία ικανοποιεί όλα τα κυριολεκτήματα του στόχου. π.χ. η κατάσταση Άνδρας $\wedge$ Ηλικιωμένος $\wedge$ Ψηλός ικανοποιεί το στόχο Άνδρας $\wedge$ Ηλικιωμένος.

Κάθε δράση αποτελείται από τις προϋποθέσεις (preconditions), οι οποίες πρέπει να ισχύουν πριν την εκτέλεση της και τις συνέπειες (effects), οι οποίες παρουσιάζονται μετά την εκτέλεση της π.χ.

Action (Kill (A, B))

Preconditions: Have (A, gun) $\wedge$ Angry_with (A, B) $\wedge$ $\neg$ Protected (B)

Effect: Dead (B)

2.2.2 Παράδειγμα με τη γλώσσα STRIPS

Σε αυτό το σημείο θα παρουσιαστεί ένα παράδειγμα προβλήματος προγραμματισμού δράσης, το οποίο μπορεί να αναπαρασταθεί με τη γλώσσα STRIPS, η οποία περιγράφηκε πιο πάνω.

Παράδειγμα:

Έστω ένας άνθρωπος σε ένα εργαστήριο. Ο άνθρωπος βρίσκεται στη θέση A και θέλει να πάρει ένα κιβώτιο, το οποίο βρίσκεται στη θέση B (ένα υψηλό σημείο). Επίσης υπάρχει μια καρέκλα στη θέση C. Ο άνθρωπος χρειάζεται την καρέκλα για να πιάσει το κιβώτιο.

Αρχική κατάσταση : $At(A), Level(low), ChairAt(C), BoxAt(B)$

Τελική κατάσταση – Στόχος : $Have(Box)$

Δράσεις:

$Move(X, Y)$ [Μετακίνηση από μια θέση σε άλλη]

Preconditions: $At(X) \wedge Level(low)$

Effects: $not\ At(X) \wedge At(Y)$

$Up_to(Location)$ [Ανάβαση του ανθρώπου στην καρέκλα]

Preconditions: $At(Location) \wedge ChairAt(Location), Level(low)$

Effects: $Level(high) \wedge not\ Level(low)$

$Down_to(Location)$ [Κατάβαση του ανθρώπου από την καρέκλα]

Preconditions: $At(Location) \wedge ChairAt(Location) \wedge Level(high)$

Effects: $Level(low) \wedge not\ Level(high)$

$MoveChair(X, Y)$ [Μετακίνηση της καρέκλας από το σημείο X στο σημείο Y]

Preconditions: $At(X) \wedge ChairAt(X) \wedge Level(low)$

Effects: $ChairAt(Y) \wedge not\ ChairAt(X) \wedge At(Y) \wedge not\ At(X)$

TakeBox(Location) [Παραλαβή κιβωτίου]
 Preconditions: $\text{At}(\text{Location}) \wedge \text{BoxAt}(\text{Location}) \wedge \text{Level}(\text{high})$
 Effects: Have (box)

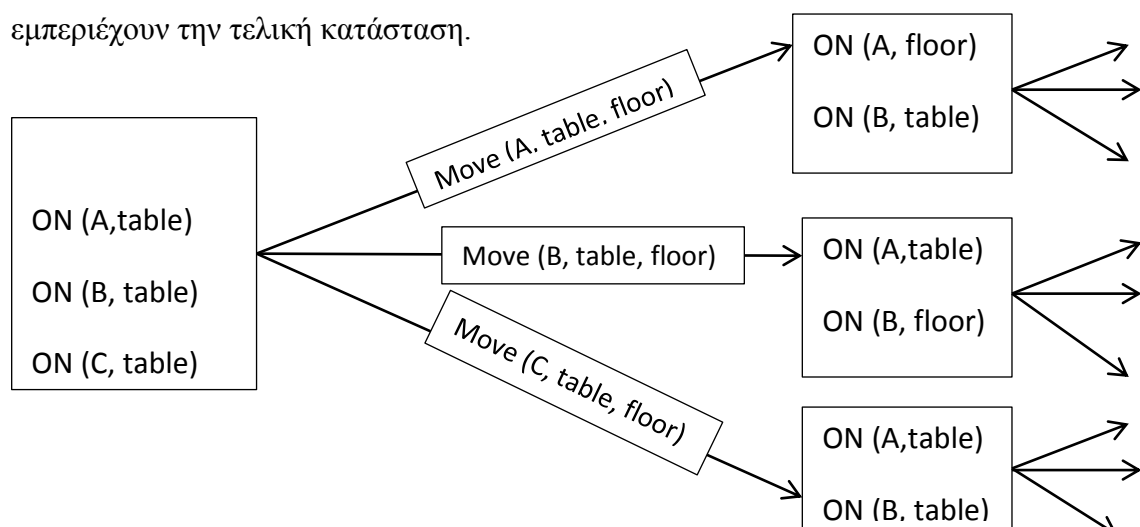
Η κάθε δράση όπως φαίνεται πιο πάνω περιγράφεται από ένα σύνολο προϋποθέσεων (preconditions) που πρέπει να ισχύουν για να εφαρμοστεί η δράση καθώς επίσης και από τις συνέπειες (effects) που είναι το αποτέλεσμα μετά την εφαρμογή της δράσης. Οι συνέπειες διακρίνονται σε θετικές (add effects) οπότε ο όρος ισχύει και σε αρνητικές (delete effects) οπότε ο όρος δεν ισχύει.

2.3 Αλγόριθμοι προγραμματισμού δράσης

Στα πλαίσια επίλυσης προβλημάτων προγραμματισμού δράσης έχουν δημιουργηθεί πολλοί αλγόριθμοι οι οποίοι μπορεί να διαφέρουν ανάλογα με την υλοποίησή τους ως προς τρόπο αναζήτησης της λύσης στο χώρο καταστάσεων. Οι αλγόριθμοι μπορεί να αναζητούν τη λύση εμπρόσθια από την αρχική κατάσταση προς την τελική ή αντίστροφα από το στόχο (τελική κατάσταση) προς την αρχική κατάσταση, λαμβάνοντας πάντοτε υπόψη τις προϋποθέσεις και τις συνέπειες κάθε δυνατής δράσης. Πιο κάτω περιγράφουμε μερικούς από τους πιο γνωστούς αλγόριθμους.

2.3.1 Εμπρόσθια αναζήτηση στο χώρο (Forward state-space search)

Ο αλγόριθμος αυτός ξεκινά από την αρχική κατάσταση και εκτελεί διαδοχικά ένα σύνολο από δράσεις μέχρι να φτάσει στον στόχο, δηλαδή οι συνέπειες των δράσεων να εμπεριέχουν την τελική κατάσταση.



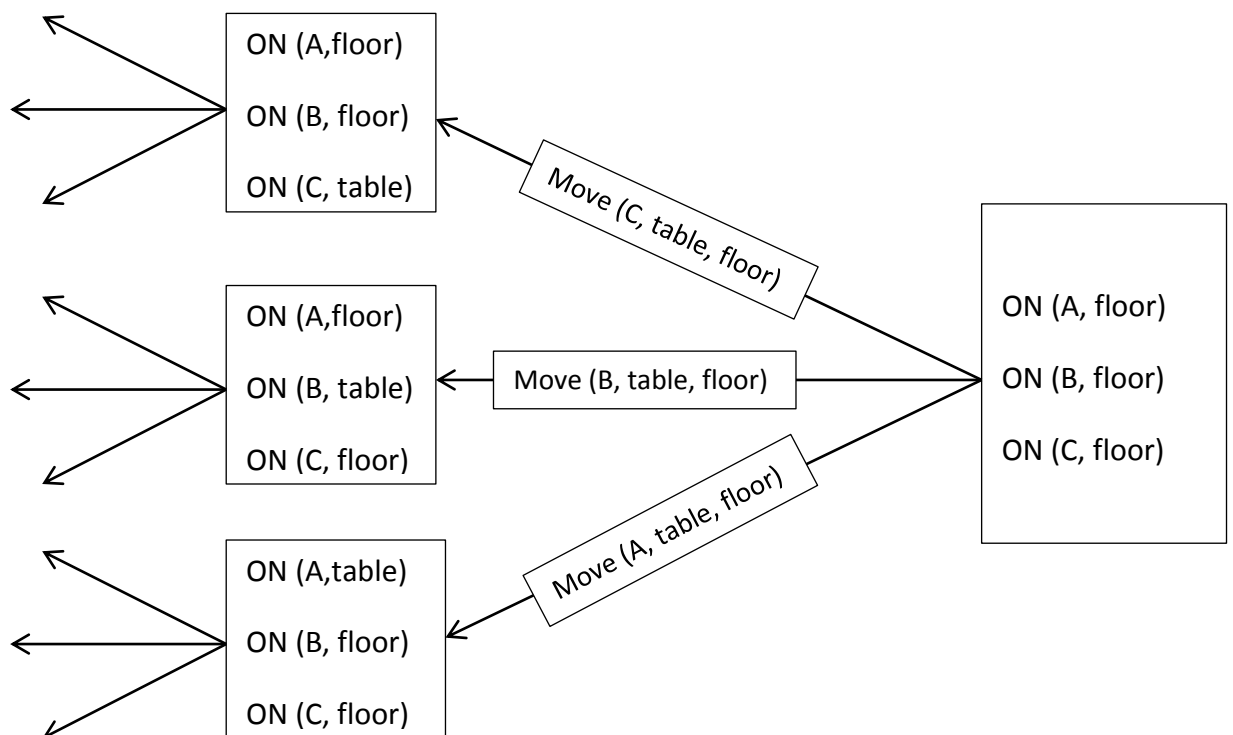
Σχήμα 2.1, Εμπρόσθια αναζήτηση στο χώρο

Στο πιο πάνω Σχήμα 2.1 ξεκινώντας από την αρχική κατάσταση εξετάζεται για όλες τις δράσεις αν είναι δυνατόν να εκτελεστούν. Ακολουθώς μέχρι να φτάσει ο αλγόριθμος σε τελική κατάσταση επαναλαμβάνει το ίδιο πράγμα από κάθε επόμενη κατάσταση. Συνεπώς, οι δυνατές δράσεις αυξάνονται εκθετικά, σε κάθε επόμενη κατάσταση.

Μειονέκτημα του αλγορίθμου είναι ότι δεν μπορεί να επιλύσει τα προβλήματα που συνδυάζουν άσχετες δράσεις αφού οι δράσεις που εκτελούνται σχετίζονται με την κάθε τρέχουσα κατάσταση. Δημιουργεί έτσι μεγάλο χώρο αναζήτησης καθώς εξετάζει όλες τις δυνατές περιπτώσεις από μια κατάσταση στην επόμενη. Ως εκ τούτου, για μεγάλα προβλήματα, με πολλές καταστάσεις και δράσεις ο αλγόριθμος δεν είναι καθόλου αποδοτικός, π.χ. για ένα πρόβλημα με 10.000 δράσεις και 100 καταστάσεις δημιουργούνται 10.000^{100} κόμβοι [1]. Υπάρχει δηλαδή πολύ μεγάλο βάθος αναζήτησης. Αναλυτικότερα χωρίς τη χρήση μιας καλής ευρετικής συνάρτησης η οποία μπορεί να αποκλείσει κάποιες δράσεις από το χώρο αναζήτησης, μπορεί να οδηγήσει και σε συνδυαστική έκρηξη μνήμης, λόγω της πιθανής έλλειψης υπολογιστικών πόρων.

2.3.2 Προς τα πίσω αναζήτηση στο χώρο (Backward state-space search)

Ο αλγόριθμος αυτός ξεκινά από την τελική κατάσταση και εκτελεί διαδοχικά ένα σύνολο από δράσεις μέχρι να φτάσει στην αρχική κατάσταση εάν αυτό είναι δυνατό. Η αρχική κατάσταση γίνεται ουσιαστικά ο στόχος.



Σχήμα 2.2, Προς τα πίσω αναζήτηση στο χώρο

Έτσι όπως παρατηρείται και στο Σχήμα 2.2 ξεκινά από το στόχο και εφαρμόζει διαδοχικά τις δράσεις, όπου κάθε δράση που εκτελείται σχετίζεται με ένα μέρος – μια σύζευξη του στόχου. Από αυτές τις δράσεις δημιουργούνται οι προηγούμενες καταστάσεις, οι οποίες αποτελούνται από τις προϋποθέσεις της συγκεκριμένης δράσης που εκτελέστηκε. Οι συνέπειες της δράσης που περιλαμβάνονται στο στόχο διαγράφονται. Η αναζήτηση τελειώνει όταν μια από της προηγούμενες καταστάσεις ικανοποιείται από την αρχική κατάσταση. Έτσι, ακολουθείται έτσι η αντίστροφη διαδικασία από αυτή του αλγορίθμου με εμπρόσθια αναζήτηση στο χώρο.

Το πλεονέκτημα είναι ότι οι δράσεις που εφαρμόζονται σχετίζονται άμεσα με το στόχο, και διατηρούν σε συνεπή κατάσταση το πρόβλημα καθώς δεν αναιρούν επιθυμητά όρους, οι οποίοι αποτελούν το στόχο. Έτσι, ο χώρος αναζήτησης είναι κατά πολύ μικρότερος σε σχέση με το χώρο αναζήτησης της εμπρόσθιας αναζήτησης στο χώρο.

2.3.3 Ευρετική αναζήτηση στο χώρο (Heuristics for state-space search)

Η ύπαρξη μιας σχετικά καλής ευρετικής συνάρτησης είναι απαραίτητη σε κάθε αλγόριθμο. Μια τέτοια συνάρτηση έχει το πλεονέκτημα να αποκλείσει περιπτώσεις στο χώρο αναζήτησης, με τον υπολογισμό της απόστασης από μια δεδομένη κατάσταση προς το στόχο, εξοικονομώντας έτσι χρονικό κόστος.

Σε μια τέτοια αναζήτηση το κόστος κάθε δράσης είναι 1, συνεπώς η απόσταση αυτή είναι ο αριθμός όλων των δράσεων. Στόχος είναι η παραδεκτή εκτίμηση, δηλαδή η μη υπερεκτίμηση. Αξίζει να σημειωθεί επίσης ότι μερικές φορές η υπερεκτίμηση οδηγεί σε ικανοποιητικά (όχι βέλτιστα) αποτελέσματα .

Υπάρχουν δύο τεχνικές που χρησιμοποιούνται για την ευρετική αναζήτηση :

Πρώτη τεχνική είναι η διάσπαση του στόχου σε υποστόχους. Το κόστος σε αυτή την τεχνική ισοδυναμεί με το άθροισμα του κόστους επίλυσης όλων των ανεξάρτητων υποστόχων. Το μειονέκτημα που εμφανίζεται είναι όταν εφαρμόζονται πλεονάζουσες δράσεις για τους υποστόχους και οι οποίες στη συνέχεια κατά την διάρκεια της συνένωσης των ανεξάρτητων πλάνων αντικαθιστώνται από μια δράση. Γίνονται δηλαδή σε αυτήν την περίπτωση αχρείαστες ενέργειες στα ανεξάρτητα υποσχήδια.

Η δεύτερη τεχνική αφορά τη «χαλάρωση» του προβλήματος (Relaxed Planning problem). Κάθε δράση με αυτήν την τεχνική θα μπορεί να εφαρμοστεί. Συγκεκριμένα, αφαιρούνται οι αρνητικές συνέπειες και ακολούθως μετράμε τον ελάχιστο αριθμό από δράσεις που πρέπει να εφαρμοστούν οι θετικές συνέπειες των οποίων θα ικανοποιούν το στόχο. Κόστος, είναι ο αριθμός των μη ικανοποιημένων στόχων που βρίσκονται στη σύζευξη, λαμβάνοντας υπόψη και την περίπτωση που μια δράση έχει ως συνέπεια τη διαγραφή ενός όρου του στόχου που πέτυχε κάποια άλλη δράση και την περίπτωση που μια δράση ικανοποιεί πέραν του ενός στόχου. Δημιουργείται όμως η ανάγκη για δυναμικό προγραμματισμό, λόγω NP δυσκολίας του προβλήματος.

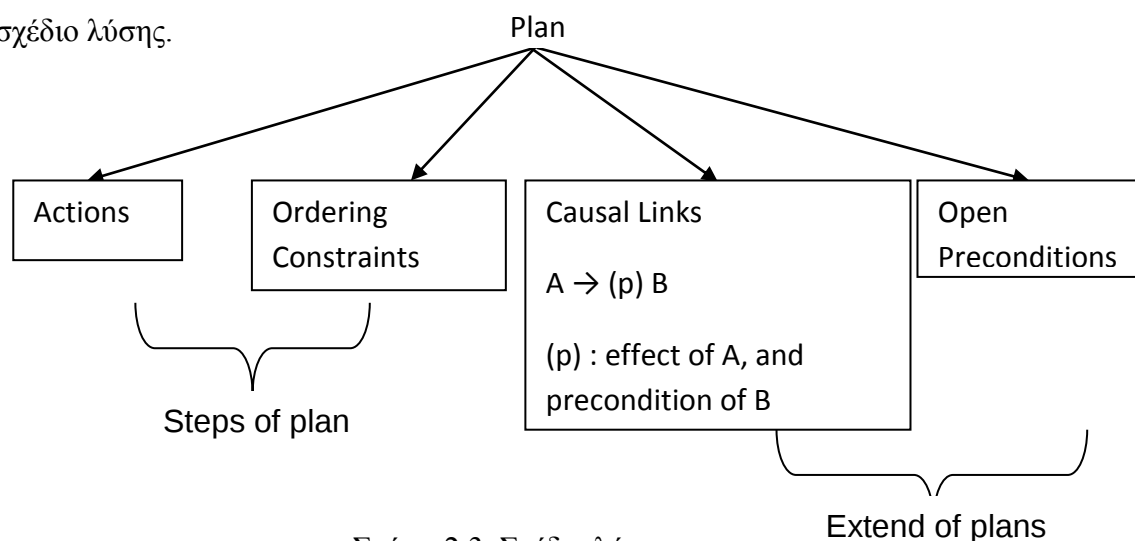
2.3.4 Προγραμματισμός με μερική σειρά (Partial-Order Planning)

Μέχρι τώρα έχουν μελετηθεί εμπρόσθιοι και οπίσθιοι αλγόριθμοι, που διερευνούν το πρόβλημα από την αρχική στην τελική κατάσταση και αντίστροφα. Εξερευνούν οι αλγόριθμοι αυτοί το χώρο, με βάση μια γραμμική ακολουθία δράσεων, οι οποίες

συνδέουν άμεσα την αρχική κατάσταση με το στόχο. Αυτή η γραμμικότητα που χρησιμοποιούν έχει ως μειονέκτημα ότι δεν μπορεί να λύσει το πρόβλημα τμηματικά, διασπώντας το δηλαδή σε υποπροβλήματα και συνενώνοντας ακολούθως τις μερικές λύσεις. Έτσι η προσέγγιση αυτή έχει περιορισμένη ευελιξία. Οι αλγόριθμοι αυτοί ονομάζονται αλγόριθμοι προγραμματισμού δράσης ολικής διάταξης.

Εισάγουμε εδώ μια νέα τεχνική-αλγόριθμο που έχει την ιδιότητα να μπορεί να λύνει ένα πρόβλημα αποσπασματικά και ακολούθως να συνδυάζει τις λύσεις. Εδώ σε αντίθεση με το προγραμματισμό δράσης ολικής διάταξης μπορούν να συνδυαστούν δυο δράσεις σε ένα σχέδιο δράσης, χωρίς να καθοριστεί η σειρά εκτέλεσής τους. Τέτοιοι αλγόριθμοι ονομάζονται αλγόριθμοι προγραμματισμού δράσης μερικής διάταξης. Σε ένα σχέδιο μερικής διάταξης μπορούν να πραγματοποιηθούν διάφορα πιθανά σχέδια.

Όπως βλέπουμε στο Σχήμα 2.3 κάθε πιθανό σχέδιο λύσης αποτελείται από τέσσερα μέρη. Στο πρώτο μέρος είναι οι δράσεις που αφορούν το κάθε σχέδιο. Ένα άδειο σχέδιο περιλαμβάνει μόνο την αρχή και το τέλος. Στο δεύτερο μέρος περιλαμβάνονται οι διάφοροι περιορισμοί σχετικά με τη σειρά εκτέλεσης ορισμένων δράσεων. Οι περιορισμοί αυτοί είναι απαραίτητοι ώστε να μην γίνονται αναιρέσεις μεταξύ τους και να μη δημιουργούνται αχρείαστοι κύκλοι ανάμεσα στις εκτελέσεις των δράσεων. Το τρίτο μέρος αφορά τους συνδέσμους αιτιότητας (causal links). Οι σύνδεσμοι αιτιότητας συσχετίζουν δύο καταστάσεις, μέσω της συνέπειας που δημιουργείται από την πρώτη κατάσταση π.χ. $A \rightarrow (p) B$, δηλαδή το A επιτυγχάνει το p για το B. Τέλος το τέταρτο μέρος αφορά το σύνολο των ανοιχτών προϋποθέσεων, όπου ανοικτή προϋπόθεση είναι οποιαδήποτε προϋπόθεση δεν έχει ικανοποιηθεί από καμία δράση που περιέχεται στο σχέδιο λύσης.



Σχήμα 2.3, Σχέδιο λύσης

Ο αλγόριθμος που εφαρμόζεται για επίλυση τέτοιου είδους προβλημάτων είναι ο εξής:

Το αρχικό σχέδιο περιλαμβάνει τις δράσεις Start και Finish, περιορισμούς διάταξης (ordering constraints) το $\text{Start} < \text{Finish}$, δεν έχει συνδέσμους αιτιότητας και έχει ανοικτές προϋποθέσεις τις συνέπειες της δράσης finish. Σε κάθε επανάληψη του αλγορίθμου επιλέγεται να εκτελεστεί μια δράση η οποία έχει ως συνέπεια αυτή την προϋπόθεση. Ο αλγόριθμος τερματίζει όταν το σύνολο των ανοικτών προϋποθέσεων είναι κενό.

Συνεπές είναι το σχέδιο στο οποίο δεν παρουσιάζονται κύκλοι κατά την εκτέλεση και δεν υπάρχουν συγκρούσεις ανάμεσα στα συνδέσμους αιτιότητας. Ταυτόχρονα συνεπές είναι το πλάνο που δεν έχει ανοικτές προϋποθέσεις. Ένα συνεπές πλάνο αποτελεί λύση του προβλήματος.

2.3.5 Ευρετικές μέθοδοι για προγραμματισμό μερικής διάταξης

Το μεγάλο πλεονέκτημα του προγραμματισμού δράσης με μερική διάταξη είναι το γεγονός ότι έχει την ιδιότητα να διασπά ένα πρόβλημα σε υποπροβλήματα και έτσι δεν είναι αναγκαία η γραμμική αναπαράσταση των καταστάσεων από την αρχική στην τελική κατάσταση. Αυτό όμως δημιουργεί παράλληλα το πρόβλημα ότι δε γνωρίζουμε πόσο κοντά ή πόσο μακριά είμαστε από την επίτευξη ενός στόχου. Δημιουργήθηκαν γι αυτό το λόγο δύο ευρετικές μέθοδοι οι οποίες προσδιορίζουν ικανοποιητικά την απόσταση από το στόχο.

Το πρώτο ευρετικό σχετίζεται με τη μέτρηση των καταστάσεων που απομένουν, και αυτό υπολογίζεται με βάση τη διαφορά μεταξύ του συνολικού αριθμού των ανοικτών προϋποθέσεων μείον τον αριθμό των ανοικτών προϋποθέσεων οι οποίες ικανοποιούνται από την αρχική κατάσταση. Αυτό ωστόσο έχει το μειονέκτημα ότι υπερεκτιμάται το κόστος σε περίπτωση που μια δράση ικανοποιεί περισσότερους από ένα στόχους. Παρατηρείται υποεκτίμηση όταν οι δράσεις έχουν αρνητικές συνέπειες, οι οποίες αλληλεπιδρούν με θετικές συνέπειες άλλων δράσεων. Το δεύτερο ευρετικό αφορά την επιλογή μιας από τις ανοικτές προϋποθέσεις, η οποία μπορεί να ικανοποιηθεί από τις πιο λίγες σε αριθμό δράσεις. Το πλεονέκτημα είναι ότι μπορεί να εντοπιστεί γρήγορα η μη δυνατότητα ικανοποίησης.

Κεφάλαιο 3

Γράφοι δράσεων και μέθοδος Satplan

3.1 Δομή – Περιγραφή του γράφου δράσεων	15
3.2 Ευρετικός υπολογισμός με βάσει το γράφημα δράσεων	18
3.3 Ο αλγόριθμος του γράφου δράσεων	19
3.4 Ο αλγόριθμος του SatPlan	20
3.4.1 Η ιδέα του SatPlan	20
3.4.2 Λειτουργία SatPlan	21
3.4.3 Κωδικοποίηση γράφου δράσεων – Κωδικοποίηση SAT-MAX-PLAN	23

3.1 Δομή – Περιγραφή του γράφου δράσεων

Διαπιστώθηκε ότι οι αλγόριθμοι με προγραμματισμό δράσης ολικής και μερικής διάταξης μπορούν να δώσουν λύση στο πρόβλημα. Ειδικά οι αλγόριθμοι με προγραμματισμό δράσης μερικής διάταξης, λόγω της διάσπασης του προβλήματος που πραγματοποιούν, αποκτούν μεγάλη ευελιξία ειδικά για μεγάλα προβλήματα και ελαχιστοποιούν το κόστος. Παράλληλα όμως τα ευρετικά που χρησιμοποιούνται δεν είναι και τόσο αποδοτικά αφού μερικές φορές είναι ανακριβή.

Με βάση αυτά τα προβλήματα δημιουργήθηκε μια νέα δομή δεδομένων, για τη διαχείριση των δράσεων η οποία ονομάζεται γράφημα δράσεων. Το γράφημα δράσεων είναι μια πιο αποτελεσματική δομή για τα προβλήματα προγραμματισμού δράσης, με ευρετικά τα οποία κάνουν πιο αξιόπιστες και ακριβείς εκτιμήσεις.

Ένα γράφημα δράσεων περιλαμβάνει μια ακολουθία από επίπεδα (levels), τα οποία αντιστοιχούν στις χρονικές στιγμές. Η αρχική κατάσταση βρίσκεται στο επίπεδο 0. Κάθε επίπεδο περιλαμβάνει ένα σύνολο από θετικούς όρους (literals) τα οποία

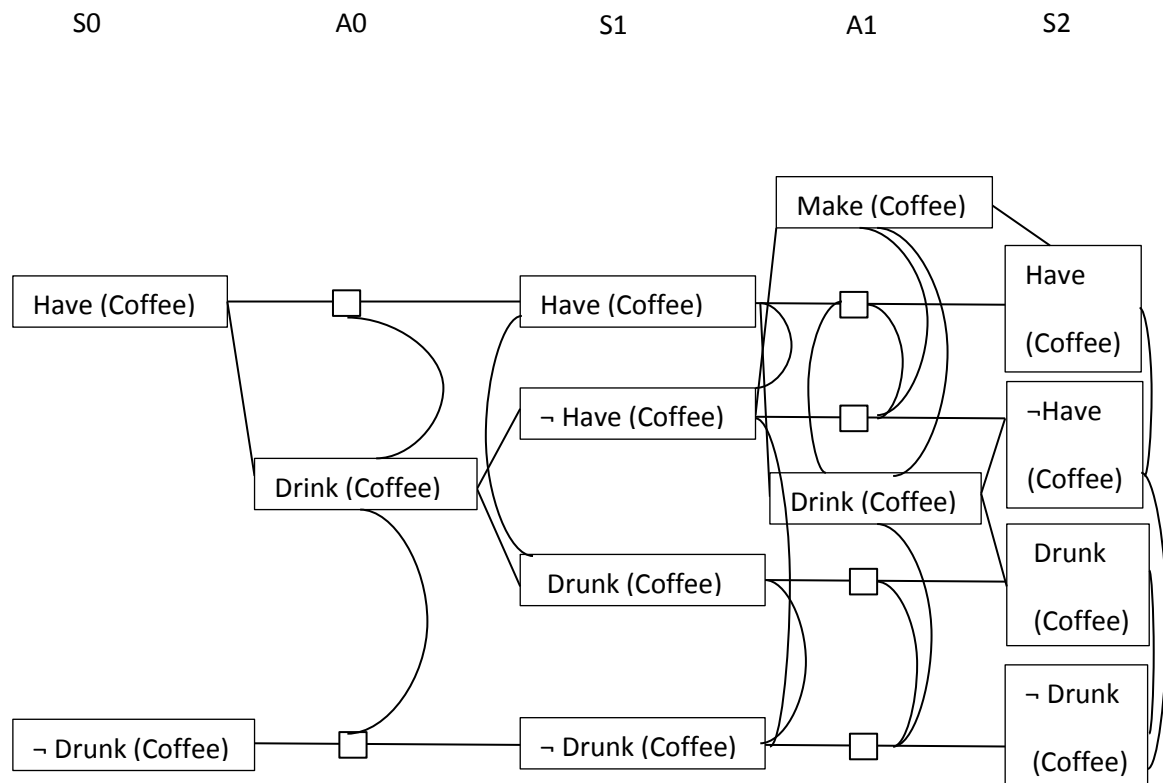
αντιστοιχούν στην τρέχουσα κατάσταση και τα οποία δημιουργήθηκαν με την εφαρμογή- εκτέλεση δράσεων του προηγούμενου επιπέδου, καθώς και ένα σύνολο από δράσεις των οποίων οι προϋποθέσεις τους ικανοποιούνται στη συγκεκριμένη κατάσταση (χρονική στιγμή). Το γράφημα δράσεων μπορεί να επιτύχει μεγάλη μείωση στο χρόνο αναζήτησης της λύσης και αυτό προκύπτει από τον τρόπο που κωδικοποιεί το όλο πρόβλημα. Σε κάθε επίπεδο καθορίζει άμεσα τους περιορισμούς, εντοπίζει δηλαδή τους αμοιβαία αποκλειόμενους όρους αλλά και τις αμοιβαία αποκλειόμενες δράσεις.

Τα επίπεδα συνεχίζουν να αυξάνονται καθώς περνούν οι αντίστοιχες χρονικές στιγμές. Σταματούν όταν διαπιστωθεί ότι δύο διαδοχικά επίπεδα είναι τα ίδια και στο τελευταίο επίπεδο δεν υπάρχει συνδυασμός από οποιουσδήποτε δύο όρους που αποτελούν το στόχο και οι οποίοι να είναι αμοιβαία αποκλειόμενοι. Σε αυτή την περίπτωση λέμε ότι το γράφημα έχει φτάσει σε κατάσταση level off.

Σημαντικές έννοιες σε ένα γράφημα δράσεων είναι οι αυτές των επίμονων δράσεων (persistence actions) καθώς και των αμοιβαία αποκλειόμενων συνδέσμων (mutual exclusion links) αλλά και των αμοιβαία αποκλειόμενων όρων. Οι επίμονες δράσεις αφορούν όρους οι οποίοι παραμένουν όπως είναι από μια κατάσταση στην επόμενη, χωρίς οποιαδήποτε δράση να τα τροποποιήσει. Οι αμοιβαία αποκλειόμενοι σύνδεσμοι, σχετίζονται με τους περιορισμούς ανάμεσα σε δύο δράσεις οι οποίες δεν μπορούν να συνυπάρχουν, δηλαδή να εκτελεστούν και οι δυο. Τέτοιοι σύνδεσμοι μεταξύ δυο δράσεων προκύπτουν από :

- i. Αντιφατικά αποτελέσματα (Inconsistent effects). Σε αυτήν την περίπτωση μια δράση έχει ως συνέπεια την αναίρεση κάποιας συνέπειας μιας άλλης δράσης.
- ii. Παρεμβολή (Interference), όταν μια δράση έχει κάποια συνέπεια η οποία αναιρεί κάποια από τις προϋποθέσεις μιας άλλης δράσης.
- iii. Ανταγωνιστικές ανάγκες (Competing needs), όταν δύο δράσεις ανταγωνίζονται για την ίδια προϋπόθεση, η μια για το θετικό αυτό όρο και η άλλη για τον αντίστοιχο αρνητικό όρο.

Σύνδεσμοι μεταξύ δύο όρων προκύπτουν όταν ο ένας όρος είναι η άρνηση του άλλου ή αν οι δυο αυτοί όροι μπορούν να είναι το αποτέλεσμα από μόνο δύο δράσεις οι οποίες είναι αμοιβαία αποκλειόμενες.



Σχήμα 3.1, Γράφημα Δράσεων

Στο Σχήμα 3.1 τα S_i είναι όροι τη χρονική στιγμή i και A_i οι δράσεις οι οποίες μπορούν να εφαρμοστούν τη χρονική στιγμή i . Οι κυρτές γραμμές μεταξύ δράσεων και μεταξύ όρων αναπαριστούν τους αμοιβαία αποκλειόμενους συνδέσμους. Επίσης οι όροι σημαίνουν:

Have (Coffee) = «Έχω καφέ»

Drink (Coffee) = «Πίνω καφέ»

Make (Coffee) = «Κάνω καφέ»

Αρχική κατάσταση (Have (Coffee))

Τελική Κατάσταση- Στόχος (Have (Coffee) $\wedge$ Drunk (Coffee))

Δράση(Drink (coffee))

Προϋπόθεση: Have (Coffee)

Συνέπειες: $\neg$ Have (Coffee) $\wedge$ Drunk (Coffee))

Δράση (Make (Coffee))

Προϋπόθεση: $\neg$ Have (Coffee)

Συνέπειες: Have (Coffee)

Στο Σχήμα 3.1 φαίνεται ότι το γράφημα έχει 3 επίπεδα. Στο πρώτο επίπεδο το S0 αντιστοιχεί στην αρχική κατάσταση, ενώ το A0 είναι οι δράσεις που μπορούν να εφαρμοστούν με βάση τις προϋποθέσεις που ισχύουν σε αυτό το επίπεδο. Στο S1 φαίνονται οι συνέπειες των δράσεων που έχουν εφαρμοστεί στο αμέσως προηγούμενο επίπεδο. Η διαδικασία αυτή επαναλαμβάνεται μέχρι δυο επίπεδα να είναι τα ίδια, να γίνει δηλαδή level off. Παράλληλα σε κάθε επίπεδο πριν δημιουργηθεί το επόμενο επίπεδο λαμβάνονται υπόψη οι αμοιβαίοι αποκλειόμενοι σύνδεσμοι ανάμεσα στις δράσεις και στους όρους.

3.2 Ευρετικός υπολογισμός με βάση το γράφημα δράσεων

Όπως και σε προηγούμενους αλγόριθμους, που έχουν μελετηθεί μέχρι τώρα η ύπαρξη ενός αποδοτικού ευρετικού μπορεί να δώσει πιο ικανοποιητικές λύσεις και να μειώσει σημαντικά το κόστος. Η ιδέα στο γράφο δράσεων είναι ότι κάθε όρος ο οποίος δεν περιλαμβάνεται στην τελική κατάσταση δεν μπορεί να συμπεριληφθεί σε κανένα σχέδιο. Με βάσει τα πιο πάνω μπορούμε να θεωρήσουμε ότι ένας όρος σε μια κατάσταση που δεν περιλαμβάνεται στην τελική κατάσταση ή μια ανοικτή προϋπόθεση έχουν κόστος $h(n)=\infty$. Από αυτήν την παρατήρηση τώρα μπορούμε να υπολογίσουμε το κόστος κάθε όρου που αποτελεί μέρος του στόχου, το οποίο ισοδυναμεί με το επίπεδο που εμφανίζεται πρώτη φορά, π.χ. στο πιο πάνω παράδειγμα το Have (Coffee) έχει κόστος 0

αφού εμφανίζεται πρώτη φορά στο επίπεδο 0 ενώ αντίστοιχα το Drunk (Coffee) έχει κόστος 1.

Παρόλο που εκ πρώτης όψεως τα ευρετικά αυτά φαίνονται να είναι ακριβή, οι εκτιμήσεις που κάνουν δεν είναι πολύ αποδοτικές, δεδομένου ότι σε κάθε επίπεδο μπορούν να εφαρμοστούν πολλές δράσεις και αφού το ευρετικό μετρά το επίπεδο και όχι το σύνολο των δράσεων στο κάθε επίπεδο. Γι' αυτό δημιουργήθηκαν μεταξύ των δράσεων αμοιβαία αποκλειόμενοι σύνδεσμοι.

Έτσι για τον υπολογισμό του κόστους δημιουργήθηκαν ευρετικά που υπολογίζουν το κόστος του στόχου που βρίσκεται σε σύζευξη. Τρία απλά στην έννοια ευρετικά είναι τα εξής:

- (i) max-level, βρίσκει τον όρο του στόχου με το μεγαλύτερο κόστος επιπέδου.
- (ii) level sum, βρίσκει το άθροισμα από τα κόστη επιπέδων των literals του στόχου.
- (iii) set-level, βρίσκει το επίπεδο στο οποίο όλα τα literals του στόχου εμφανίζονται, χωρίς οποιαδήποτε δυο ζεύγη να είναι αμοιβαία αποκλειόμενα.

3.3 Ο αλγόριθμος του γράφου δράσεων

Ο αλγόριθμος του γράφου δράσεων γίνεται ουσιαστικά σε δύο φάσεις. Στην πρώτη φάση προσπαθεί να διαπιστωθεί κατά πόσο υπάρχει ή όχι λύση για ένα δεδομένο πρόβλημα. Στη δεύτερη φάση και εφόσον υπάρχει κάποια λύση, δηλαδή το πρόβλημα μπορεί να ικανοποιηθεί, υπολογίζει την ίδια τη λύση. Πιο συγκεκριμένα, η λειτουργία του αλγορίθμου περιγράφεται πιο κάτω.

Αρχικοποιεί τον γράφο με την αρχική κατάσταση και προσδιορίζει το στόχο. Στη συνέχεια επαναλαμβάνει για ένα συγκεκριμένο αριθμό επιπέδων (maxlevel) το εξής: σε κάθε επίπεδο έστω k ελέγχει αν εμφανίζονται οι όροι του στόχου και αν δεν υπάρχει αμοιβαίος αποκλεισμός μεταξύ οποιονδήποτε δυο. Εάν αυτό ισχύει, τότε εξάγει τη λύση από το τρέχον γράφημα αν υπάρχει, αλλιώς επεκτείνει το γράφημα στο επόμενο επίπεδο, έστω $k+1$ (εφαρμόζει δηλαδή τις δράσεις που μπορούν να εκτελεστούν στην τρέχουσα κατάσταση).

Εάν και εφόσον υπάρχει λύση αυτή εξάγεται με τρόπο ανάλογο του αλγορίθμου αναζήτησης προς τα πίσω στο χώρο, δηλαδή ξεκινάμε από το τελευταίο επίπεδο που είναι ο στόχος και πάμε στην αρχική κατάσταση, εφαρμόζοντας τις αντίστοιχες δράσεις. Ο αλγόριθμος του γράφου δράσεων είναι αποδεδειγμένο ότι τερματίζει. Επίσης κατά την εξαγωγή της λύσης μπορούν να χρησιμοποιηθούν ευρετικά για την επιλογή των δράσεων κατά τη διαδικασία της αναζήτησης προς τα πίσω όπως: ικανοποίηση πρώτα του όρου με το μεγαλύτερο κόστος επιπέδου ή επιλογή δράσης της οποίας οι προϋποθέσεις της έχουν το μικρότερο συνολικό άθροισμα του κόστους όλων των επιπέδων.

3.4 Ο αλγόριθμος του SatPlan

3.4.1 Η ιδέα του SatPlan

Το 1992, οι Kautz & Selman εισήγαγαν την ιδέα τους για μετασχηματισμό του προβλήματος προγραμματισμού δράσης σε πρόβλημα ικανοποιησιμότητας προτασιακής λογικής [4]. Με βάση αυτήν την ιδέα εμφανίστηκε το ολοκληρωμένο σύστημα SatPlan το 1996, από τους Kautz & Selman, που στηρίζεται σε προτασιακή λογική. Απέδειξαν ότι είναι εξίσου αποδοτικό με τις υπόλοιπες τεχνολογίες προβλημάτων δράσης και μπορεί να είναι ανταγωνιστικό [5].

Το 2006 δημιουργήθηκε μια νέα έκδοση του SatPlan, το SatPlan-2006, το οποίο βελτίωσε την αποδοτικότητα της λύσης με πιο γρήγορους υπολογισμούς, λαμβάνοντας υπόψη τους αμοιβαία αποκλειόμενους περιορισμούς που αφορούν στις συνθήκες και συνέπειες των δράσεων [4]. Το SatPlan-2006 παίρνει σαν είσοδο τα προβλήματα που ανήκουν στο υποσύνολο STRIPS της γλώσσας PDDL (Planning Domain Definition Language). Προσπαθεί να βρει λύσεις με βάση το μικρότερο παράλληλο μήκος. Αυτό που προσπαθεί να κάνει είναι να υπολογίζει το μικρότερο σύνολο χρονικών βημάτων μέσα από τις πολλές πράξεις που μπορεί να εμφανίζονται παράλληλα σε κάθε χρονικό διάστημα [4]. Μέχρι σήμερα παραμένει ένα από τα κορυφαία συστήματα όσο αφορά την επίλυση προβλημάτων προγραμματισμού δράσης και διάφοροι ερευνητές προσπαθούν να το επεκτείνουν βελτιώνοντας την αποδοτικότητα του με την εφαρμογή νέων μεθόδων ή την τροποποίηση των απλούστερων υπολογιστικών «κομματιών» του.

3.4.2 Λειτουργία SatPlan

Το ολοκληρωμένο σύστημα SatPlan λειτουργεί σε συνδυασμό με το γράφημα δράσεων, το οποίο περιγράφηκε προηγουμένως. Συγκεκριμένα ο SatPlan κατασκευάζει αρχικά το γράφημα μέχρι κάποιο επίπεδο k . Κατά την κατασκευή του γράφου σε κάθε επίπεδο k , παράγει από τους περιορισμούς που προκύπτουν από το πρόβλημα μια συζευκτική κανονική μορφή (cnf). Η cnf μορφή στην ουσία είναι η μετάφραση των περιορισμών αυτών σε συζευκτική μορφή, δηλαδή πολλές προτάσεις (διαζεύξεις προτάσεων) που η καθεμία έχει ένα σύνολο από όρους. Οι όροι είναι μεταβλητές που αποτιμούνται σε true ή false και αντιπροσωπεύουν τους ίδιους τους περιορισμούς σε κάθε βήμα. Στόχος εφόσον έχει δημιουργήσει αυτές τις προτάσεις είναι να εξετάσει κατά πόσο υπάρχει λύση δηλαδή αν είναι δυνατό να ικανοποιηθούν αυτές οι προτάσεις (clauses) και αν ναι να βρει μια τέτοια ανάθεση στις μεταβλητές (ποιές μεταβλητές θα πάρουν τιμή true και ποιές false). Η διεκπεραίωση αυτής της λειτουργίας του γίνεται με τη βοήθεια ενός ανεξάρτητου προτασιακού επιλυτή (solver). Ο κάθε επιλυτής με δικούς του αλγοριθμικούς μηχανισμούς εξετάζει κατά πόσο το πρόβλημα μπορεί να ικανοποιηθεί ή όχι. Αν το πρόβλημα μπορεί να ικανοποιηθεί, τότε δημιουργεί μια λύση η οποία περιλαμβάνει μια τέτοια ανάθεση στις μεταβλητές ώστε όλες οι συζεύξεις (clauses) να γίνονται true και επιστρέφει τη λύση στο SatPlan. Εφόσον ο SatPlan παίρνει τη λύση, εκτελεί μια ακόμη επιπλέον επεξεργασία αφαιρώντας αχρείαστες δράσεις. Η λειτουργία αυτή είναι αναγκαία για το λόγο ότι η μια δράση μπορεί να μην είναι απαραίτητη για την επίτευξη των στόχου του αρχικού προβλήματος. Αν το πρόβλημα δεν μπορεί να ικανοποιηθεί ο επιλυτής επιστρέφει UNSAT στον SatPlan και επαναλαμβάνεται η ίδια διαδικασία για το επόμενο επίπεδο του γράφου [4].

Όσο αφορά στον τρόπο και τη μεθοδολογία με την οποία μεταφράζονται οι περιορισμοί, ώστε να είναι σε συζευκτική μορφή έχουν δημιουργηθεί διάφορες κωδικοποιήσεις. Αξίζει να σημειωθεί ότι η καλή απόδοση που έχει ο SatPlan προκύπτει λόγω του συνδυασμού της απόδοσης που έχουν οι προτασιακοί επιλυτές (solvers) και των περιορισμών που έχουν δημιουργηθεί από το γράφημα σε κάθε επίπεδο. Σημαντική είναι επίσης η έννοια της διάδοσης περιορισμών (constraint propagation), που αφορά την ανάθεση τιμών στις νέες μεταβλητές από μεταβλητές ήδη γνωστές. Ισχυρές μορφές διάδοσης μεταβλητών δίνουν περισσότερες τιμές για τις μεταβλητές.

Αυτό δίνει το πλεονέκτημα διότι γίνεται αναζήτηση σε μικρότερο χώρο πλέον και έτσι αυξάνεται η απόδοση. Πολλοί προτασιακοί επιλυτές (solvers) όπως ο precosat και ο minisat χρησιμοποιούν την έννοια της μοναδιαίας ανάλυσης (unit resolution) για να κάνουν πιο αποδοτική διάδοση των περιορισμών. Η μοναδιαία ανάλυση κάνει ανάλυση ανάμεσα στους περιορισμούς των διαζεύξεων μέχρι το σημείο που αυτοί οι περιορισμοί είναι αδύνατο να αναλυθούν σε πιο απλή μορφή.

Παράδειγμα Unit Resolution :

1^ο βήμα : $\neg X \vee Y \vee Z$ X (δεδομένου του X)
 2^ο βήμα : $Y \vee Z$ $\neg Y$ (δεδομένου του $\neg Y$)
 3^ο βήμα : Z

Με βάση τα πιο πάνω ο αλγόριθμος για το SatPlan είναι ο ακόλουθος :

```
function SATPLAN(problem,T max) returns solution or failure
  inputs: problem, a planning problem
  T max, an upper limit for plan length
  for T = 0 to T max do
    cnf ← TRANSLATE-TO-SAT(problem,T)
    assignment SAT-SOLVER(cnf)
    if assignment is not null then
      return EXTRACT-SOLUTION(assignment, mapping)
  end for
  return failure
```

3.4.3 Κωδικοποίηση γράφου δράσεων – Κωδικοποίηση SAT-MAX-PLAN

Καθώς το γράφημα δράσεων κτίζεται σε κάθε επίπεδο μπορούν να εξαχθούν από την αναπαράσταση αυτή, οι διάφορες μορφές από διαζευκτικές προτάσεις. Οι προτάσεις αυτές αποτελούν την άμεση κωδικοποίηση από το ίδιο το γράφημα. Έχουν δημιουργηθεί κατά καιρούς διάφορες κωδικοποιήσεις. Οι κωδικοποιήσεις στηρίζονται στις ακόλουθες προτάσεις:

1. Μοναδιαίες διαζεύξεις για την αρχική και τελική κατάσταση.
2. $A(T) \rightarrow f(T)$, για κάθε δράση A και για κάθε όρο f , όπου $f \in \text{pre}(A)$.
3. $A(T) \rightarrow f(T + 1)$, για κάθε δράση A και για κάθε όρο f , όπου $f \in \text{add}(A)$.
4. $A(T) \rightarrow \neg f(T + 1)$, για κάθε δράση A και για κάθε όρο f , όπου $f \in \text{del}(A)$.
5. $f(T) \rightarrow A_1(T - 1) \vee \dots \vee A_m(T - 1)$, για κάθε όρο f και για όλες τις δράσεις A_i , όπου $1 \leq i \leq m$ (συμπεριλαμβανομένου και των αδρανών (noop) δράσεων) και όπου $f \in \text{add}(A_i)$.
6. $\neg f(T) \rightarrow A_1(T - 1) \vee \dots \vee A_m(T - 1) \vee \neg f(T - 1)$, για κάθε όρο f και για όλες τις δράσεις A_i , όπου $1 \leq i \leq m$ και όπου $f \in \text{del}(A_i)$.
- 7.1 $\neg A_1(T) \vee \neg A_2(T)$, για κάθε ζεύγος δράσεων A_1, A_2 τέτοιο ώστε το σύνολο $\text{del}(A_1) \cap \text{pre}(A_2) \neq \text{κενού συνόλου}$.
- 7.2 $\neg A_1(T) \vee \neg A_2(T)$, για κάθε ζεύγος δράσεων A_1, A_2 τέτοιο ώστε το σύνολο $\text{del}(A_1) \cap \text{add}(A_2) \neq \text{κενού συνόλου}$.
- 7.3 $\neg A_1(T) \vee \neg A_2(T)$, εάν υπάρχει ζεύγος όρων f_1, f_2 όπου $f_1 \in \text{pre}(A_1)$, $f_2 \in \text{pre}(A_2)$, τέτοιο ώστε οι f_1 και f_2 να είναι αμοιβαία αποκλειόμενοι τη χρονική στιγμή T .
- 8 $\neg f_1(T) \vee \neg f_2(T)$, για κάθε ζεύγος όρων f_1, f_2 οι οποίοι είναι αμοιβαία αποκλειόμενοι τη χρονική στιγμή T .

Επεξηγήσεις συμβολισμών :

$\text{pre}(A)$ = preconditions (προϋποθέσεις) της δράσης A

$\text{add}(A)$ = θετικές συνέπειες της δράσης A

$\text{del}(A)$ = αρνητικές συνέπειες της δράσης A

Η Blackbox είναι μια κωδικοποίηση για προβλήματα planning η οποία δημιουργήθηκε το 1999 από τους Kautz και Selman. Σε σχέση με την κωδικοποίηση SatPlan06 που έγινε το 2006, η Blackbox είναι μια πιο ισχυρή μορφή. Ωστόσο η κωδικοποίηση την οποία θα χρησιμοποιήσουμε κατά κύριο λόγο στη συνέχεια για την επίλυση προβλημάτων είναι η SAT-MAX-PLAN (SMP). Η SMP θεωρείται η καλύτερη κωδικοποίηση καθώς επιτυγχάνει μεγαλύτερη διάδοση περιορισμών τόσο από αυτή της Blackbox όσο και από αυτή της SatPlan06. Αυτό διότι η διάδοση γίνεται από σύνολα που δεν περιέχουν πλεονασμούς [2]. Η SMP ως κωδικοποίηση χρησιμοποιεί τις προτάσεις: 1, 2, 3, 4, 5, 6, 7.1, 8 όπως αυτές περιγράφηκαν πιο πάνω. Επίσης, όλες οι προτάσεις που αφορούν δύο δράσεις έστω $A1$, $A2$ που είναι αμοιβαία αποκλειόμενες και περιλαμβάνουν θετικές (add) συνέπειες $p1$ και $p2$ αντίστοιχα που είναι επίσης αμοιβαία αποκλειόμενες τη στιγμή $T+1$, μπορούν να παραβλεφθούν.

Θα χρησιμοποιήσουμε την SMP κυρίως σε συνδυασμό με τον προτασιακό επιλυτή minisat+ στα πλαίσια της προσπάθειας της ψευδοδυαδικής βελτιστοποίησης, αλλά και με τον προτασιακό επιλυτή precosat για λόγους σύγκρισης μεταξύ των δύο [2].

Κεφάλαιο 4

Ψευδοδυαδικοί περιορισμοί και περιορισμοί πληθικότητας

4.1 Γενικά	25
4.2 Ψευδοδυαδική συνάρτηση	26
4.3 Ψευδοδυαδικοί περιορισμοί	26
4.3.1 Γραμμικοί Ψευδοδυαδικοί περιορισμοί	27
4.3.2 Μη Γραμμικοί Ψευδοδυαδικοί περιορισμοί	27
4.4 Περιορισμοί Πληθικότητας	28
4.4.1 Κανονικοποίηση Ψευδοδυαδικών περιορισμών	30
4.5 Ικανοποιησιμότητα και Βελτιστοποίηση	31
4.6 Βελτιστοποίηση με ψευδοδυαδικούς περιορισμούς	31
4.7 Λειτουργία αλγορίθμου των ψευδοδυαδικών επιλυτών	32
4.8 Η σημασία των ψευδοδυαδικών περιορισμών	33

4.1 Γενικά

Οι ψευδοδυαδικοί περιορισμοί και οι ψευδοδυαδικές συναρτήσεις ξεκίνησαν τα τελευταία χρόνια να έχουν μεγάλη εφαρμογή σε προβλήματα προγραμματισμού δράσης, λόγω μεγαλύτερης ευελιξίας στην έκφραση των δυνατών περιπτώσεων του προβλήματος. Είναι μια μορφή, η οποία δίνει τη δυνατότητα έκφρασης τέτοιων προβλημάτων με μεγαλύτερη εκφραστικότητα αλλά και με πιο φυσικό τρόπο σε σχέση με τα άλλα είδη περιορισμών. Αυτό γιατί με αυτή τη μορφή η ανάθεση στις τιμές των μεταβλητών γίνεται και πάλι από το δυαδικό σύνολο $(0,1)$, αλλά η αποτίμηση μιας πρότασης με ψευδοδυαδικούς περιορισμούς επεκτείνεται στο σύνολο όλων των ακεραίων αριθμών. Μια τέτοια αποτίμηση έχει ως αποτέλεσμα οι περιορισμοί να εκφράζονται με περισσότερη απλότητα και να είναι πιο κοντά στο φυσικό τρόπο σκέψης του ανθρώπου. Διάφοροι ερευνητές όπως οι Marques-Silva & Sakallah 1996 Een & Sorensson το 2003, εστίασαν την έρευνα τους γύρω από επεκτάσεις που μπορούσαν να

γίνουν σε προβλήματα προτασιακής ικανοποιησιμότητας (SAT), με τη χρήση ψευδοδυαδικής βελτιστοποίησης (Pseudo-Boolean Optimization).

4.2 Ψευδοδυαδική συνάρτηση

Η ψευδοδυαδική συνάρτηση είναι μια συνάρτηση, η οποία αφού αντιστοιχήσει στις μεταβλητές δυαδικές τιμές (Boolean), μπορεί να κάνει αποτίμηση μιας πρότασης με τιμή από το σύνολο των όλων ακεραίων αριθμών και όχι κατά ανάγκη με 0 ή 1. Οι ψευδοδυαδικές συναρτήσεις έγιναν κατά κύριο λόγο αντικείμενο μελέτης κοντά στα μέσα του 1960 στο πλαίσιο της μελέτης των τελεστών του 0-1 προγραμματισμού. Ο ορισμός «ψευδοδυαδικοί» προέρχεται από το γεγονός ότι αν και δεν είναι δυαδικές συναρτήσεις έχουν μεγάλες ομοιότητες ως προς τον τρόπο λειτουργίας τους [6].

Οι ψευδοδυαδικές συναρτήσεις αποτελούν ένα ενδιαφέρον θέμα προς μελέτη και ένα ανοικτό θέμα για έρευνα, καθώς πολλά από τα προβλήματα μπορούν να εκφραστούν σε προβλήματα βελτιστοποίησης της τιμής μιας ψευδοδυαδικής συνάρτησης. Μπορεί επίσης να εξασφαλιστεί πιο εύκολα σε κάποιες περιπτώσεις η ικανοποίηση των περιορισμών από ότι στην περίπτωση που οι μεταβλητές των προτάσεων πρέπει να ικανοποιηθούν με ένα συνδυασμό αναθέσεων στις μεταβλητές με τιμές στους συντελεστές μόνο μεταξύ 0 και 1. Παρέχουν με αυτό τον τρόπο οι ψευδοδυαδικές συναρτήσεις μεγαλύτερη ευελιξία στην προσπάθεια ικανοποίησης των περιορισμών του προβλήματος.

4.3 Ψευδοδυαδικοί περιορισμοί

Μέχρι τώρα η ανάθεση τιμών στις μεταβλητές γινόταν με μόνο δύο τιμές 0 ή 1 (false ή true). Κάθε όρος έστω i σε κάθε πρόταση μπορούσε να πάρει τιμή X_i ή $\overline{X_i}$. Με αυτή την ανάθεση ένας θετικός όρος δηλαδή X_i ισοδυναμεί με true εάν και μόνο εάν το X_i είναι true, ενώ αντίστοιχα ένα αρνητικός όρος δηλαδή $\overline{X_i}$ ισοδυναμεί με true εάν και μόνο εάν το X_i είναι false.

Στους ψευδοδυαδικούς περιορισμούς ένας όρος παίρνει τιμή 0 ή 1. Ο όρος αυτός αφού αποτιμάται σε true ή false, πολλαπλασιάζεται με ένα ακέραιο αριθμό, που είναι ο συντελεστής του. Δηλαδή μαθηματικά αυτό ορίζεται ως εξής :

$$\forall a \in \mathbb{Z}, a.T = a \text{ και } a.F = 0$$

4.3.1 Γραμμικοί Ψευδοδυαδικοί περιορισμοί

Ένας γραμμικός ψευδοδυαδικός περιορισμός έχει την μορφή :

$$\sum_j a_j l_j \blacktriangleright b,$$

όπου τα a_j και b είναι ακέραιοι αριθμοί, το l_j είναι όρος και το $\blacktriangleright$ προσδιορίζει ένα σχεσιακό τελεστή όπως $=, >, <, \geq, \leq$. Το δεξιό μέρος χαρακτηρίζει το βαθμό του περιορισμού. Ο περιορισμός που είναι αυτής της γραμμικής ψευδοδυαδικής μορφής ικανοποιείται όταν το δεξί και αριστερό μέρος του περιορισμού πάρουν τέτοιες τιμές ακεραίων τέτοιες ώστε να ικανοποιείται ο σχεσιακός τελεστής.

Παραδείγματα γραμμικών ψευδοδυαδικών περιορισμών :

$$\blacktriangleright 2.X_1 + X_2 \geq 3$$

Ικανοποιείται για $X_1=1$ και $X_2=1$.

Για οποιουδήποτε άλλους συνδυασμούς τιμών δεν ικανοποιείται.

$$\blacktriangleright 3.\overline{X_1} + 2.X_2 < 0$$

Δεν ικανοποιείται ποτέ.

$$\blacktriangleright 4.X_1 - 3.\overline{X_2} \geq 0$$

Ικανοποιείται για $X_1=1$ και $X_2=1$.

Δεν ικανοποιείται για $X_1=0$ και $X_2=0$.

4.3.2 Μη Γραμμικοί Ψευδοδυαδικοί περιορισμοί

Ένας μη γραμμικός ψευδοδυαδικός περιορισμός έχει την μορφή :

$$\sum_j a_j . (\prod_k l_{j,k}) \blacktriangleright b$$

Όπου τα a, j και b είναι ακέραιοι αριθμοί, το l, j, k είναι όροι και το $\blacktriangleright$ προσδιορίζει ένα σχεσιακό τελεστή όπως $=, >, <, \geq, \leq$. Σε ένα τέτοιο ψευδοδυαδικό περιορισμό, το γινόμενο του κάθε αθροίσματος, περιλαμβάνει ένα σύνολο από όρους. Ο κάθε όρος του γινομένου μπορεί να πάρει μόνο τις δυαδικές τιμές $\{0,1\}$. Το αποτέλεσμα του γινομένου ισοδυναμεί με τη λογική πράξη του AND, απλά εδώ αντί false και true, οι όροι παίρνουν τιμές 0 και 1, που είναι ουσιαστικά το ίδιο. Ο περιορισμός που είναι αυτής της μη γραμμικής ψευδοδυαδικής μορφής ικανοποιείται όταν το δεξί και αριστερό μέρος του περιορισμού πάρουν τέτοιες τιμές ακεραίων ώστε να ικανοποιείται ο σχεσιακός τελεστής.

Παραδείγματα μη γραμμικών ψευδοδυαδικών περιορισμών :

$$\blacktriangleright 5.X_1.X_2 + 2.X_2 \geq 6$$

Ικανοποιείται για $X_1=1$ και $X_2=1$.

Δεν ικανοποιείται για $X_1=0, X_2=1$.

$$\blacktriangleright 4. \overline{X_1} + X_1 + 2.X_2 + \overline{X_1}.X_2.\overline{X_3} \geq 7$$

Ικανοποιείται για $X_1=0, X_2=1, X_3=0$.

Δεν ικανοποιείται για $X_1=1, X_2=1, X_3=1$.

$$\blacktriangleright 2.X_1.X_2 + X_2 \geq 4$$

Δεν ικανοποιείται με κανένα συνδυασμό τιμών – Δεν ικανοποιείται ποτέ.

$$\blacktriangleright 2.X_1.X_2 + X_2 \geq -1$$

Ικανοποιείται για όλους τους συνδυασμούς τιμών για τα X_1 και X_2 .

4.4 Περιορισμοί Πληθικότητας

Οι περιορισμοί πληθικότητας σχετίζονται με το πλήθος των όρων που απαιτείται να έχουν τιμή «αληθές» σε μια συγκεκριμένη πρόταση. Ανάλογα με την πρόταση, υπάρχουν τρεις διαφορετικοί τύποι περιορισμών πληθικότητας.

Πρώτο είδος περιορισμών είναι οι περιορισμοί at least. Σε αυτή την περίπτωση ο περιορισμός ικανοποιείται (είναι δηλαδή true), εάν και μόνο εάν τουλάχιστο k όροι, από

ένα σύνολο $x_1, x_2, \dots, x_n$ με πλήθος n όρους, είναι true. Ο περιορισμός αυτός έχει την μορφή $\text{atleast}(k, \{x_1, x_2, \dots, x_n\})$. Σε ψευδοδυαδική μορφή, ο περιορισμός αυτός γίνεται $x_1 + x_2 + \dots + x_n \geq k$.

Δεύτερο είδος περιορισμών είναι οι περιορισμοί at most. Ο περιορισμός ικανοποιείται (είναι δηλαδή true), εάν και μόνο εάν το πολύ k όροι, από ένα σύνολο $x_1, x_2, \dots, x_n$ με πλήθος n όρους, είναι true. Η μορφή αυτού του περιορισμού είναι $\text{atmost}(k, \{x_1, x_2, \dots, x_n\})$. Η αντίστοιχη ψευδοδυαδική μορφή είναι $x_1 + x_2 + \dots + x_n \leq k$.

Τέλος, ο τελευταίος τύπος περιορισμού είναι οι περιορισμοί exact. Ο περιορισμός εδώ είναι true, εάν και μόνο εάν ακριβώς k όροι, από ένα σύνολο $x_1, x_2, \dots, x_n$ με πλήθος n όρους, είναι true. Η μορφή αυτού του περιορισμού είναι $\text{exactly}(k, \{x_1, x_2, \dots, x_n\})$. Σε ψευδοδυαδική μορφή αυτό μεταφράζεται με βάση τη σχέση :

$$\sum_{j=1}^n ax_j \geq b \equiv \text{atleast}(\lceil b/a \rceil, \{x_1, x_2, \dots, x_n\})$$

Με βάση τα πιο πάνω εξάγονται κάποια συμπεράσματα που αφορούν ισοδυναμίες ανάμεσα στους διαφορετικούς τύπους των 3 πιο πάνω περιορισμών. Είναι φανερή η ισοδυναμία $\text{atmost}(k, \{x_1, x_2, \dots, x_n\}) \equiv \text{atleast}(n-k, \{\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}\})$, αφού αυτό ερμηνεύεται ως εξής : όχι περισσότεροι από $n-k$ όροι να είναι false. Επίσης μπορούμε να συμπεράνουμε ότι $\text{exactly}(k, \{x_1, x_2, \dots, x_n\}) \equiv \text{atmost}(k, \{x_1, x_2, \dots, x_n\}) \wedge \text{atleast}(k, \{x_1, x_2, \dots, x_n\})$. Με βάση τα πιο πάνω είναι σαφές ότι ο περιορισμός atleast μπορεί να εκφράσει οποιοδήποτε περιορισμό πληθικότητας. Μπορεί να μεταφραστεί παράλληλα σε διαζεύξεις από προτάσεις (clauses). Ο περιορισμός είναι ισοδύναμος με το συνδυασμό όλων των πιθανών προτάσεων που λαμβάνονται όταν επιλεγούν $n - k + 1$ όροι μέσα από το σύνολο $\{x_1, x_2, \dots, x_n\}$, δηλαδή $\binom{n}{n-k+1}$. Αυτό ισχύει καθώς από τον περιορισμό $\text{atleast}(k, \{x_1, x_2, \dots, x_n\})$ προκύπτει ότι είναι αδύνατο περισσότεροι από $n - k$ όροι να πάρουν την τιμή false και συνεπώς από τους $n - k + 1$ όρους, τουλάχιστο ένας θα είναι true [3].

4.4.1 Κανονικοποίηση Ψευδοδυαδικών περιορισμών

Η κανονικοποίηση των περιορισμών χρειάζεται ώστε να υπάρχει ένας κοινός συμβολισμός, ανάμεσα στους περιορισμούς που έχουν οι διάφορες προτάσεις. Αυτό μπορεί να βοηθήσει ώστε να ληφθεί πιο γρήγορα η απόφαση αν μπορεί ή όχι να ικανοποιηθεί μια συγκεκριμένη πρόταση. Νοούμενου ότι μια πρόταση μπορεί να ικανοποιηθεί, εάν είναι κανονικοποιημένη βοηθά τον επιλυτή που θα ψάξει να βρει για μια τέτοια ανάθεση στους όρους, να την εντοπίσει πιο γρήγορα λόγω πιο ευέλικτων μηχανισμών που θα μπορεί να χρησιμοποιήσει. Θα μελετήσουμε στη συνέχεια τον τρόπο με τον οποίο ένας ψευδοδυαδικός περιορισμός μπορεί να κανονικοποιηθεί. Πιο συγκεκριμένα, η μορφή στην οποία θα κανονικοποιήσουμε τον περιορισμό είναι:

$\sum_{j=1}^n a_j l_j \geq b$, $a_j, b \in \mathbb{N}^+$, όπου τα a_j και b είναι μη αρνητικοί φυσικοί αριθμοί, το l_j είναι είτε το x_j είτε το $\bar{x}_j$.

Για την κανονικοποίηση ενός οποιουδήποτε περιορισμού ακολουθούνται τα πιο κάτω βήματα :

Αν ο τελεστής είναι $>$, τότε προσθέτουμε 1 από το δεξιό μέλος και αλλάζουμε τον τελεστή σε $\geq$.

Αν ο τελεστής είναι $<$, αφαιρούμε 1 από το δεξιό μέλος και αλλάζουμε τον τελεστή σε $\leq$.

Αν ο τελεστής είναι $=$, αντικαθιστούμε την παράσταση με δύο ίδιες παραστάσεις η μια με τελεστή $\geq$ και η άλλη με τελεστή $\leq$.

Αν ο περιορισμός έχει το σύμβολο $\leq$, πολλαπλασιάζουμε και τα δύο μέλη με -1 και αλλάζουμε τον τελεστή σε $\geq$.

Αλλάζουμε τέλος τους όρους που έχουν αρνητικούς συντελεστές δηλαδή,

$$x_j = 1 - (\bar{x}_j) \quad (\bar{x}_j) = 1 - x_j$$

4.5 Ικανοποιησιμότητα και Βελτιστοποίηση

Υπάρχουν δύο κατηγορίες όσον αφορά τα προβλήματα ικανοποίησης ψευδοδυαδικών περιορισμών. Έχουμε καταρχήν τα προβλήματα απόφασης, όπου η απάντηση είναι μια. Με βάση δηλαδή τους ψευδοδυαδικούς περιορισμούς μια φόρμουλα αναγνωρίζει κατά πόσο το πρόβλημα ικανοποιείται ή όχι, δηλαδή αν υπάρχει μια τέτοια ανάθεση που κάνει την πρόταση αληθή, σύμφωνα με αυτούς τους περιορισμούς. Αυτά τα προβλήματα αποφάσεων ονομάζονται Pseudo-Boolean Solving.

Επιπλέον, υπάρχουν όπως είναι αναμενόμενο και τα προβλήματα βελτιστοποίησης. Σε αυτήν την κατηγορία προβλημάτων εξετάζεται κατά πόσο ένα πρόβλημα έχει λύση και εάν έχει λύση ευρίσκεται η πιο ικανοποιητική- καλύτερη λύση για τη συγκεκριμένη φόρμουλα περιορισμών. Η απάντηση δηλαδή που δίνει ένας προτασιακός επιλυτής σε αυτήν την περίπτωση δεν είναι μόνο αν οι περιορισμοί μπορούν ή δεν μπορούν να ικανοποιηθούν αλλά περιλαμβάνει μαζί και τη βέλτιστη λύση. Αυτά τα προβλήματα αποφάσεων ονομάζονται Pseudo-Boolean Optimization.

4.6 Βελτιστοποίηση με ψευδοδυαδικούς περιορισμούς

Με βάση τους πιο πάνω ορισμούς για τη βελτιστοποίηση ενός προβλήματος, θα δοθεί έμφαση στη συνέχεια στη βελτιστοποίηση των προβλημάτων με τη χρήση ψευδοδυαδικών περιορισμών. Η μέθοδος που θα αναφερθεί και θα χρησιμοποιηθεί κατά κύριο λόγο και στις εφαρμογές σε προβλήματα προγραμματισμού δράσης στη συνέχεια είναι η γραμμική αναζήτηση. Στην γραμμική αναζήτηση στόχος είναι η συνάρτηση βελτιστοποίησης $\text{optimum}(g_1, g_2, g_3, \dots, g_n)$ (όπου $g_1, g_2, g_3, \dots, g_n$ οι στόχοι δεδομένου προβλήματος), να ελαχιστοποιηθεί. Μετατρέπεται δηλαδή αυτή η συνάρτηση σε περιορισμό που έχει τη μορφή $\text{optimum}(g_1, g_2, g_3, \dots, g_n) \leq \text{Cost}$ όπου Cost είναι μια ακέραια τιμή και ενημερώνεται σε κάθε επανάληψη του αλγορίθμου. Το cost είναι αρχικοποιημένο με μια μέγιστη τιμή. Η τιμή αυτή σε κάθε επανάληψη μειώνεται με σκοπό να ελαχιστοποιηθεί το κόστος της συνάρτησης βελτιστοποίησης. Η μείωση αυτή του κόστους ερμηνεύεται ως εξής : καθώς το κόστος μειώνεται μπορούν να ικανοποιηθούν περισσότεροι στόχοι του προβλήματος. Έτσι σε κάθε επανάληψη σκοπός είναι η μεγιστοποίηση των στόχων που μπορούν να ικανοποιηθούν ελαχιστοποιώντας ταυτόχρονα το κόστος. Η διαδικασία επαναλαμβάνεται μέχρι να μη

μπορεί να βρεθεί καλύτερη λύση με βάση πάντοτε τους περιορισμούς του προβλήματος, δηλαδή με απλά λόγια να μην μπορούν να ικανοποιηθούν περισσότεροι στόχοι για το συγκεκριμένο πρόβλημα.

Επίσης γνωστή είναι και μια παραλλαγή της γραμμικής αναζήτησης, η δυαδική αναζήτηση. Στη δυαδική αναζήτηση η διαφορά έγκειται στο γεγονός ότι η συνάρτηση βελτιστοποίησης χρησιμοποιεί δύο περιορισμούς αντί ένα όπως η γραμμική αναζήτηση. Οι περιορισμοί δηλαδή αφορούν τη μέγιστη και ελάχιστη τιμή της συνάρτησης βελτιστοποίησης. Δηλαδή η συνάρτηση βελτιστοποίησης σε αυτήν την περίπτωση είναι της μορφής:

$\text{MinCost} \leq \text{optimum}(g_1, g_2, g_3, \dots, g_n) \leq \text{MaxCost}$, όπου προφανώς οι μεταβλητές MinCost και MaxCost καθορίζουν την ελάχιστη και τη μέγιστη τιμή της συνάρτησης βελτιστοποίησης αντίστοιχα.

Οι διάφοροι προτασιακοί επιλυτές ψευδοδυαδικών περιορισμών υποστηρίζουν αρκετές συναρτήσεις βελτιστοποιήσεις, όπως η $\min$ και η $\max$. Στα προβλήματα που θα μελετηθούν στη συνέχεια θα χρησιμοποιηθεί η συνάρτηση $\min$ σε συνδυασμό με τον προτασιακό επιλυτή minisat+ .

4.7 Λειτουργία αλγορίθμου των ψευδοδυαδικών επιλυτών

Οι περισσότεροι προτασιακοί επιλυτές ακολουθούν μια συγκεκριμένη μεθοδολογία στην προσπάθεια να βρουν μια ανάθεση τιμών στις προτάσεις, ώστε να ικανοποιηθεί στο τέλος το πρόβλημα. Αρχικά οι μεταβλητές των προτάσεων δεν έχουν συγκεκριμένη τιμή, γι αυτό γίνεται ανάθεση στις διάφορες μεταβλητές (δηλαδή ανάθεση true ή false). Αφού γίνει η ανάθεση σε οποιαδήποτε μεταβλητή, γίνεται έλεγχος στη συνέχεια για τις συνέπειες που επηρεάζουν τις υπόλοιπες μεταβλητές, έχοντας ως δεδομένο ότι η τιμή που έχει ανατεθεί στη συγκεκριμένη μεταβλητή ισχύει. Για κάθε ανάθεση τιμής σε μεταβλητή εάν ο αλγόριθμος δε βρει κάποια σύγκρουση συνεχίζει με τη δημιουργία μιας νέας ανάθεσης τιμών σε μια άλλη μεταβλητή που δεν είχε μέχρι τώρα συγκεκριμένη τιμή. Εάν όμως εντοπίσει σύγκρουση τότε ο αλγόριθμος επιστρέφει προς τα πίσω και αφαιρεί από την μεταβλητή τη δοθείσα τιμή και συνεχίζει με την εύρεση καινούργιας τιμής στη μεταβλητή.

Η διαδικασία συνεχίζεται μέχρι να βρεθεί μια τέτοια ανάθεση στις μεταβλητές που να ικανοποιεί όλους τους περιορισμούς, νοουμένου πάντα ότι υπάρχει μια τέτοια ανάθεση.

4.8 Η σημασία των ψευδοδυαδικών περιορισμών

Με όσα έχουμε δει μέχρι στιγμής είναι φανερό ότι οι ψευδοδυαδικοί περιορισμοί έχουν μεγαλύτερη εκφραστικότητα από ότι οι προτάσεις με τους γνωστούς περιορισμούς. Η δομή τους διευκολύνει στην πιο απλή έκφραση των προτάσεων. Ενώ για μεγάλα προβλήματα με μεγάλη πολυπλοκότητα, απαιτείται μεγάλος αριθμός από προτάσεις (εκθετικό μέγεθος) ώστε να καλυφθούν όλες οι δυνατές περιπτώσεις που απορρέουν από τους περιορισμούς, με ένα σαφή ψευδοδυαδικό περιορισμό καλύπτονται όλες αυτές οι περιπτώσεις. Επίσης έχει αποδειχθεί ότι ο φορμαλισμός των ψευδοδυαδικών περιορισμών είναι πιο μικρός εκθετικά από ότι ο φορμαλισμός των διαζευκτικών προτάσεων. Κλασσικό παράδειγμα που αποδεικνύει αυτό το γεγονός είναι ο περιορισμός $\text{atleast}(k, \{x_1, x_2, \dots, x_n\})$, ο οποίος αντιστοιχεί στην πραγματικότητα σε $\binom{n}{n-k+1}$ διαφορετικά clauses.

Ένα πιθανό πρόβλημα που μπορεί να δημιουργηθεί αφορά στους συντελεστές μπροστά από κάθε όρο. Αν οι συντελεστές αυτοί έχουν μεγάλο μέγεθος, κατά την ανάλυση της πολυπλοκότητας του προβλήματος και αν το εργαλείο που χρησιμοποιείται έχει αριθμητική ακρίβεια αυτό κοστίζει σημαντικά ή μπορεί να οδηγήσει σε σφάλμα υπερχείλισης το οποίο μπορεί να είναι είτε compile error είτε runtime error. Λόγω δηλαδή της μεγάλης ακρίβειας που απαιτείται και σε συνδυασμό του μεγάλου εύρους των συντελεστών, υπάρχει η πιθανότητα να βγούμε εκτός των ορίων του χώρου αποθήκευσης των αριθμών και να δημιουργηθεί έτσι σφάλμα.

Κεφάλαιο 5

Ψευδοδυαδική Ικανοποίηση και Προγραμματισμός Δράσης

5.1 Επίλυση προβλημάτων προγραμματισμού δράσης	35
5.2 Γνωστά προβλήματα προγραμματισμού δράσης	36
5.3 Προτασιακοί επιλυτές και κωδικοποιήσεις	39
5.4 Ο προτασιακός επιλυτής Minisat+	42
5.4.1 Γενικά περί του Minisat+	42
5.4.2 Η κωδικοποίηση SMP σε cnf μορφή για επίλυση προβλημάτων με το εργαλείο Minisat+	43
5.5 Ψευδοδυαδική Βελτιστοποίηση	47
5.5.1 Η ψευδοδυαδική συνάρτηση βελτιστοποίησης του minisat+	48
5.5.2 Επίλυση προβλημάτων μέσω ενδιάμεσης κατάστασης βάσει ενός κάτω φράγματος – κατωφλίου	49
5.5.3 Τμηματική επίλυση προβλημάτων	52
5.5.3.1 Τεχνική διατήρησης ικανοποιηθέντων περιορισμών	55
5.5.3.2 Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών	56
5.5.3.3 Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών – συνδυασμός διατήρησης και ενημέρωσης περιορισμών	56

5.1 Σημασία προβλημάτων προγραμματισμού δράσης

Ο προγραμματισμός των δράσεων για ένα πρόβλημα προγραμματισμού δράσης είναι καθοριστικής σημασίας στις μέρες μας. Η συνεχής εξέλιξη της τεχνολογίας τόσο σε υλικό όσο και σε λογισμικό, καθώς επίσης και οι αυξανόμενες απαιτήσεις στο χώρο εργασίας του ανθρώπου καθιστούν την αποδοτική και αποτελεσματική επίλυση τέτοιων προβλημάτων επιτακτική ανάγκη.

Πριν κάποια χρόνια σκοπός για ένα οποιοδήποτε πρόβλημα, ήταν ουσιαστικά η εύρεση μιας σχετικά ικανοποιητικής λύσης, χωρίς να δίνεται μεγάλη έμφαση στο κόστος που θα είχε αυτή η λύση. Οι έρευνες όμως που γίνονται τις τελευταίες δεκαετίες, έδωσαν ώθηση ώστε να λαμβάνεται υπόψη ο μηχανισμός με τον οποίο ένα εργαλείο επίλυσης οδηγεί στη λύση συγκεκριμένου προβλήματος. Με απλά λόγια δε μας ενδιαφέρει να βρούμε απλά μια λύση αλλά να βρούμε τη βέλτιστη σε χρόνο και κόστος. Θέλουμε να αποφύγουμε στις λύσεις μας τους οποιουσδήποτε πλεονασμούς, οι οποίοι οδηγούν σε άσκοπη σπατάλη των διαθέσιμων πόρων του συστήματός μας. Χρειάζεται συγκεκριμένη μεθοδολογία και μηχανισμοί, οι οποίοι θα εξασφαλίζουν μια αποδοτική και αποτελεσματική σειρά από δράσεις που θα οδηγούν στην τελική λύση του προβλήματος.

Τα προβλήματα με τα οποία έχει να διαπραγματευτεί ο άνθρωπος έχουν ως κρίσιμες παραμέτρους το χρόνο και το κόστος (μνήμη, επεξεργαστές). Είναι προβλήματα με αρκετά μεγάλη πολυπλοκότητα, τα οποία απαιτούν μεγάλο ποσοστό από τους διαθέσιμους πόρους του συστήματος. Το να καταφέρεις για παράδειγμα να επιλύσεις ένα πρόβλημα σε 5 ώρες το οποίο χρειαζόταν 8-10 ώρες να επιλυθεί, αποκτά προφανώς μεγάλη σημασία. Πόσο μάλλον όταν το χρονικό πλαίσιο των προβλημάτων αυτών διεκπεραιώνεται όχι μόνο μέσα σε ώρες αλλά και σε ημέρες ή ακόμα και μήνες. Έτσι, σε προβλήματα που έχουν μεγάλο όγκο δεδομένων να επεξεργαστούν και συνδυασμούς περιπτώσεων, έχει μεγάλη απήχηση η εύρεση μιας ακολουθίας από ενέργειες – δράσεις τέτοιας ώστε η εφαρμογή της να οδηγεί στο στόχο του προβλήματος βελτιώνοντας ασφαλώς τις κρίσιμες παραμέτρους που αναφέραμε (χρόνος, κόστος).

5.2 Γνωστά προβλήματα προγραμματισμού δράσης

Θα γίνει περιγραφή στη συνέχεια μερικών από τα προβλήματα, τα οποία χρησιμοποιούνται σε διαγωνισμούς Προγραμματισμού Δράσης και τα οποία θα χρησιμοποιηθούν ακολούθως για πειραματικές μετρήσεις και εξαγωγή συμπερασμάτων. Μερικά από αυτά καταπιάνονται με εφαρμογές του πραγματικού κόσμου όπως αποθήκευση, μεταφορές με φορτηγά και διαβάσεις, κάποια άλλα στοχεύουν να ερευνήσουν τη δυνατότητα εφαρμογής και την αποτελεσματικότητα του αυτοματοποιημένου προγραμματισμού για νέες εφαρμογές και άλλα σχετίζονται με γνωστά προβλήματα που έχουν εξεταστεί σε άλλους τομείς της πληροφορικής (όπως τα προβλήματα TPP και openstacks).

Προβλήματα TPP

Τα προβλήματα προγραμματισμού δράσης TPP, έχουν σχέση με μεταβάσεις και αγορές αγαθών από επιλεγμένες αγορές τα οποία ελαχιστοποιούν το κόστος. Συγκεκριμένα σε αυτά τα προβλήματα έχουμε ένα σύνολο προϊόντων και ένα σύνολο αγορών, όπου η κάθε αγορά παρέχει ένα όριο στο ποσό του κάθε προϊόντος σε γνωστή τιμή. Σκοπός του TPP είναι η επιλογή ενός υποσυνόλου των αγορών έτσι ώστε μια δεδομένη απαίτηση κάθε προϊόντος να μπορεί να αγοραστεί, ελαχιστοποιώντας την ίδια ώρα το κόστος δρομολόγησης και το κόστος αγοράς.

Προβλήματα Openstacks

Τα προβλήματα προγραμματισμού δράσης Openstacks, είναι συνδυαστικά προβλήματα βελτιστοποίησης στο σχεδιασμό παραγωγής. Σε αυτό το πρόβλημα ένας κατασκευαστής έχει διάφορες παραγγελίες, κάθε μια για έναν συνδυασμό διαφορετικών προϊόντων, και μπορεί να κατασκευάσει μόνο ένα προϊόν κάθε φορά. Τέτοια προβλήματα είναι γνωστό ότι ανήκουν στην κατηγορία NP-complete προβλημάτων.

Προβλήματα Storage

Τα προβλήματα προγραμματισμού δράσης Storage, σχετίζονται με τη μετακίνηση και αποθήκευση πακέτων που περιέχουν αγαθά, μέσω των ανελκυστήρων από τα

εμπορευματοκιβώτια στις αποθήκες σύμφωνα με κάποιους χάρτες. Συγκεκριμένα μέσα σε μια αποθήκη, κάθε ανελκυστήρας μπορεί να κινηθεί σύμφωνα με ένα δεδομένο χωρικό χάρτη που συνδέει τις διαφορετικές περιοχές της αποθήκης.

Η διερεύνηση αυτού του προβλήματος γίνεται με βάση διαφορετικούς αριθμούς αποθηκών, ανελκυστήρων, πακέτων, εμπορευματοκιβωτίων, και περιοχών αποθηκών. Για κάθε πρόβλημα, δίνουμε ως είσοδο το χωρικό χάρτη των περιοχών των αποθηκών. Το ζητούμενο για κάθε περιοχή όπως είναι αναμενόμενο είναι να παραχθεί ένα σχέδιο καλής ποιότητας, ωστόσο σε κάποια προβλήματα είναι δύσκολη απλά και μόνο η προσπάθεια να εξαχθεί η οποιαδήποτε λύση.

Προβλήματα Pathways

Σκοπός στα προβλήματα προγραμματισμού δράσης Pathways είναι να βρεθεί μια ακολουθία – μονοπάτι βιοχημικών αντιδράσεων σε έναν οργανισμό που παράγει ορισμένες ουσίες. Τα προβλήματα έχουν τη βάση τους στη μοριακή βιολογία. Ένα μονοπάτι είναι μια ακολουθία χημικών αντιδράσεων σε έναν βιολογικό οργανισμό. Αυτά τα μονοπάτια περιγράφουν τους μηχανισμούς που εξηγούν πώς τα κύτταρα πραγματοποιούν σημαντικές από τις λειτουργίες τους με τη βοήθεια των μορίων και των αντιδράσεων που παράγουν τις φυσιολογικές αλλαγές. Ατέλειες στα μονοπάτια μπορούν να εξηγήσουν πολλές από τις ασθένειες. Περαιτέρω έρευνες περιλαμβάνουν συχνά την εύρεση των φαρμάκων τα οποία διορθώνουν τις ατέλειες αυτές στα μονοπάτια.

Προβλήματα Trucks

Αυτά τα προβλήματα καταπιάνονται με τη μεταφορά πακέτων από μια τοποθεσία σε κάποια άλλη μέσω φορτηγών. Οι μεταφορές αυτές γίνονται λαμβάνοντας υπόψη περιορισμούς, που αφορούν κυρίως στο χωρικό πλαίσιο του εκάστοτε φορτηγού αλλά και στις χρονικές προθεσμίες παράδοσης του πακέτου. Συγκεκριμένα, ένα πακέτο μπορεί να φορτωθεί και να εκφορτωθεί σε ένα μέρος του φορτηγού μόνο εάν η περιοχή που θα μεταφερθεί το πακέτο και η πόρτα του φορτηγού είναι ελεύθερες. Η εύρεση της λύσης είναι αρκετές φορές επίσης δύσκολη λόγω των πολλών συνδυασμών των

περιπτώσεων ανάμεσα στα φορτηγά και στις περιοχές στις οποίες προορίζονται τα πακέτα.

Προβλήματα Rovers

Τα προβλήματα προγραμματισμού δράσης Rovers σχετίζονται με οχήματα τα οποία μεταβαίνουν σε κάποιον πλανήτη. Εκεί περιπλανώνται με σκοπό να συλλέγουν στοιχεία και να στέλνουν δείγματα στη γη. Στα οχήματα αυτά δεν μπαίνουν άνθρωποι για να τα καθοδηγούν [7].

Προβλήματα Pipes World

Τα pipes world προβλήματα ασχολούνται με την μεταφορά προϊόντων πετρελαίου σε ένα δίκτυο αγωγών. Σε αυτά τα προβλήματα μπορεί να λαμβάνονται υπόψη και περιορισμοί που αφορούν στη χωρητικότητα αποθήκευσης του πετρελαίου σε δεξαμενές [7].

Κάθε ένα από αυτά τα προβλήματα περιγράφεται από 2 αρχεία το domain και το problem:

Το αρχείο του domain περιγράφει ουσιαστικά τους ορισμούς όλων των δυνατών δράσεων για το συγκεκριμένο πρόβλημα. Ορίζει δηλαδή το όνομα της κάθε δράσης, τις προϋποθέσεις που πρέπει να ισχύουν για να εκτελεστεί η συγκεκριμένη δράση και τις συνέπειες που παρουσιάζονται με την εφαρμογή της. Είναι δηλαδή με απλά λόγια το πεδίο ορισμού του προβλήματος.

Το αρχείο του problem ορίζει το ίδιο το πρόβλημα. Αναφέρει αρχικά το όνομα του προβλήματος και το αντίστοιχο όνομα του domain που σχετίζεται με αυτό το πρόβλημα. Ακολούθως, περιγράφει την αρχική κατάσταση που βρισκόμαστε τώρα, δηλαδή τις προϋποθέσεις που ισχύουν καθώς επίσης και την τελική κατάσταση- στόχο, δηλαδή τους όρους που πρέπει να ισχύουν για να ικανοποιηθεί το πρόβλημα.

Τα δύο αυτά αρχεία είναι χρήσιμα για το γράφο δράσεων, ο οποίος «κτίζει» το αντίστοιχο γράφημα ανάλογα με το domain του προβλήματος. Δημιουργεί δηλαδή τις

κατάλληλες δομές δεδομένων που θα χρησιμοποιηθούν και αργότερα από τον Satplan για την επίλυση του προβλήματος.

5.3 Προτασιακοί επιλυτές και κωδικοποιήσεις

Όπως αναφέραμε και κατά την περιγραφή της λειτουργίας του γράφου δράσεων, το γράφημα αυτό δίνει μια άμεση κωδικοποίηση του προβλήματος σε προτασιακή λογική με τη μορφή clauses, τα οποία περιγράφουν τις δράσεις του προβλήματος τις προϋποθέσεις και τις συνέπειες σε κάθε χρονική στιγμή. Οι προτάσεις της άμεσης κωδικοποίησης του γράφου δράσεων είναι οι ακόλουθες:

1. Μοναδιαίες διαζεύξεις για την αρχική και τελική κατάσταση.
2. $A(T) \rightarrow f(T)$, για κάθε δράση A και για κάθε όρο f , όπου $f \in \text{pre}(A)$.
3. $A(T) \rightarrow f(T + 1)$, για κάθε δράση A και για κάθε όρο f , όπου $f \in \text{add}(A)$.
4. $A(T) \rightarrow \neg f(T + 1)$, για κάθε δράση A και για κάθε όρο f , όπου $f \in \text{del}(A)$.
5. $f(T) \rightarrow A_1(T - 1) \vee \dots \vee A_m(T - 1)$, για κάθε όρο f και για όλες τις δράσεις A_i , όπου $1 \leq i \leq m$ (συμπεριλαμβανομένου και των αδρανών (noop) δράσεων) και όπου $f \in \text{add}(A_i)$.
6. $\neg f(T) \rightarrow A_1(T - 1) \vee \dots \vee A_m(T - 1) \vee \neg f(T - 1)$, για κάθε όρο f και για όλες τις δράσεις A_i , όπου $1 \leq i \leq m$ και όπου $f \in \text{del}(A_i)$.
- 7.1 $\neg A_1(T) \vee \neg A_2(T)$, για κάθε ζεύγος δράσεων A_1, A_2 τέτοιο ώστε το σύνολο $\text{del}(A_1) \cap \text{pre}(A_2) \neq \text{κενού συνόλου}$.
- 7.2 $\neg A_1(T) \vee \neg A_2(T)$, για κάθε ζεύγος δράσεων A_1, A_2 τέτοιο ώστε το σύνολο $\text{del}(A_1) \cap \text{add}(A_2) \neq \text{κενού συνόλου}$.
- 7.3 $\neg A_1(T) \vee \neg A_2(T)$, εάν υπάρχει ζεύγος όρων f_1, f_2 όπου $f_1 \in \text{pre}(A_1)$, $f_2 \in \text{pre}(A_2)$, τέτοιο ώστε οι f_1 και f_2 να είναι αμοιβαία αποκλειόμενοι τη χρονική στιγμή T .
- 8 $\neg f_1(T) \vee \neg f_2(T)$, για κάθε ζεύγος όρων f_1, f_2 οι οποίοι είναι αμοιβαία αποκλειόμενοι τη χρονική στιγμή T .

Επεξηγήσεις συμβολισμών :

$\text{pre}(A) = \text{preconditions}$ (προϋποθέσεις) της δράσης A

$\text{add}(A) = \text{θετικές συνέπειες της δράσης } A$

$\text{del}(A)$ = αρνητικές συνέπιες της δράσης A

Κατά καιρούς δημιουργήθηκαν πολλά διαφορετικά εργαλεία που υποστηρίζουν διαφορετικές κωδικοποιήσεις, έχοντας ως βάση ασφαλώς την άμεση κωδικοποίηση που απορρέει από το γράφημα δράσεων, δηλαδή τις οκτώ πιο πάνω μορφές προτάσεων.

Το πρώτο σύστημα, το οποίο δημιουργήθηκε και σχετίζεται με την προτασιακή κωδικοποίηση ενός προβλήματος προγραμματισμού δράσης με βάση την άμεση κωδικοποίηση που προκύπτει από το γράφημα δράσεων ήταν το BLACKBOX (BB). Το σύστημα αυτό ενσωμάτωσε τρεις διαφορετικές κωδικοποιήσεις την BB-7, την BB-31 και την BB-32. Η BB-7 χρησιμοποιεί τα clauses 1,2,5,7.1,7.2,7.3 όπως περιγράφηκαν πιο πάνω, η BB-31 χρησιμοποιεί τις προτάσεις 1,2,3,4,5,7.1,8 και η BB-32 χρησιμοποιεί τις προτάσεις 1,2,3,4,5,7.1,7.2,7.3,8.

Ένα δεύτερο εργαλείο που δημιουργήθηκε ήταν το SATPLAN06, το οποίο υποστηρίζει 2 κωδικοποιήσεις τις SATPLAN06-4 και SATPLAN06-3. Η SATPLAN06-04 χρησιμοποιεί τα clauses 1,2,5,7.1,7.2,8, ενώ η SATPLAN06-3 χρησιμοποιεί τα clauses 1,2,5,7.1,7.2,7.3,8.

Αυτό όμως που έχει μεγάλη σημασία για τις διάφορες κωδικοποιήσεις των συστημάτων που περιγράφηκαν πιο πάνω είναι η διάδοση των περιορισμών (constraint propagation) . Η μοναδιαία διάδοση (unit propagation) δηλώνει την κλειστότητα της θεωρίας T κάτω από τη διάχυση των περιορισμών. Με βάση αυτό τον ορισμό μπορούμε εύκολα να εντοπίσουμε τις προτάσεις που είναι πλεονασμοί σε μερικές κωδικοποιήσεις.

Αρχικά μπορούμε να παρατηρήσουμε ότι $\text{SATPLAN06-4} \subset \text{SATPLAN06-3} \subset \text{BB-32}$, και $\text{BB-7} \subset \text{SATPLAN06-3}$.

Επίσης, εντοπίζεται ότι μερικά αμοιβαία αποκλειόμενες προτάσεις είναι πλεονάζουσες σε κάποιες κωδικοποιήσεις. Συγκεκριμένα, προτάσεις σε κωδικοποιήσεις που περιγράφονται από προτάσεις της μορφής 7.3 είναι πλεονάζουσες (UP-redundant) αν η κωδικοποίηση έχει και τα σύνολα 2 και 8. Επίσης, προτάσεις σε κωδικοποιήσεις που περιγράφονται από προτάσεις της μορφής 7.2 είναι πλεονάζουσες αν η κωδικοποίηση έχει και τα σύνολα 3 και 4 [2].

Έτσι, με βάση τις πιο πάνω παρατηρήσεις, αφού οι προτάσεις 7.2 και 7.3 φάνηκε ότι αποτελούσαν πλεονασμό, αποφασίσθηκε η αφαίρεση των δύο αυτών προτάσεων. Η αφαίρεση των δύο αυτών συνόλων οδήγησε στη δημιουργία μιας νέας κωδικοποίησης της SATMAXPLAN. Η κωδικοποίηση SATMAXPLAN χρησιμοποιεί τις προτάσεις της μορφής 1,2,3,4,5,6,7.1,8.

Εφόσον οι προτάσεις της μορφής 7.2 και 7.3 είναι πλεονάζουσες, η κωδικοποίηση BB-32 όπως και η BB-7 θα μπορούσε να απλοποιηθεί με την αφαίρεση των δύο αυτών προτάσεων. Επίσης, προκύπτει ότι το σύνολο των προτάσεων της μορφής 8 σε συνδυασμό με τις υπόλοιπες προτάσεις 1 έως 7.3 (εκτός του 7.2 και 7.3) δεν είναι πλεονάζον. Με αυτό καταλήγουμε όσον αφορά τις κωδικοποιήσεις του BLACKBOX, ότι η κωδικοποίηση BB-31 είναι πιο ισχυρή από τις κωδικοποιήσεις BB-7, BB-32 και SATPLAN06-3.

Ωστόσο, λόγω του ότι η κωδικοποίηση SATMAXPLAN χρησιμοποιεί προτάσεις της μορφής 6, αυτό την καθιστά πιο ισχυρή από την κωδικοποίηση BB-31 και συνεπώς πιο ισχυρή και από οποιαδήποτε άλλη κωδικοποίηση από τις πιο πάνω.

Όσο αφορά την κωδικοποίηση SATMAXPLAN, το μέγεθος των προτάσεων που την αποτελούν είναι δυνατόν να μειωθεί ακόμη πιο πολύ. Αυτό επειδή υπάρχουν προτάσεις της μορφής 7.1 – ζεύγη αμοιβαία αποκλειόμενων δράσεων, που μπορεί να περιλαμβάνονται και σε άλλα αμοιβαία αποκλειόμενα σύνολα που είναι ήδη πλεονάζοντα. Με βάση αυτό, το μέγεθος των προτάσεων της μορφής 7.1 στην κωδικοποίηση SATPLANMAX, μπορεί να μειωθεί με την αφαίρεση των προτάσεων που ανήκουν ταυτόχρονα και στα σύνολα των προτάσεων 7.2 και 7.3 (τα οποία είναι πλεονάζοντα). Τέλος μπορούν να αφαιρεθούν, όλες οι προτάσεις των αμοιβαία αποκλειόμενων δράσεων πάνω σε δράσεις A1 και A2 με χρόνο T που έχουν ως αποτέλεσμα τις συνέπειες p1 και p2 αντίστοιχα, έτσι ώστε οι p1 και p2 να είναι αμοιβαία αποκλειόμενα τη χρονική στιγμή T+1. Έτσι και με αυτές τις τροποποιήσεις καταλήγουμε στην τελική μορφή της κωδικοποίησης SMP (SAT – MAX - PLAN), η οποία είναι αναμφισβήτητα η πιο ισχυρή κωδικοποίηση [2].

Πολλοί από τους προτασιακούς επιλυτές προβλημάτων προγραμματισμού δράσης μπορούν να χρησιμοποιηθούν με διάφορες κωδικοποιήσεις που περιγράφηκαν πιο πάνω.

Φυσικά όπως είναι προφανές καλύτερα αποτελέσματα εξάγονται όταν χρησιμοποιείται η κωδικοποίηση SMP, που είναι και η πιο ισχυρή. Μερικοί από τους πιο συνηθισμένους προτασιακούς επιλύτες (solvers) που χρησιμοποιούνται από τους ερευνητές είναι οι siege, precosat, minisat, minisat+. Οι πειραματικές μετρήσεις που θα πραγματοποιηθούν θα είναι αρχικά με τον επιλύτη precosat, που θεωρείται και ο πιο αποδοτικός και στη συνέχεια θα δοκιμαστούν διάφορες εφαρμογές με τον minisat+, που είναι και το κύριο μέρος των πειραματικών μετρήσεων της διπλωματικής εργασίας.

5.4 Ο προτασιακός επιλύτης Minisat+

5.4.1 Γενικά περί του Minisat+

Η εξέλιξη στα εργαλεία επίλυσης σε προβλήματα προτασιακής ικανοποιησιμότητας (solvers) τα τελευταία χρόνια είναι ραγδαία. Τέτοια εργαλεία έχουν μεγάλες εφαρμογές στο πεδίο του αυτοματοποιημένου ηλεκτρονικού σχεδιασμού. Παράλληλα, η μεγάλη συμβολή που είχαν τέτοια εργαλεία στις πιο πάνω εφαρμογές, είχε ως αποτέλεσμα να επεκταθούν και πέραν από τα όρια της προτασιακής λογικής.

Τα εργαλεία επίλυσης προβλημάτων ικανοποιησιμότητας προτασιακής λογικής έχουν τη δυνατότητα να χειριστούν και άλλα είδη περιορισμών και μπορούν να εμπλακούν σε διάφορα πεδία ορισμών, όπως για παράδειγμα τα πεπερασμένα σύνολα [8].

Ο minisat+ είναι ένα εργαλείο επίλυσης προτασιακής ικανοποιησιμότητας. Αποτελεί μία επέκταση του προτασιακού επιλύτη minisat. Βασικός σκοπός του ήταν να μπορεί να υποστηρίξει ψευδοδυαδικούς περιορισμούς όπως για παράδειγμα γραμμικούς περιορισμούς της μορφής : $3.\overline{X1} + 2.X2 < 3$.

Αυτό το εργαλείο επίλυσης προβλημάτων ικανοποιησιμότητας προτασιακής λογικής μπορεί να χρησιμοποιηθεί για την επίλυση ψευδοδυαδικών προβλημάτων με προτάσεις όπως οι πιο πάνω, μεταφράζοντας αυτές τις προτάσεις σε περιορισμούς. Τέτοιου είδους προβλήματα είναι γνωστά και ως προβλήματα ακέραιου γραμμικού 0-1 προγραμματισμού, το λεγόμενο integer linear Programming - ILP. Στον ακέραιο γραμμικό 0-1 προγραμματισμό, επιχειρείται να ικανοποιηθεί ένας περιορισμός κάνοντας ανάθεση στις μεταβλητές μόνο τις τιμές 0 και 1 που ισοδυναμούν στις false

και true αντίστοιχα. Είναι μια ανίσωση δηλαδή που προσπαθεί να επιλυθεί με εύρος δυνατών τιμών το πεδίο $\{0,1\}$ και όχι οποιαδήποτε τιμή και λόγω του ότι οι τιμές αντιστοιχούν σε αληθείς ή ψευδείς γι' αυτό ονομάζονται και ψευδοδυαδικοί. Επομένως, γίνεται προσπάθεια από τον επιλυτή να βρεθεί μια τέτοια ανάθεση αν υπάρχει, που ικανοποιεί το πρόβλημα. Αυτή είναι και η βάση στην οποία θα στηριχθεί το εργαλείο αυτό για να επιλύσει τα προβλήματα που θα μελετηθούν πιο κάτω.

Πρόσφατα έχουν δημιουργηθεί εκτός από το εργαλείο minisat+, αρκετά εργαλεία επίλυσης ψευδοδυαδικών περιορισμών. Τα νέα αυτά εργαλεία επεκτείνουν μέσα από την υλοποίηση τους τις τεχνικές επίλυσης προτασιακής ικανοποιησιμότητας (SAT) για την επίλυση ψευδοδυαδικών περιορισμών. Μερικά από αυτά είναι οι PBS, PUEBLO, GALENA και ο OPBDP [8].

5.4.2 Η κωδικοποίηση SMP σε cnf μορφή για επίλυση προβλημάτων με το εργαλείο Minisat+

Η κωδικοποίηση SMP, όπως προαναφέρθηκε είναι η πιο ισχυρή μορφή κωδικοποίησης των περιορισμών ενός προβλήματος, η οποία λειτουργεί σε συνδυασμό με την άμεση κωδικοποίηση του graphplan. Ωστόσο, η κωδικοποίηση αυτή από μόνη της δεν μπορεί να επικοινωνήσει άμεσα με τον επιλυτή (solver) minisat+.

Ο προτασιακός επιλυτής minisat+, όπως και ο κάθε επιλυτής καθορίζει συγκεκριμένη μορφή εισόδου για το αρχείο που περιγράφει τους περιορισμούς και τους οποίους στη συνέχεια θα προσπαθήσει να επιλύσει. Συγκεκριμένα, παίρνει ως παράμετρο εισόδου ένα αρχείο στο οποίο υπάρχουν όλα οι προτάσεις, που σχετίζονται με τις δράσεις που έχουν εκτελεστεί ή όχι, τους στόχους του προβλήματος (συνήθως ως unit clauses), την αρχική και τελική κατάσταση, οποιουσδήποτε περιορισμούς αφορούν στα γεγονότα (facts-literals), τους αμοιβαία αποκλειόμενους συνδέσμους ανάμεσα στις δράσεις και ανάμεσα στα γεγονότα σε κάθε επίπεδο του γράφου. Η μορφή του αρχείου είναι σε CNF (συζευκτική κανονική μορφή). Έτσι, ο satplan που επεκτείνει το γράφημα με τη βοήθεια του graphplan δημιουργεί την αντίστοιχη μορφή CNF, που θα πάρει ως είσοδο ο minisat+. Η συγκεκριμένη μορφή που ο satplan πρέπει να δημιουργήσει για το αρχείο CNF που θα δοθεί ως είσοδο στο minisat+ περιγράφεται ως ακολούθως :

Για κάθε πρόταση που προκύπτει από την SMP της μορφής : $X \vee \neg Y \vee \neg W \vee Z$, μετράμε το πλήθος των αρνητικών όρων (αυτών που είναι false) της πρότασης και αφαιρούμε από το 1 το πλήθος αυτών των όρων. Το αποτέλεσμα της αφαίρεσης αυτής καθορίζει το συνολικό αριθμό των όρων που πρέπει να ικανοποιούνται. Επίσης, μπροστά από κάθε θετικό όρο προστίθεται το +1* και μπροστά από κάθε αρνητικό όρο αφαιρούμε το $\neg$ και προσθέτουμε το -1*. Τέλος, οι όροι μεταξύ τους διαχωρίζονται με κενό διάστημα.

Έτσι το $X \vee \neg Y \vee \neg W \vee Z$ μετατρέπεται σε :

$$+1*X \ -1*Y \ -1*W \ +1*Z \ >= (1-\text{πλήθος αρνητικών όρων})$$

δηλαδή

$$+1*X \ -1*Y \ -1*W \ +1*Z \ >= -1$$

Αξιίζει να σημειωθεί ότι το CNF αρχείο που παίρνει ως είσοδο ο επιλυτής δεν του δίνει τη δυνατότητα να γνωρίζει τα ονόματα των δράσεων, στόχων αφού οι προτάσεις δίνονται ως μεταβλητές με αριθμούς. Ωστόσο, δεν υπάρχει ούτε και νόημα ο επιλυτής να γνωρίζει την περιγραφή των περιορισμών, καθώς σκοπός του, αφού στην ουσία είναι μια ανεξάρτητη υπορουτίνα, είναι να διαπιστώσει κατά πόσο το πρόβλημα είναι ικανοποιήσιμο και αν ναι να δώσει τη λύση (δηλαδή τις τιμές στην κάθε μεταβλητή – true ή false) στον satplan. Η περιγραφή των μεταβλητών βρίσκεται σε ξεχωριστό αρχείο, το οποίο διαχειρίζεται εξ' ολοκλήρου ο satplan και είναι το λεγόμενο varfile. Στο αρχείο αυτό υπάρχει αντιστοιχία ανάμεσα στις μεταβλητές και στα ονόματα των περιορισμών του προβλήματος (δράσεις, στόχοι, κτλ.), το οποίο χρησιμοποιεί στο τέλος ο satplan για να εξάγει την τελική λύση εάν και εφόσον υπάρχει.

```

c VARS:
c var v 1 fact      (STACKS-AVAIL-N0()) 0
c var v 2 fact      (NOT-MADE-P1()) 0
c var v 3 fact      (NOT-MADE-P2()) 0
c var v 4 fact      (NOT-MADE-P3()) 0
c var v 5 fact      (NOT-MADE-P4()) 0
c var v 6 fact      (NOT-MADE-P5()) 0
c var v 7 fact      (MACHINE-AVAILABLE-()) 0
c var v 8 fact      (WAITING-01()) 0
c var v 9 fact      (WAITING-02()) 0
c var v10 fact      (WAITING-03()) 0
c var v11 fact      (WAITING-04()) 0
c var v12 fact      (WAITING-05()) 0
c var v13 action    (SETUP-MACHINE-P1-N0) 0
c var v14 action    (SETUP-MACHINE-P2-N0) 0
c var v15 action    (SETUP-MACHINE-P3-N0) 0
c var v16 action    (SETUP-MACHINE-P4-N0) 0
c var v17 action    (SETUP-MACHINE-P5-N0) 0
c var v18 action    (OPEN-NEW-STACK-N0-N1) 0
c var v19 actionNOOP (NOOP-STACKS-AVAIL-N0()) 0
c var v20 actionNOOP (NOOP-NOT-MADE-P1()) 0
c var v21 actionNOOP (NOOP-NOT-MADE-P2()) 0
c var v22 actionNOOP (NOOP-NOT-MADE-P3()) 0
c var v23 actionNOOP (NOOP-NOT-MADE-P4()) 0
c var v24 actionNOOP (NOOP-NOT-MADE-P5()) 0
c var v25 actionNOOP (NOOP-MACHINE-AVAILABLE-()) 0
c var v26 actionNOOP (NOOP-WAITING-01()) 0
c var v27 actionNOOP (NOOP-WAITING-02()) 0
c var v28 actionNOOP (NOOP-WAITING-03()) 0
c var v29 actionNOOP (NOOP-WAITING-04()) 0
c var v30 actionNOOP (NOOP-WAITING-05()) 0
c var v31 fact      (STACKS-AVAIL-N1()) 1
c var v32 fact      (MACHINE-CONFIGURED-P5()) 1
c var v33 fact      (MACHINE-CONFIGURED-P4()) 1
c var v34 fact      (MACHINE-CONFIGURED-P3()) 1

```

Σχήμα 6.1 – Αντιστοιχία μεταβλητών δράσεων στο αρχείο VARFILE.

```

+1*v1    >= 1;
+1*v2    >= 1;
+1*v3    >= 1;
+1*v4    >= 1;
+1*v5    >= 1;
+1*v6    >= 1;
+1*v7    >= 1;
+1*v8    >= 1;
+1*v9    >= 1;
+1*v10   >= 1;
+1*v11   >= 1;
+1*v12   >= 1;
- 1*v13 +1*v1    >= 0;
- 1*v13 +1*v2    >= 0;
- 1*v13 +1*v7    >= 0;
- 1*v14 +1*v1    >= 0;
- 1*v14 +1*v3    >= 0;
- 1*v14 +1*v7    >= 0;
- 1*v15 +1*v1    >= 0;
- 1*v15 +1*v4    >= 0;
- 1*v15 +1*v7    >= 0;
- 1*v16 +1*v1    >= 0;
- 1*v16 +1*v5    >= 0;
- 1*v16 +1*v7    >= 0;
- 1*v17 +1*v1    >= 0;
- 1*v17 +1*v6    >= 0;
- 1*v17 +1*v7    >= 0;
- 1*v18 +1*v1    >= 0;
- 1*v19 +1*v1    >= 0;
- 1*v20 +1*v2    >= 0;
- 1*v21 +1*v3    >= 0;
- 1*v22 +1*v4    >= 0;
- 1*v23 +1*v5    >= 0;
- 1*v24 +1*v6    >= 0;
- 1*v25 +1*v7    >= 0;
- 1*v26 +1*v8    >= 0;
- 1*v27 +1*v9    >= 0;
- 1*v28 +1*v10   >= 0;
- 1*v29 +1*v11   >= 0;
- 1*v30 +1*v12   >= 0;
- 1*v31 +1*v18   >= 0;
- 1*v32 +1*v17   >= 0;
- 1*v33 +1*v16   >= 0;
- 1*v34 +1*v15   >= 0;

```

Σχήμα 6.2 – Περιορισμοί λογικών προτάσεων στο αρχείο CNF.

Τα πιο πάνω Σχήματα 6.1 και 6.2 αποτελούν ένα παράδειγμα αρχείου varfile και αρχείου cnf αντίστοιχα, τα οποία αφορούν το πρόβλημα stacks (πρόβλημα p04.pddl).

Στο αρχείο varfile τα v1,v2,v3,v4 είναι μεταβλητές που αντιστοιχούν στα γεγονότα (facts) (STACKS-AVAIL-N0()),(NOT-MADE-P1()),(NOT-MADE-P2()) και (NOT-MADE-P3()) αντίστοιχα. Στο αρχείο του cnf το clause $-1*v13 + 1*v7 \geq 0$, ικανοποιείται με οποιοδήποτε συνδυασμό τιμών εκτός του συνδυασμού με : v13=1 (true) και v7=0 (false).

5.5 Ψευδοδυαδική Βελτιστοποίηση

Οι λύσεις που οι προτασιακοί επιλυτές ικανοποιησιμότητας δίνουν πολλές φορές δεν είναι οι βέλτιστες. Συχνά οι επιλυτές αυτοί έδιναν μια λύση, η οποία είχε μεγάλο χρονικό κόστος αλλά ήταν μια ορθή λύση. Παρά την ορθότητα της λύσης, η λύση αυτή μερικές φορές έχανε τη χρησιμότητά της ειδικά όταν για μια εφαρμογή είναι επιτακτική ανάγκη να επιλυθεί κάτι πολύ σύντομα. Για παράδειγμα, σε προβλήματα με χρονικές προθεσμίες (deadlines), αν ο επιλυτής μας δώσει μια λύση στο πρόβλημά μας, η οποία ικανοποιεί όλους τους περιορισμούς (στόχους) μετά το πέρας του προκαθορισμένου χρόνου, η λύση αυτή είναι δεν είναι λειτουργική, παρόλο που ικανοποιεί όλους τους περιορισμούς. Η βελτιστοποίηση μπορεί να ικανοποιήσει ένα υποσύνολο στόχων αλλά σε σημαντικότερο χρονικό διάστημα. Έτσι, ανάλογα και με τις απαιτήσεις του προβλήματος τίθεται το εξής ερώτημα: Είναι προτιμότερο να λύσουμε ένα πρόβλημα εξολοκλήρου ικανοποιώντας όλους τους στόχους αλλά μετά από μεγάλο χρονικό διάστημα ή να λύσουμε ένα πρόβλημα ικανοποιώντας ένα μεγάλο υποσύνολο των στόχων αλλά σε σαφώς μικρότερο χρονικό διάστημα;

Κάτω από αυτές τις συνθήκες, η βελτιστοποίηση των λύσεων είναι απαραίτητη. Έτσι λοιπόν η ανάπτυξη νέων τεχνικών επίλυσης προτασιακής ικανοποιησιμότητας όπως και η πρόοδος των ψευδοδυαδικών αλγόριθμων στοχεύουν στην εύρεση βέλτιστων λύσεων. Αρκετοί ερευνητές μέσω του minisat+ αλλά και μέσω άλλων εργαλείων, διερευνούν αποδοτικές αλγοριθμικές μεθόδους στις οποίες μπορεί να συνδυάζουν ευρετικά ή άλλες τεχνικές, προκειμένου να καταλήξουν στο βέλτιστο.

Σε κάθε πρόβλημα, στο οποίο γίνεται προσπάθεια βελτιστοποίησης, εισάγεται μια συνάρτηση βελτιστοποίησης. Στόχος της συνάρτησης βελτιστοποίησης είναι να βρει μια βέλτιστη λύση, δηλαδή μια ανάθεση στις τιμές των μεταβλητών που ικανοποιούν τους περιορισμούς, αλλά και να ελαχιστοποιεί το κόστος της συνάρτησης. Το πρόβλημα εύρεσης της βέλτιστης λύσης είναι μια συνεχής διαδικασία η οποία τερματίζεται μόνο όταν διαπιστωθεί ότι δεν μπορεί να βρεθεί καλύτερη λύση, με βάση πάντοτε τους υφιστάμενους περιορισμούς.

Για παράδειγμα, έστω ότι έχουμε ένα ψευδοδυαδικό πρόβλημα ελαχιστοποίησης και μια συνάρτηση βελτιστοποίησης έστω $f(x)$. Μια ανάθεση στις τιμές των μεταβλητών, η

οποία ελαχιστοποιεί την τιμή της συνάρτησης, μπορεί εύκολα να βρεθεί με επαναληπτικές κλήσεις στον προτασιακό επιλυτή. Ως πρώτο βήμα θα καλέσουμε τον επιλυτή και θα του δώσουμε ως είσοδο το σύνολο των περιορισμών χωρίς να λάβουμε υπόψη τη συνάρτηση βελτιστοποίησης. Αυτό θα μας οδηγήσει σε μια αρχική λύση, έστω $f(x_0) = \kappa$. Ακολουθώντας, προσθέτουμε τον περιορισμό $f(x) < \kappa$ και καλούμε ξανά τον επιλυτή, χρησιμοποιώντας όμως τώρα τη συνάρτηση βελτιστοποίησης που αναφέραμε. Εάν το πρόβλημα δεν ικανοποιείται (unsatisfiable), η κ είναι η βέλτιστη λύση. Αντίθετα, αν το πρόβλημα ικανοποιείται η διαδικασία επαναλαμβάνεται με μικρότερη λύση. Έτσι, με αυτήν την επαναληπτική διαδικασία καταλήγουμε σε μια βέλτιστη λύση, η οποία ελαχιστοποιεί το κόστος για το συγκεκριμένο πρόβλημα. Αυτός ο αλγόριθμος βελτιστοποίησης σε συνδυασμό με τη συνάρτηση βελτιστοποίησης θα αποτελέσει και τη βάση πάνω στην οποία θα δοκιμάσουμε αργότερα διάφορες τεχνικές βελτιστοποίησης. [8]

5.5.1 Η ψευδοδυαδική συνάρτηση βελτιστοποίησης του minisat+

Λόγω της μεγάλης σημασίας της βελτιστοποίησης έχουν αναπτυχθεί αρκετές διαφορετικές συναρτήσεις βελτιστοποίησης, που λειτουργούν όμως με την ίδια λογική. Όσο αφορά το εργαλείο του minisat+ είναι γνωστές δύο συναρτήσεις. Η συνάρτηση minimum, η οποία προσπαθεί να βρει την ελάχιστη τιμή της συνάρτησης και η συνάρτηση maximum, η οποία προσπαθεί να βρει τη μέγιστη τιμή της συνάρτησης. Και οι δύο συναρτήσεις ουσιαστικά δουλεύουν με τον ίδιο τρόπο αφού προσπαθούν να ελαχιστοποιήσουν ή να μεγιστοποιήσουν το κόστος αντίστοιχα, ικανοποιώντας όσο το δυνατό περισσότερους στόχους (goals) με βάσει τους συνολικούς περιορισμούς ολόκληρου του προβλήματος. Η διαφορά τους έγκειται στον τρόπο εισαγωγής των στόχων που δίνονται ως παράμετροι μέσα στην κάθε συνάρτηση.

Η συνάρτηση minimum που δίνεται στο cnf αρχείο που παίρνει ο minisat+ γράφεται ως εξής :

min: -1*goal1 -1*goal2 -1*goal3-1*goaln

Αντίστοιχα η συνάρτηση maximum γράφεται ως εξής:

max: +1*goal1 +1*goal2 +1*goal3.....+1*goaln

Στις μεθόδους και τεχνικές βελτιστοποιήσεις, που θα εφαρμόσουμε με τη βοήθεια του minisat+, θα χρησιμοποιήσουμε τη συνάρτηση min. Αρχικά, μέσω της κωδικοποίησης SMP, θα δημιουργήσουμε το cnf αρχείο όπως περιγράφηκε πιο πάνω. Επιπλέον αφού θα εισάγουμε τη συνάρτηση βελτιστοποίησης θα πρέπει να εντοπίσουμε και τους στόχους του προβλήματος για το συγκεκριμένο επίπεδο στο οποίο αναζητούμε λύση. Οι στόχοι αυτοί αφού εντοπιστούν θα δοθούν ως παράμετρος στη συνάρτηση min. Θα προστεθεί έτσι ακόμη μια γραμμή στην αρχή του cnf αρχείου που θα έχει τη συνάρτηση min. Παράλληλα λόγω του ότι θα γίνεται προσπάθεια από τη συνάρτηση βελτιστοποίησης να ικανοποιηθούν όσο περισσότεροι στόχοι και όχι αναγκαστικά όλοι, θα αφαιρέσουμε όλους τους στόχους οι οποίοι προστέθηκαν στο αρχείο cnf ως μοναδιαίες προτάσεις (unit clauses). Σε μερικές τεχνικές που θα χρησιμοποιήσουμε στη συνέχεια οι στόχοι θα αφαιρούνται μεν από μοναδιαίες προτάσεις, ωστόσο είναι δυνατό να προσθέτουμε μια πρόταση, που θα περιλαμβάνει όλους τους στόχους μαζί για να εκφράσουμε με αυτόν τον τρόπο κάποιο κάτω φράγμα, που αφορά τους στόχους και που πρέπει οπωσδήποτε να ικανοποιηθεί (ελάχιστο ποσοστό στόχων που απαιτείται να ικανοποιηθεί).

5.5.2 Επίλυση προβλημάτων μέσω ενδιάμεσης κατάστασης βάσει ενός κάτω φράγματος – κατωφλίου

Με βάση την έννοια της βελτιστοποίησης, η οποία επεξηγήθηκε προηγουμένως θα δοκιμαστεί μια νέα μορφή βελτιστοποίησης με τη χρήση της συνάρτησης min. Η μέθοδος αυτή, εκμεταλλεύεται το γεγονός ότι η συνάρτηση βελτιστοποίησης προσπαθεί να ικανοποιήσει όσο περισσότερους στόχους είναι δυνατόν, δοκιμάζοντας όλους τους συνδυασμούς από στόχους που μπορούν να ικανοποιηθούν.

Η μέθοδος αυτή, δεν προσπαθεί να φτάσει στην τελική κατάσταση απευθείας. Αρχικός στόχος είναι να φτάσουμε σε μια ενδιάμεση κατάσταση ή κάποια άλλη κατάσταση που δίνει ο χρήστης και προηγείται της τελικής κατάστασης. Στη συνέχεια αφού φτάσουμε σε αυτήν την ενδιάμεση κατάσταση, δηλαδή ικανοποιούνται οι συγκεκριμένα στόχοι γι' αυτήν την κατάσταση, προσπαθούμε να φτάσουμε στην τελική κατάσταση του προβλήματος έχοντας ως δεδομένο όμως τις συνέπειες των δράσεων μέχρι την ενδιάμεση κατάσταση. Η ισχύς αυτής της μεθόδου στηρίζεται στο γεγονός ότι ενώ ψάχνουμε την τελική κατάσταση έχουμε ήδη γνωστή σημαντική πληροφορία όσο

αφορά τους περιορισμούς του προβλήματος. Με απλά λόγια η ενδιάμεση λύση δίνει τη δυνατότητα στον προτασιακό επιλυτή minisat+ να πάρει πιο αποδοτικά και πιο εύκολα αποφάσεις σχετικά με τις αναθέσεις (true ή false) που πρέπει να κάνει στις μεταβλητές που αφορούν τους όρους των προτάσεων. Μετά δηλαδή που έχει βρεθεί μια ενδιάμεση λύση, για κάθε επόμενο επίπεδο για το οποίο δημιουργούμε ένα νέο cnf αρχείο, προσθέτουμε σε αυτό ως μοναδιαίες προτάσεις, τις τιμές οι οποίες ανατέθηκαν στις μεταβλητές και με τις οποίες ο minisat+ κατέληξε στην ικανοποίηση της ενδιάμεσης αυτής λύσης. Έτσι η ενδιάμεση λύση είναι σημαντική, καθώς στη συνέχεια αποκλείει συνδυασμούς περιπτώσεων, για τις οποίες ο επιλυτής θα σπαταλούσε χρόνο για να αποφασίσει την κατάλληλη ανάθεση.

Αξιίζει να σημειωθεί ότι τα χαρακτηριστικά της ενδιάμεσης κατάστασης καθορίζονται από το χρήστη. Ο χρήστης δηλαδή έχει τη δυνατότητα να δώσει ένα κατώφλι, το οποίο μόνο όταν ξεπεραστεί μπορούμε να φτάσουμε στην ενδιάμεση λύση. Πιο συγκεκριμένα, ο χρήστης εισάγει στο πρόγραμμα μέσω της παραμέτρου –userthreshold ένα ακέραιο αριθμό από 0-100, ο οποίος αντιστοιχεί στο ποσοστό των στόχων του προβλήματος, που επιθυμεί να ικανοποιηθούν μέχρι την ενδιάμεση κατάσταση. Αν για παράδειγμα σε ένα πρόβλημα έχουμε 6 στόχους, και ο χρήστης δώσει κατώφλι 50, αυτό σημαίνει ότι επιθυμεί να ικανοποιηθεί ποσοστό 50% των στόχων, δηλαδή 3 στόχοι. Για ποσοστά όπου δεν υπάρχει ακέραια αντιστοιχία ανάμεσα σε στόχους και ποσοστό, γίνεται στρογγυλοποίηση όσο αφορά το πλήθος των στόχων που πρέπει να ικανοποιηθούν (π.χ. με ποσοστό 25% πρέπει να ικανοποιηθούν 2 στόχοι). Όταν ικανοποιηθεί το κατώφλι αυτό στη συνέχεια το αυξάνουμε στο 100%, αφού σημαίνει ότι έχουμε φτάσει στην ενδιάμεση λύση και θα ψάξουμε για την τελική λύση στην οποία θα πρέπει να ικανοποιηθούν όλα οι στόχοι.

Παράδειγμα: Ο χρήστης δίνει κατώφλι 80% σε ένα πρόβλημα με 4 στόχους έστω g1,g2,g3,g4.

Με απλούς υπολογισμούς προκύπτει ότι πρέπει να ικανοποιηθούν 3 οποιοιδήποτε στόχοι για την ενδιάμεση κατάσταση. Η πρόταση στην τελευταία γραμμή δείχνει το πλήθος των στόχων που πρέπει οπωσδήποτε να ικανοποιηθούν. Έτσι η μορφή του cnf αρχείου που θα δίδεται μέχρι να βρεθεί η ενδιάμεση λύση θα είναι η ακόλουθη:

```

min: -1*g1 -1*g2 -1*g3 -1*g4
+1*v1>=1
-1*v2>=0
.....
.....
.....
+1*g1 +1*g2 +1*g3 +1*g4 >= 3

```

Όταν βρεθεί η ενδιαμέση λύση ξεκινά η αναζήτηση της τελικής λύσης. Το κατώφλι αυξάνεται και πάει στο μέγιστο δηλαδή στο 100%, αφού θέλουμε τώρα να ικανοποιηθούν όλοι οι στόχοι και στη συγκεκριμένη περίπτωση και οι στόχοι. Έτσι στην τελευταία γραμμή που αφορά το πλήθος των στόχων που πρέπει να ικανοποιηθούν, βάζουμε 4 αφού πρέπει να ικανοποιηθούν όλοι οι στόχοι. Τώρα, η μορφή του cnf αρχείου που θα δίδεται μέχρι να βρεθεί η τελική λύση θα είναι η ακόλουθη :

```

min: -1*g1 -1*g2 -1*g3 -1*g4
+1*v1>=1
-1*v2>=0
.....
.....
.....
+1*g1 +1*g2 +1*g3 +1*g4 >= 4

```

Τέλος, όταν ικανοποιηθεί και το ανώτατο αυτό κατώφλι, αυτό υποδηλώνει ότι έχουμε φτάσει στην τελική λύση του προβλήματος. Έτσι ο minisat+ εξάγει την τελική λύση (ανάθεση στις τιμές των μεταβλητών) και την επιστρέφει στο satplan. Ο satplan με βάση πάντοτε το varfile που δίνει την αντιστοιχία μεταβλητών και περιγραφής των μεταβλητών δημιουργεί τη λύση. Η λύση περιλαμβάνει τις δράσεις που πρέπει να γίνουν με τη σωστή σειρά, ώστε να οδηγηθούμε στο στόχο του προβλήματος.

5.5.3 Τμηματική επίλυση προβλημάτων

Η επίλυση των προβλημάτων βελτιστοποίησης μέσω μιας ενδιάμεσης κατάστασης την οποία περιγράψαμε προηγουμένως, λύνει το πρόβλημα ουσιαστικά σε δύο ξεχωριστές φάσεις. Στην πρώτη φάση γίνεται προσπάθεια δηλαδή να βρεθεί μια ενδιάμεση λύση, και η δεύτερη φάση στηριζόμενη κατά κάποιο τρόπο στη γνώση που παρέχει η ενδιάμεση λύση, προσπαθεί να φτάσει στην τελική κατάσταση, όπου ικανοποιούνται όλοι οι περιορισμοί.

Στην ενότητα αυτή θα δοκιμαστεί μια διαφορετική μέθοδος, η οποία δε λύνει το πρόβλημα σε 2 στάδια. Η μέθοδος αυτή προσπαθεί να λύσει το όλο πρόβλημα τμηματικά. Το πρόβλημα δηλαδή διασπάται και γίνεται προσπάθεια να επιλυθεί σε ένα σύνολο από βήματα και όχι σε δύο βήματα όπως γινόταν στη μέθοδο με την ενδιάμεση κατάσταση. Ξεκινώντας από το πρώτο βήμα, προσπαθούμε να ικανοποιήσουμε (συνήθως) ένα πολύ μικρό αριθμό από στόχους. Στο δεύτερο βήμα, προσπαθούμε να ικανοποιήσουμε ένα μεγαλύτερο ποσοστό από τους στόχους του προβλήματος. Η διαδικασία αυτή επαναλαμβάνεται για ένα σύνολο από βήματα, μέχρι να ικανοποιηθούν όλοι οι στόχοι του προβλήματος.

Με αυτήν τη νέα στρατηγική γίνεται προσπάθεια να φτάσουμε σταδιακά στο στόχο. Η διάσπαση του προβλήματος δε γίνεται σε δύο επιμέρους προβλήματα αλλά σε ένα σύνολο από k προβλήματα. Με αυτό τον τρόπο, γίνεται και πάλι εκμετάλλευση της πληροφορίας που προκύπτει από τις επιμέρους λύσεις, ωστόσο αυτή η πληροφορία τώρα δε γίνεται γνωστή αφού ικανοποιηθεί ένα μεγάλο μέρος του προβλήματος (δηλαδή οι μισοί στόχοι) αλλά γίνεται σε πιο σύντομο χρονικό σημείο, αφού τα βήματα είναι μικρά (μικρός αριθμός από στόχους). Σε κάθε επόμενο βήμα, προστίθεται η πληροφορία του αμέσως προηγούμενου βήματος αλλά και όλων των προηγούμενων βημάτων που έδωσαν ένα μέρος της ολικής λύσης. Καθώς δηλαδή τα βήματα αυξάνονται μέχρι να φτάσουμε στο τελικό βήμα που θα ικανοποιήσει όλους τους στόχους, αυξάνονται και οι στόχοι που έχουν ικανοποιηθεί στις διάφορες χρονικές στιγμές.

Ακολουθεί η διαδικασία η οποία εφαρμόζεται σε κάθε βήμα της επίλυσης του προβλήματος. Αξίζει να σημειωθεί ότι όπως και στην τεχνική με την ενδιάμεση λύση, έτσι και εδώ ο χρήστης έχει την δυνατότητα να εισάγει ένα κατώφλι (ποσοστό) για τους

στόχους που στο αρχικό βήμα απαιτεί να ικανοποιηθούν, δηλαδή ένα κάτω φράγμα το οποίο προσαρμόζεται όπως πάντα με το πλήθος του τρέχοντος προβλήματος. Συγκεκριμένα ξεκινώντας από το πρώτο βήμα, ο satplan δημιουργεί με βάση την κωδικοποίηση SMP το cnf αρχείο, όπως ακριβώς το δημιουργούσε και στην περίπτωση της ενδιάμεσης λύσης: τοποθετεί στην πρώτη γραμμή τη συνάρτηση $\min$ που παίρνει ως παραμέτρους τους στόχους του προβλήματος τη δεδομένη χρονική στιγμή, δηλαδή την $t=1$. Εισάγει στη συνέχεια τους υπόλοιπους περιορισμούς του προβλήματος σε μορφή πρότασης (clause) και στην τελευταία γραμμή του αρχείου εισάγει τον περιορισμό που αφορά τον ελάχιστο αριθμό (κατώφλι που έδωσε ο χρήστης) από στόχους που πρέπει να ικανοποιηθούν. Το cnf αρχείο δίνεται στο minisat+, ο οποίος προσπαθεί να βρεί λύση. Αν βρει λύση την επιστρέφει στον satplan ενημερώνοντας τον ταυτόχρονα ποιοι από τους στόχους έχουν ικανοποιηθεί για το συγκεκριμένο αρχικό κατώφλι και ο satplan ξεκινά να δημιουργήσει το δεύτερο βήμα. Αν δε βρει λύση ο minisat+, ο satplan παραμένει στο πρώτο βήμα, επεκτείνοντας το γράφημα στο επόμενο επίπεδο και δημιουργεί με την ίδια λογική νέο cnf με το ίδιο αρχικό κατώφλι (αφού δεν έχει ικανοποιηθεί), με τους στόχους να αφορούν τώρα τη χρονική στιγμή $t=2$. Η διαδικασία επαναλαμβάνεται μέχρι $t=k$, όπου k η χρονική στιγμή – επίπεδο στο οποίο έχει βρεθεί λύση που ικανοποιεί το αρχικό κατώφλι.

Ακολούθως, προχωράμε στο δεύτερο βήμα, κατά το οποίο ο satplan δημιουργεί το cnf αρχείο με τον ίδιο τρόπο, αλλά τώρα το κατώφλι που προσθέτει στην τελευταία γραμμή, είναι τέτοιο ώστε οι στόχοι που πρέπει να ικανοποιηθούν να είναι ίσοι με το πλήθος των στόχων του βήματος 1 αυξημένοι με 1. Επίσης, στο αρχείο τοποθετεί ως περιορισμούς τους στόχους που ικανοποιήθηκαν στο πρώτο βήμα.

Αυτή η διαδικασία συνεχίζεται για ένα αριθμό από βήματα, μέχρι το τελικό κατώφλι που αφορά το συνολικό αριθμό των στόχων να ικανοποιηθεί. Γίνεται δηλαδή σταδιακή επίλυση του προβλήματος, προσπαθώντας να ικανοποιήσουμε σε κάθε βήμα τους επιμέρους στόχους.

Παράδειγμα: Το πρόβλημα trucks 1 έχει τρεις στόχους. Ο χρήστης δίνει κατώφλι 25%, εισάγοντας τον αριθμό 25.

Ο satplan δημιουργεί το γράφημα μέχρι το 5^ο επίπεδο, επίπεδο στο οποίο κωδικοποιούνται-εμφανίζονται για πρώτη φορά οι στόχοι του προβλήματος. Έτσι, στο επίπεδο 5 ξεκινά το 1^ο βήμα και η μορφή του cnf αρχείου είναι ως εξής:

```
min: -1*683 -1*665 -1*666
+1*v1>=1
-1*v2>=0
.....
.....
.....
+1*683 +1*665 +1*666 >= 1
```

Όπως διαφαίνεται από το αρχείο οι στόχοι στο επίπεδο 5 είναι οι : 683,665 και 666. Το αρχικό κατώφλι είναι 1, καθώς το ποσοστό 25% που έδωσε για κατώφλι ο χρήστης αντιστοιχεί στο σύνολο των τριών στόχων του προβλήματος σε ένα στόχο. Καλείται ακολούθως ο minisat+, ο οποίος ψάχνει για λύση. Ο minisat+ βρίσκει λύση ικανοποιώντας το στόχο 683 – επιτυγχάνει δηλαδή να ικανοποιήσει ένα στόχο από τους τρεις που είναι και το αρχικό κάτω φράγμα. Έτσι αφού βρέθηκε λύση για αυτό το κατώφλι προχωράμε στο βήμα 2. Ο satplan επεκτείνει το γράφημα μέχρι το 6^ο επίπεδο και η μορφή του cnf αρχείου τώρα είναι ως εξής :

```
min: -1*1053 -1*1035 -1*1036
+1*v683
+1*v1>=1
-1*v2>=0
.....
.....
.....
+1*1053 +1*1035 +1*1036 >= 2
```

Εισαγωγή του στόχου που ικανοποιήθηκε στο βήμα 1 ως περιορισμός στο βήμα 2

Αύξηση του κατωφλίου από 1 σε 2

Οι αντίστοιχοι στόχοι στο επίπεδο 6 είναι οι : 1053,1035 και 1036. Ο στόχος που ικανοποιήθηκε στο βήμα 1 τίθεται ως περιορισμός (ο στόχος 683), ενώ το κατώφλι αυξάνεται από 1 σε 2 (προσπάθεια ικανοποίησης 2 στόχων).

Στο επίπεδο 6 ο minisat+ δε βρίσκει λύση που να ικανοποιεί αυτό το κατώφλι, έτσι παραμένουμε στο βήμα 2. Αφού παραμένουμε στο βήμα 2, ο satplan επεκτείνει το γράφημα στο επίπεδο 7, διατηρώντας όμως τώρα το κατώφλι στο 2 αφού δεν έχει ακόμη ικανοποιηθεί. Το cnf αρχείο τώρα είναι ως εξής :

```
min: -1*1446 -1*1428 -1*1429
+1*v683
+1*v1>=1
-1*v2>=0
.....
.....
.....
+1*1446 +1*1428 +1*1429 >= 2
```

Οι στόχοι 1446,1428 και 1429 είναι οι αντίστοιχοι στόχοι τη χρονική στιγμή $t=7$, ενώ το κατώφλι είναι ίσο με 2. Στο επίπεδο 7, ο minisat+ βρίσκει λύση αφού καταφέρνει να ικανοποιήσει 2 στόχους, τους 1446 και 1428. Έτσι, πάμε στο επόμενο βήμα (3). Το κατώφλι αυξάνεται από 2 σε 3 (που είναι και το τελικό κατώφλι αφού αντιστοιχεί σε ποσοστό 100% των στόχων), τίθενται ως περιορισμοί οι στόχοι που ικανοποιηθήκαν μέχρι τώρα και επαναλαμβάνεται με τον ίδιο τρόπο η διαδικασία. Μέχρι το επίπεδο 11 ο minisat+, ικανοποιεί και τους 3 στόχους και είναι σε θέση να δώσει την τελική λύση του προβλήματος. Έχοντας δημιουργήσει αυτήν τη νέα μεθοδολογία, θα δοκιμαστούν τώρα τρεις διαφορετικές τεχνικές που σχετίζονται με τον τρόπο εισαγωγής των περιορισμών (στόχων) που ικανοποιούνται σε ένα επίπεδο και εισάγονται στο αμέσως επόμενο επίπεδο του αμέσως επόμενου βήματος.

5.5.3.1 Τεχνική διατήρησης ικανοποιηθέντων περιορισμών

Η τεχνική διατήρησης ικανοποιηθέντων περιορισμών, διατηρεί τους στόχους όπως αυτοί ικανοποιήθηκαν στο εκάστοτε επίπεδο και τους εισάγει ως περιορισμούς στα επόμενα επίπεδα που ακολουθούν. Στο πιο πάνω παράδειγμα, εφαρμόστηκε η τεχνική διατήρησης ικανοποιηθέντων περιορισμών. Οι στόχος δηλαδή 683, ο οποίος ικανοποιήθηκε τη χρονική στιγμή $t=5$ εισήχθη ως περιορισμός και στις επόμενες

χρονικές στιγμές – επίπεδα που ακολούθησαν. Παρομοίως, οι στόχοι 1446 και 1428 που ικανοποιήθηκαν στο επίπεδο 7.

5.5.3.2 Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών

Η τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών, αποτελεί παραλλαγή της προηγούμενης τεχνικής. Οι στόχοι οι οποίοι ικανοποιούνται σε μια χρονική στιγμή έστω $t=k$, εισάγονται ως περιορισμοί τις επόμενες χρονικές στιγμές $k+1, k+2, \dots, k+n$, αλλά όχι με τη μορφή που ικανοποιήθηκαν τη στιγμή $t=k$ αλλά ως περιορισμοί στόχων της δεδομένης χρονικής στιγμής. Δηλαδή, αν στο πιο πάνω παράδειγμα εφαρμόζοταν αυτή η τεχνική στο επίπεδο 6 δεν θα εισαγόταν ο στόχος 683 ως περιορισμός αλλά ο στόχος 1053 που αντιστοιχεί στη χρονική στιγμή 6. Το ίδιο και με τους στόχους 1446 και 1428 που ικανοποιήθηκαν στο επίπεδο 7.

Έτσι σε κάθε επίπεδο τοποθετούμε τους στόχους ως περιορισμούς που σχετίζονται με τη συγκεκριμένη χρονική στιγμή και όχι με τη χρονική στιγμή που ικανοποιήθηκαν αυτοί οι στόχοι.

5.5.3.3 Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών – συνδυασμός διατήρησης και ενημέρωσης περιορισμών

Μια τρίτη τεχνική τμηματικής επίλυσης του προβλήματος, η οποία θα δοκιμαστεί, σχετίζεται με το συνδυασμό της ενημέρωσης και της διατήρησης των περιορισμών. Σε αυτήν την τεχνική, οι στόχοι οι οποίοι ικανοποιούνται τη χρονική στιγμή $t=k$, διατηρούνται για ένα δεδομένο αριθμό χρονικών στιγμών (όπως ακριβώς η λειτουργία της τεχνικής διατήρησης που περιγράψαμε πιο πάνω) και ενημερώνονται μόνο όταν περάσει ένας συγκεκριμένος αριθμός χρονικών στιγμών (όπως η τεχνική ενημέρωσης που περιγράψαμε πιο πάνω). Ο αριθμός των χρονικών στιγμών που διατηρούνται οι στόχοι, μπορεί να καθοριστεί από μια παράμετρο (την -stepgoals) που δίνει ο χρήστης στο πρόγραμμα. Έτσι, με βάση τον τρόπο λειτουργίας αυτής της τεχνικής, η τεχνική είναι ένας συνδυασμός των 2 προηγούμενων τεχνικών που περιγράφηκαν. Αξίζει να σημειωθεί ότι όταν ο αριθμός των χρονικών στιγμών που διατηρούνται οι στόχοι είναι ίσος με 1, η τεχνική αυτή συμπεριφέρεται όπως ακριβώς η τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών. Αποτελεί σε αυτήν την περίπτωση η συνδυασμένη αυτή

τεχνική κατά κάποιο τρόπο γενίκευση της τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών.

Κεφάλαιο 6

Πειραματική Ανάλυση

6.1 Επίλυση προβλημάτων και μετρήσεις με το εργαλείο Precosat – χρήση της κωδικοποίησης SMP και BlackBox	58
6.2 Μετρήσεις από την επίλυση προβλημάτων μέσω ενδιάμεσης κατάστασης βάσει ενός κάτω φράγματος – κατωφλίου	60
6.3 Μετρήσεις από την τμηματική επίλυση των προβλημάτων	65

6.1 Επίλυση προβλημάτων και μετρήσεις με το εργαλείο Precosat – χρήση της κωδικοποίησης SMP και BlackBox

Ο πιο κάτω πίνακας 6.1.1 αφορά τις μετρήσεις που έγιναν από την επίλυση των προβλημάτων Pathways και Rovers, χρησιμοποιώντας τον προτασιακό επιλυτή Precosat. Για κάθε μια από αυτές τις δύο διαφορετικές κατηγορίες προβλημάτων δοκιμάστηκε η επίλυση του κάθε προβλήματος με δύο διαφορετικές κωδικοποιήσεις: την SMP και την BlackBox. Όπως είναι εμφανές από τον πίνακα η κωδικοποίηση SMP καταφέρνει να επιλύσει σε μικρότερο χρόνο όλα τα προβλήματα σε αντίθεση με την κωδικοποίηση BlackBox, η οποία σε μερικά προβλήματα καθυστερεί πολύ περισσότερο από την SMP. Μεγαλύτερη διαφορά στην χρονική επίλυση των προβλημάτων παρατηρείται στα προβλήματα Pathways παρά στα προβλήματα Rovers, για το λόγο ότι η κατηγορία προβλημάτων Pathways, περιλαμβάνει προβλήματα με περισσότερους συνδυασμούς περιπτώσεων και γενικά πιο πολλούς περιορισμούς ανάμεσα στις διάφορες μορφές προτάσεων. Και οι δύο κωδικοποιήσεις όπως ήταν αναμενόμενο δίνουν τη λύση στο ίδιο αριθμό επιπέδων που κτίζεται από το γράφημα δράσεων.

	Προβλήματα Pathways				Προβλήματα Rovers			
	Κωδικοποίηση <i>SMP</i>		Κωδικοποίηση <i>BlackBox</i>		Κωδικοποίηση <i>SMP</i>		Κωδικοποίηση <i>BlackBox</i>	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0.05	5	0.05	5	0.02	5	0.03
p02	7	0.06	7	0.07	4	0.02	4	0.02
p03	8	0.19	8	0.24	7	0.06	7	0.06
p04	8	0.28	8	0.32	4	0.02	4	0.03
p05	9	0.75	9	0.83	5	0.06	5	0.06
p06	12	4.41	12	6.78	9	0.23	9	0.35
p07	13	11.67	13	21.59	5	0.09	5	0.11
p08	17	248.03	17	2053.3	6	0.41	6	0.45
p09	17	2623.89	-	Aborted Solver	10	0.86	10	1.48
p10	15	4.71	15	7.08	6	0.39	6	0.45
p11	17	147.59	17	2109.11	10	0.96	10	1.13
p12	19	3099.8	-	Aborted Solver	5	0.23	5	0.27
p13	18	629.76	18	1954.86	8	1.97	8	2.53
p14	-	Aborted Solver	-	Aborted Solver	8	0.89	8	1.28
p15	18	59.48	18	99	9	3.47	9	6.5
p16	21	891.06	21	4156.26	8	2.23	8	2.91
p17	21	1135.52	21	2938.34	11	10.38	11	15.32
p18	-	Aborted Solver	-	Aborted Solver	6	3.79	6	5.96
p19	-	Aborted Solver	-	Aborted Solver	9	33.12	9	47.99
p20	-	Aborted Solver	-	Aborted Solver	10	97.22	10	139.03

Πίνακας 6.1.1 : Κωδικοποίηση SMP και BlackBox

6.2 Μετρήσεις από την επίλυση προβλημάτων μέσω ενδιάμεσης κατάστασης βάσει ενός κάτω φράγματος – κατώφλιου

Σε αυτό το στάδιο επίκεντρο της μελέτης είναι η μεθοδολογία που περιγράφηκε πιο πάνω που σχετίζεται με την επίλυση ενός προβλήματος μέσω ενδιάμεσης κατάστασης. Η μεθοδολογία αυτή δοκιμάστηκε σε μερικά προβλήματα προγραμματισμού δράσης με τρία διαφορετικά ποσοστά κατώφλιου: 10% στον πίνακα 6.2.1, 25% στον πίνακα 6.2.2 και 50% στον πίνακα 6.2.3.

Στην κατηγορία των Pathways μπορούμε να δούμε ότι το κατώφλι 10% επιτυγχάνει να βελτιώσει σε αρκετά προβλήματα το χρόνο εκτέλεσης αλλά περισσότερο τον αριθμό των επιπέδων που επεκτείνεται το γράφημα μέχρι να βρεθεί λύση. Χαρακτηριστικά στο πρόβλημα 9 επιλύει το πρόβλημα σε 22 επίπεδα και χρόνο 45.57 s (πίνακας 6.2.1), ενώ το ίδιο πρόβλημα επιλύεται με κατώφλι 25% σε 23 επίπεδα και χρόνο 4056.38 s, ενώ ακόμη χειρότερα με το 50% όπου το πρόβλημα δεν είναι δυνατό να ικανοποιηθεί (Unsat) – παρόμοια και για το πρόβλημα 16. Παράλληλα, σε μερικά προβλήματα αρκετά καλά συμπεριφέρεται και το κατώφλι με 25%, όπως τα προβλήματα 14 και 15. Το κατώφλι 50% είναι ίσως η χειρότερη επιλογή γι' αυτήν την κατηγορία προβλημάτων, καθώς σε μερικές περιπτώσεις δεν επιλύει ούτε καν το πρόβλημα, ενώ σε άλλες καθυστερεί πολύ περισσότερο σε σχέση με τα αντίστοιχα των 10% και 25% κατώφλια.

Στην κατηγορία TPP, λόγω της σχετικά μικρής πολυπλοκότητας των προβλημάτων δεν υπάρχουν μεγάλες διαφορές για το κάθε κατώφλι. Δεδομένου μια συγκεκριμένης παραγγελίας οι μόνες παράμετροι που πρέπει να ληφθούν υπόψη είναι το κόστος δρομολόγησης και κόστος αγοράς. Ένα αποδοτικό κατώφλι γι αυτήν την κατηγορία μπορεί να θεωρηθεί το 50%, όπως μπορεί να διαφανεί και από τους αντίστοιχους πίνακες. Με ένα ποσοστό 50% επιλύεται αρχικά το μισό πρόβλημα, και λόγω του ότι οι δυνατές περιπτώσεις είναι μειωμένες, οι περιορισμοί από την πρώτη «μισή» λύση (που οπωσδήποτε πρέπει να ικανοποιούνται) δεν «περιορίζουν» τον επιλυτή στην επίλυση του υπόλοιπου προβλήματος.

Στα προβλήματα PipeWorlds το κατώφλι 50% είναι δυνατόν να οδηγήσει σε πιο γρήγορη χρονικά λύση σε σχέση με αυτά των 10% και 25%, κυρίως σε μεγάλα προβλήματα. Χαρακτηριστικό παράδειγμα το πρόβλημα 13, όπου επιλύεται σε μόλις

78.05 s και σε 13 επίπεδα (με κατώφλι 50%), ενώ τα άλλα δύο κατώφλια το επιλύουν σε 21 επίπεδα και με περίπου έξι φορές μεγαλύτερο χρόνο, ≈ 480 s. Παράλληλα σε πολλά μικρότερα προβλήματα το κατώφλι 50% επιτυγχάνει μικρότερο ή ίδιο χρόνο σε σχέση με τα άλλα δύο κατώφλια.

Στα Depots προβλήματα, το κατώφλι με ποσοστό 25% τείνει να συμπεριφέρεται πιο αποδοτικά από τα άλλα δύο. Είναι προτιμότερο δηλαδή σε τέτοιου είδους προβλήματα να ικανοποιηθεί ένας μέσος αριθμός στόχων αρχικά, και στη συνέχεια να συνεχίσουμε με τους υπόλοιπους στόχους.

Όσον αφορά στην κατηγορία Storage τόσο το κατώφλι 25% όσο και 50% είναι εξίσου αποδοτικά. Σε μερικά προβλήματα με το κατώφλι 50% μειώνεται αισθητά ο χρόνος, ενώ σε μερικά άλλα το κατώφλι 25% επιτυγχάνει μικρή μείωση στα επίπεδα που επεκτείνεται το γράφημα, για παράδειγμα στο πρόβλημα 13 το κατώφλι 50% εξασφαλίζει χρόνο 18,13 s σε 22 επίπεδα, ενώ το κατώφλι 25% χρόνο 60.22 s αλλά σε μόλις 20 επίπεδα.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η κατηγορία των Trucks. Εδώ για τα περισσότερα προβλήματα όχι μόνο δε βρίσκουμε αποδοτική λύση, αλλά δεν είναι δυνατό να βρούμε μία οποιοδήποτε λύση. Οι τρεις αυτές τεχνικές με το μεταβαλλόμενο κατώφλι αποτυγχάνουν εξίσου. Ο λόγος είναι ότι αυτά τα προβλήματα διέπονται από αυστηρούς χρονικούς και χωρικούς περιορισμούς. Βάση αυτού μια οποιαδήποτε ενδιάμεση λύση, η οποία στη συνέχεια κατά την αναζήτηση της τελικής λύσης προσθέτει επιπλέον περιορισμούς σε συγκεκριμένες προηγούμενες χρονικές στιγμές, καθιστά την προσπάθεια του επιλυτή ακόμη πιο δύσκολη αφού πρέπει να ικανοποιήσει για παράδειγμα δύο στόχους σε 2 διαφορετικές χρονικές στιγμές ταυτόχρονα.

Κατώφλι 10%						
	Προβλήματα Pathways		Προβλήματα TPP		Προβλήματα PipeWorlds	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	6	0,23	6	0,04	4	0,21
p02	8	0,26	6	0,05	13	2,7
p03	11	0,82	6	0,06	8	4,64
p04	9	0,69	6	0,07	8	4,77
p05	-	UNSAT	8	0,18	7	3,57
p06	-	UNSAT	13	1,3	7	3,65
p07	20	26,1	15	2,19	6	14,36
p08	23	45,08	14	1,86	8	37,34
p09	22	45,57	19	5,71	13	187,42
p10	17	12,13	18	6,12	14	214,17
p11	22	52,33	22	14,26	19	90,67
p12	24	58,62	22	18,66	24	529,75
p13	21	236,42	17	24,24	21	494,45
p14	22	2017,68	16	25,93	-	Aborted Solver
p15	21	887,86	17	36,63	-	Aborted Solver
p16	26	457,01	11	28,1	-	Aborted Solver
p17	26	414,34	12	69,74	-	Aborted Solver
p18	-	Aborted Solver	23	530,36	22	1531,02
p19	24	2592,58	18	403,45	-	Aborted Solver
p20	-	Aborted Solver	13	177,21	-	Aborted Solver

Κατώφλι 10%						
	Προβλήματα Depots		Προβλήματα Storage		Προβλήματα Trucks	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	6	0,12	4	0,03	-	Unsat
p02	9	0,51	4	0,04	-	Unsat
p03	18	5,77	3	0,04	-	Unsat
p04	25	23,45	10	0,43	21	60,94
p05	30	65,17	6	0,13	-	Unsat
p06	40	457,53	6	0,16	-	Unsat
p07	12	2,79	16	2,44	-	Unsat
p08	17	18,63	10	1,51	-	Unsat
p09	-	Aborted Solver	7	0,8	22	638,22
p10	16	12,12	20	12,22	-	Aborted Solver
p11	29	274,91	11	4,27	-	Aborted Solver
p12	-	Aborted Solver	9	3,03	-	Aborted Solver
p13	10	2,34	20	60,86	-	Aborted Solver
p14	20	80,39	13	10,18	-	Aborted Solver
p15	-	Aborted Solver	11	8,43	-	Aborted Solver
p16	11	7,49	11	20,62	-	Aborted Solver
p17	11	29,39	13	2679,7	-	Aborted Solver
p18	13	269,66	-	Aborted Solver	-	Aborted Solver
p19	16	53,32	-	Aborted Solver	-	Aborted Solver
p20	27	2106,51	-	Aborted Solver	-	Aborted Solver

Πίνακας 6.2.1 : Μέθοδος ενδιάμεσης κατάστασης με κατώφλι 10%

Κατώφλι 25%						
	Προβλήματα Pathways		Προβλήματα TPP		Προβλήματα PipeWorlds	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	6	0,21	6	0,04	4	0,22
p02	8	0,28	6	0,04	13	2,89
p03	11	0,82	6	0,05	8	4,73
p04	9	0,65	6	0,07	8	4,7
p05	12	2,07	8	0,17	7	3,65
p06	56	77,92	17	2,3	7	3,76
p07	38	68,36	15	2,03	6	14,17
p08	20	29,51	14	1,93	8	37,62
p09	23	4056,38	21	7,42	12	135,95
p10	18	14,04	20	7,94	14	341
p11	20	30,86	22	15,11	19	90,61
p12	24	58,82	20	13,82	-	Aborted Solver
p13	20	272,08	14	14,19	21	495,28
p14	23	352,87	15	20,49	-	Aborted Solver
p15	21	232,96	16	30,99	-	Aborted Solver
p16	29	4938,8	11	28,41	-	Aborted Solver
p17	-	Aborted Solver	12	67,86	-	Aborted Solver
p18	26	3098,81	18	296,26	22	1574,77
p19	26	887,83	17	346,44	-	Aborted Solver
p20	-	Aborted Solver	13	185,93	-	Aborted Solver

Κατώφλι 25%						
	Προβλήματα Depots		Προβλήματα Storage		Προβλήματα Trucks	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	6	0,1	4	0,02	-	Unsat
p02	9	0,41	4	0,04	-	Unsat
p03	20	8,01	4	0,03	-	Unsat
p04	23	18,43	10	0,4	-	Unsat
p05	30	63,64	6	0,11	-	Unsat
p06	40	481,54	6	0,17	-	Unsat
p07	12	2,75	16	2,44	-	Unsat
p08	18	22,36	10	1,44	-	Unsat
p09	-	Aborted Solver	7	0,86	-	Unsat
p10	16	12,03	20	12,11	-	Aborted Solver
p11	26	136,02	11	4,33	-	Aborted Solver
p12	-	Aborted Solver	9	3,03	-	Aborted Solver
p13	10	2,42	20	60,22	-	Aborted Solver
p14	20	81,28	13	10,1	-	Aborted Solver
p15	29	1108,94	11	8,12	-	Aborted Solver
p16	11	7,49	11	13,48	-	Aborted Solver
p17	12	38,72	13	143,35	-	Aborted Solver
p18	15	439,51	-	Aborted Solver	-	Aborted Solver
p19	16	54,31	-	Aborted Solver	-	Aborted Solver
p20	25	1769,92	-	Aborted Solver	-	Aborted Solver

Πίνακας 6.2.2 : Μέθοδος ενδιάμεσης κατάστασης με κατώφλι 25%

Κατώφλι 50%						
	Προβλήματα Pathways		Προβλήματα TPP		Προβλήματα PipeWorlds	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	6	0,24	6	0,02	4	0,17
p02	8	0,32	6	0,05	13	2,98
p03	11	1	6	0,04	7	3,08
p04	9	0,81	6	0,05	8	4,43
p05	46	38,5	8	0,15	7	3,59
p06	16	9,93	14	1,31	7	3,66
p07	17	19,09	15	1,99	6	14,22
p08	21	54,05	16	2,64	8	36,25
p09	-	UNSAT	19	5,72	13	212,5
p10	18	14,75	20	7,73	14	223,09
p11	21	35,54	20	10,68	19	88,34
p12	-	UNSAT	20	13,63	-	Aborted Solver
p13	20	143,65	15	17,38	13	78,05
p14	-	Aborted Solver	16	25,18	-	Aborted Solver
p15	21	1173,28	16	33,92	-	Aborted Solver
p16	25	5298,5	11	27,74	-	Aborted Solver
p17	-	Aborted Solver	12	77,53	-	Aborted Solver
p18	-	Aborted Solver	17	306,36	22	1653,76
p19	-	Aborted Solver	18	451,06	-	Aborted Solver
p20	-	Aborted Solver	13	223,07	-	Aborted Solver

Κατώφλι 50%						
	Προβλήματα Depots		Προβλήματα Storage		Προβλήματα Trucks	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	6	0,11	4	0,04	-	Unsat
p02	10	0,66	4	0,02	-	Unsat
p03	20	8,18	4	0,04	-	Unsat
p04	22	15,32	10	0,44	20	39,55
p05	30	63,89	11	0,11	-	Unsat
p06	40	715,94	6	0,17	-	Unsat
p07	11	2,06	16	1,92	-	Unsat
p08	15	12,22	8	0,68	-	Unsat
p09	-	Aborted Solver	7	0,81	-	Unsat
p10	16	12	20	6,51	-	Aborted Solver
p11	24	179,48	11	3,17	-	Aborted Solver
p12	33	4355,22	9	2,45	-	Aborted Solver
p13	10	2,31	22	18,13	-	Aborted Solver
p14	20	80,33	13	5,44	-	Aborted Solver
p15	31	1311,38	11	6,98	-	Aborted Solver
p16	15	19,87	11	13,46	-	Aborted Solver
p17	13	49,03	13	68,82	-	Aborted Solver
p18	19	860,64	16	1735,77	-	Aborted Solver
p19	18	75,67	-	Aborted Solver	-	Aborted Solver
p20	23	1507,05	-	Aborted Solver	-	Aborted Solver

Πίνακας 6.2.3 : Μέθοδος ενδιάμεσης κατάστασης με κατώφλι 50%

6.3 Μετρήσεις από την τμηματική επίλυση των προβλημάτων

Μετά από τα πάνω πειραματικά αποτελέσματα, δοκιμάστηκε η παραλλαγή της μεθόδου που επίλυε το πρόβλημα μέσω ενδιάμεσης κατάστασης. Προσπαθήσαμε να επιλύσουμε τα ίδια προβλήματα προγραμματισμού δράσης, αλλά αυτήν την φορά τμηματικά, δηλαδή σε φάσεις και όχι μόνο σε δύο φάσεις. Στον πίνακα 6.3.1 ως πρώτη τεχνική χρησιμοποιήθηκε η τεχνική διατήρησης ικανοποιηθέντων περιορισμών, στον πίνακα 6.3.2 ως δεύτερη τεχνική χρησιμοποιήθηκε η τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα και στον πίνακα 6.3.3 ως τρίτη τεχνική χρησιμοποιήθηκε η τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 3 βήματα, όπως αυτές περιγράφηκαν στο κεφάλαιο 5.

Με βάση τα αποτελέσματα στα διάφορα προβλήματα, εξάγεται το συμπέρασμα ότι δεν υπάρχει κάποια τεχνική η οποία να είναι καλύτερη από τις υπόλοιπες για οποιαδήποτε κατηγορία προβλημάτων, ακόμα και για οποιοδήποτε πρόβλημα συγκεκριμένης κατηγορίας.

Για παράδειγμα για την κατηγορία των Pathways είναι εύκολο να διακρίνουμε ότι τα προβλήματα 8 και 9 λύνονται σε μικρότερο αριθμό επιπέδων με τη δεύτερη και τρίτη τεχνική από ότι στην πρώτη αλλά η πρώτη τεχνική επιτυγχάνει πολύ χαμηλότερους χρόνους εκτέλεσης. Επίσης το πρόβλημα 15 επιλύεται σε σύνολο μόλις 19 επιπέδων και με τις τρεις τεχνικές αλλά η πρώτη τεχνική επιτυγχάνει καλύτερο χρόνο εκτέλεσης. Συνεπώς, νοούμενου ότι μας ενδιαφέρει το πλήθος των χρονικών βημάτων (επιπέδων) η δεύτερη και η τρίτη τεχνική είναι πιο αποδοτικές και έχουν λιγότερο κόστος.

Η μικρή πολυπλοκότητα των προβλημάτων TPP (εξαιρουμένου του προβλήματος 19), έχει ως αποτέλεσμα και πάλι η δεύτερη και τρίτη τεχνική να θεωρούνται ως οι πιο κατάλληλες με ελαφρώς πιο αποδοτική τη δεύτερη τεχνική. Γενικά, όμως και οι τρεις τεχνικές μπορούν να εφαρμοστούν εξίσου και να δώσουν ικανοποιητικά αποτελέσματα, αφού οι διαφορές τους στο χρόνο εκτέλεσης και στον αριθμό των επιπέδων μέχρι να φτάσουν σε λύση είναι πολύ μικρές.

Όσον αφορά τα προβλήματα PipeWorlds, από τις μετρήσεις δεν είναι δυνατόν να εξαχθεί ασφαλές συμπέρασμα εξαιτίας του ότι η κάθε τεχνική επιλύει καλύτερα ένα

υποσύνολο από αυτά τα προβλήματα και όχι όλα. Μερικές τεχνικές έχουν κατά πολύ καλύτερο χρόνο εκτέλεσης, ενώ άλλες έχουν σημαντική μείωση στον αριθμό των επιπέδων.

Σημαντική είναι επίσης η βελτίωση που δίνει η τεχνική ενημέρωσης των ικανοποιηθέντων περιορισμών μετά από 2 βήματα στη κατηγορία προβλημάτων Depots. Η τεχνική αυτή καταφέρνει να μειώσει κατά πολύ τον αριθμό των επιπέδων στα οποία επεκτείνεται το γράφημα σε σχέση με την τεχνική διατήρησης των ικανοποιηθέντων περιορισμών. Αυτή η βελτίωση έχει ως συνεπακόλουθο την αισθητή μείωση του κόστους, καθώς η επέκταση του γραφήματος προϋποθέτει και την ανάλογη αύξηση στην χρονική πολυπλοκότητα. Χαρακτηριστικά παραδείγματα είναι το πρόβλημα 5 όπου η τεχνική διατήρησης των ικανοποιηθέντων περιορισμών το επιλύει επεκτείνοντας το γράφημα μέχρι 56 επίπεδα ενώ η τεχνική ενημέρωσης των ικανοποιηθέντων περιορισμών το επιλύει σε σύνολο 29 επιπέδων, ο αριθμός των επιπέδων είναι σχεδόν ο μισός σε σχέση με την πρώτη τεχνική, όπως επίσης και τα προβλήματα 16,17,18,19. Επίσης, στα προβλήματα 6 και 15 η πρώτη τεχνική δεν μπορεί ούτε καν να τα ικανοποιήσει (Unsat). Παράλληλα, η τρίτη τεχνική συμπεριφέρεται κάπου στο ενδιάμεσο αυτών των δυο τεχνικών.

Πέραν από τις προαναφερθείσες κατηγορίες προβλημάτων, σημαντική βελτίωση παρατηρείται στην επίλυση των προβλημάτων κατηγορίας trucks. Η πρώτη τεχνική και πολύ περισσότερο η δεύτερη και η τρίτη τεχνική επιτυγχάνουν να ικανοποιήσουν πολλά από τα προβλήματα, για τα οποία οι τρεις μεθοδολογίες (με κατώφλι 10%, 25% και 50%) που είδαμε προηγουμένως δεν μπορούσαν. Αυτό συμβαίνει διότι η δεύτερη και η τρίτη τεχνική παρακάμπτουν κατά κάποιο τρόπο τους χρονικούς περιορισμούς των στόχων, οι οποίοι ικανοποιήθηκαν σε διαφορετικές χρονικές στιγμές. Οι στόχοι δηλαδή σε αυτές τις τεχνικές ενημέρωσης τοποθετούνται ως περιορισμοί της τρέχουσας χρονικής στιγμής και όχι της χρονικής στιγμής που ικανοποιηθήκαν. Αυτό διευκολύνει τον επιλυτή, καθώς τα προβλήματα αυτά έχουν σημαντικούς χρονικούς και χωρικούς περιορισμούς.

Τέλος, στον πίνακα 6.3.4, έχει δοκιμαστεί η τεχνική διατήρησης τεχνική ενημέρωσης των ικανοποιηθέντων περιορισμών μετά από 2 βήματα με κατώφλι 10% και 25% αντί 50% στις κατηγορίες προβλημάτων Pathways, PipeWorld και Depots. Αυτό έγινε με σκοπό να παρατηρηθεί η συμπεριφορά της τεχνικής αυτής όταν ξεκινά από μικρότερο κάτω φράγμα στόχων – περιορισμών.

Στα προβλήματα Pathways, τόσο το κατώφλι 10% όσο και το κατώφλι 25% δεν επιτυγχάνουν ουσιαστική βελτίωση σε σχέση με το αντίστοιχο του 50% με την ίδια πάντα τεχνική (Πίνακας 6.3.2). Οι χρόνοι εκτέλεσης και ο αριθμός επιπέδων είναι πολύ κοντά. Κάποια μικρή βελτίωση στο χρόνο εκτέλεσης μπορεί να διαπιστωθεί ότι γίνεται με το κατώφλι 10% σε μεγάλα κυρίως προβλήματα όπως για παράδειγμα: 8,9,11,13 και 15. Ειδικά στο πρόβλημα 9 επιτυγχάνεται μεγάλη μείωση του χρόνου από 4042.34 σε 522.82, αλλά και στο 15 μείωση από 1305.9 σε 757.84.

Στα προβλήματα PipeWorlds δεν παρατηρείται καμία ουσιαστική διαφορά με αυτές τις τρεις διαφορετικές τιμές στο κατώφλι (10%,25% και 50%) για την τεχνική διατήρησης τεχνική ενημέρωσης των ικανοποιηθέντων περιορισμών μετά από 2 βήματα. Παράλληλα, κάποια μικρή βελτίωση παρατηρείται και πάλι με το κατώφλι 10% κυρίως σε μεγάλα προβλήματα.

Τέλος, όσο αφορά τα προβλήματα Depots, αποτυγχάνουν τόσο το κατώφλι 10% όσο και το κατώφλι 25%. Παρατηρείται συγκεκριμένα σε μερικά προβλήματα ότι ο αριθμός επιπέδων αλλά και ο χρόνος εκτέλεσης της λύσης με αυτά τα κατώφλια είναι μεγαλύτερος από το αντίστοιχο του 50%. Αυτό οφείλεται κατά κύριο λόγο ότι η κατηγορία Depots περιλαμβάνει προβλήματα με μικρή πολυπλοκότητα κυρίως και γι' αυτό είναι πιο αποδοτικό, να ικανοποιηθεί μεγάλο ποσοστό περιορισμών αρχικά (δηλαδή 50% αντί 10% και 25%), ώστε να φτάσουμε πιο γρήγορα στην τελική λύση.

Τεχνική διατήρησης ικανοποιηθέντων περιορισμών - Κατώφλι 50%						
	Προβλήματα Pathways		Προβλήματα TPP		Προβλήματα PipeWorlDs	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0,18	5	0,02	3	0,13
p02	7	0,22	5	0,02	12	3,39
p03	8	0,69	5	0,04	7	5,87
p04	8	0,56	5	0,05	8	8,12
p05	9	2	7	0,12	7	6,21
p06	13	12,76	9	0,6	7	6,16
p07	16	30,21	9	0,7	6	23,22
p08	19	113,8	9	0,82	7	38,16
p09	20	122,48	11	1,82	12	232,94
p10	17	25,32	11	2,08	16	443,78
p11	19	57,95	14	7,1	19	1027,97
p12	21	528,82	14	7,73	16	156,43
p13	19	159,79	11	12,42	13	153,96
p14	-	Aborted Solver	13	82,65	-	Aborted Solver
p15	19	1030,17	13	68,65	-	Aborted Solver
p16	26	4920,53	10	17,28	-	Aborted Solver
p17	-	Aborted Solver	11	39,36	-	Aborted Solver
p18	-	Aborted Solver	11	68,01	16	1001,07
p19	-	Aborted Solver	11	4816,23	-	Aborted Solver
p20	-	Aborted Solver	12	104,02	-	Aborted Solver

Τεχνική διατήρησης ικανοποιηθέντων περιορισμών - Κατώφλι 50%						
	Προβλήματα Depots		Προβλήματα Storage		Προβλήματα Trucks	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0,05	3	0,01	-	Unsat
p02	8	0,41	3	0,01	17	13,44
p03	24	22,13	3	0,01	-	Unsat
p04	32	79,11	10	0,47	23	91,76
p05	56	478,59	6	0,15	23	350,34
p06	-	UNSAT	6	0,23	29	449,04
p07	10	2,85	16	2,75	-	Unsat
p08	15	22,76	8	0,91	-	Unsat
p09	59	3109,1	7	1,12	-	Unsat
p10	18	27,43	22	13,33	-	Aborted Solver
p11	33	430,91	12	6,13	-	Aborted Solver
p12	-	Aborted Solver	9	3,94	-	Aborted Solver
p13	9	1,48	22	27,61	-	Aborted Solver
p14	22	144,06	15	13,17	-	Aborted Solver
p15	-	UNSAT	11	10,79	-	Aborted Solver
p16	15	31,15	13	33,49	-	Aborted Solver
p17	13	68,6	13	106,37	-	Aborted Solver
p18	18	811,15	17	1869,33	-	Aborted Solver
p19	22	175,59	20	1415,89	-	Aborted Solver
p20	31	3916,15	-	Aborted Solver	-	Aborted Solver

Πίνακας 6.3.1 – Τεχνική διατήρησης ικανοποιηθέντων περιορισμών με κατώφλι 50%.

Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα - Κατώφλι 50%							Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα - Κατώφλι 50%						
Προβλήματα Pathways			Προβλήματα TPP		Προβλήματα PipeWorlds			Προβλήματα Depots		Προβλήματα Storage		Προβλήματα Trucks	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης		Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0,15	5	0,03	3	0,14	p01	5	0,07	3	0,01	15	7,79
p02	7	0,18	5	0,02	11	2,89	p02	8	0,53	3	0,01	15	10,7
p03	8	0,59	5	0,03	7	5,76	p03	15	5,97	3	0,02	22	85,03
p04	8	0,42	5	0,04	8	8,1	p04	14	6,29	8	0,25	18	57,92
p05	9	1,88	7	0,1	7	6,16	p05	29	144,45	7	0,28	20	306,8
p06	13	13,52	9	0,61	7	6,26	p06	29	1182,43	7	0,39	24	794,01
p07	14	26,88	9	0,69	6	23,18	p07	10	2,84	14	1,83	25	345,75
p08	18	671,18	9	0,79	7	38,54	p08	15	22,91	9	1,4	19	546,14
p09	18	4042,34	11	1,82	11	230,89	p09	-	Aborted Solver	8	1,75	27	4054,83
p10	15	14,38	11	2,04	12	310,79	p10	10	11,11	18	8,03	-	Aborted Solver
p11	18	398,53	13	4,62	13	40,62	p11	-	Aborted Solver	11	5,1	-	Aborted Solver
p12	20	2571,98	13	5,51	17	1409,98	p12	-	Aborted Solver	10	6,46	-	Aborted Solver
p13	18	180,87	10	8,75	13	151,53	p13	9	1,47	18	21,06	-	Aborted Solver
p14	-	Aborted Solver	10	18,38	-	Aborted Solver	p14	17	74,49	14	35,7	-	Aborted Solver
p15	19	1305,9	10	48,32	-	Aborted Solver	p15	28	9140,83	10	8,91	-	Aborted Solver
p16	-	Aborted Solver	10	17,44	-	Aborted Solver	p16	9	6,14	11	79,56	-	Aborted Solver
p17	-	Aborted Solver	11	39,49	-	Aborted Solver	p17	8	14,47	13	8259,24	-	Aborted Solver
p18	-	Aborted Solver	11	68,79	18	1570,58	p18	14	472,62	-	Aborted Solver	-	Aborted Solver
p19	-	Aborted Solver	11	5150,38	-	Aborted Solver	p19	11	28,91	-	Aborted Solver	-	Aborted Solver
p20	-	Aborted Solver	12	100,67	-	Aborted Solver	p20	-	Aborted Solver	-	Aborted Solver	-	Aborted Solver

Πίνακας 6.3.2 – Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα με κατώφλι 50%.

Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 3 βήματα - Κατώφλι 50%						
	Προβλήματα Pathways		Προβλήματα TPP		Προβλήματα PipeWorlds	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0,13	5	0,01	3	0,14
p02	7	0,17	5	0,03	13	4,04
p03	8	0,57	5	0,02	7	5,79
p04	8	0,44	5	0,03	8	8,18
p05	9	1,88	7	0,09	7	6,16
p06	13	12,7	9	0,48	7	6,32
p07	15	27,34	9	0,55	6	23,14
p08	18	489,26	9	0,65	7	38,52
p09	18	173,82	11	1,68	11	189,76
p10	16	20,28	11	2,03	13	296,07
p11	18	64,34	14	7,06	12	29,57
p12	21	1756,3	14	7,5	16	201,49
p13	18	154,22	11	12,34	12	124,19
p14	-	Aborted Solver	11	35,08	-	Aborted Solver
p15	19	1157,38	11	79,88	-	Aborted Solver
p16	-	Aborted Solver	10	17,17	-	Aborted Solver
p17	-	Aborted Solver	11	40,03	-	Aborted Solver
p18	-	Aborted Solver	11	73,67	17	1360,91
p19	-	Aborted Solver	11	5468,77	-	Aborted Solver
p20	-	Aborted Solver	12	101,15	-	Aborted Solver

Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 3 βήματα - Κατώφλι 50%						
	Προβλήματα Depots		Προβλήματα Storage		Προβλήματα Trucks	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0,07	3	0,01	18	13,77
p02	8	0,53	3	0,03	15	10,24
p03	18	10,55	3	0,02	18	117,91
p04	15	8,14	8	0,34	20	70,92
p05	33	205,36	6	0,19	22	360,18
p06	33	964,58	6	0,27	24	374,28
p07	10	2,84	14	1,98	22	354,97
p08	15	22,67	8	1,1	20	551,76
p09	-	Aborted Solver	7	1,25	31	7117,57
p10	15	17,04	18	7,72	-	Aborted Solver
p11	-	Aborted Solver	11	4,88	-	Aborted Solver
p12	-	Aborted Solver	10	5,48	-	Aborted Solver
p13	9	1,49	20	24,36	-	Aborted Solver
p14	21	128,44	15	16,38	-	Aborted Solver
p15	-	Aborted Solver	10	8,31	-	Aborted Solver
p16	10	9,06	12	33,75	-	Aborted Solver
p17	9	22,09	13	493,37	-	Aborted Solver
p18	16	590,62	14	2568,95	-	Aborted Solver
p19	13	46,46	-	Aborted Solver	-	Aborted Solver
p20	-	Aborted Solver	-	Aborted Solver	-	Aborted Solver

Πίνακας 6.3.3 – Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 3 βήματα με κατώφλι 50%.

Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα - Κατώφλι 10%						
	Προβλήματα Pathways		Προβλήματα PipeWorlds		Προβλήματα Depots	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0,17	3	0,11	5	0,05
p02	7	0,21	12	3,96	9	0,74
p03	8	0,71	6	3,6	15	5,76
p04	8	0,57	7	5,58	14	6,11
p05	9	1,97	7	5,99	29	146,56
p06	13	13,03	7	6,02	29	930,65
p07	14	22,57	6	22,57	10	2,77
p08	18	496,72	7	37,7	15	22,56
p09	18	522,82	11	222,42	30	4243,25
p10	15	15,49	12	303,11	13	11,25
p11	18	231,1	13	39,45	-	Aborted Solver
p12	20	3726,29	17	1401,69	-	Aborted Solver
p13	18	252,6	13	156,78	9	1,48
p14	-	Aborted Solver	-	Aborted Solver	17	77,19
p15	19	757,84	-	Aborted Solver	28	3734,27
p16	-	Aborted Solver	-	Aborted Solver	8	3,87
p17	-	Aborted Solver	-	Aborted Solver	7	8,38
p18	-	Aborted Solver	18	1557,11	15	532,72
p19	-	Aborted Solver	-	Aborted Solver	13	48,02
p20	-	Aborted Solver	-	Aborted Solver	-	Aborted Solver

Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα - Κατώφλι 25%						
	Προβλήματα Pathways		Προβλήματα PipeWorlds		Προβλήματα Depots	
	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης	Αριθμός επιπέδων	Χρόνος Εκτέλεσης
p01	5	0,17	3	0,13	5	0,07
p02	7	0,22	12	4,25	9	0,85
p03	8	0,68	6	3,62	15	5,75
p04	8	0,55	7	5,65	14	6,36
p05	9	2,02	7	6,07	29	150,53
p06	13	13,06	7	6,28	29	938,64
p07	14	26,36	6	22,64	10	2,84
p08	19	5187,05	7	38,05	18	39,52
p09	18	616,74	11	225,02	-	Aborted Solver
p10	15	15,85	12	493,46	13	11,25
p11	18	230,92	13	39,86	-	Aborted Solver
p12	20	3386,3	17	1364,47	-	Aborted Solver
p13	19	408,78	13	160,18	9	1,48
p14	-	Aborted Solver	-	Aborted Solver	17	74,61
p15	19	284,84	-	Aborted Solver	-	Aborted Solver
p16	-	Aborted Solver	-	Aborted Solver	8	3,98
p17	-	Aborted Solver	-	Aborted Solver	7	8,34
p18	-	Aborted Solver	18	1545,64	15	516,53
p19	-	Aborted Solver	-	Aborted Solver	13	47,67
p20	-	Aborted Solver	-	Aborted Solver	-	Aborted Solver

Πίνακας 6.3.4 – Τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα με κατώφλι 10% και 25%

Κεφάλαιο 7

Συμπεράσματα

7.1 Σύνοψη και συμπεράσματα από τη Διπλωματική Εργασία	72
7.2 Επεκτάσεις και Μελλοντική Εργασία	77

7.1 Σύνοψη και συμπεράσματα από τη Διπλωματική Εργασία

Το πρόβλημα του προγραμματισμού δράσεων ξεκίνησε από πολύ παλιά και αποτελεί μέχρι και σήμερα ένα ενδιαφέρον θέμα προς έρευνα. Ο Προγραμματισμός Δράσης δεν ασχολείται απλά με την εύρεση μιας λύσης, αλλά με την εύρεση μιας αποδοτικής λύσης σε χρόνο. Δηλαδή μιας συγκεκριμένης ακολουθίας από δράσεις, οι οποίες όταν εφαρμοστούν να οδηγούν σε κάποιο στόχο, ελαχιστοποιώντας ταυτόχρονα το κόστος σε χρόνο και υλικό (πόρου του συστήματος).

Οι διάφοροι αλγόριθμοι που αναπτύχθηκαν τα τελευταία χρόνια αλλά και η ραγδαία και συνεχής εξέλιξη της τεχνολογίας, ήταν το έναυσμα για τα προβλήματα προγραμματισμού δράσης. Σημαντική ήταν και η συνεισφορά της τεχνητής νοημοσύνης, με την ανάπτυξη διάφορων αλγοριθμικών και ευρετικών μεθόδων που στοχεύουν στο να φτάσουν σε μια ικανοποιητική λύση.

Μέσα στα πλαίσια αυτής της διπλωματικής εργασίας μελετήθηκαν αρχικά οι πρώτοι αλγόριθμοι που αναπτύχθηκαν για προβλήματα προγραμματισμού δράσης. Η επίλυση των προβλημάτων με αυτούς τους αλγορίθμους μπορεί να ξεκινούσε από την αρχική κατάσταση μέχρι να φτάσει στο στόχο (Forward state-space search) ή αντίστροφα (Backward state-space search). Τέτοιου είδους αλγόριθμοι ονομάζονται αλγόριθμοι προγραμματισμού δράσης με ολική διάταξη, αφού ο τρόπος με τον επιλύουν το πρόβλημα χαρακτηρίζεται από γραμμικότητα, δηλαδή η κάθε επόμενη δράση

σχετίζεται με την προηγούμενη. Ακολούθως αναπτύχθηκαν και αλγόριθμοι οι οποίοι είχαν τη δυνατότητα διάσπασης του προβλήματος σε υποπροβλήματα οι λεγόμενοι αλγόριθμοι προγραμματισμού δράσης με ολική διάταξη, καθώς και πολλά ευρετικά τα οποία έκαναν μια εκτίμηση της απόστασης της εκάστοτε κατάστασης από το στόχο. Στη συνέχεια και λόγω της περιορισμένης εμβέλειας των ευρετικών δημιουργήθηκε μια πιο ολοκληρωμένη δομή, το λεγόμενο γράφημα δράσεων, μέσα από το οποίο η αναζήτηση και η οργάνωση πληροφοριών γίνεται με πιο δομημένο τρόπο – καθορίζονται επακριβώς οι δράσεις που είναι δυνατόν να εφαρμοστούν σε κάθε χρονική στιγμή, οι αμοιβαία αποκλειόμενες δράσεις καθώς και οι συνέπειες των δράσεων. Παράλληλα ως γλώσσα αναπαράστασης των προβλημάτων προγραμματισμού δράσης αναπτύχθηκε η γλώσσα STRIPS.

Σταθμός στον προγραμματισμό δράσης θεωρείται στη συνέχεια η ανάπτυξη του ολοκληρωμένου συστήματος SatPlan, το οποίο μπορούσε να επιλύσει ένα πρόβλημα που ήταν σε μορφή προτασιακής λογικής. Το SatPlan είναι από τα πιο ανταγωνιστικά συστήματα Προγραμματισμού Δράσεων μέχρι σήμερα καθώς μπορεί να υποστηρίξει πληθώρα κωδικοποιήσεων που περιλαμβάνουν η καθεμία διαφορετικές συζευκτικές κανονικές μορφές. Πιο αποδοτική κωδικοποίηση είναι η SAT-MAX-PLAN (SMP), η οποία αφαιρεί όλους τους πλεονασμούς από τις προτάσεις και στηρίζεται στην μετάφραση των περιορισμών ενός προβλήματος που προκύπτουν από το γράφημα δράσεων σε μορφή προτασιακής λογικής. Οι διάφοροι προτασιακοί επιλυτές όπως Precosat και Minisat+ που αναπτύχθηκαν αποτελούν το «εκτελεστικό» κομμάτι του Satplan. Λαμβάνουν δηλαδή ως είσοδο την κάθε κωδικοποίηση (π.χ. SMP) η οποία περιγράφει τους περιορισμούς του προβλήματος και δίνουν ως έξοδο τη λύση σε αυτούς τους περιορισμούς εάν και εφόσον υπάρχει.

Έχουν μελετηθεί στη συνέχεια οι ψευδοδυαδικοί περιορισμοί. Οι ψευδοδυαδικοί περιορισμοί αποτελούν γενίκευση της προτασιακής λογικής, καθώς οι περιορισμοί λαμβάνουν και πάλι τιμές από το δυαδικό σύνολο $\{0,1\}$. Έχουν μεγάλη συμβολή στα προβλήματα προγραμματισμού δράσης εξαιτίας της μεγάλης εκφραστικότητας αλλά και απλότητας που τους χαρακτηρίζει. Για τέτοιου είδους περιορισμούς έχουν δημιουργηθεί και τα αντίστοιχα εργαλεία (Pseudo-Boolean solvers), όπως minisat+, τα οποία υποστηρίζουν αυτές τις μορφές περιορισμών.

Στη συνέχεια η διπλωματική εργασία εστιάζεται σε διάφορες μεθόδους επίλυσης προβλημάτων προγραμματισμού δράσης, που στηρίζονται στην βελτιστοποίηση, μέσω συνάρτησης βελτιστοποίησης του εργαλείου minisat+ (συνάρτηση min). Αρχικά έγινε ανάπτυξη μιας μεθοδολογίας η οποία προσπαθεί να λύσει το πρόβλημα μέσω κάποιας ενδιάμεσης κατάστασης, δηλαδή σε δύο ξεχωριστά στάδια και όχι απευθείας, με βάσει ένα κάτω φράγμα που καθορίζει το ποσοστό ικανοποιησιμότητας στόχων που θα πρέπει να ικανοποιεί η ενδιάμεση αυτή λύση. Ως μια παραλλαγή αυτής της μεθόδου ήταν η δημιουργία τριών διαφορετικών τεχνικών τμηματικής επίλυσης του προβλήματος, οι οποίες λύνουν το πρόβλημα τμηματικά σε ένα σύνολο από στάδια, όπου στο καθένα ικανοποιείται μικρός αριθμός περιορισμών και αυτό συνεχίζεται μέχρι να ικανοποιηθούν όλοι οι περιορισμοί. Οι τεχνικές αυτές όπως και η μεθοδολογία με την ενδιάμεση λύση αξιολογήθηκαν πειραματικά σε ένα μεγάλο εύρος προβλημάτων από διαφορετικές κατηγορίες.

Από τις πειραματικές μετρήσεις που έγιναν μέσα από αυτές τις μεθοδολογίες προκύπτουν κάποια συμπεράσματα που συνδέονται με τη χρονική πολυπλοκότητα και τους χρονικούς αλλά και χωρικούς περιορισμούς των προβλημάτων, τα οποία σαφώς επηρεάζουν την αποδοτικότητα της λύσης.

Από την μελέτη της μεθοδολογίας της ενδιάμεσης λύσης με κατώφλια 10%, 25% και 50% μπορεί να εξαχθεί το συμπέρασμα ότι το κατώφλι 10% είναι συνήθως κατάλληλο για προβλήματα με μεγάλη πολυπλοκότητα, που περιλαμβάνουν ένα μεγάλο σύνολο δυνατών περιπτώσεων όπως για παράδειγμα τα προβλήματα Pathways. Υπάρχει δηλαδή σε τέτοιου είδους προβλήματα λόγω του μεγάλου χώρου αναζήτησης τους, η ανάγκη να ικανοποιηθεί σύντομα ένα μικρό μέρος των περιορισμών και δεδομένου αυτών των περιορισμών να προσπαθεί ο επιλυτής να ικανοποιήσει τους υπόλοιπους περιορισμούς χωρίς να χρειάζεται να ικανοποιήσει τους περιορισμούς που έχουν ήδη ικανοποιηθεί. Το αρχικό αυτό σύνολο των περιορισμών θα πρέπει να είναι μικρό, ώστε να μην καθυστερήσει να ικανοποιηθεί και δεύτερον και πιο σημαντικό εφόσον ικανοποιηθεί αυτό το σύνολο να μην αυξάνει δραματικά τους περιορισμούς στην προσπάθεια ικανοποίησης των υπολοίπων στόχων (συγκρούσεις ανάμεσα στους αρχικούς ικανοποιηθέντες περιορισμούς και στους υπόλοιπους περιορισμούς στους οποίους γίνεται προσπάθεια ικανοποίησης τους).

Αντίθετα για πιο μικρά προβλήματα με λιγότερους συνδυασμούς περιπτώσεων κατάλληλο μπορεί να θεωρηθεί το κατώφλι 50%, όπως και το κατώφλι 25%. Σε τέτοια προβλήματα λόγω της μικρής πολυπλοκότητας, είναι δυνατό να ικανοποιηθεί αρχικά μεγάλο μέρος των περιορισμών – στόχων και στη συνέχεια να φτάσουμε στην τελική λύση σύντομα, εκμεταλλευόμενοι την μεγάλη πληροφορία της ενδιάμεσης λύσης και η οποία δεν δημιουργεί επιπρόσθετους περιορισμούς, λόγω όπως είπαμε του μικρού αριθμού δυνατών περιπτώσεων.

Αξίζει να σημειωθεί ότι το κατώφλι με ποσοστό 25% αποτελεί μια σχετικά ικανοποιητική επιλογή τόσο για προβλήματα μικρής αλλά και μεγάλης πολυπλοκότητας.

Η μεθοδολογία όμως με την ενδιάμεση κατάσταση διαπιστώθηκε ότι αποτυγχάνει να επιλύσει τη συγκεκριμένη κατηγορία προβλημάτων trucks. Αυτή η κατηγορία προβλημάτων περιγράφεται από αυστηρούς χρονικούς περιορισμούς λόγω των προθεσμιών που έχουν τα πακέτα να παραδοθούν στον προορισμό τους (κόστος χρονικών μονάδων). Η ενδιάμεση λύση εισάγοντας με μορφή περιορισμών τους στόχους που έχουν ικανοποιηθεί σε προηγούμενες χρονικές στιγμές, στις επόμενες χρονικές στιγμές κατά τις οποίες γίνεται η αναζήτηση της τελικής λύσης, δημιουργεί συγκρούσεις ανάμεσα σε στόχους που ικανοποιηθήκαν προηγουμένως και σε στόχους που πρέπει να ικανοποιηθούν τώρα, λόγω επικάλυψης των στόχων αυτών μέσω κάποιας κοινής παραμέτρου – περιορισμού που έχει εξορισμού το πρόβλημα.

Μετά από αυτές τις μετρήσεις με την μέθοδο της ενδιάμεσης κατάστασης, πραγματοποιήθηκαν οι ίδιες μετρήσεις στο ίδιο ευρύ φάσμα προβλημάτων αλλά με τη μέθοδο της τμηματικής επίλυσης και τις τρεις διαφορετικές τεχνικές οι οποίες έχουν προαναφερθεί. Όπως και με την ενδιάμεση κατάσταση έτσι και εδώ διαπιστώθηκε ότι δεν υπάρχει τεχνική η οποία να είναι ικανή να επιλύσει οποιοδήποτε πρόβλημα αποδοτικά. Η διαφορά με αυτήν την τμηματική μεθοδολογία είναι ότι το πρόβλημα λύνεται σε στάδια όπου σε κάθε στάδιο γίνεται προσπάθεια να επιλυθεί μικρός αριθμός περιορισμών. Αυτό προσδίδει κάποια αποδοτικότητα καθώς ο επιλυτής δεν «εγκλωβίζεται» σε ένα μεγάλο σύνολο περιπτώσεων, αλλά προσπαθεί σε κάθε στάδιο να ικανοποιήσει τον αμέσως επόμενο μεγαλύτερο αριθμό από στόχους και όχι όλους απευθείας.

Με βάση αυτό σε αρκετές κατηγορίες προβλημάτων όπως Pathways και Pipeworlds, οι τρεις αυτές τεχνικές τείνουν να έχουν καλύτερα αποτελέσματα σε σχέση με τη μέθοδο της ενδιάμεσης λύση. Κυρίως η τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα ή 3 βήματα, παρατηρούμε ότι επιτυγχάνει σε μερικά προβλήματα αισθητή μείωση του αριθμού επιπέδων του γραφήματος. Χαρακτηριστικό είναι το πρόβλημα 11 της κατηγορίας PipeWorlds, όπου η μέθοδος την ενδιάμεσης κατάστασης δίνει λύση μετά από 19 επίπεδα ενώ οι τεχνικές ενημέρωσης ικανοποιηθέντων περιορισμών δίνουν λύση μετά από μόλις 12 και 13 επίπεδα αντίστοιχα.

Επίσης οι τεχνικές διατήρησης και ενημέρωσης των στόχων που έχουν ικανοποιηθεί κατάφεραν να δώσουν σε μεγάλο βαθμό λύση στα προβλήματα trucks, στα οποία η ενδιάμεση λύση είχε αποτύχει σχεδόν εντελώς. Αυτό οφείλεται στο γεγονός ότι οι τεχνικές αυτές ενημέρωσης, εισάγουν μόνο τους στόχους που ικανοποιήθηκαν, ως περιορισμούς που αφορούν την τρέχουσα χρονική στιγμή και όχι ως περιορισμούς στη χρονική στιγμή στην οποία ικανοποιήθηκαν. Εξαιρέση φυσικά η τεχνική διατήρησης που εισάγει τους στόχους ως περιορισμούς, με βάση τη χρονική στιγμή που ικανοποιήθηκαν και όχι την τρέχουσα χρονική στιγμή. Αυτός είναι και ο λόγος που η τεχνική ενημέρωσης συμπεριφέρεται παρόμοια με αυτή της ενδιάμεσης λύσης για τα προβλήματα trucks, με ελαφρώς καλύτερα αποτελέσματα.

Συνοψίζοντας όλες τις τεχνικές που έχουν δοκιμαστεί και με βάση τα πειραματικά αποτελέσματα που έδωσαν στα προβλήματα που έχουν δοκιμαστεί μπορεί να διατυπωθεί ότι μια κατάλληλη τεχνική για προβλήματα μεγάλης πολυπλοκότητας είναι η τεχνική ενημέρωσης ικανοποιηθέντων περιορισμών μετά από 2 βήματα και με κατώφλι 10 %. Η τεχνική αυτή λόγω του ότι ξεκινά από χαμηλό κατώφλι προσπαθεί να ικανοποιήσει τους περιορισμούς ξεκινώντας από ένα μικρό υποσύνολο. Όταν ικανοποιηθεί αυτό το αρχικό χαμηλό κατώφλι συνεχίζει σταδιακά με μικρά βήματα και προσπαθεί να ικανοποιήσει και τους υπόλοιπους περιορισμούς. Επίσης λόγω της ενημέρωσης των περιορισμών που κάνει στις αντίστοιχες χρονικές στιγμές, αυτό δεν την περιορίζει σε χρονικούς περιορισμούς (δε ξεοδεύει χρονικές μονάδες) όπως γινόταν με τις τεχνικές διατήρησης περιορισμών αλλά και της τεχνικής με την ενδιάμεση λύση οι οποίες προσπαθούν να ικανοποιήσουν τους περιορισμούς στην τρέχουσα χρονική στιγμή αλλά και στην προηγούμενη χρονική στιγμή στην οποία ικανοποιήθηκαν αρχικά.

Αντίθετα για προβλήματα με μικρή πολυπλοκότητα ένα κατάλληλο κατώφλι μπορεί να θεωρηθεί ότι είναι το 50% με την μέθοδο της ενδιάμεσης λύσης ή και της τεχνικής διατήρησης ικανοποιηθέντων περιορισμών. Λόγω του μικρού συνόλου από περιορισμούς σε τέτοια προβλήματα, είναι προτιμότερο να ικανοποιηθεί μεγάλο ποσοστό αρχικά, αφού ο επιλυτής δεν περιορίζεται τόσο πολύ στις αποφάσεις που παίρνει (ανάθεση στις μεταβλητές) για να ικανοποιήσει αυτούς τους περιορισμούς.

Τέλος θα πρέπει να σημειωθεί όπως φαίνεται και από τους πίνακες για όλες τις τεχνικές το γεγονός της εξόδου του επιλυτή με το μήνυμα «Aborted Solver». Το μήνυμα αυτό που δίνει ο επιλυτής σε μερικές περιπτώσεις προβλημάτων, εμποδίζει κατά κάποιο τρόπο να διαφανεί ακριβώς κατά πόσο μια τεχνική είναι αποδοτική, αφού δεν εξάγεται κάποιο αποτέλεσμα για να μπορεί να συγκριθεί. Το μήνυμα αυτό εμφανίζεται τόσο από τον επιλυτή precosat όσο και από τον minisat+. Το πρόβλημα που δημιουργείται με αυτό το μήνυμα προϋπήρχε και πριν την έναρξη αυτής της διπλωματικής εργασίας. Παρά τις συνεχείς προσπάθειες που έχουν γίνει δεν έχει βρεθεί κάποια εξήγηση για την εμφάνιση αυτού του μηνύματος που δίνει ο επιλυτής. Ίσως να δημιουργείται λόγω κάποιων παραμέτρων που δίνονται στον κάθε επιλυτή, π.χ. δέσμευσης μνήμης. Ακόμη και στον ενημερωτικό αρχείο που δόθηκε από τους δημιουργούς του Satplan, δεν γράφει οτιδήποτε το οποίο να δίνει μια εξήγηση για την αιτία εμφάνισης αυτού του μηνύματος.

7.2 Επεκτάσεις και Μελλοντική Εργασία

Στα πλαίσια αυτής της διπλωματικής εργασίας έγινε μελέτη διάφορων τεχνικών μέσω των οποίων προσπαθήσαμε να φτάσουμε σε αποδοτικές λύσεις, που να ελαχιστοποιούν το χρόνο εκτέλεσης και γενικότερα το κόστος. Η συνάρτηση βελτιστοποίησης, αποτέλεσε το κυριότερο εργαλείο σε αυτή την προσπάθεια. Μέσω αυτής ήταν κατορθωτό να ικανοποιείται ένα οποιοδήποτε υποσύνολο από στόχους υπό την προϋπόθεση ότι ήταν δυνατό να ικανοποιηθεί τη συγκεκριμένη χρονική στιγμή.

Με βάση τη μέθοδο της ενδιάμεσης λύσης, θα μπορούσε να δοκιμαστεί η ίδια μέθοδο αλλά με μια μικρή διαφοροποίηση. Η ενδιάμεση λύση όπως περιγράφηκε προσθέτει ένα μεγάλο σύνολο από περιορισμούς στην αμέσως επόμενη χρονική στιγμή. Αυτοί οι περιορισμοί (unit clause) αντικατοπτρίζουν στην ουσία όλους τους περιορισμούς που

ικανοποιηθήκαν στην ενδιάμεση λύση (δηλαδή ποιες μεταβλητές – περιορισμοί είναι true και ποιες false). Με την εισαγωγή όλων αυτών των περιορισμών όμως περιορίζεται κατά κάποιο τρόπο η αυτοδυναμία του επιλυτή καθώς δεν μπορεί ικανοποιήσει με διαφορετικό τρόπο (ανάθεση στις μεταβλητές) τους περιορισμούς αυτούς στην προσπάθεια που κάνει για να φτάσει στην τελική λύση. Έτσι αντί αυτού θα μπορούσε να δοκιμαστεί κάτι διαφορετικό μετά που φτάνουμε στην ενδιάμεση λύση. Θα μπορούσε να δημιουργηθεί νέο γράφημα δράσεων (planning graph) το οποίο να έχει ως αρχική κατάσταση όμως την ενδιάμεση λύση στην οποία έχουμε φτάσει. Έτσι το γράφημα θα ξεκινά να επεκτείνεται και πάλι από το πρώτο επίπεδο και ο χώρος αναζήτησης θα είναι σαφώς μικρότερος από ότι προηγουμένως. Θα αφαιρούνται με αυτόν τον τρόπο όλοι αυτοί οι περιορισμοί που εισάγονταν μετά την ενδιάμεση λύση, καθώς αυτοί θα εισάγονται έμμεσα πλέον μέσω της νέας αρχικής κατάστασης.

Επιπρόσθετα μια δεύτερη τροποποίηση θα μπορούσε να γίνει ως προς τις τεχνικές της τμηματικής επίλυσης που προτείναμε. Με βάση δηλαδή τις τεχνικές διατήρησης και ενημέρωσης των ικανοποιηθέντων περιορισμών θα ήταν δυνατόν να δημιουργηθεί μια νέα παραλλαγή αυτών των μεθόδων. Η δημιουργία μιας υβριδικής μεθόδου θα μπορούσε να συνδυάσει τα δυνατά σημεία των τεχνικών αυτών. Δηλαδή ανάλογα και με την κατηγορία προβλημάτων που επιλύουμε θα μπορούσε να ξεκινά με την τεχνική διατήρησης των ικανοποιηθέντων στόχων. Ενώσω παρατηρείται ότι αυτή η τεχνική δίνει λύση (τμηματικά πάντοτε) μέσα σε ένα συγκεκριμένο χρονικό όριο συνεχίζουμε με αυτήν. Μόλις εντοπιστεί ότι δε βρίσκεται λύση εντός συγκεκριμένου χρονικού ορίου, θα συνεχίζουμε με την τεχνική ενημέρωσης των ικανοποιηθέντων στόχων. Η τεχνική της ενημέρωσης θα στοχεύει να συσχετίζει τους περιορισμούς με τις τρέχουσες χρονικές στιγμές, στις οποίες υπάρχει μεγαλύτερη πιθανότητα να ικανοποιούνται.

Μπορούμε με αυτήν την τελευταία τεχνική να δοκιμάσουμε και διαφορετικά κατώφλια και όχι μόνο με ποσοστό 50%. Καλό θα ήταν να ξεκινούσαμε από ένα μικρότερο ποσοστό περίπου 25% ή και 10% ανάλογα με το χρόνο εκτέλεσης που παρατηρούμε ότι δίνει λύση η πρώτη τεχνική της υβριδικής αυτής μεθόδου. Ταυτόχρονα μια δοκιμή με κατώφλι μεγαλύτερο του 50%, π.χ. 75% με όλες τις τεχνικές που αναφέρθηκαν μπορεί να οδηγήσει σε μια καλύτερη και πιο ακριβή εικόνα σχετικά με την αποδοτικότητα αυτών των τεχνικών, αλλά και τυχόν επεκτάσεις που μπορούν να γίνουν.

Πέραν όμως από αυτά σημαντική θεωρείται και η επίλυση του προβλήματος με το μήνυμα «Aborted Solver». Καλό θα είναι η συνέχεια αυτής της εργασίας να επιλύσει πρώτα από όλα αυτό το πρόβλημα. Έτσι θα υπάρχει μια πιο σαφής εικόνα στα αποτελέσματα των μεθόδων.

Εν κατακλείδι θα πρέπει να λαμβάνουμε πάντοτε υπόψη ότι η βελτιστοποίηση του χρόνου είναι αποτέλεσμα πολλών παραμέτρων, κάθε μια από τις οποίες πρέπει να εξεταστεί ξεχωριστά.

Βιβλιογραφία

- [1] Stuart Russel and Peter Norvig, Artificial Intelligence: A Modern Approach(Second Edition), Chapter 11 Planning, Prentice Hall, 2002.
- [2] Andreas Sideris, Yannis Dimopoulos, "Constraint Propagation in Propositional Planning",20th International Conference on Automated Planning and Scheduling, (ICAPS'10), "Toronto, Canada, 2010.
- [3] Olivier Roussel and Vasco Manquinho, "Pseudo-Boolean and Cardinality Constraints", Chapter 22, Handbook of Satisfiability Armin Biere, Marijn Heule, Hans van Maaren and Toby Walsh (Eds.) IOS Press, 2009.
- [4] Henry Kautz, Bart Selman, Joerg Hoffmann, "SatPlan: Planning as Satisfiability"
- [5] Henry Kautz, "Deconstructing Planning as Satisfiability" , Proceedings of the 21st National Conference on Artificial Intelligence. (AAAI-06) , Boston, MA, 2006.
- [6] Federico Heras, Vasco Manquinho, Joao Marques-Silva "On Applying Unit Propagation-Based Lower Bounds in Pseudo-Boolean Optimization", Proceedings of the Twenty-First International FLAIRS Conference(2008), Association for the Advancement of Artificial Intelligence (www.aaai.org) ,2008.
- [7] Yannis Dimopoulos, Alfonso Gerevini, Patrik Haslum, Alessandro Saetti, "The Benchmark Domains of the Deterministic Part of IPC-5",Booklet of the 2006 Planning Competition, ICAPS'06, 2006
- [8] Niklas Een, Niklas Sorensson, "Translating Pseudo-Boolean Constraints into SAT", Journal on Satisfiability, Boolean Modeling and Computation 2, 2006.